When Nash met Markov: Novel results for pure Nash equilibria and the switch Markov chain

Pieter Kleer

Promotores:	Prof.dr. Guido Schäfer
	Prof.dr. Alexander Schrijver
Other members:	Prof.dr. Leen Stougie (chair)
	Prof.dr. Catherine Greenhill
	Dr. Evdokia Nikolova
	Prof.dr. Britta Peis
	Prof.dr. Marc Uetz

This research has been carried out in the Networks and Optimization group at the Centrum Wiskunde & Informatica in Amsterdam, supported by NWO Gravitation Project Networks, Grant Number 024.002.003.

Printed and bound by Ipskamp Printing

ISBN 978-94-028-1601-3

VRIJE UNIVERSITEIT

When Nash met Markov: Novel results for pure Nash equilibria and the switch Markov chain

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor
aan de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. V. Subramaniam,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de School of Business and Economics
op maandag 9 september 2019 om 15.45 uur
in de aula van de universiteit,
De Boelelaan 1105

door

Pieter Sybrand Kleer

geboren te 's-Gravenhage

promotoren: prof.dr. G. Schäfer
prof.dr. A. Schrijver

Acknowledgements

First of all, I would like to thank my supervisor Guido for his guidance and support throughout my time at CWI. I have very much enjoyed all our talks, writing papers together, and having you as my supervisor. You have taught me many valuable lessons and your door was always open. Many thanks for everything! For their willingness to read and approve this thesis, I want to thank Britta, Catherine, Evdokia, Leen and Marc, and, in particular, Catherine for her extensive feedback on Chapter 4. I also want to thank Lex for being my second promotor.

It has been a great pleasure to carry out this research in the Networks project. It inspired me to think about, and work on, problems outside of algorithmic game theory. A special thanks goes to Yorgos Amanatidis and Jacobien Carstens for letting me meet with various Markov Chain Monte Carlo approaches for the generation of graphs with given degrees, and for working with me on the results presented in Chapter 4 of this thesis. I also want to thank all the other students and staff of the Networks project for the many nice project events.

I want to thank Tim Roughgarden for hosting me for two months at Columbia University in the City of New York, and the Networks project for funding this visit. Many thanks for this great opportunity and sharing your extensive knowledge on all kinds of topics in theoretical computer science! Also, I want to thank Vaggos Chatziafratis for many nice discussions and informal chats, and, finally, everybody at the Data Science Institute and department of Computer Science of Columbia for the many nice lunches and dinners.

There have been many people at CWI over the years that have made working there very enjoyable; I would like to thank all of you! Special mention goes to Sander who has been around since the first year of university in Delft, and with whom I have discussed lots of research and have played many ping-pong matches. A special thanks also goes to Bart, Carla, Daniel, Ilan, Irving (for letting me borrow your bat more or less permanently), Isabella, Makrand, Mathé, Monique, Neil, Nikhil, Ruben (for getting us the first place in the CWI doubles ping-pong tournament), Sophie (for never complaining about me walking in with yet another smoothed analysis question), Susanne, Sven, Viresh, Yllka, and Yorgos. Finally, thanks to all colleagues I have had the pleasure to play with, and against, in the activity room over the years! Also a shout-out to the PhD activity committee for organizing many cool events over the years.

Moreover, I also want to thank all the people that I have met over the years at conferences, visits and workshops. Thanks for all the nice games, dinners and beers. Special thanks to Veerle, Tim, and the many Germans I have had the pleasure to meet; and to Britta and Veerle for letting me spend a nice week in Aachen!

Finally, I would like to thank all my friends and family for getting my mind off of research every now and then. Pap en mam, bedankt voor al jullie hulp en steun met vele niet werkgerelateerde zaken in het leven. Dank dat jullie altijd voor me klaar staan!

Contents

Chapter 1

Introduction

Traffic congestion. On December 18, 2015, the first motorists were allowed on an extended stretch of the A4 highway in the Netherlands, that provides a new connection between the cities of Delft and Schiedam [149]. One of the main reasons for this extended stretch was to alleviate the traffic congestion on the nearby A13 highway. Indeed, one might be inclined to believe that extending the road network will always improve the traffic situation. After all, there is more area of road available for the same amount of motorists. This reasoning is mostly true if there would be a centralized entity, say, the Dutch government, that has the power to prescribe the route every motorist has to take in order to get to its destination. The government could assign routes in such a way that the traffic is nicely spread out over all roads, minimizing the total traffic congestion.

Figure 1.1: The A4 highway goes from Amsterdam to the town of Ossendrecht near the Belgium border, with a break around Rotterdam. The extended stretch connecting Delft and Schiedam is the lowest part of the upper blue trajectory (source: www.wegenwiki.nl).

Unfortunately, real-life is far from this ideal situation. Every motorist is free to take any route, and, most likely, will choose the fastest one available at the moment of departure (which is easy to obtain nowadays using, e.g., Google Maps). That is, motorists tend to be selfish: Their objective is to get to their destination as quickly as possible, not caring about the travel time of other motorists using the road network. Selfish behavior has turned out to be very important in the analysis of traffic congestion. In particular, because of selfishness, building more roads is not always the solution to traffic congestion. Equivalently, closing down road sections can sometimes improve the traffic situation in the presence of selfish players! This phenomenon is called the *Braess paradox*, named after Dietrich Braess, who gave the first mathematical description of it in 1969 [18].

Selfish behavior is one of the main aspects studied in the field of *algorithmic game theory*, that lies at the intersection of (theoretical) computer science, economics and mathematics. Algorithmic game theory is concerned with designing and analyzing algorithms in *strategic* settings, and has received considerable attention especially since the rise of the Internet at the end of the last century. An excellent book on many topics in algorithmic game theory is [140].

A fundamental problem in algorithmic game theory is to *quantify the inefficiency* of selfish behavior. For example, in the traffic example above we would like to know how much higher the traffic congestion is, as a result of selfish behavior, compared to the traffic congestion in the ideal situation in which we are allowed to assign routes to motorists. If the inefficiency is significant, there is a need to come up with *mechanisms* that alleviate the problem. In the case of traffic congestion, such a mechanism could be a *tolling scheme* for busy road sections.

A large part of this thesis is concerned with quantifying the inefficiency of selfish behavior in two congestion models: Wardrop's routing model [175] and Rosenthal's congestion model [150].

Darwin's finches. Apart from the analysis of congestion models, this thesis also focuses on a different topic at the intersection of theoretical computer science and mathematics, that of *uniformly sampling (bipartite) graphs with given degrees*. We will illustrate this problem, as well as its relevance, with an example from the field of ecology.

In the nineteenth century, Darwin travelled to the Galápagos islands and observed the presence of many different species of finches on different islands (see Figure 1.2). In particular, these species showed significant differences in their beaks, both in form and function.[1] A question of great interest concerning Darwin's finch data is formulated in the following quote from [158]:

"Birds with differing beaks may live side by side because they can eat different things, whereas similarly endowed animals may not occupy the same territory because they compete with one another for the same kinds of food. Ecologists have long debated whether such competition between similar species controls their distribution on island groups or whether the patterns found simply reflect chance events in the distant past."

[1] We refer the interested reader to [30] for more details.

	Island																
Finch	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*
Large ground finch	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
Medium ground finch	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0
Small ground finch	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0
Sharp-beaked ground finch	0	0	1	1	1	0	0	1	0	1	0	1	1	0	1	1	1
Cactus ground finch	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	0	0
Large cactus ground finch	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
Large tree finch	0	0	1	1	1	1	1	1	1	0	0	1	0	1	1	0	0
Medium tree finch	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Small tree finch	0	0	1	1	1	1	1	1	1	1	0	1	0	0	1	0	0
Vegetarian finch	0	0	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0
Woodpecker finch	0	0	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0
Mangrove finch	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Warbler finch	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

NOTE: Island name code: A = Seymour, B = Baltra, C = Isabella, D = Fernandina, E = Santiago, F = Rábida, G = Pinzón, H = Santa Cruz, I = Santa Fe, J = San Cristóbal, K = Española, L = Floreana, M = Genovesa, N = Marchena, O = Pinta, P = Darwin, Q = Wolf.

Figure 1.2: Darwin's finch data (source: screenshot of Table 1 in [30].)

In order to address the question of whether or not such patterns can be explained by chance events, we can use the statistical tool of *hypothesis testing* that we briefly explain next (see [30] for more details). The idea is to generate (or sample) many random 0-1 tables[2] with *similar characteristics* as the observed data in Figure 1.2, and, based on these random samples, determine how likely it is that finches with different types of beaks appear on the same island. What do we mean by similar characteristics? In our setting the characteristics of interest are the row and column sums of the observed data in Figure 1.2. These so-called *marginals* reflect the fact that certain islands can accommodate more species than others, as well as that certain species appear more often than others. Thus, we are interested in the problem of generating tables with the same marginals as the observed data, and, perhaps most importantly, we would like to be able to do this *efficiently* for practical purposes. The reason that we have to resort to sampling is that computing all possible tables with the same marginals is often not possible as there might be a tremendous amount of them.

One prominent line of work to obtain such random samples is the *Markov Chain Monte Carlo method* that roughly works as follows. We start with the observed 0-1 table and repeatedly make small random changes to it, according to some probabilistic procedure, while preserving the marginals. The idea is that if one makes sufficiently many random changes, then the resulting table corresponds to a random sample from the set of all possible tables with the same marginals. Perhaps one of the most natural examples of such a probabilistic procedure is to repeatedly *switch (or swap)* two 1-entries uniformly at random while preserving the marginals.

We illustrate such a *switch operation* with an example; see Figure 1.3. We perform a *switch operation* on 1-entries $(1,1)$ and $(3,3)$ as follows. If entries $(1,3)$ and $(3,1)$ contain a zero: interchange the 1-entries in $(1,1)$ and $(3,3)$ with the 0-entries from $(1,3)$ and $(3,1)$. Note that this operation preserves the marginals.

A fundamental question that arises here is the following: How many of these

[2] A 0-1 table can naturally be interpreted as the adjacency matrix of a bipartite graph.

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

Figure 1.3: Example of switch operation.

switches do we have to perform before our table is close to a random sample? This is one of the problems we study in Chapter 4.

1.1 About this thesis

This thesis is the result of four years of research carried out as a Ph.D. student in the Networks and Optimization group at CWI in Amsterdam, the Netherlands. The Ph.D. position has been a part of the Networks project, a 10-year program (2014–2024) funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organization for Scientific Research. The project focuses on network-related research in many different fields in mathematics and theoretical computer science, ranging from random graph theory to quantum computing, and more.

As described above, this thesis focuses on two areas at the intersection of theoretical computer science and mathematics. As these two areas do not have any direct overlap, we continue this chapter with some general background information for both. In Section 1.2 we discuss *(finite) strategic games* and some of the fundamental aspects of interest that have been studied in the last twenty years. The model in Chapter 3 is such a strategic game. The model in Chapter 2 can roughly be seen as a strategic game with an infinite number (or continuum) of players, which is explained in Section 1.3. In Section 1.4 we present some background on the topic of *(uniformly) sampling finite objects* and its relation to *counting finite objects*. We illustrate all concepts using the problem of sampling perfect matchings from a given undirected graph, which is a generalization of the problem of sampling graphs with given degrees that we consider in Chapter 4. Results for which no proof or reference are given can be found in most standard textbooks on the area of discussion. For an overview of the contributions in this thesis, see Section 1.5.

Sections 1.2, 1.3 and 1.4 are not meant to be a complete overview of their respective area, but rather a gentle introduction for readers who are unfamiliar with them. We do not give technical definitions of all notions discussed (e.g., we do not formally define complexity classes).

1.2 Strategic games

A basic concept in (algorithmic) game theory is the notion of a strategic game (or non-cooperative game).

Definition 1.1. A *strategic game* is a tuple $\Gamma = (N, (\mathcal{S}_i)_{i \in N}, (C_i)_{i \in N})$ where $N = \{1, \dots, n\}$ is a finite set of *players*,[3] $\mathcal{S}_i$ a finite set of *strategies* for $i \in N$, and $C_i : \times_{j \in N} \mathcal{S}_j \to \mathbb{R}$ a *cost function* for $i \in N$. A vector $s = (s_1, \dots, s_n) \in \times_i \mathcal{S}_i$ is called a *strategy profile.*

We write $s_{-i} = (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$ for the vector of length $n - 1$ in which the strategy of player $i \in N$ is left out of the strategy profile s. We use (s_{-i}, s'_i) to denote the strategy profile in which all players play their strategy in s, except for player i, that plays strategy $s'_i \in \mathcal{S}_i$.[4]

A cost minimization game is a finite strategic game in which the objective of every player is to choose a strategy from her set of strategies that minimizes her cost (where randomization over strategies is allowed). Note that her cost not only depends on the strategy she chooses, but also on the chosen strategies of the other players. We assume that our cost minimization game is a so-called *one shot game*, meaning that all players have to declare a strategy simultaneously. Furthermore, players have *full information* meaning that they know the strategy sets and cost functions of all the other players.

We present an example of a finite strategic game that is a special case of the congestion model studied in Chapter 3.

Example 1.2. An *atomic network congestion game* is a strategic game on a directed graph $G = (V, A)$. For every player $i \in N$ her set of strategies $\mathcal{S}_i$ consists of all (o_i, d_i)-paths in the graph G for some $o_i, d_i \in V$. Moreover, every arc $a \in A$ is equipped with a cost function $c_a : \mathbb{N} \to \mathbb{R}$ so that for a given strategy profile $s = (s_1, \dots, s_n)$ and $i \in N$, we have

$$C_i(s) = \sum_{a \in s_i} c_a(x_a),$$

where x_a is the number of players using arc a in the profile s. That is, every player controls one unit of flow that has to be routed through the network, and the goal is to choose an (o_i, d_i)-path that minimizes her total cost through the network. The game is called *symmetric* if all players have the same origin o and destination d.

The notion of a cost minimization game gives rise to the following question: which strategy profiles can we expect to see as an outcome? Intuitively, we might expect the outcome of the game to satisfy some form of 'stability' in the sense that no player, in retrospect, would like to have chosen a different strategy. This

[3]Throughout this thesis, we often use the notation $[n] = \{1, \dots, n\}$ for $n \in \mathbb{N}$.

[4]For functions $f : \times_i \mathcal{S}_i \to \mathbb{R}$ we slightly abuse notation and write $f(s_{-i}, s'_i)$ for the evaluation of f in the strategy profile (s_{-i}, s'_i), instead of $f((s_{-i}, s'_i))$.

can be formalized by means of a *solution concept* as a prediction for the outcome of a game. We give two such examples, named after John Forbes Nash jr., but many more exist in the literature (see, e.g., [140]).

Definition 1.3. Let $\Gamma = (N, (\mathcal{S}_i)_{i\in N}, (C_i)_{i\in N})$ be a strategic game. A *pure Nash equilibrium* is a strategy profile $s = (s_1, \ldots, s_n) \in \times_i \mathcal{S}_i$ with the property that for every $i \in N$, we have

$$C_i(s) \leq C_i(s_{-i}, s_i') \tag{1.1}$$

for all $s_i' \in \mathcal{S}_i$.

A pure Nash equilibrium is a strategy profile in which no player has an incentive to unilaterally deviate to another strategy, i.e., no player can strictly decrease her cost by switching to another strategy.

As mentioned before, players may also be allowed to randomize over strategies, which we now formalize. A *mixed strategy* of player i is a probability distribution over $\mathcal{S}_i$. A collection of mixed strategies $\sigma = (\sigma_1, \ldots, \sigma_n)$ is called a *mixed strategy profile*, where σ_i is the mixed strategy of player i that is independent of the mixed strategies of the other players. We use $\mathbb{E}_{t\sim\sigma}[C_i(t)]$ to denote the *expected cost* of the random variable $C_i(t)$ under strategy profile $t = (t_1, \ldots, t_n)$ where $t_i \sim \sigma_i$ for $i \in N$.[5] Also, we use $\mathbb{E}_{s_{-i}\sim\sigma_{-i}}[C_i(s_{-i}, s_i')]$ to denote the expected cost of player i when choosing the strategy s_i' with probability one (where the expectation is taken with respect to the random choices of the other players).

Definition 1.4. Let $\Gamma = (N, (\mathcal{S}_i)_{i\in N}, (C_i)_{i\in N})$ be a strategic game. A *mixed Nash equilibrium* is a mixed strategy profile σ with the property that for every $i \in N$, we have

$$\mathbb{E}_{s\sim\sigma}[C_i(s)] \leq \mathbb{E}_{s_{-i}\sim\sigma_{-i}}[C_i(s_{-i}, s_i')]$$

for all $s_i' \in \mathcal{S}_i$,

A pure Nash equilibrium is a mixed Nash equilibrium where each player chooses precisely one strategy with probability one, and all her other strategies with probability zero. We next focus on three fundamental aspects of Nash equilibria: existence, computation and inefficiency. In particular, the computation and inefficiency aspects are two of the main topics of interest in algorithmic game theory (as opposed to classical game theory).

1.2.1 Existence

The following fundamental theorem in game theory is due to Nash [137]. Its proof relies on a fixed-point theorem argument.

Theorem 1.5 ([137])**.** *Every finite strategic game has a mixed Nash equilibrium.*

Unfortunately, this result does not hold true for pure Nash equilibria, as can been seen from the classical *matching pennies game* given in Example 1.6. Moreover, in general, mixed Nash equilibria do not have to be unique.

[5] Here $t \sim \sigma$ indicates that t is a random variable with distribution σ.

Example 1.6 (Matching pennies)**.** The matching pennies game is played between two players 1 and 2. Both players have the strategy set $\{H, T\}$ corresponding to *heads (H)* and *tails (T)*. The cost functions of the players are described in the matrix in Table 1.1. For example, if we consider the strategy profile $s = (H, T)$ then $C_1(s) = 0$ and $C_2(s) = 1$. It is not hard to see that the

1 \ 2	H	T
H	1, 0	0, 1
T	0, 1	1, 0

Table 1.1: Costs for the game of matching pennies.

game does not possess a pure Nash equilibrium. Player 1 would like to choose a strategy different from that of player 2, whereas player 2 prefers to choose the same strategy as player 1. On the other hand, a mixed Nash equilibrium is given by the mixed strategy profile in which both players randomize (with equal probability) over both strategies.

Although a pure Nash equilibrium is in general not guaranteed to exist, many special classes of games do have a pure equilibrium. Very often, existence is shown by proving that a strategic game is a so-called *potential game*.

Definition 1.7. A finite strategic game $\Gamma = (N, (\mathcal{S}_i)_{i \in N}, (C_i)_{i \in N})$ is an *(exact) potential game* if there exists a *potential function* $\Phi : \times_i \mathcal{S}_i \to \mathbb{R}$ with the property that

$$C_i(s) - C_i(s_{-i}, s_i') = \Phi(s) - \Phi(s_{-i}, s_i')$$

for every strategy profile $s \in \times_i \mathcal{S}_i$ and every unilateral deviation $s_i' \in \mathcal{S}_i$ for every $i \in N$.

It is not hard to see that every potential game possesses a pure Nash equilibrium: with every unilateral deviation that improves the cost of some player, the potential function decreases in value. As there are only finitely many strategy profiles, there must exist some strategy profile satisfying the definition of a pure Nash equilibrium. Such a sequence of unilateral deviations is called a *best response sequence* if every player deviates to a strategy that improves her cost as much as possible. The congestion games considered in Chapter 3 are exact potential games. The reverse is also true: every exact potential game is 'isomorphic' to such a congestion game, see [134].

1.2.2 Computation

Given the existence of a mixed Nash equilibrium, can we compute one *efficiently*, i.e., in polynomial time? In general, the answer to this question is believed to be negative. In particular, computing a mixed Nash equilibrium is a complete problem for the complexity class PPAD. Note that the problem cannot be NP-complete as NP is a class of decision problems. Theorem 1.5 tells us that a mixed equilibrium always exists, so the decision problem always has the answer YES. We refer the reader to, e.g., [140] for details and further references.

For the computation of a pure Nash equilibrium in potential games there is also bad news, when interpreted as a so-called *local search problem.* The existence of a potential function naturally gives rise to the following algorithm for finding a pure Nash equilibrium: choose some initial strategy profile, and, as long as there is some player that can improve its cost by unilaterally deviating to some other strategy, let this player deviate. We know that this process terminates as the potential function decreases in every step of this procedure. This puts the computation of a pure Nash equilibrium, in exact potential games, in the complexity class PLS of *polynomial local search* problems. The problem of finding a pure Nash equilibrium is known to be complete for this class.

Remark 1.8. Although computing both pure and mixed Nash equilibria are believed to be hard problems in their respective complexity classes, many positive results are known for special games. In particular, we give some unifying results in Chapter 3 for the polynomial time computation of pure Nash equilibria in congestion games with some combinatorial structure.

1.2.3 Inefficiency

How 'inefficient' are Nash equilibria with respect to some 'centralized outcome' of a strategic game? In order to quantify this question we need the notion of a *social cost function* $C : \times_i \mathcal{S}_i \to \mathbb{R}_{\geq 0}$ assigning a real value to every strategy profile. A *(socially) optimal outcome* is a strategy profile minimizing the social cost function. Note that computing an optimal outcome is just a discrete optimization problem (without any strategic aspects).

How much worse can the social cost of a Nash equilibrium be compared to that of a socially optimal outcome? Two popular notions to quantify this inefficiency, and which have been studied extensively since their introduction, are the *price of anarchy* [117] and the *price of stability* [7]. We define these notions for pure Nash equilibria, but their definitions naturally extend to, e.g., mixed Nash equilibria. The price of anarchy (PoA) and the price of stability (PoS) of a finite strategic game Γ are defined as

$$\mathrm{PoA}(\Gamma) = \frac{\max_{s \in \mathrm{NE}(\Gamma)} C(s)}{\min_{s^* \in \times_i \mathcal{S}_i} C(s^*)} \quad \text{and} \quad \mathrm{PoS}(\Gamma) = \frac{\min_{s \in \mathrm{NE}(\Gamma)} C(s)}{\min_{s^* \in \times_i \mathcal{S}_i} C(s^*)},$$

where $\mathrm{NE}(\Gamma)$ denotes the set of all pure Nash equilibria of the game Γ. That is, the price of anarchy compares the worst Nash equilibrium to an optimal outcome, whereas the price of stability compares the best Nash equilibrium to an optimal outcome. Furthermore, for a *class (or collection) of games* $\mathcal{H}$ we define

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

Note that both the price of anarchy and price of stability are worst-case notions for a class of games.

A powerful tool when studying the inefficiency of equilibria is the *smoothness framework* formalized by Roughgarden [155].

Definition 1.9 ([155]). Let $\Gamma = (N, (\mathcal{S}_i)_{i\in N}, (C_i)_{i\in N})$ be a finite strategic game and C a social cost function. For $\lambda \geq 0$ and $\mu < 1$, we say that Γ is *(λ, μ)-smooth with respect to C* if for all $s, s^* \in \times_i \mathcal{S}_i$, we have

$$\sum_{i\in N} C_i(s_{-i}, s_i^*) \leq \mu C(s) + \lambda C(s^*). \tag{1.2}$$

Now, suppose that the social cost C is defined as $C(s) = \sum_{i\in N} C_i(s)$. If s is a pure Nash equilibrium and s^* a socially optimal outcome with respect to C, then it follows that (see [155])

$$C(s) = \sum_{i\in N} C_i(s) \leq \sum_{i\in N} C_i(s_{-i}, s_i^*) \leq \mu C(s) + \lambda C(s^*).$$

Here we use the Nash conditions in (1.1) for the first inequality, and (1.2) in the second inequality. Rewriting gives

$$\frac{C(s)}{C(s^*)} \leq \frac{\lambda}{1-\mu}. \tag{1.3}$$

Showing the existence of a feasible combination of λ and μ is called a *smoothness argument*. Note that, as the bound of $\lambda/(1-\mu)$ applies to any pure Nash equilibrium, we obtain a bound on the price of anarchy.

The main feature of this approach is that the bound of $\lambda/(1-\mu)$ on the price of anarchy for pure Nash equilibria automatically extends to a bound on the price of anarchy for mixed Nash equilibria, as well for more general solution concepts like *(coarse) correlated equilibria* (see, e.g., [140] for a definition). Many price of anarchy analyses in the literature can be cast in the form of a smoothness argument [155]. In particular, these arguments often provide *tight* price of anarchy bounds for classes of finite strategic games (although not always [155]).

1.3 Congestion models

We informally describe various *congestion models* in order to provide some context for the models we study in Chapters 2 and 3. A congestion model is defined by a tuple $(N, E, (\mathcal{S}_i)_{i\in N}, (c_e)_{e\in E}, (d_i)_{i\in N})$, where N is a set of players, E a ground set of *resources*, and $\mathcal{S}_i \subseteq 2^E$ for all $i \in N$, i.e., every strategy is a subset of resources. Resources are equipped with a cost function $c_e : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, and, finally, the quantity $d_i \in \mathbb{N}$ for $i \in N$ is the *demand* of player i.

The goal of player i is to spread out her demand over the available strategies in $\mathcal{S}_i$. Let d_i^e be the total demand of player i assigned to strategies that contain resource e. The cost of player i in strategy profile s is then defined as

$$C_i(s) = \sum_{e \in E} d_i^e c_e(x_e),$$

where $x_e = \sum_{i=1}^n d_i^e$ is the total demand assigned to resource e by all players. The main difference between the models described below, is the way in which players are allowed to 'spread out' their demand over different strategies, as well as whether or not they have a finite or *infinitesimally small* demand (for the latter setting we need a slightly different definition of a congestion model). We will address for all models the question of whether or not a game is guaranteed to possess a pure Nash equilibrium, that is, under the natural definition of a pure Nash equilibrium for the respective model.

- **Finite unsplittable demand.** Here every player $i \in N$ has to assign her full demand $d_i \in \mathbb{N}$ to one of her strategies in $\mathcal{S}_i$. These games are also known as *unsplittable atomic congestion games.* There are two important cases in the literature, depending on the demands of the players.

  - *Weighted case.* For general demands d_i, a description of this model can be found in, e.g., [132], where it is also shown that in general a pure Nash equilibrium is not guaranteed to exist. For certain special classes, with some additional (combinatorial) structure on the strategy sets, a pure Nash equilibrium does exist, see, e.g., [2].
  - *Unweighted case.* If all players have unit demand, i.e., $d_i = 1$ for all $i \in N$, we obtain the model in [150], which is studied in Chapter 3 of this thesis. As mentioned earlier, these games are so-called potential games, and therefore a pure Nash equilibrium is always guaranteed to exist [150].

  Both these models can be seen as strategic games as in Definition 1.1.

- **Finite splittable demand.** Here every player $i \in N$ is allowed to divide her demand over multiple strategies. There are three important special cases in the literature, depending on the way in which the demand may be spread out over different paths.

  - *Integer-splittable case.* Here player $i \in N$ can divide her demand d_i over multiple strategies, under the restriction that every strategy gets assigned an integral amount of demand, see, e.g., [151]. In general a pure Nash equilibrium is not guaranteed to exist [151], although there exist special classes of games where this is the case, see, e.g., [169, 100] and references therein.
  - *Atomic splittable case.* Here every player $i \in N$ is allowed to divide her demand d_i over multiple strategies in an arbitrary way, non-integral [143]. A pure Nash equilibrium is guaranteed to exist, e.g., when the

cost functions are continuous and convex: this follows from a fixed point theorem of Kakutani [112].

- *k-splittable case.* Here every player $i \in N$ is allowed to divide her demand d_i over at most k strategies, see, e.g., [24]. This model serves as an interpolating version of unsplittable and splittable atomic congestion games, for respectively $k = 1$ and k sufficiently large. In general, a pure Nash equilibrium is not guaranteed to exist, as the case $k = 1$ corresponds to the unsplittable weighted case given above.

- **Infinitesimally small demand.** Here, instead of a finite set of players N, we are given a finite number of *commodities* $j = 1, \dots, q$ and every commodity j contains a *continuum* of players represented by the interval $[0, p_j]$ where $p_j > 0$ is the size of the continuum of commodity j. The players of commodity j have finite strategy set $\mathcal{S}_j \subseteq 2^E$ for $j = 1, \dots, q$. A strategy profile f here is a collection of non-negative functions $f_j : \mathcal{S}_j \to \mathbb{R}_{\geq 0}$ with the property that $\sum_{s \in \mathcal{S}_j} f_j(s) = p_j$ for $j = 1, \dots q$. A pure Nash equilibrium is guaranteed to exist under some mild assumptions on the cost functions, see, e.g., [159]. When the strategy sets represent paths in a given directed network, we obtain Wardrop's routing model that we study in Chapter 2.[6]

1.4 Sampling and counting

We will illustrate the notion of sampling[7] and counting using the example of a perfect matching in a graph. We choose this example for multiple reasons. Sampling and counting perfect matchings is a generalization of the problem of sampling and counting graphs with given degrees, which will be explained later on. Furthermore, it is a classical example in the field of sampling and counting. Finally, in Chapter 4 we rely on various ideas introduced in the context of sampling perfect matchings.

1.4.1 Perfect matchings

Given an undirected graph $G = (V, E)$ with $|V| = n$, a *perfect matching* $M \subseteq E$ is a set of edges such that every node in V is adjacent to precisely one edge in M, i.e., for every $v \in V$ it holds that $\{v\} \cap \{x, y\} \neq \emptyset$ for precisely one $\{x, y\} \in M$. The set of all perfect matchings of the graph G is denoted by $\mathcal{P}_G$.

Four fundamental problems, in non-decreasing order of difficulty,[8] concerning

[6] Interestingly, this model can also been seen as the limiting case of an atomic (splittable) congestion game, see, e.g., [101].

[7] In this thesis *(approximate) sampling* always refers to *(approximately) uniform sampling*, i.e., generating an element from a finite set according to the uniform distribution over the set.

[8] The only non-trivial claim here is the fact that counting is not easier than sampling. For a standard reduction in the case of perfect matchings, see [111]. For general finite objects it is believed that counting is more difficult than sampling. Evidence for this claim follows,

perfect matchings in a given graph G are the following.

- *Existence*: Is $\mathcal{P}_G \neq \emptyset$?
- *Construction*: Can we compute a perfect matching $H \in \mathcal{P}_G$, or decide that $\mathcal{P}_G = \emptyset$, in polynomial time?
- *Sampling*: Can we generate in polynomial time a perfect matching uniformly at random from $\mathcal{P}_G$?
- *Counting*: Can we compute the number $|\mathcal{P}_G|$ in polynomial time?

The first two problems are well-understood in mathematics and computer science: Edmond's *blossom algorithm* [63] can be used to compute a perfect matching in polynomial time, or decide that no perfect matching exists. However, it is not known what the answers to the sampling and counting problems are. In particular, the problem of exactly computing the number of perfect matchings is a complete problem in the (counting) complexity class $\#\mathsf{P}$ [172]. Exact counting is therefore believed to be a hard problem, and, thus, attention has shifted to *approximate counting*. More precisely, we are interested in a so-called *fully polynomial randomized approximation scheme* for computing the number of perfect matchings in a given graph.

Definition 1.10. Let G be an undirected graph on n nodes. A *fully polynomial randomized approximation scheme (FPRAS)* for counting the number of perfect matchings in G is a randomized algorithm that, for every $\epsilon,\ \delta > 0$, outputs the number of perfect matchings up to a multiplicative factor $(1 \pm \epsilon)$ with probability at least $1 - \delta$, in time polynomial in n, $1/\epsilon$ and $\log(1/\delta)$.

It is well-known that for perfect matchings, and more generally for so-called *self-reducible problems*, see, e.g., [111], the existence of an FPRAS is equivalent to the existence of an *approximate sampler*, or so-called *fully polynomial almost uniform sampler*. Roughly speaking, an approximate sampler for the (approximate) uniform sampling of perfect matchings of a given graph G is a randomized algorithm that outputs every perfect matching $M \in \mathcal{P}_G$ with probability *close to* $1/|\mathcal{P}_G|$. That is, the output distribution of the algorithm should be close to the uniform distribution over $\mathcal{P}_G$. In order to quantify this notion of closeness, the *total variation distance* is used as a measure for the distance between two probability distributions.[9] The total variation distance between two probability distributions q, q' over $\mathcal{P}_G$ is given by

$$d_{TV}(q, q') = \frac{1}{2} \sum_{M \in \mathcal{P}_G} |q(M) - q'(M)|.$$

e.g., from a comparison between (a corollary of) Toda's theorem [168] and results of Bellare, Goldreich and Petrank [13].

[9] There exist many choices to quantify the distance between two probability distributions, see, e.g., the survey of Gibbs and Su [91]. "Closeness" in total variation distance implies closeness in many other distance metrics.

Definition 1.11. A *fully polynomial almost uniform sampler (FPAUS)* for perfect matchings is a randomized algorithm that, with probability at least $1-\delta$ in time polynomial in n, $\log(1/\epsilon)$ and $\log(1/\delta)$ outputs a perfect matching from the set of all perfect matchings $\mathcal{P}_G$ according to a distribution $\tilde{u}$ with total variation distance at most ϵ from the uniform distribution u over $\mathcal{P}_G$.

It is interesting to note that the existence of an FPAUS/FPRAS for the problem of approximate sampling/counting perfect matchings in a general undirected graph G is still an open problem. For bipartite graphs the existence has been shown by Jerrum, Sinclair and Vigoda [110] in their work on the approximation of the permanent[10] of a non-negative matrix. Remember that an undirected graph $G=(V,E)$ is *bipartite* if there exists a partition $V = A \cup B$ so that every $e = \{a,b\} \in E$ has one endpoint in A and one endpoint in B, i.e., $A \cap e \neq \emptyset$ and $B \cap e \neq \emptyset$.

Remark 1.12 (Tutte's construction [171]). The problem of *sampling graphs with given degrees*, that we consider in Chapter 4, is a special case of sampling perfect matchings in a general undirected graph. Here we are given a sequence of non-negative integers $d = (d_1,\dots,d_n)$ and the goal is to sample a simple undirected labelled graph G with degree sequence d from the set $\mathcal{G}(d)$ of all such graphs G.

The fact that this problem is a special case of the problem of sampling perfect matchings in a given undirected graph follows from a reduction due to Tutte [171] that we describe next. Let $d = (d_1,\dots,d_n)$ be a sequence of non-negative integers. We construct the auxiliary graph $T(d) = (V,E)$ as follows. We let

$$V = \{(i,j) : i,j \in [n], i \neq j\} \cup \big(\cup_{i\in[n]} V_i \big),$$

where $V_i = \{v_i^1,\dots,v_i^{d_i}\}$. Moreover, we have

$$E = \Big\{ \{(i,j),(j,i)\} : i,j \in [n], i \neq j \Big\} \cup \big(\cup_{i\in[n]} E_i \} \big),$$

where $E_i = \{\{v_i^k,(i,j)\} : j \in [n]\setminus\{i\}, k = 1,\dots,d_i\}$. It then follows that every perfect matching in $T(d)$ corresponds to a graph G with degree sequence d, where $\{a,b\}$ is an edge in G if and only if $\{(a,b),(b,a)\}$ is *not* an edge in the perfect matching. An example is given in Figure 1.4.

In particular, for every graph G there are precisely $\Pi_i d_i!$ perfect matchings in $T(d)$ corresponding to it. This can be seen as follows. Let $i \in [n]$ and subset $J \subseteq [n]\setminus\{i\}$ be given. If there is a perfect matching M in $T(d)$ corresponding to a graphical realization G, that in particular forms a perfect matching between the nodes in V_i and $\{(i,j) : j \in J\}$, then any other perfect matching M' obtained

[10] The permanent of an $n \times n$-matrix A is defined as $\text{per}(A) = \sum_{\pi\in S_n} \Pi_{i=1}^n a_{i\pi(i)}$ where S_n denotes the set of all permutations π of $\{1,\dots,n\}$. If A is a matrix with only entries in $\{0,1\}$ then the permanent corresponds to the number of perfect matchings in the bipartite graph with adjacency matrix A.

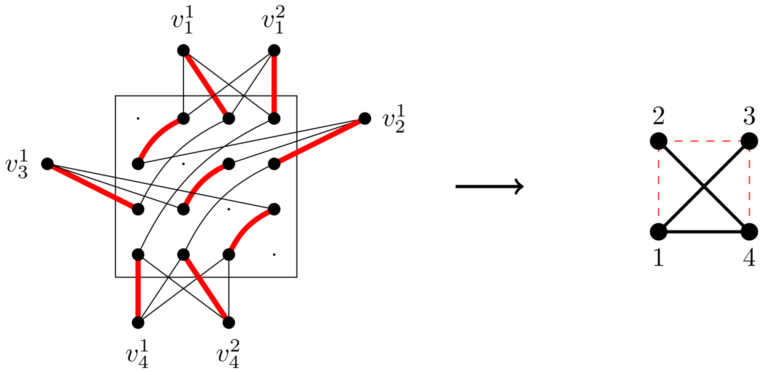

Figure 1.4: Tutte's construction $T(d)$ for $d = (2, 1, 1, 2)$ where the square contains all nodes corresponding to the pairs (i, j) for $1 \leq i \neq j \leq 4$ in the natural (matrix) way: the left bottom corner point corresponds to $(4, 1)$ and the right top corner point to $(1, 4)$. On the right we see the graphical realization G (solid black edges) corresponding to the (red) perfect matching in Tutte's construction. Note that the dashed red edges in the square correspond to the edges in the complement of G.

by taking a different (perfect) matching between V_i and $\{(i, j) : j \in J\}$ in M corresponds to the same graphical realization G. For example, in Figure 1.4, if we replace the edges $\{v_1^1, (1, 3)\}$ and $\{v_1^2, (1, 4)\}$ with $\{v_1^1, (1, 4)\}$ and $\{v_1^2, (1, 3)\}$, then the resulting perfect matching in $T(d)$ still corresponds to the graphical realization G in Figure 1.4 on the right. This implies that, with $\mathcal{G}(d)$ denoting the set of all graphical realizations of the sequence d,

$$\Pi_i d_i! \cdot |\mathcal{G}(d)| = |\mathcal{P}_{T(d)}|.$$

These observations are sufficient to argue that, in order to (approximately) uniformly sample a graph with degree sequence d, it suffices to (approximately) uniformly sample a perfect matching from $T(d)$. To see this, note that the probability of obtaining a perfect matching corresponding to a given graph $G \in \mathcal{G}(d)$ is $\Pi_i d_i!/|\mathcal{P}_{T(d)}|$, which is independent of G (as all graphical realizations have degree sequence d). Moreover, note that the size of Tutte's construction $T(d)$ is quadratic in n. A similar relation holds for the counting problem.[11]

Remark 1.13 (Exact matching). A generalization of the perfect matching problem is the *exact matching* problem [144]. Here we are given an undirected graph with every edge colored red or blue, and we are interested in perfect matchings with exactly k red edges for $k \in \mathbb{N}$. A randomized polynomial time algorithm

[11]Note that $\ln(|\mathcal{P}_{T(d)}|)$ is at most a polynomial factor larger than $\ln(|\mathcal{G}(d)|)$, which is also necessary for the reduction to be valid.

is known for the construction problem [136], but it remains an open problem if a deterministic polynomial time algorithm exists. We refer to [15] for some sampling and counting results regarding exact matchings. Some of the techniques developed in [15] are used in Chapter 4.

1.4.2 Markov Chain Monte Carlo method

One of the most successful approaches for designing an approximate sampler for the sampling of finite objects is the *Markov Chain Monte Carlo (MCMC) method.*

A Markov chain $\mathcal{M} = (\Omega, P)$ is a process that moves between the states of a (finite) set Ω, called the *state space*, where the probability of *transitioning* to state y, given that the chain is in state x, is given by $P(x, y)$.[12] A Markov chain is called *irreducible* if for every pair of states x and y, it is possible to reach y from x with strictly positive probability in a finite number of transitions in $\mathcal{M}$; it is called *aperiodic* if the greatest common divisor of the set $\mathcal{T}(x) = \{t \geq 1 : P^t(x, x) > 0\}$ equals one for all $x \in \Omega$. Moreover, the chain $\mathcal{M}$ is called *reversible with respect to a probability distribution* π if the *detailed balanced equations* in (1.4) are satisfied:

$$\pi(x)P(x, y) = \pi(y)P(y, x), \qquad \forall x, y \in \Omega. \tag{1.4}$$

It is well-known that if an irreducible, aperiodic Markov chain $\mathcal{M}$ is reversible with respect to distribution π, then (the row vector) π is its (unique) *stationary distribution*, i.e.,

$$\pi = \pi P. \tag{1.5}$$

Roughly speaking, if we simulate the Markov chain for a (very) large number of steps, then $\pi(x)$ is the probability that the chain will be in state $x \in \Omega$ independent of where the process started.

The idea of the MCMC method is to *design* an irreducible Markov chain on the set of all finite objects of interest in such a way that the stationary distribution is precisely the uniform distribution over the set of objects. This is usually done by designing a *symmetric Markov chain* for which $P(x, y) = P(y, x)$ for all $x, y \in \Omega$. The detailed balance equations then directly imply that the chain is reversible with respect to the uniform distribution, as desired. The goal is to show that the Markov chain *mixes rapidly*, meaning that we only have to simulate it for a polynomial number of steps in order to get an output that is 'close' to the (stationary) uniform distribution. Full details are given in Chapter 4, here our objective is to only explain the idea.

Often, the state space Ω is augmented to a set Ω' consisting of Ω and a set of auxiliary states, as it is sometimes easier to design a Markov chain on an augmented set Ω' than it is on Ω directly. For example, in the case of perfect matchings, the state space is usually augmented with the set of all *near-perfect matchings*, which are matchings of size $n/2 - 1$. In order to get a sample from Ω, we can then use *rejection sampling*: repeatedly compute samples from Ω' until

[12] For formal definitions, see, e.g., [118].

one obtains a sample that lies in Ω. Roughly speaking, we only need a polynomial number of samples for this procedure if Ω' is at most a polynomial factor larger than Ω.[13]

The MCMC method has been successfully applied for the design of an FPAUS (which is in turned used for the design of an FPRAS) for many finite objects, such as perfect matchings in dense undirected graphs [108] and general bipartite graphs [110]; Hamilton cycles in dense undirected graphs [59]; 0-1 knapsack solutions [135]; and contingency tables with a constant number of rows [47]. We refer the reader to the respective references for a description of these problems. Moreover, for some of these problems there also exist approaches not using the Markov Chain Monte Carlo method in order to obtain an FPRAS.

1.5 Overview and publications

This thesis continues with three chapters. All chapters have a similar structure. We start with an introduction of the problem and the model under consideration, followed by all the necessary technical preliminaries.[14] Then, in the subsequent sections the main results are presented. For the reader interested in only obtaining an overview of the results, it should be sufficient to read the introductory parts (assuming familiarity with the model and problems under discussion).

We next give an overview of the publications, either in journals or (peer-reviewed) conference proceedings, that the Chapters 2, 3 and 4 are based on, together with a short description.

In Chapter 2 we study the effect of deviations (or perturbations) on the latency functions in non-atomic network routing games [175]. In particular, we are interested in how these perturbations affect the quality of a Nash equilibrium. The main result of this section is a tight analysis of this quality deterioration for routing games on common-source multi-commodity network topologies. This builds on, and extends, the work of Nikolova and Stier-Moses [139] who study risk aversion in non-atomic network routing games. Chapter 2 is based on the following two publications.

- Pieter Kleer and Guido Schäfer: The impact of worst-case deviations in non-atomic network routing games. *Theory of Computing Systems,* 63(1): 54–89, 2019.
- Pieter Kleer and Guido Schäfer. Path deviations outperform approximate stability in heterogeneous congestion games. *Lecture Notes in Computer*

[13] There are ways to overcome this issue when there is no polynomial relation between Ω and Ω'. For example, this is one of the crucial points in the work of Jerrum, Sinclair and Vigoda [110] on sampling perfect matchings in bipartite graphs.

[14] There is sometimes minor overlap with the content discussed in the current chapter, for self-containment.

Science (LNCS), 10504:212–224, 2017. *Proceedings of SAGT 2017.*

In Chapter 3 we focus on Rosenthal's congestion game model [150]. We provide a unified framework for the polynomial time computation of *good* pure Nash equilibria for games in which the strategy sets have some combinatorial structure. That is, we show that a Nash equilibrium can be computed with an inefficiency guarantee that is strictly better than that of an arbitrary Nash equilibrium (in the form of a tight price of stability bound). We also provide a unified framework for various extensions of Rosenthal's model that have been proposed in recent years. Chapter 3 is based on the following two publications.

- Pieter Kleer and Guido Schäfer. Tight inefficiency bounds for perception-parameterized affine congestion games. *Theoretical Computer Science*, 754: 65-87, 2019.

- Pieter Kleer and Guido Schäfer. Potential function minimizers of combinatorial congestion games: Efficiency and computation. *Proceedings of the 18th ACM Conference on Economics and Computation*, pages 223–240, 2017.

In Chapter 4 we study the switch Markov chain for the approximately uniform sampling of graphs with given degrees. We provide a novel proof technique for showing rapid mixing of the switch Markov chain that unifies many results in the literature. We also use this novel proof idea to make some progress on a related sampling problem. Chapter 4 is based on the following two publications.

- Georgios Amanatidis and Pieter Kleer. Rapid mixing of the switch Markov chain for strongly stable degree sequences and 2-class joint degree matrices. *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms,* pages 966–985, 2019.

- Corrie Jacobien Carstens and Pieter Kleer. Speeding up switch Markov chains for sampling bipartite graphs with given degree sequence. *Leibniz International Proceedings in Informatics (LIPIcs),* 116(36):1–18, 2018. *Proceedings of APPROX-RANDOM 2018.*

Chapter 2

Worst-case latency deviations in non-atomic routing games

2.1 Introduction

In 1952, Wardrop [175] introduced a simple model to study outcomes of selfish route choices in traffic networks which are affected by congestion. In this model, there is a continuum of non-atomic players, each controlling an infinitesimally small amount of flow, whose goal is to choose a path in a given network to minimize their own travel time. The latency (or delay) of each edge is prescribed by a non-negative, non-decreasing latency function which depends on the total flow on that edge. Ever since its introduction, Wardrop's model has been used extensively, in operations research, algorithmic game theory, and traffic engineering studies, to investigate various aspects of selfish routing in networks.

An important topic of interest concerns the inefficiency of selfish outcomes, so-called *Wardrop (or Nash) flows*. These are equilibrium situations in which no player has an incentive to deviate to a different path. A popular approach for quantifying the inefficiency is by means of the *price of anarchy*, that compares the total latency of all players in a Nash flow to that of a *socially optimal outcome*. The latter is a flow that minimizes the total latency of all players over all feasible flows in the network. In a seminal work, Roughgarden and Tardos [156] show that for non-negative affine[1] latency functions the total latency of a Wardrop flow can exceed that of a socially optimal flow by a multiplicative factor of at most $\frac{4}{3}$. This bound is tight already for the well-known *Pigou network* consisting of two parallel edges between two nodes, essentially showing that the price of anarchy is *independent of the network topology.* That is, there exists an upper bound that holds for all instances with affine latency functions, and it is attained already on the smallest non-trivial network topology. The result of [156] was later extended to more general classes of functions [152, 46] in terms of a so-called

[1] A function $l : \mathbb{R} \to \mathbb{R}$ is *affine* if $l(x) = ax + b$ for some $a, b \in \mathbb{R}$.

smoothness parameter, as in the approach sketched in Section 1.2.3.[2] Given the possible inefficiency of Wardrop flows, a natural question that arises is how this phenomenon might be alleviated. One way to do this is by means of network design.

In the *network design problem* the goal is to find a subgraph of the given network for which the common[3] total latency of all Wardrop flows is minimal. This problem is inspired by the *Braess paradox* [18], that shows that removing an arc from a network can improve the quality of the resulting Wardrop flow. Roughgarden [154] shows that the *trivial algorithm*, which simply returns the original network, gives an $\lfloor n/2 \rfloor$-approximation algorithm for single-commodity networks and that this is best possible, unless $\mathsf{P} = \mathsf{NP}$. In order to show tightness of the approximation guarantee, Roughgarden [154] introduces the class of *generalized Braess graphs* that play an important role in this work. This class forms a generalization of the *Braess graph*, originally used by Braess to illustrate the Braess paradox.

More recently, Wardrop's classical model has been extended in various ways to capture more complex player behaviors. Examples include the incorporation of uncertainty attitudes (e.g., risk-aversion, risk-seeking), cost alterations (e.g., latency perturbations, road pricing), other-regarding dispositions (e.g., altruism, spite) and player biases (e.g., responsiveness, bounded rationality). Several of these extensions can be viewed as defining some modified cost for each path which combines the original latency with some 'deviation' (or perturbation) along that path. The player objective then becomes to minimize the combined cost of latency and deviation along a path. The deviations might be given explicitly, e.g., as in the altruism model of Chen et al. [28]; or be defined implicitly, e.g., as in the risk-aversion model of Nikolova and Stier-Moses [139]. Furthermore, players might perceive these deviations differently, i.e., players might be *heterogeneous* with respect to the deviations.

The main purpose of this work is to study how much the quality of a *deviated (or perturbed) Nash flow* deteriorates in the worst case under bounded deviations of the latency functions. There are two natural choices two quantify this inefficiency. One can again compare the deviated Nash flow to a socially optimal flow, as is done, e.g., in the behavioral bias model of Meir and Parkes [129]. This quantity corresponds to the *biased price of anarchy* [129]. Secondly, one can compare the deviated Nash flow to a classical Nash flow in which there are no deviations, as is done in, e.g., the non-atomic risk aversion model of Nikolova and Stier-Moses [138, 139]. This quantity is called the *price of risk aversion (PRA)*. In turns out that both these quantities can be characterized in completely different ways. For the biased price of anarchy one can generalize the proof techniques for the anal-

[2]When bounding the inefficiency of a Nash flow with respect to an optimal outcome, some assumptions on the latency functions have to be made. In general the ratio can be unbounded already on the Pigou network.

[3]It is a well-known fact that all Wardrop flows have the same total latency, see, e.g., [140].

ysis of the price of anarchy [156, 152, 46], and provide bounds characterized by the latency functions, again showing that this quantity is essentially independent of the network topology. On the other hand, the price of risk aversion heavily depends on the network topology, and can be characterized independent of the latency functions [139, 120].

In this work we introduce and quantify the *deviation ratio (DR)* that compares a perturbed Nash flow to an unperturbed Nash flow. This notion is inspired by, and builds on, the price of risk aversion of Nikolova and Stier-Moses [139]. From a technical point of view these notions are roughly the same. The reason for introducing a new ratio, is that the ideas introduced in [139] for the analysis of the price of risk aversion are not inherent to the notion of risk aversion, but can be applied more broadly. We next introduce some notation in order to explain our contributions. Formal definitions can be found in Section 2.2.

Given an instance of a selfish routing game with latency functions $(l_a)_{a \in A}$ on the arcs, we define the *deviation ratio (DR)* as the worst case ratio $C(f^\delta)/C(f^0)$ of the social cost of a Nash flow f^δ with respect to deviated latency functions $(l_a + \delta_a)_{a \in A}$, where $(\delta_a)_{a \in A}$ are arbitrary deviation functions from a feasible set of bounded deviations, and the social cost of a Nash flow f^0 with respect to the unaltered latency functions $(l_a)_{a \in A}$. Here the social cost function C refers to the total average latency (without the deviations). Our motivation for studying this social cost function is that a central designer usually cares about the long-term performance of the system accounting for the average latency (or pollution). On the other hand, the players typically do not know the exact latencies and use estimates or include 'safety margins' in their planning. Similar viewpoints are adopted in [129, 139].

2.1.1 Our contributions

In order to model bounded deviations, we extend an idea previously put forward by Bonifaci, Salek and Schäfer [17] in the context of the *restricted network toll problem*: We assume that for every arc $a \in A$ we are given (flow-dependent) lower and upper bound restrictions $\theta_a^{\min}$ and $\theta_a^{\max}$, respectively, and call a deviation δ_a *feasible* if $\theta_a^{\min}(x) \leq \delta_a(x) \leq \theta_a^{\max}(x)$ for all $x \geq 0$. Our contributions mostly apply to a specific class of deviations, which we term (α, β)*-deviations.* Here the latency restrictions are of the form $\theta_a^{\min} = \alpha l_a$ and $\theta_a^{\max} = \beta l_a$ with $-1 < \alpha \leq 0 \leq \beta$.

1. In Section 2.3, we show that for (α, β)-deviations the deviation ratio is at most
$$1 + \frac{\beta - \alpha}{1 + \alpha} \left\lceil \frac{n-1}{2} \right\rceil r, \tag{2.1}$$
where n is the number of nodes of the network and r is the sum of the demands of the commodities (Theorem 2.5). In particular, this reveals that

the deviation ratio depends linearly on the size of the underlying network (among other parameters).

In order to prove this bound, we first generalize a result by Bonifaci et al. [17], characterizing the inducibility of a fixed flow by δ-deviations, to multi-commodity networks with a common source (Theorem 2.7). This characterization naturally gives rise to the concept of an *alternating path*, which also plays a crucial role in the work by Nikolova and Stier-Moses [139]. By using this result, we obtain a bound on the price of risk aversion (Theorem 2.27) which generalizes the one in [139] (see Section 2.7.1 for a description of their model). Our bound generalizes their result in two ways: (i) it holds for multi-commodity networks with a common source, and (ii) it also holds for negative risk-aversion parameters (capturing *risk-taking* players). Further, we show that our result can be used to bound the relative error in social cost of Nash flows incurred by small latency perturbations (Theorem 2.28), which is of independent interest.

2. In Section 2.4, we prove that our bound on the deviation ratio for (α, β)-deviations is best possible for multi-commodity networks with a common source. We also show that it does not extend to general multi-commodity networks. To be more precise, we show the following.
   - For *single-commodity networks* we show that our bound is tight in *all* its parameters. Our lower bound construction holds for arbitrary $n \in \mathbb{N}$ and is based on the *generalized Braess graph* [154] (Example 2.13). In particular, this also complements a result by Lianeas, Nikolova and Stier-Moses [119] who show that the upper bound on the price of risk aversion in [139] is tight for single-commodity networks with $n = 2^j$ nodes for all $j \in \mathbb{N}$.
   - For *multi-commodity networks with a common source* we show that our bound is tight in all parameters if n is odd, while a small gap remains if n is even (Theorem 2.14).
   - For *general multi-commodity networks* we establish a lower bound showing that the deviation ratio can be exponential in n (Theorem 2.16). In particular, this shows that there is an exponential gap between the cases of multi-commodity networks with and without a common source. In our proof, we adapt a graph structure used by Lin et al. [122] in their lower bound construction for the network design problem on multi-commodity networks.
3. In Section 2.5, we improve (and slightly generalize) *smoothness bounds* on the price of risk aversion given by Meir and Parkes [129] and, independently, by Lianeas et al. [119]. In particular, we derive tight bounds for the *biased price of anarchy (BPoA)* [129] for *arbitrary* $(0, \beta)$-deviations (Theorem 2.19). We show that the biased price of anarchy is at most

$$\frac{1+\beta}{1-\hat{\mu}(\mathcal{L}, \beta)}$$

for latency functions in class $\mathcal{L}$, where $\hat{\mu}$ is a so-called *smoothness parameter* that naturally generalizes that of [45]; see Section 2.5 for details. In particular, for instances with affine latency functions, we obtain a tight bound of

$$\frac{4(1+\beta)^2}{3+4\beta}$$

that naturally generalizes the $\frac{4}{3}$ bound in [156] for $\beta = 0$. Note that the biased price of anarchy always yields an upper bound on the deviation ratio (and price of risk aversion).

4. Finally, in Section 2.6, we also study a more general type of path deviations, that are not necessarily formed by the aggregation of arc deviations, but rather defined as a general real-valued function on the set of all feasible flows in the instance. We provide tight inefficiency bounds for single-commodity instances on *series-parallel* graphs for the deviation ratio with a heterogeneous player population. That is, the population is split into h classes, each with demand $r_j > 0$ for $j \in [h]$ with $\sum_{j\in[h]} r_j = 1$. Every class $j \in [h]$ in the population has its own sensitivity parameter $\gamma_j > 0$ representing to what extent players in a class take into account the deviations. We show that for single-commodity non-atomic network routing games on a series-parallel graph, the deviation ratio is upper bounded by

$$1 + \beta \cdot \max_{j\in[h]} \left\{ \gamma_j \left(\sum_{p=j}^{h} r_p \right) \right\},$$

assuming, without loss of generality, that $\gamma_1 < \gamma_2 < \cdots < \gamma_h$. This bound is tight for arbitrary demand vectors r and sensitivity distributions γ.

Our first two results answer a question posed by Nikolova and Stier-Moses in [139] regarding possible relations between their risk aversion model, the restricted network toll problem [17], and the network design problem [154]. In particular, our results show that the analysis in [139] is not inherent to the used variance function, but rather depends on the restrictions imposed on the feasible deviations.

2.1.2 Related work

The routing model of Wardrop [175] has been studied extensively since its introduction. Basic properties are derived by Beckmann, McGuire and Winsten [12]; see also [153] for more references.

In particular, in [12] it is shown that the social optimum under latencies $(l_a)_{a\in A}$ is a Nash flow with respect to the latency functions $(xl_a(x))' = l_a(x) + xl_a'(x)$ for $a \in A$, assuming that $xl_a(x)$ is differentiable. The terms $xl_a'(x)$ for $a \in A$ are referred to as *marginal tolls.* As the tolls might become arbitrarily large, Bonifaci et al. [17] introduce the *restricted network toll problem*, to which our model is conceptually related, in which the problem is to compute non-negative tolls that have to obey some upper bound restrictions $(\theta_a)_{a\in A}$ such that the cost of the resulting Nash flow is minimized. This is tantamount to computing best-case deviations in our model with $\theta_a^{\min} = 0$ and $\theta_a^{\max} = \theta_a$. The problem is known to be NP-complete, as was shown in a special setting by Hoefer, Olbrich and Skopalik [103], where every arc has $\theta_a^{\max} \in \{0, \infty\}$. In [103], the authors also present a polynomial time algorithm for parallel arc networks, which Bonifaci et al. [17] improved to a polynomial time algorithm for computing the best tolls for networks consisting of parallel arcs. The authors of [17] also present a specific set of toll functions, a scaled version of the marginal tolls described above, and bound the resulting inefficiency of these tolls compared to a socially optimal outcome. See also the work of Fotakis, Kalimeris and Lianeas [81]. These results are also related to price of stability bounds for approximate Wardrop equilibria given by Christodoulou, Koutsoupias and Spirakis [35].

Furthermore, as also mentioned earlier, Roughgarden [154] studies the *network design problem* of finding a subnetwork that minimizes the latency of all flow-carrying paths of the resulting Nash flow. He introduces the *Braess ratio* which relates the common latency of a Nash flow in the original graph to the common latency of a Nash flow in an (optimal) subgraph. He shows that the *trivial algorithm* (which simply returns the original network) gives an $\lfloor n/2 \rfloor$-approximation algorithm for single-commodity networks and that this is best possible, unless $\mathsf{P} = \mathsf{NP}$. Later, Lin et al. [122] show that this algorithm can be exponentially bad for multi-commodity networks. The instances that we use in our lower bound constructions are based on the ones used in [154, 122].

The modeling and study of uncertainties in routing games has received a lot of attention in recent years. An extensive survey on this topic is given by Cominetti [40].

Englert, Franke and Olbrich [66] study the sensitivity of Nash flows in non-atomic network routing games. They investigate the relative change in social cost with respect to two alterations: (i) when the demand is perturbed by an additive constant $\epsilon > 0$, and (ii) when an edge with only an ϵ-fraction of flow is removed. For single-commodity instances with polynomial latency functions of degree at most p, they show that the ratio of the social cost of a Nash flow in the original instance and the social cost of a Nash flow in the instance with demand increased by $\epsilon > 0$, is at most $(1 + \epsilon)^p$. They also show that this bound is tight.

Finally, Cole, Lianeas and Nikolova [38] investigate whether diversity, i.e., a heterogeneous population with respect to deviations, can improve the outcome of selfish routing games (compared to an averaged homogeneous population). In their model, they take into account the deviations in the social cost function. Therefore, although conceptually related, their results do not apply to our setting. Chen et al. [28] study heterogeneous populations in the context of altruism. They derive price of anarchy bounds on instances with parallel-arc network topologies.

Some recent directions in the study of non-atomic routing games concern the quantification of the price of anarchy on a fixed instance as the demand grows large [39], and the computation of Wardrop equilibria for instances with piece-wise linear functions, parameterized by the demand [116].

2.1.3 Outline

In Section 2.2 we present all the necessary preliminaries. We then continue in Section 2.3 with our upper bounds on the deviation ratio, and provide complementing lower bounds in Section 2.4. Tight bounds for the biased price of anarchy are discussed in Section 2.5. In Section 2.6 we provide our bounds for heterogeneous populations. Finally, in Section 2.7 we provide applications of our results, in particular, we discuss the implications for the price of risk aversion [139].

2.2 Preliminaries

In this section, we introduce our bounded deviation model for non-atomic network routing games, we define the deviation ratio, and elaborate on some related notions. We also derive some preliminary results that are used later.

2.2.1 Non-atomic network routing games

An instance of a non-atomic network routing game is given by a tuple $\mathcal{I} = (G = (V, A), (l_a)_{a\in A}, (s_i, t_i)_{i\in[k]}, (r_i)_{i\in[k]})$. Here, $G = (V, A)$ is a directed graph with node set V and arc set $A \subseteq V \times V$, where each arc $a \in A$ has a non-negative, non-decreasing and continuous latency function $l_a : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. Each commodity $i \in [k]$ is associated with a source-destination pair (s_i, t_i) and has a demand of $r_i \in \mathbb{R}_{>0}$. We assume without loss of generality that $t_i \neq t_j$ if $i \neq j$ for $i, j \in [k]$. If all commodities share a common source node, i.e., $s_i = s_j = s$ for all $i, j \in [k]$, we call $\mathcal{I}$ a *common source multi-commodity instance (with source s)*. We assume without loss of generality that $1 = r_1 \leq r_2 \leq \cdots \leq r_k$ and define $r = \sum_{i\in[k]} r_i$.

We denote by $\mathcal{P}_i$ the set of all simple (s_i, t_i)-paths of commodity $i \in [k]$ in G, and we define $\mathcal{P} = \cup_{i\in[k]}\mathcal{P}_i$. An outcome of the game is a feasible flow $f : \mathcal{P} \to \mathbb{R}_{\geq 0}$ such that $\sum_{P\in\mathcal{P}_i} f_P = r_i$ for every $i \in [k]$. The set of all feasible flows f is denoted by $\mathcal{F}$. Given a flow $f = (f^i)_{i\in[k]}$, we use f^i_a to denote the total flow on arc $a \in A$ of commodity $i \in [k]$, i.e., $f^i_a = \sum_{P\in\mathcal{P}_i : a\in P} f_P$. The total flow on arc $a \in A$ is defined as $f_a = \sum_{i\in[k]} f^i_a$. The latency of a path $P \in \mathcal{P}$ with respect to f is defined as $l_P(f) := \sum_{a\in P} l_a(f_a)$.[4] The *social cost* $C(f)$ of a flow f is given by its total latency, i.e.,

$$C(f) = \sum_{P\in\mathcal{P}} f_P l_P(f) = \sum_{a\in A} f_a l_a(f_a).$$

[4]For different objectives, such as ℓ_p-norms, see, e.g., [37].

A flow that minimizes $C(\cdot)$ is called *(socially) optimal*. We use $A_i^+ = \{a \in A : f_a^i > 0\}$ to refer to the support of f^i for commodity $i \in [k]$ and define $A^+ = \cup_{i\in[k]} A_i^+$ as the support of f. We say that f is a *Nash flow* (or *Wardrop flow*) if

$$\forall i \in [k], \forall P \in \mathcal{P}_i, f_P > 0 : \qquad l_P(f) \leq l_{P'}(f) \ \ \forall P' \in \mathcal{P}_i. \tag{2.2}$$

More general, for $\epsilon \geq 0$, we say that f is an *ϵ-approximate Nash flow* if

$$\forall i \in [k], \forall P \in \mathcal{P}_i, f_P > 0 : \qquad l_P(f) \leq (1+\epsilon) l_{P'}(f) \ \ \forall P' \in \mathcal{P}_i. \tag{2.3}$$

Remark 2.1. If the population is heterogeneous (with respect to deviations as defined later on), then each commodity $i \in [k]$ is further partitioned in h_i *sensitivity classes*, where class $j \in [h_i]$ has demand r_{ij} such that $r_i = \sum_{j\in[h_i]} r_{ij}$. Given a path $P \in \mathcal{P}_i$, we use $f_{P,j}$ to refer to the amount of flow on path P of sensitivity class j (so that $\sum_{j\in[h_i]} f_{P,j} = f_P$).

Existence of equilibria is guaranteed for (heterogeneous) non-atomic congestion games under the assumptions made on the latency functions above, see, e.g., [159].

2.2.1.1 Price of anarchy

The (ϵ-approximate) price of anarchy of an instance $\mathcal{I}$ is defined as

$$\epsilon\text{-PoA}(\mathcal{I}) = \sup\left\{\frac{C(f)}{C(f^*)} : f \text{ is } \epsilon\text{-appproximate Nash flow}\right\}$$

where f^* denotes a socially optimal flow. For a class of instances $\mathcal{H}$, the (ϵ-approximate) price of anarchy is defined as

$$\epsilon\text{-PoA}(\mathcal{H}) = \sup_{\mathcal{I}\in\mathcal{H}} \epsilon\text{-PoA}(\mathcal{I}).$$

When $\epsilon = 0$ we simply refer to this quantity the price of anarchy (of either an instance or a class of instances).

In [45] it is shown that, for latency functions in a class $\mathcal{D}$, the price of anarchy of an instance is at most

$$\rho(\mathcal{D}) := (1 - \beta(\mathcal{D}))^{-1}, \quad \text{where} \quad \beta(\mathcal{D}) = \sup_{d\in\mathcal{D}} \sup_{x,y\in\mathbb{R}: x\geq y>0} \frac{y(d(x)-d(y))}{xd(x)}. \tag{2.4}$$

The value of $\rho(\mathcal{D})$ is well-understood for many important classes of latency functions. For example, let

$$\mathcal{D}_d = \{g : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} : g(\mu x) \geq \mu^d g(x) \ \forall \mu \in [0,1]\}.$$

In particular, $\mathcal{D}_d$ contains all polynomial latency functions with non-negative coefficients and maximum degree d. We have

$$\rho(\mathcal{D}_d) = \left(1 - \frac{d}{(d+1)^{(d+1)/d}}\right)^{-1}.$$

2.2.2 Bounded deviation model

In its most general form, we have for every path $P \in \mathcal{P}$ a continuous deviation (or perturbation) function $\delta_P : \mathcal{F} \to \mathbb{R}$. For a heterogeneous population we introduce a sensitivity parameter $\gamma_{ij} \geq 0$ for every sensitivity class $j \in [h_i]$ for every commodity $i \in [k]$. The deviated latency along a path P under flow f for a player in commodity i with sensitivity γ_{ij} is then given by $q_P^j(f) = \ell_P(f) + \gamma_{ij}\delta_P(f)$. We say that f is *δ-inducible* if and only if it is a *Wardrop flow* (or *Nash flow*) with respect to q, i.e.,

$$\forall i \in [k], \forall j \in [h_i], \forall P \in \mathcal{P}_i, f_{P,j} > 0 : \qquad q_P^j(f) \leq q_{P'}^j(f) \;\; \forall P' \in \mathcal{P}_i. \tag{2.5}$$

Note that under a Nash flow f all flow-carrying paths $P \in \mathcal{P}_i$ of sensitivity class j in commodity $i \in [k]$ have the same deviated latency. If f is δ-inducible, we also write $f = f^\delta$. Also note that a Nash flow f for the unaltered latencies $(l_a)_{a \in A}$ is 0-inducible, i.e., $f = f^0$. We define

$$\Delta(\theta) = \{(\delta_P)_{P \in \mathcal{P}} : \forall P \in \mathcal{P} : \theta_P^{\min}(f) \leq \delta_P(f) \leq \theta_P^{\max}(f) \text{ for all } f \in \mathcal{F}\},$$

for given threshold functions $\theta_P^{\min}$ and $\theta_P^{\max}$ for all $P \in \mathcal{P}$. For $-1 < \alpha \leq 0 \leq \beta$, we call $\delta \in \Delta(\theta)$ an *(α, β)-deviation* if $\theta^{\min} = \alpha l$ and $\theta^{\max} = \beta l$, and also write $\theta = (\alpha, \beta)$. We then say that for $\delta \in \Delta(\theta)$, the flow f^δ is an (α, β)-deviated Nash flow. If $\alpha = 0$, we often write β-deviated Nash flow.

There is a close relation to the notion of an approximate Nash flow. In particular for $(0, \beta)$-deviated flows, the Nash condition in (2.5) implies that

$$\forall i \in [k], \forall j \in [h_i], \forall P \in \mathcal{P}_i, f_{P,j} > 0 : \qquad l_P^j(f) \leq (1 + \epsilon_{ij}) l_{P'}^j(f) \;\; \forall P' \in \mathcal{P}_i, \tag{2.6}$$

where $\epsilon_{ij} = \beta\gamma_{ij}$ for all pairs (i, j).[5] The statement in (2.6) is the definition of f being a (heterogeneous) ϵ-approximate Nash flow for $\epsilon = \beta\gamma$.

We are often interested in a more special type of path deviations, namely those induced as an aggregation of *arc deviations*. We describe this model next for homogeneous populations. We use similar notation and terminology but it will always be clear if we consider general path deviations, or those induced by arc deviations.

Additive path deviations. For every arc $a \in A$, we have a continuous function $\delta_a : \mathbb{R}_{\geq 0} \to \mathbb{R}$ modeling the *deviation* on arc a, and we write $\delta = (\delta_a)_{a \in A}$. Note that the deviation δ_a on arc a can be positive or negative. We define the deviation of a path $P \in \mathcal{P}$ as $\delta_P(f) = \sum_{a \in P} \delta_a(f_a)$. The *deviated latency* on arc $a \in A$ is given by $l_a(f_a) + \delta_a(f_a)$; similarly, the deviated latency on path $P \in \mathcal{P}$ is given by $l_P(f) + \delta_P(f)$.

Let $\theta^{\min} = (\theta_a^{\min})_{a \in A}$ and $\theta^{\max} = (\theta_a^{\max})_{a \in A}$ be given threshold functions, where for every $a \in A$, $\theta_a^{\min} : \mathbb{R}_{\geq 0} \to \mathbb{R}$ is a continuous, non-increasing function

[5] This follows directly by applying the threshold functions as bounds for δ_P in (2.5).

and $\theta_a^{\max} : \mathbb{R}_{\geq 0} \to \mathbb{R}$ is a continuous, non-decreasing function. Further, we assume that $\theta_a^{\min}(x) \leq 0 \leq \theta_a^{\max}(x)$ for all $x \geq 0$ and $a \in A$, and let $\theta = (\theta^{\min}, \theta^{\max})$. We define

$$\Delta(\theta) = \{(\delta_a)_{a\in A} : \forall a \in A : \theta_a^{\min}(x) \leq \delta_a(x) \leq \theta_a^{\max}(x),\ \forall x \geq 0\}$$

as the set of *feasible deviations*. Note that $0 \in \Delta(\theta)$ for all threshold functions $\theta^{\min}$ and $\theta^{\max}$. We say that $\delta \in \Delta(\theta)$ is a *θ-deviation*. Furthermore, f is *θ-inducible* if there exists a $\delta \in \Delta(\theta)$ such that f is δ-inducible. For $-1 < \alpha \leq 0 \leq \beta$, we call $\delta \in \Delta(\theta)$ an *(α, β)-deviation* if $\theta^{\min} = \alpha l$ and $\theta^{\max} = \beta l$, and also write $\theta = (\alpha, \beta)$.

We make the following assumption:

Assumption 2.2. We assume that the function $l_a + \theta_a^{\min}$ is non-negative and non-decreasing for every arc $a \in A$.

Intuitively, the non-negativity property ensures that the deviated latencies $l + \delta$ remain non-negative for all feasible deviations $\delta \in \Delta(\theta)$. The non-decreasingness property is exploited in our upper bound proof on the deviation ratio. Note that (α, β)-deviations naturally satisfy this assumption.

Throughout this chapter, we (implicitly) only consider deviations δ for which a Nash flow exists. The existence of such flows is always guaranteed under some mild conditions on the threshold functions. As an example, we elaborate on the existence when $\theta_a^{\min} = 0$ and $\theta_a^{\max}$ is non-negative, non-decreasing and continuous for all $a \in A$. It is not hard to see that for a deviated Nash flow f^δ with $\delta \in \Delta(\theta)$ there exists some $0 \leq \lambda_a \leq 1$ for every arc $a \in A$ such that $\delta_a(f_a^\delta) = \lambda_a \theta_a^{\max}(f_a^\delta)$. In particular, this means that the deviations δ' defined as $\delta'_a = \lambda_a \theta_a^{\max}$ satisfies $\delta' \in \Delta(\theta)$ and also induces f^δ. Therefore it is sufficient to consider deviations of the form $\delta_a = \lambda_a \theta_a^{\max}$, where $0 \leq \lambda_a \leq 1$ for all $a \in A$. For such deviations, the deviated latency function $l_a + \delta_a$ is non-negative, non-decreasing and continuous for every $a \in A$. It is well-known that for these types of functions, the existence of a Nash flow is guaranteed, see, e.g., [140].

The following lemma shows an equivalence between (α, β)-deviations with $-1 < \alpha \leq 0 \leq \beta$ and $(0, \frac{\beta-\alpha}{1+\alpha})$-deviations. In particular, it allows us to assume without loss of generality that $\alpha = 0$.

Lemma 2.3. *Let $-1 < \alpha \leq 0 \leq \beta$ be fixed. Then f is inducible with an (α, β)-deviation if and only if it is inducible with a $(0, \frac{\beta-\alpha}{1+\alpha})$-deviation.*

Proof. Let f be δ-inducible for some $\alpha l \leq \delta \leq \beta l$, and for $a \in A$, write $\delta_a(f_a) = d_a l_a(f_a)$. Without loss of generality we may assume that $\delta_a(x) = d_a l_a(x)$ (since by definition $d_a l_a(x)$ also induces f). From the equilibrium conditions (2.5), we know that

$$\forall i \in [k], \forall P \in \mathcal{P}_i, f_P > 0 : \quad \sum_{a\in P} l_a(f_a) + \delta_a(f_a) \leq \sum_{a\in P'} l_a(f_a) + \delta_a(f_a) \ \ \forall P' \in \mathcal{P}_i.$$

This is equivalent to $\forall i \in [k], \forall P \in \mathcal{P}_i, f_P > 0$:

$$\sum_{a \in P} \left(1 + \frac{d_a - \alpha}{1 + \alpha}\right) l_a(f_a) \leq \sum_{a \in P'} \left(1 + \frac{d_a - \alpha}{1 + \alpha}\right) l_a(f_a) \quad \forall P' \in \mathcal{P}_i$$

which can be seen by writing

$$l_a(f_a) + \delta_a(f_a) = (1 + d_a) l_a(f_a) = (1 + \alpha + d_a - \alpha) l_a(f_a),$$

and then dividing the inequality by $1 + \alpha > 0$. We then see that δ', defined by $\delta'_a(x) = \frac{d_a - \alpha}{1+\alpha} l_a(x)$ for all $a \in A$ and $x \geq 0$, also induces f since

$$\alpha l_a(x) \leq d_a l_a(x) \leq \beta l_a(x) \quad \Leftrightarrow \quad 0 \leq \frac{d_a - \alpha}{1 + \alpha} l_a(x) \leq \frac{\beta - \alpha}{1 + \alpha} l_a(x).$$

□

This reduction does not work for heterogeneous populations, but a similar statement is true for general path deviations under a homogeneous population.

2.2.3 Inefficiency measures

Given an instance $\mathcal{I}$ and threshold functions $\theta = (\theta^{\min}, \theta^{\max})$, we define the *deviation ratio* as the worst-case ratio of the cost of a θ-inducible flow and the cost of a 0-inducible flow; more formally,

$$\mathrm{DR}(\mathcal{I}, \theta) = \sup_{\delta \in \Delta(\theta)} \left\{ \frac{C(f^\delta)}{C(f^0)} \;\middle|\; f^\delta \text{ is } \delta\text{-inducible} \right\}.$$

This definition applies to both general path deviations as well as additive path deviations. Intuitively, $\mathrm{DR}(\mathcal{I}, \theta)$ measures the worst-case deterioration of the social cost of a Nash flow due to (feasible) latency deviations. In case of (α, β)-deviations, we sometimes write (α, β)-$\mathrm{DR}(\mathcal{I})$. Note that for a fixed deviation $\delta \in \Delta(\theta)$, there might be multiple Nash flows that are δ-inducible. Unless stated otherwise, we adopt the convention that $C(f^\delta)$ refers to the social cost of the worst Nash flow that is δ-inducible.

We introduce a similar notion for ϵ-approximate Nash flows. We define the ϵ-stability ratio as

$$\epsilon\text{-}\mathrm{SR}(\mathcal{I}) = \sup_{f^\epsilon} \left\{ \frac{C(f^\epsilon)}{C(f^0)} \;\middle|\; f^\epsilon \text{ is an } \epsilon\text{-approximate Nash flow} \right\}.$$

2.3 Upper bounds on the deviation ratio

In this section we derive an upper bound on the deviation ratio for additive path deviations, in particular, for (α, β)-deviations. All results hold for multi-commodity instances with a common source. The following notion of alternating paths turns out to be crucial. It was first introduced by Lin et al. [122] in the context of the network design problem, see [154], and is also used by Nikolova and Stier-Moses [139].

Definition 2.4 (Alternating path [122, 139])**.** Let x and z be feasible flows. We partition $A = X \cup Z$, where $Z = \{a \in A : z_a \geq x_a \text{ and } z_a > 0\}$ and $X = \{a \in A : z_a < x_a \text{ or } z_a = x_a = 0\}$. We say that $\pi_i = (a_1, \ldots, a_r)$ is an *alternating* (s, t_i)*-path* if the arcs in $\pi_i \cap Z$ are oriented in the direction of t_i, and the arcs in $\pi_i \cap X$ are oriented in the direction of s.

An *alternating path tree* π is a tree, rooted at the common source s, which contains an alternating (s, t_i)-path π_i for every commodity i. We show below that an alternating path tree always exists for multi-commodity networks with a common source.

The main theorem which we prove in this section is as follows.

Theorem 2.5. *Let x be θ-inducible and let z be 0-inducible. Further, let $A = X \cup Z$ be a partition of A as in Definition 2.4. Let π be an alternating path tree, where π_i denotes the alternating (s, t_i)-path in π. Suppose $\theta = (\theta^{\min}, \theta^{\max})$. Let X_i be a flow-carrying path of commodity $i \in [k]$ maximizing $l_P(x)$ over all $P \in \mathcal{P}_i$.*[6] *Then*

$$C(x) \leq C(z) + \sum_{i \in [k]} r_i \Bigg(\sum_{a \in Z \cap \pi_i} \theta_a^{\max}(z_a) - \sum_{a \in X \cap \pi_i} \theta_a^{\min}(z_a) - \sum_{a \in X_i} \theta_a^{\min}(x_a) \Bigg).$$

We give some interpretation: Theorem 2.5 relates the social cost of a θ-inducible Nash flow x to the social cost of an original Nash flow z. More specifically, it shows that $C(x) - C(z)$ is at most

$$\sum_{i \in [k]} r_i \Bigg(\sum_{a \in Z \cap \pi_i} \theta_a^{\max}(z_a) - \sum_{a \in X \cap \pi_i} \theta_a^{\min}(z_a) - \sum_{a \in X_i} \theta_a^{\min}(x_a) \Bigg),$$

where X_i is a flow-carrying (s, t_i)-path with respect to x and π_i is an alternating (s, t_i)-path for commodity i. Intuitively, the contribution of commodity i to the above term can be seen as the total θ*-cost* of sending r_i units of flow along the directed cycle C_i which we obtain from π_i and X_i by reversing all arcs in $X \cap \pi_i$ and X_i. Here the θ*-cost* is defined as $\theta_a^{\max}(z_a)$ for a forward arc $a \in Z \cap \pi_i$, $-\theta_a^{\min}(z_a)$ for a reversed arc $a \in X \cap \pi_i$, and $-\theta_a^{\min}(x_a)$ for a reversed arc $a \in X_i$. If we can bound the total θ-cost by $\lambda C(z) - \mu C(x)$ with $\lambda \geq 0$ and $\mu > -1$, then we obtain an upper bound of $(1 + \lambda)/(1 + \mu)$ on the deviation ratio.

In particular, for (α, β)-deviations the θ-cost can naturally be related to the latencies. In this case, we obtain the bound stated below.

[6] Note that the values $l_P(x) + \delta_P(x)$ are the same for all flow-carrying paths, but this is not necessarily true for the values $l_P(x)$.

Corollary 2.6. *Let x be θ-inducible and let z be 0-inducible. Further, let $A = X \cup Z$ be a partition of A as in Definition 2.4. Let π be an alternating path tree, where π_i denotes the alternating (s, t_i)-path in π.*

Suppose $\theta = (\alpha, \beta)$ with $-1 < \alpha \leq 0 \leq \beta$. Let η_i be the number of disjoint segments of consecutive arcs in Z on the alternating (s, t_i)-path π_i for $i \in [k]$.[7] *Then*

$$C(x) \leq \left(1 + \frac{\beta - \alpha}{1 + \alpha} \sum_{i \in [k]} r_i \eta_i\right) C(z) \leq \left(1 + \frac{\beta - \alpha}{1 + \alpha} \cdot \left\lceil \frac{n-1}{2} \right\rceil r\right) C(z).$$

In order to prove Theorem 2.5 we proceed as follows: We first derive a characterization of when a given flow f is θ-inducible (Theorem 2.7). As it turns out, this reduces to a non-negative cycle condition in a suitably defined auxiliary graph $\hat{G}(f)$ with cost function c. In particular, this non-negative cycle condition allows us to relate the cost of a flow-carrying path F_i of f to arbitrary (s, t_i)-paths and (t_i, s)-paths in the auxiliary graph $\hat{G}(f)$ (Lemma 2.9). We then turn to relating the social cost of a θ-inducible flow x to that of a 0-inducible flow z. We show that an alternating path tree π with respect to x and z always exists (Lemma 2.11). With the help of this alternating tree we can then relate the costs of (carefully chosen) flow-carrying paths under x and z for each commodity. Basically, for each commodity i we bound the cost of a flow-carrying path X_i of x by the cost of the alternating path π_i (by applying Lemma 2.9 to X_i and π_i). The latter in turn can then be bounded by the cost of a flow-carrying path Z_i of z (by applying Lemma 2.9 to Z_i and π_i).

2.3.1 Characterization of θ-inducible flows

We provide a characterization of the inducibility of a given flow. Let f be a feasible flow. We define an *auxiliary graph* $\hat{G} = \hat{G}(f) = (V, \hat{A})$ with $\hat{A} = A \cup \bar{A}$, where $\bar{A} = \{(v, u) : a = (u, v) \in A^+\}$, i.e., $\hat{A}$ consists of the set of arcs in A, which we call *forward* arcs, and the set $\bar{A}$ of arcs (v, u) with $(u, v) \in A^+$, which we call *reversed arcs*. Further, we define a cost function $c : \hat{A} \to \mathbb{R}$ as follows:

$$c_a = \begin{cases} l_{(u,v)}(f_{(u,v)}) + \theta^{\max}_{(u,v)}(f_{(u,v)}) & \text{if } (u, v) \in A \\ -l_{(u,v)}(f_{(u,v)}) - \theta^{\min}_{(u,v)}(f_{(u,v)}) & \text{if } (v, u) \in \bar{A}. \end{cases} \tag{2.7}$$

Theorem 2.7 below generalizes a characterization result for single-commodity networks in [17] to multi-commodity networks with a common source.

Theorem 2.7. *Let f be a feasible flow. f is θ-inducible if and only if $\hat{G} = \hat{G}(f)$ does not contain a cycle of negative cost with respect to c.*

[7]Note that $\eta_i \leq \lceil (n-1)/2 \rceil$.

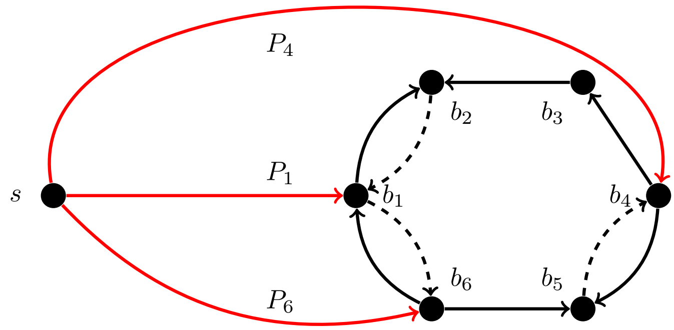

Figure 2.1: The dashed arcs are the reversed arcs in $\hat{G}$. The black bold arcs indicate the cycle B. We have $(h_0, h_1, h_2, h_3) = (1, 4, 6, 1)$. Note that, for example, it could be the case that $P_1 = P_6 \cup (b_6, b_1)$.

Proof. Suppose that f is an inducible flow and let δ be a vector of deviations that induce f. Throughout the proof all latency, deviation and threshold functions are evaluated with respect to f. For notational convenience, we omit the explicit reference to f.

Let $\hat{B}$ be a directed cycle in $\hat{G}$. If $\hat{B}$ only consists of forward arcs, then $\sum_{a\in\hat{B}}(l_a+\theta_a^{\max}) \geq \sum_{a\in\hat{B}}(l_a+\theta_a^{\min}) \geq 0$, where the last inequality holds because of Assumption 2.2. Next, suppose that there is a reversed arc $a = (v, u) \in \hat{B} \cap \bar{A}$. Then $(u, v) \in A_i^+$ for some commodity $i \in [k]$. Let $B = (b_1, \ldots, b_q, b_1)$ be the cycle that we obtain from $\hat{B}$ if all arcs $(v, u) \in \hat{B} \cap \bar{A}$ are replaced by $a = (u, v) \in A^+$ (note that B is contained in G and that it is not a directed cycle). For every arc $b = (b_l, b_{l+1}) \in B \cap A^+$, there is a flow-carrying path P_l from s to b_l for some commodity i (here we use the fact that all commodities share the same source).[8]

Intuitively, the proof is as follows. For all nodes $b \in V(B)$ with two incoming arcs of B, we can can find two paths Q_1 and Q_2 leading to that node, using the paths P_l and the cycle B (see also Figure 2.1). Furthermore, one of those paths is flow-carrying by construction. We then apply the Nash conditions to those flow-carrying paths (exploiting the common source) and add up the resulting inequalities. The contributions of the paths P_l cancel out in the aggregated inequality, leading to the desired result. We now give a formal proof.

Without loss of generality, we may assume that $(b_1, b_2) \in A^+$. Let $h_1 \in \{2, \ldots, q+1\}$ be the smallest index for which $(b_{h_1}, b_{h_1+1}) \in A^+$ (here we take $b_{q+1} := b_1$ and $P_{q+1} := P_1$). Note that the concatenation of the path P_{h_1} and

[8]Note that the paths P_l can overlap, use parts of B, or even be subpaths of each other.

$(b_{h_1}, b_{h_1-1}, \ldots, b_2)$ is a directed path from s to b_2. Then we have

$$l_{(b_1,b_2)} + \delta_{(b_1,b_2)} + \sum_{a \in P_1} (l_a + \delta_a) \leq \sum_{j=3}^{h_1} (l_{(b_j,b_{j-1})} + \delta_{(b_j,b_{j-1})}) + \sum_{a \in P_{h_1}} (l_a + \delta_a).$$

This follows by using the fact that a subpath $(s, \ldots, u)$ of a shortest (s, t_i)-path $(s, \ldots, u, \ldots, t_i)$ is a shortest (s, u)-path if G does not contain negative cost cycles under the cost function $l + \delta$ (which is true because of Assumption 2.2).

We can now repeat this procedure by letting $h_2 \in \{h_1 + 1, \ldots, q + 1\}$ be the smallest index for which $(b_{h_2}, b_{h_2+1}) \in A^+$. We then have

$$\begin{aligned} l_{(b_{h_1},b_{h_1+1})} + \delta_{(b_{h_1},b_{h_1+1})} &+ \sum_{a \in P_{h_1}} (l_a + \delta_a) \\ &\leq \sum_{j=h_1+2}^{h_2} (l_{(b_j,b_{j-1})} + \delta_{(b_j,b_{j-1})}) + \sum_{a \in P_{h_2}} (l_a + \delta_a). \end{aligned}$$

Continuing this procedure, we find a sequence $1 = h_0 < h_1 < \cdots < h_p = q + 1$ such that, for every $0 \leq w \leq p - 1$,

$$\begin{aligned} l_{(b_{h_w},b_{h_w+1})} + \delta_{(b_{h_w},b_{h_w+1})} &+ \sum_{a \in P_{h_w}} (l_a + \delta_a) \\ &\leq \sum_{j=h_w+2}^{h_{w+1}} (l_{(b_j,b_{j-1})} + \delta_{(b_j,b_{j-1})}) + \sum_{a \in P_{h_{w+1}}} (l_a + \delta_a). \end{aligned} \tag{2.8}$$

Note that p is the number of reversed arcs on the cycle $\hat{B}$.

Summing up these inequalities for $0 \leq w \leq p - 1$, we obtain

$$\sum_{(v,u) \in \hat{B} \cap \bar{A}} (l_{(u,v)} + \delta_{(u,v)}) \leq \sum_{a \in \hat{B} \cap A} (l_a + \delta_a),$$

since all the contributions of the path P_l cancel out. Now using the definition of a θ-deviation, we find

$$\begin{aligned} \sum_{a \in \hat{B} \cap A} (l_a + \theta_a^{\max}) - &\sum_{(v,u) \in \hat{B} \cap \bar{A}} (l_{(u,v)} + \theta_{(u,v)}^{\min}) \\ &\geq \sum_{a \in \hat{B} \cap A} (l_a + \delta_a) - \sum_{(v,u) \in \hat{B} \cap \bar{A}} (l_{(u,v)} + \delta_{(u,v)}) \geq 0. \end{aligned}$$

We have shown that $\hat{B}$ has non-negative cost. Note that $\hat{B}$ has zero cost if all the arcs on the cycle are reversed.

For the other direction of the proof, consider the set $\mathcal{H}(\theta)$ of θ-deviations $\delta \in \Delta(\theta)$ that induce $f = (f_a^i)_{i \in [k], a \in A}$ (see also [122, 154]):

$$\mathcal{H}(\theta) = \{ \ (\delta_a)_{a \in A} \ \big| \ \pi_{i,v} - \pi_{i,u} \leq l_a + \delta_a \quad \forall a = (u, v) \in A, \forall i \in [k]$$

$$
\begin{aligned}
\pi_{i,v} - \pi_{i,u} = l_a + \delta_a & \quad \forall a = (u,v) \in A_i^+, \forall i \in [k] \\
\delta_a \geq \theta_a^{\min} & \quad \forall a \in A \\
\delta_a \leq \theta_a^{\max} & \quad \forall a \in A \ \}.
\end{aligned} \tag{2.9}
$$

That is, f is θ-inducible if and only if the linear system defining $\mathcal{H}(\theta)$ in (2.9) has a feasible solution. Now suppose that $\hat{G}$ does not contain a cycle of negative cost. Then we can determine the shortest path distance π_u from s to every node $u \in V$. We define $\pi_{i,u} := \pi_u$ for all $u \in V$ and $i \in [k]$. Furthermore, for $a = (u,v) \in A$, we define $\delta_a := \max\{\theta_a^{\min}, \pi_v - \pi_u - l_a\}$. We will now show that δ induces f by showing that we have constructed a feasible solution for (2.9). First of all, for all $i \in [k]$ and $a \in A \setminus A_i^+$, we have $\delta_a \geq \pi_v - \pi_u - l_a$, which is equivalent to $\pi_{i,v} - \pi_{i,u} \leq l_a + \delta_a$. Secondly, if $a = (u,v) \in A_i^+$, then $\pi_u - \pi_v \leq -l_a - \theta_a^{\min}$ (which we derive using the reversed arc (v,u)). But this is equivalent to $\pi_{i,v} - \pi_{i,u} - l_a \geq \theta_a^{\min}$. We can conclude that $\delta_a = \pi_{i,v} - \pi_{i,u} - l_a$. Furthermore, we clearly have $\delta_a \geq \theta_a^{\min}$. Lastly, for all $a = (u,v) \in A$ we have $\pi_v - \pi_u \leq l_a + \theta_a^{\max}$ which is equivalent to $\pi_v - \pi_u - l_a \leq \theta_a^{\max}$. Combining this with the trivial inequality $\theta_a^{\min} \leq \theta_a^{\max}$ we can conclude that $\delta_a \leq \theta_a^{\max}$. This completes the proof. □

The characterization of Theorem 2.7 applies if all commodities share a common source. In fact, in Example 2.8 we show that this characterization does not hold if this assumption is dropped.

Example 2.8. Consider the graph $G = (V, A)$ in Figure 2.2 and suppose that $r_1 = r_2 = 1$. Then the flow f that routes one unit of flow over both paths $(s_1, v_1, 1, 2, t_1)$ and $(s_2, v_2, 3, 4, t_2)$ is feasible and inducible (take $\delta = 0$). However, looking at the graph $\hat{G}(f)$, we obtain a negative cost cycle $(1,4,3,2,1)$ (by using the reversed arcs of $(1,2)$ and $(3,4)$). □

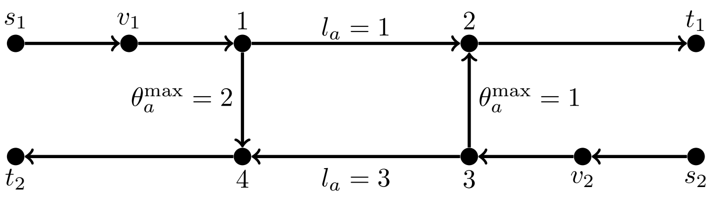

Figure 2.2: All the values of l_a, $\theta_a^{\min}$ and $\theta_a^{\max}$ that are not explicitly stated are zero.

By exploiting the non-negative cycle condition of Theorem 2.7, we can now establish the following bounds on the cost of a flow-carrying path F_i of a θ-inducible flow f.

Lemma 2.9. *Let f be θ-inducible and let F_i be a flow-carrying (s,t_i)-path for commodity $i \in [k]$ in G. Let χ and ψ be any (s,t_i)-path and (t_i,s)-path in $\hat{G}(f)$, respectively. Then*

$$\sum_{a \in F_i} l_a(f_a) + \theta_a^{\min}(f_a) \leq \sum_{a \in \chi \cap A} l_a(f_a) + \theta_a^{\max}(f_a) - \sum_{a \in \chi \cap \bar{A}} l_a(f_a) + \theta_a^{\min}(f_a)$$
$$\sum_{a \in F_i} l_a(f_a) + \theta_a^{\max}(f_a) \geq \sum_{a \in \psi \cap \bar{A}} l_a(f_a) + \theta_a^{\min}(f_a) - \sum_{a \in \psi \cap A} l_a(f_a) + \theta_a^{\max}(f_a).$$

We need the following proposition to prove Lemma 2.9.

Proposition 2.10. *Let $G = (V, A)$ be a non-empty, directed multigraph with the property that $\delta^-(v) = \delta^+(v)$ for all $v \in V$.*[9] *Then G is the union of arc-disjoint directed (simple) cycles $C_1, \ldots, C_l$ for some l such that $\cup_{j=1}^{l} V(C_j) = V$ and $\cup_{j=1}^{l} A(C_j) = A$.*

Proof. If G is non-empty then we can find a (simple) directed cycle C in G. Removing the arcs of this cycle leads to the graph $G \setminus C := (V, A \setminus A(C))$ that also satisfies $\delta^-(v) = \delta^+(v)$ for all $v \in V$ (note that if there are multiple arcs between two nodes, we only remove the copy on the cycle). By repeating this procedure until G becomes empty, we decompose G into a series of arc-disjoint directed (simple) cycles $C_1, \ldots, C_l$ as claimed. □

Proof of Lemma 2.9. Since F_i is a flow-carrying path, we know that for every $a = (u, v) \in F_i$ we have a reversed arc $(v, u) \in \hat{A}$ in $\hat{G}$. Let $\bar{F}_i$ denote the reversed path of F_i. Define $\hat{H}$ as the graph consisting of the (t_i, s)-path $\bar{F}_i$ and the (s, t_i)-path χ, where we add a copy of an arc if it is used in both paths (i.e., $\hat{H}$ can be a multigraph). Note that $\hat{H}$ satisfies the conditions of Proposition 2.10. Thus, $\hat{H}$ can be decomposed into arc-disjoint directed cycles $C_1, \ldots, C_l$ for some l. By Theorem 2.7, each such cycle C_j has non-negative cost with respect to c (as defined in (2.7)). Thus, we have

$$c(C_j) = \sum_{a \in A \cap C_j} (l_a(x_a) + \theta_a^{\max}(x_a)) - \sum_{a \in \bar{A} \cap C_j} (l_a(x_a) + \theta_a^{\min}(x_a)) \geq 0.$$

By adding these inequalities for all $j = 1, \ldots, l$ and rearranging terms, we obtain the first inequality.

The second inequality is proven analogously (applying the same arguments to the graph $\hat{H}$ consisting of paths F_i and ψ). □

2.3.2 Existence of alternating path tree

Let x and z be feasible flows. Recall the definition of an alternating (s, t_i)-path π_i (Definition 2.4). The following lemma establishes the existence of an alternating

[9] We use the standard notation $\delta^-(v)$ and $\delta^+(v)$ to refer to the set of outgoing and incoming edges of a node v, respectively.

path tree π, rooted at the common source s, which contains an alternating (s, t_i)-path π_i for every commodity $i \in [k]$. It is a direct generalization of Lemma 4.6 in [122] and Lemma 4.5 in [139].

Lemma 2.11. *Let x and z be feasible flows and let $A = X \cup Z$ be a partition of A as in Definition 2.4. Then there exists an alternating path tree.*

Proof. Let $G' = (V', A')$ be the graph defined by $V = V \cup \{t\}$ and $A' = A \cup \{(t_i, t) : i \in [k]\}$. Let x', z' be the flows defined by

$$x'_a = \begin{cases} x_a & \text{for } a = (u, v) \in A \\ r_i & \text{for } a = (t_i, t) \text{ with } i \in [k], \end{cases}$$

$$z'_a = \begin{cases} z_a & \text{for } a = (u, v) \in A \\ r_i & \text{for } a = (t_i, t) \text{ with } i \in [k]. \end{cases}$$

Then x' and z' are feasible (s, t)-flows in G'. We can write $A = Z' \cup X'$ with $Z' = Z \cup \{(t_i, t) : i \in [k]\}$ and X' having the same properties as Z and X in G (which follows from $x'_a = z'_a = r_i > 0$ for all $a = (t_i, t)$).

We can now apply the same argument as in the proof of Lemma 4.5 in [139] of which we will give a short summary (for sake of completeness). For any s-t cut defined by $S \cup V'$ with $s \in S$ we claim that we can cross S with an arc in Z', or a reversed arc in X'. Suppose that this would not be the case, i.e., all arcs into S are in the set Z' and all the outgoing arcs of S are in X'. Let $x_{Z'}$ and $z_{Z'}$ be the total incoming flows from S, and $x_{X'}$ and $z_{X'}$ the total outgoing flows from S (for flows x and z, respectively). From the definition of Z' it follows that $x_{Z'} \leq z_{Z'}$. From conservation of flow it follows that $x_{X'} - x_{Z'} = z_{X'} - z_{Z'}$. Combining these two observations, we find that $x_{X'} \leq z_{X'}$. However, by definition of X', we have $x_{X'} > z_{X'}$ (since we removed all arcs a with $z_a = x_a = 0$). We find a contradiction.

Having proved the claim that we can always cross with an arc in Z' or a reversed arc in X', we can now easily construct a spanning tree π' consisting of alternating paths, by starting with the cut $(S, G \setminus S)$ given by $S = \{s\}$.

Note that t cannot be an interior point of π', since t is only adjacent to incoming arcs of the set Z'. This means that if we remove (t_j, t) from π' (where j is the index for which (t_j, t) is in the tree π'), we have found an alternating path tree π for the graph G, under the flows x and z. □

2.3.3 Proofs of Theorem 2.5 and Corollary 2.6

We now have all the ingredients to prove Theorem 2.5. Throughout this section, let x be a θ-inducible flow and let z be a 0-inducible flow. Let π be an alternating path tree (which exists by Lemma 2.11). Without loss of generality we may remove all arcs with $z_a = x_a = 0$ (as they do not contribute to the social cost). Note that if along the alternating (s, t_i)-path π_i we reverse the arcs of Z then the resulting path is a directed (t_i, s)-path in $\hat{G}(z)$ (which we call the *s-oriented version of* π_i); similarly, if we reverse the arcs of X then the resulting path is an

(s, t_i)-path in $\hat{G}(x)$ (which we call the *t_i-oriented version of π_i*). We start with the proof of Theorem 2.5.

Proof of Theorem 2.5. Let X_i be a flow-carrying path of commodity $i \in [k]$ maximizing $l_P(x)$ over all $P \in \mathcal{P}_i$. Note that by our choice of X_i, we have

$$C(x) = \sum_{i \in [k]} \sum_{P \in \mathcal{P}_i} x_P^i l_P(x) \leq \sum_{i \in [k]} r_i \sum_{a \in X_i} l_a(x_a).$$

Let Z_i be an arbitrary flow-carrying path of commodity $i \in [k]$ with respect to z. We have

$$C(z) = \sum_{i \in [k]} r_i \sum_{a \in Z_i} l_a(z_a).$$

By applying the first inequality of Lemma 2.9 to the flow x in the graph $\hat{G}(x)$, where we choose χ to be the t_i-oriented version of π_i, we obtain

$$\sum_{a \in X_i} l_a(x_a) + \theta_a^{\min}(x_a) \leq \sum_{a \in Z \cap \pi_i} l_a(x_a) + \theta_a^{\max}(x_a) - \sum_{a \in X \cap \pi_i} l_a(x_a) + \theta_a^{\min}(x_a). \tag{2.10}$$

By applying the second inequality of Lemma 2.9 to the flow z in the graph $\hat{G}(z)$ with $\theta^{\max} = \theta^{\min} = 0$, where we choose ψ to be the s-oriented version of π_i, we obtain

$$\sum_{a \in Z_i} l_a(z_a) \geq \sum_{a \in Z \cap \pi_i} l_a(z_a) - \sum_{a \in X \cap \pi_i} l_a(z_a). \tag{2.11}$$

By combining these inequalities, we obtain

$$\begin{aligned}
\sum_{a \in X_i} l_a(x_a) + \theta_a^{\min}(x_a) &\leq \sum_{a \in Z \cap \pi_i} l_a(x_a) + \theta_a^{\max}(x_a) - \sum_{a \in X \cap \pi_i} l_a(x_a) + \theta_a^{\min}(x_a) \\
&\leq \sum_{a \in Z \cap \pi_i} l_a(z_a) + \theta_a^{\max}(z_a) - \sum_{a \in X \cap \pi_i} l_a(z_a) + \theta_a^{\min}(z_a) \\
&\leq \sum_{a \in Z_i} l_a(z_a) + \sum_{a \in Z \cap \pi_i} \theta_a^{\max}(z_a) - \sum_{a \in X \cap \pi_i} \theta_a^{\min}(z_a).
\end{aligned}$$

Here the first inequality follows from (2.10). The second inequality holds because of the definition of X and Z and the non-decreasingness of $l_a + \theta_a^{\max}$ and $l_a + \theta_a^{\min}$ (Assumption 2.2) for every $a \in A$. The last inequality holds because of (2.11).

The claim now follows by multiplying the above inequality with r_i and summing over all commodities $i \in [k]$. $\square$

We need the following proposition for the proof of Theorem 2.6.

Proposition 2.12. *Let $z = f^0$ be a Nash flow for a multi-commodity instance with a common source. Let $v \in V$ and let $i, j \in [k]$ be two commodities for which there exist flow-carrying (s, v)-paths $P_1 \in \mathcal{P}_i$ and $P_2 \in \mathcal{P}_j$, respectively. Then there exists a feasible Nash flow $\bar{z}$ with $\bar{z}_a = z_a$ for all $a \in A$ such that both paths P_1, P_2 are flow-carrying for commodity i, and both paths P_1, P_2 are flow-carrying for commodity j, i.e., we have $\bar{z}_{P_1}^i, \bar{z}_{P_2}^i, \bar{z}_{P_1}^j, \bar{z}_{P_2}^j > 0$.*

Proof. Intuitively, we shift an ϵ amount of flow of commodity i to path P_2 and an ϵ amount of flow of commodity j to path P_1. Formally, choose $\epsilon > 0$ small enough such that $z^i_{P_1} - \epsilon, z^j_{P_2} - \epsilon > 0$. We define

$$\bar{z}^l_P = \begin{cases} z^i_{P_1} - \epsilon & \text{if } P = P_1 \text{ and } l = i \\ z^j_{P_1} + \epsilon & \text{if } P = P_1 \text{ and } l = j \\ z^i_{P_2} + \epsilon & \text{if } P = P_2 \text{ and } l = i \\ z^j_{P_2} - \epsilon & \text{if } P = P_2 \text{ and } l = j \end{cases}$$

and let all the other flow-carrying paths remain unchanged. It then immediately follows that $z_a = \bar{z}_a$ for all $a \in A$, and in the resulting feasible flow $\bar{z}$, both commodities i and j are flow-carrying for both paths P_1 and P_2. The feasibility of $\bar{z}$ follows because both commodities have the same source. Moreover, the common source also implies that if z is a Nash flow, then $\bar{z}$ is also a Nash flow (since commodity i implies that $l_{P_1}(z) \leq l_{P_2}(z)$, and commodity j implies that $l_{P_2}(z) \leq l_{P_1}(z)$). □

We now give the proof of Corollary 2.6.

Proof of Corollary 2.6. By Lemma 2.3 we can assume without loss of generality that for every arc $a \in A$:

$$\theta_a^{\min} = 0 \qquad \text{and} \qquad \theta_a^{\max} = \frac{\beta - \alpha}{1 + \alpha} l_a.$$

Fix a commodity i and consider the alternating (s, t_i)-path π_i. Let a *segment* of π be a maximal sequence of consecutive arcs on π_i which belong to Z. Suppose π consists of η_i segments. Let A_{ij} denote the j-th segment of π_i.

Using Theorem 2.5 and the definition of A_{ij}, we obtain

$$\begin{aligned} C(x) &\leq C(z) + \frac{\beta - \alpha}{1 + \alpha} \sum_{i \in [k]} r_i \sum_{a \in Z \cap \pi_i} l_a(z_a) \\ &\leq C(z) + \frac{\beta - \alpha}{1 + \alpha} \sum_{i \in [k]} r_i \left(\eta_i \cdot \max_{j=1,\ldots,\eta_i} \sum_{a \in A_{ij}} l_a(z_a) \right) \end{aligned}$$

Note that the claim follows if we can prove that $\sum_{a \in A_{ij}} l_a(z_a) \leq C(z)$ for all $j = 1, \ldots, \eta_i$ and $i \in [k]$.

Fix a segment A_{ij}. Below we argue that there always exists a commodity $w \in [k]$ (possibly $w \neq i$) such that every $a \in A_{ij}$ is flow-carrying for commodity w, i.e., $z^w_a > 0$ for every $a \in A_{ij}$. By choosing a suitable path decomposition of z for commodity w, we can thus assume that A_{ij} is contained in some flow-carrying path $P \in \mathcal{P}_w$ and thus $\sum_{a \in A_{ij}} l_a(z_a) \leq l_P(z)$. Recall that $C(z) = \sum_{i \in [k]} r_i l_{Z_i}(z)$, where $Z_i \in \mathcal{P}_i$ is an arbitrary flow-carrying path for commodity $i \in [k]$. By exploiting that $r_i \geq 1$ for every $i \in [k]$, we obtain

$$\sum_{a \in A_{ij}} l_a(z_a) \leq l_P(z) \leq \sum_{i \in [k]} r_i l_{Z_i}(z) = C(z).$$

We now prove that there always exists a commodity w as claimed above. Suppose there are two consecutive edges $a_1 = (u, v)$ and $a_2 = (v, w)$ in A_{ij} that are flow-carrying for commodities w_1 and w_2 in z, respectively. Then there are two (s, v)-paths W_1 and W_2 which are flow-carrying with respect to commodities w_1 and w_2, respectively. The existence of W_1 is clear. The existence of W_2 follows from flow-conservation applied to commodity w_2 (because some positive amount of flow leaves node v). But then, by Proposition 2.12, we may assume that a_1 is also flow-carrying for commodity w_2. By applying this argument repeatedly, starting with the last two arcs on A_{ij} and proceeding towards the front, we can show that there is a commodity for which the whole segment A_{ij} is flow-carrying. □

2.4 Lower bounds on the deviation ratio

In this section, we give lower bounds on the deviation ratio for (α, β)-deviations. We first consider single-commodity instances and prove that the bound given in Theorem 2.5 is tight in all its parameters. We then extend this result to instances with a common source. In contrast, for general multi-commodity instances the situation is much worse. In particular, we establish an exponential lower bound on the deviation ratio.

2.4.1 Single-commodity instances

Our instance is based on the generalized Braess graph [154]. The *m-th Braess graph* $G^m = (V^m, A^m)$ is defined by $V^m = \{s, v_1, \ldots, v_{m-1}, w_1, \ldots, w_{m-1}, t\}$ and A^m as the union of three sets: $E_1^m = \{(s, v_j), (v_j, w_j), (w_j, t) : 1 \leq j \leq m-1\}$, $E_2^m = \{(v_j, w_{j-1}) : 2 \leq j \leq m\}$ and $E_3^m = \{(v_1, t) \cup \{(s, w_{m-1}\}\}$. See Figure 2.3 for an example.

The rough idea behind the lower bound construction is that in the unaltered Nash flow all players spread out evenly over the m paths *not* involving the arcs of the form (v_i, w_i). However, as a result of introducing deviations on the arcs of the form (v_i, w_{i-1}) the players switch to the paths involving the arcs (v_i, w_i), but this increases the latencies on all arcs adjacent to s and t.

Example 2.13. By Lemma 2.3, we can assume without loss of generality that $\alpha = 0$. Let $\beta \geq 0$ be a fixed constant and let $n = 2m \geq 4 \in \mathbb{N}$.[10] Let G^m be the m-th Braess graph. Furthermore, let $y_m : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a non-decreasing,

[10]Note that the value $\lceil (n-1)/2 \rceil$ is the same for $n \in \{2m, 2m+1\}$ with $m \in \mathbb{N}$. The example shows tightness for $n = 2m$. The tightness for $n = 2m+1$ then follows trivially by adding a dummy node.

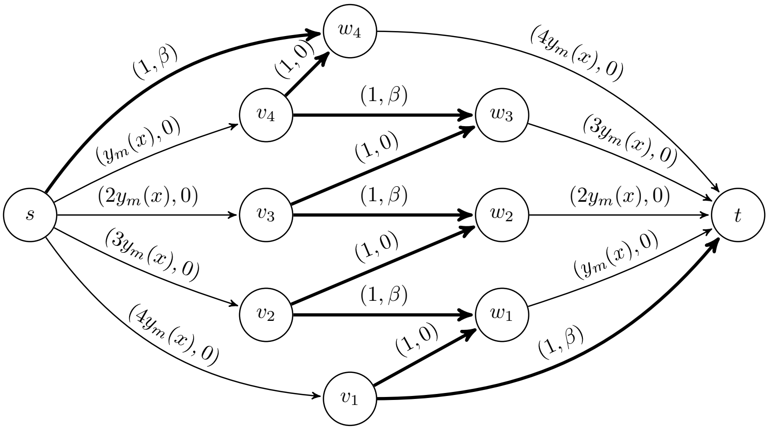

Figure 2.3: The fifth Braess graph with (l_a^5, δ_a^5) on the arcs as defined in Example 2.13. The bold arcs indicate the alternating path π_1.

continuous function[11] with $y_m(1/m) = 0$ and $y_m(1/(m-1)) = \beta$. We define

$$l_a^m(g) = \begin{cases} (m-j)\cdot y_m(g) & \text{for } a \in \{(s,v_j) : 1 \le j \le m-1\} \\ j \cdot y_m(g) & \text{for } a \in \{(w_j,t) : 1 \le j \le m-1\} \\ 1 & \text{otherwise.} \end{cases}$$

Furthermore, we define $\delta_a^m(g) = \beta$ for $a \in E_2^m$, and $\delta_a^m(g) = 0$ otherwise. Note that $0 \le \delta_a^m(g) \le \beta l_a^m(g)$ for all $a \in A$ and $g \ge 0$ (see Figure 2.3).

A Nash flow $z = f^0$ is given by routing $1/m$ units of flow over the paths (s, w_{m-1}, t), (s, v_1, t) and the paths in $\{(s, v_j, w_{j-1}, t) : 2 \le j \le m-1\}$. Note that all these paths have latency one, and the path (s, v_j, w_j, t), for some $1 \le m \le j$, also has latency one. We conclude that $C(z) = 1$.

A Nash flow $x = f^\delta$, with δ as defined above, is given by routing $1/(m-1)$ units of flow over the paths in $\{(s, v_j, w_j, t) : 1 \le j \le m-1\}$. Each such path P then has a latency of $l_P(x) = 1 + \beta m$. It follows that $C(x) = 1 + \beta m$. Note that the deviated latency of path P is $q_P(x) = 1+\beta m$ because all deviations along this path are zero. Each path $P' = (s, v_j, w_{j-1}, t)$, for $2 \le j \le m-1$, has a deviated latency of $q_{P'}(x) = 1 + \beta + (m-1)y_m(1/(m-1)) = 1 + \beta + (m-1)\beta = 1 + \beta m$. The same argument holds for the paths (s, w_{m-1}, t) and (s, v_1, t). We conclude that x is δ-inducible. It follows that $C(x)/C(z) = 1 + \beta m = 1 + \beta n/2$.

2.4.2 Common-source instances

By adapting the construction in Example 2.13, we obtain the following result.

---

[11] For example $y_m(g) = m(m-1)\beta \max\{0, (g - \frac{1}{m})\}$. That is, we define y_m to be zero for $0 \le g \le 1/m$ and we let it increase with constant rate to β in $1/(m-1)$.

Theorem 2.14. *There exist common source two-commodity instances $\mathcal{I}$ such that*

$$DR(\mathcal{I},(\alpha,\beta)) \geq \begin{cases} 1 + \dfrac{\beta-\alpha}{1+\alpha} \cdot \dfrac{n-1}{2} r & \text{for } n = 2m+1 \in \mathbb{N}_{\geq 5} \\ 1 + \dfrac{\beta-\alpha}{1+\alpha} \cdot \left[\left(\dfrac{n}{2} - 1\right) r + 1\right] & \text{for } n = 2m \in \mathbb{N}_{\geq 4}. \end{cases}$$

Proof. We first prove the claim for n odd. Let $r \in \mathbb{R}_{\geq 1}$ and $n = 2m+1 \in \mathbb{N}_{\geq 5}$. We modify the graph G^m by adding one extra node t_2 (the node t will be referred to as t_1 from here on). We add the arcs (s,t_2) and (t_2,t_1) (see the dotted arcs in Figure 2.3). We take one commodity with sink t_1 and $r_1 = 1$, and one commodity with sink t_2 and demand $r_2 = r-1$. Note that the latter commodity only has one (s,t_2)-path.

The pairs $(l_a^m(g), \delta_a^m(g))$, for all a except (s,t_2) and (t_2,t_1), are defined as in Example 2.13, but with y a non-decreasing, non-negative, continuous function satisfying $y_m(1/m) = 0$ and $y_m((1-\epsilon_m)/(m-1)) = \beta$, where we choose $0 < \epsilon_m < 1/m$ so that $1/m < (1-\epsilon_m)/(m-1)$. For $a = (s,t_2)$, we take $(l_a^m(g), \delta_a^m(g)) = (y_m^*(x'), 0)$, where y^* is a non-decreasing, non-negative, continuous function satisfying $y_m^*(r-1) = 0$ and $y_m^*(r-1+\epsilon_m) = \beta$. For $a = (t_2,t_1)$ we take $(l_a^m(g), \delta_a^m(g)) = (1,0)$. See Figure 2.4 for an example.

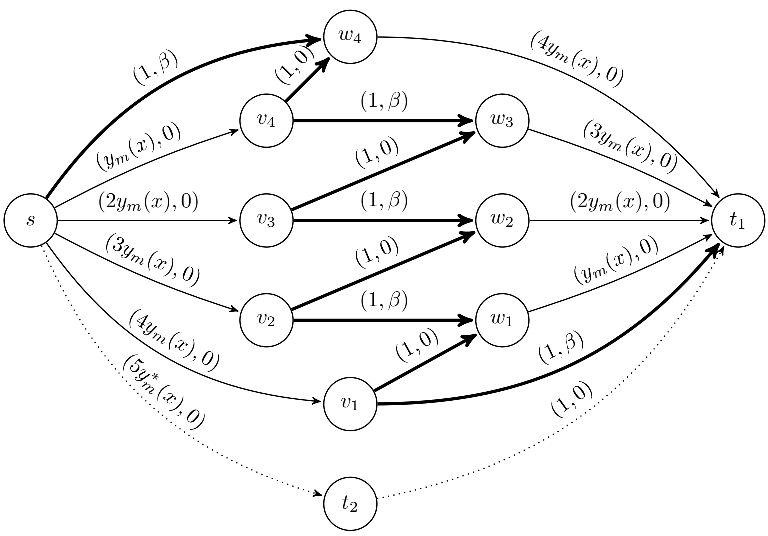

Figure 2.4: The fifth (odd) Braess graph with (l_a^5, δ_a^5) on the arcs as defined above, where $t = t_1$. The thick edges indicate the alternating path π_1.

A Nash flow z for this instance is given by routing $1/m$ units of flow over the paths $(s, w_{m-1}, t_1), (s, v_1, t_1)$ and the paths in $\{(s, v_j, w_{j-1}, t_1) : 2 \leq j \leq m-1\}$ for the first commodity, and $r-1$ units of flow over (s,t_2) for the second commodity. This claim is true since all the paths for the first commodity have latency one, as well as the paths (s, v_j, w_j, t), for $1 \leq m \leq j$. This is also true for (s, t_2, t_1). The latency for the other commodity is zero. We may conclude that $C(z) = 1$.

A Nash flow x under deviation δ, as defined here, is given by, for the first commodity, routing $(1-\epsilon_m)/(m-1)$ units of flow over the paths in $\{(s, v_j, w_j, t) : 1 \leq j \leq m-1\}$, and ϵ_m units of flow over the path (s, t_2, t_1). Note that the perceived latency on all these paths p is $q_P(x) = 1 + \beta m$ (which is also the true latency, since all the deviations are zero on the arcs of these paths). Using the same reasoning as in Example 2.13 it can be seen that the perceived latency on the paths $P' = (s, v_j, w_{j-1}, t)$, for $2 \leq j \leq m-1$, is also $q_{P'}(x) = 1 + \beta m$, from which we may conclude that x is indeed a Nash flow under the deviation δ. We have $C(x) = 1 + \beta m + (r-1)\beta m = 1 + \beta r m$, since for the first commodity the (true) latency along every path is $1 + \beta m$, and for the other commodity the latency along (s, t_2) is βm.

We next prove the claim for n even. Let $r \in \mathbb{R}_{\geq 1}$ and $n = 2m \in \mathbb{N}_{\geq 4}$. We use the same Braess graphs as in Example 2.13, without modifications. We introduce another commodity with demand $r_2 = r - 1$, for which we choose $t_2 = v_1$. We replace the pair $((m-1)y_m(x'), 0)$ on $a = (s, v_1)$ by the pair $((m-1)y'_m(g), 0)$ where y'_m satisfies $y'_m(1/m + r - 1) = 0$ and $y'_m(1/(m-1) + r - 1) = \beta$. Note that the flows x and z, as defined in Example 2.13 with the extension that the second commodity uses the arc (s, v_1) in both cases, still form feasible Nash flows for their respective deviations. We obtain

$$\begin{aligned} C(x) &= \sum_i \sum_{q \in \mathcal{P}_i} x_q^i l_q(x) = 1 + \beta m + (r-1)(m-1)\beta \\ &= 1 + \beta m + \beta(r-1)(m-1) = (1 + \beta r m) - \beta(r-1). \end{aligned}$$

This completes the proof. □

Remark 2.15. For two-commodity instances with n even, we can actually improve the upper bound in Theorem 2.5 to the lower bound stated in Theorem 2.14: Suppose the upper bound of Theorem 2.5 is tight. Then we need to have $\eta_1 = \eta_2 = n/2$. This means that the alternating path tree is actually a path, in the sense that all nodes are adjacent to at most two arcs of the alternating path tree, that alternates between arcs in X and Z, starting and ending with an arc in Z (see Figure 2.3). However, because $t_1 \neq t_2$ this means that at least one of the two commodities has no more than $n/2 - 1$ arcs in Z, which is a contradiction.

2.4.3 Multi-commodity instances

For general multi-commodity instances we establish the following exponential lower bound on the deviation ratio. In particular, this proves that there is an exponential gap between the cases of multi-commodity networks with and without a common source.

Theorem 2.16. *For every $p = 2q+1 \in \mathbb{N}$, there exists a two-commodity instance $\mathcal{I}$ whose size is polynomially bounded in p such that*

$$DR(\mathcal{I}, (0, \beta)) \geq 1 + \beta F_{p+1} \approx 1 + 0.45 \cdot \beta \cdot \phi^{p+1},$$

where F_p is the p-th Fibonacci number and $\phi \approx 1.618$ is the golden ratio.

The instance used in the proof of Theorem 2.16 is based on the following graph introduced in [122].

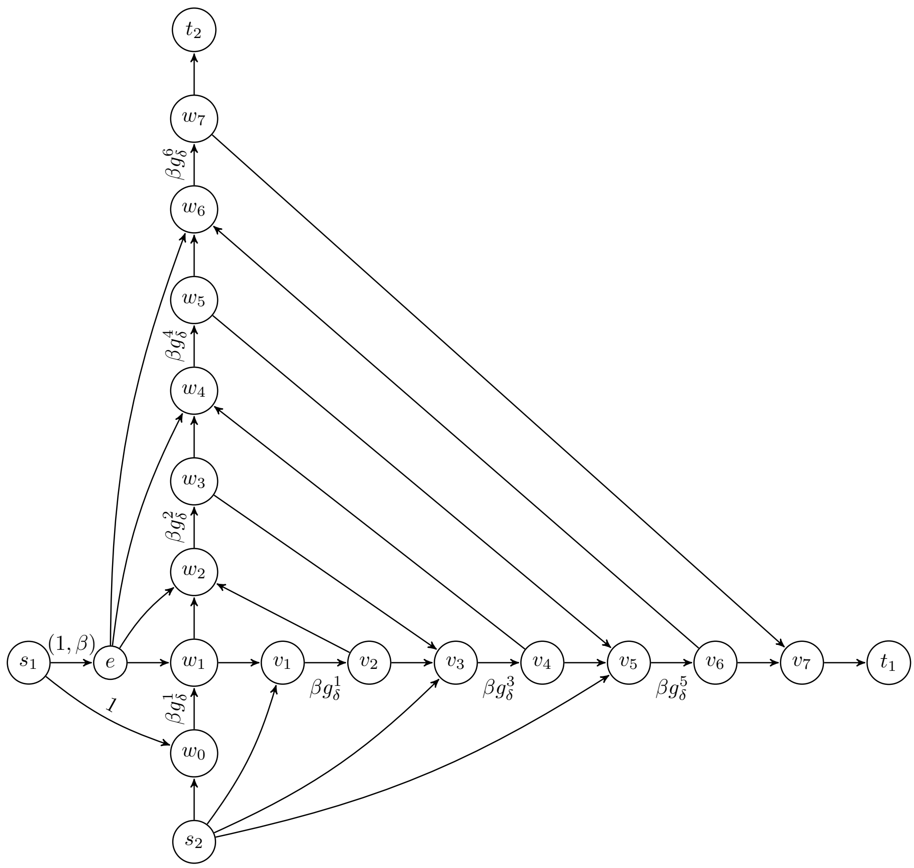

Figure 2.5: The graph G^p for $p = 7$ (this is a reproduction of Figure 4 in [122]). The arc $a = (s_1, e)$ has $\delta_a = \beta$, whereas all the other arcs have $\delta_a = 0$.

Definition 2.17 ([122])**.** For $p = 2q+1 \in \mathbb{N}$, the graph $G^p = (V^p, A^p)$ is defined by

$$V^p = \{s_1, s_2, t_1, t_2, e, w_0, \dots, w_p, v_1, \dots, v_p\},$$

and $A^p = A(P_1^p) \cup A(P_2^p) \cup A_1^p \cup A_2^p \cup \{s_1, w_0\}$ where

$$P_1^p = (s_1, e, w_1, v_1, v_2, \ldots, v_p, t_1) \text{ and } P_2^p = (s_2, w_0, w_1, \ldots, w_p, t_2)$$

are the horizontal (s_1, t_1)-path and vertical (s_2, t_2)-path, respectively; see Figure 2.5. Further,

$$A_1^p = \{(s_2, v_i) : i = 1, 3, 5, 7, \ldots, p-2\} \cup \{(e, w_i) : i = 2, 4, 6, 8, \ldots, p-1\}$$

and

$$A_2^p = \{(w_i, v_i) : i = 3, 5, 7, \ldots, p\} \cup \{(v_i, w_i) : i = 2, 4, 6, 8, \ldots, p-1\}.$$

Lastly, the paths T_i are denoted by

$$T_i = \begin{cases} (s_1, w_0, w_1, v_1, \ldots, v_p, t_1) & i = 0 \\ (s_1, e, w_i, w_{i+1}, v_{i+1}, \ldots, v_p, t_1) & i = 2, 4, 6, \ldots, p-1 \\ (s_2, v_1, v_{i+1}, w_{i+1}, \ldots, w_p, t_2) & i = 1, 3, 5, \ldots, p \end{cases}$$

These paths can be seen as 'shortcuts' for the paths P_1 and P_2.

Proof of Theorem 2.16. We consider instances $(G^p, l^p, \delta^p, r^p)_{p=1,3,5,7,\ldots}$ with G^p as in Definition 2.17. It is not hard to see that $|V^p|$, $|A^p| \in \mathcal{O}(p)$. The latency functions l^p are given as follows:

$$l_a^p(x') = \begin{cases} \beta g_\delta^i(x') & \text{for } a \in \{(v_i, v_{i+1}) : i = 1, 3, 5, \ldots, p-2\} \\ \beta g_\delta^i(x') & \text{for } a \in \{(w_i, w_{i+1}) : i = 0, 2, 4, 6, \ldots, p-1\} \\ 1 & \text{for } a \in \{(s_1, e), (s_1, w_0)\} \\ 0 & \text{otherwise.} \end{cases}$$

Here

$$g_\delta^i(x') = \begin{cases} 0 & x' \leq 1 \\ h_\delta^i(x') & 1 \leq x' \leq 1 + \delta \\ F_i & x' \geq 1 + \delta, \end{cases}$$

where F_i is the i-th Fibonacci number, and $h_\delta^i(x')$ is some non-decreasing, non-negative, continuous function satisfying $h_\delta^i(1) = 0$ and $h_\delta^i(1 + \delta) = F_i$ (so that $g_\delta^i(x')$ is also non-decreasing, non-negative and continuous). Furthermore, we take $\delta_a = \beta$ for $a = (s_1, e)$ and $\delta_a = 0$ for all $a \in A \setminus \{(s_1, e)\}$. Finally, we have $r_1^p = r_2^p = 1$.

Let z be defined by sending one unit of flow over the paths P_1 and P_2. We claim that z is a Nash flow with respect to the latencies l^p, and that $C(z) = 1$. By construction, the latency along the path P_1 is $l_{P_1}(z) = 1$. It is not hard to see that any (s_1, t_1)-path has latency greater or equal than one (because every path for commodity 1 uses either (s_1, e) or (s_1, w_0)). For commodity 2 the latency along P_2 is $l_{P_2}(z) = 0$, which is clearly a shortest path. This proves that z is a Nash flow. Further, $C(z) = 1$.

We use Lemma 2.18 (given below) to describe a Nash flow x with respect to the deviated latencies $l^p + \delta^p$. It follows that $C(x) = C(x)/C(z) \geq 1 + \beta F_{p-1} + \beta F_p = 1 + \beta F_{p+1}$. This concludes the proof (since $F_p \approx c \cdot \phi^p$ where $c \approx 0.4472$ and $\phi \approx 1.618$). $\square$

The following lemma is similar to Lemma 5.4, Lemma 5.5 and Lemma 5.6 in [122].

Lemma 2.18. *There exists a $\delta > 0$ and a feasible flow x satisfying the following properties:*

1. $x_a \geq 1 + \delta$ *for all* $a \in \{(v_i, v_{i+1}) : i = 1, 3, 5, \ldots, p-2\} \cup \{(w_i, w_{i+1}) : i = 0, 2, 4, 6, \ldots, p-1\}$.
2. $l_P(x) \geq 1 + \beta F_{p-1}$ *for all* $P \in \mathcal{P}_1$*, with equality if and only if* $P = T_i$ *for some* $i = 2, 4, 6, \ldots, p-1$.
3. $l_P(x) \geq \beta F_p$ *for all* $P \in \mathcal{P}_2$*, with equality if and only if* $P = T_i$ *for some* $i = 1, 3, 5, \ldots, p$.
4. x *is a Nash flow under the perceived latencies* $l^p + \delta^p$.

Proof. The statements (i)–(iii) follow from Lemma 5.4, Lemma 5.5 and Lemma 5.6 in [122]. The last statement is clearly true for commodity 2 (since this commodity is not affected by the deviation on arc (s_1, e)). For commodity 1, all the flow-carrying paths T_i have a perceived latency of $Q_{T_i}(x) = 1+\beta(F_p+1)$, and the perceived latency along any other (s_1, t_1)-path is greater or equal than that. The actual latencies along these paths are $l_{T_i}(x) = 1 + \beta F_{p-1}$ for $i = 2, 4, 6, \ldots, p-1$, and $l_{T_0}(x) = 1 + \beta(F_{p-1} + 1)$. □

2.5 Biased price of anarchy

In this section we consider the *biased price of anarchy (BPoA)* introduced by Meir and Parkes [129]. Adapted to our setting, given an instance $\mathcal{I}$ and threshold functions θ, the biased price of anarchy is defined as

$$\mathrm{BPoA}(\mathcal{I}, \theta) = \sup_{\delta \in \Delta(\theta)} C(f^\delta)/C(f^*),$$

where f^* is a socially optimal flow. Note that because $C(f^*) \leq C(f)$ for every feasible flow f, we have $\mathrm{DR}(\mathcal{I}, \theta) \leq \mathrm{BPoA}(\mathcal{I}, \theta)$.

In this section we derive tight smoothness bounds on the biased price of anarchy for $(0, \beta)$-deviations. Our approach is a generalization of the framework of Correa, Schulz and Stier-Moses [46], which we now first give. For a class of latency functions $\mathcal{L}$ and $l \in \mathcal{L}$ we define

$$\mu(l) = \sup_{x,z \geq 0} \left\{ \frac{z[l(x) - l(z)]}{xl(x)} \right\} \quad \text{and} \quad \mu = \mu(\mathcal{L}) = \sup_{l \in \mathcal{L}} \mu(l).$$

In [46] it is shown that the price of anarchy is upper bounded by $1/(1-\mu)$, and that this bounds tight. Meir and Parkes [129] and independently Lianeas et al. [119] show (implicitly) that for non-atomic network routing games with latency functions in $\mathcal{L}$ it holds that BPoA $\leq (1+\beta)/(1-\mu)$.[12] However, this bound is not tight in general. Here we improve the result of [129] by obtaining a tight bound on the biased price of anarchy. Instead of providing a bound in terms of the original smoothness parameter μ, we include the parameter β in the definition.

Let $\mathcal{L}$ be a given set of latency functions and $\beta \geq 0$ fixed. For $l \in \mathcal{L}$, define

$$\hat{\mu}(l,\beta) = \sup_{x,z\geq 0}\left\{\frac{z[l(x)-(1+\beta)l(z)]}{xl(x)}\right\} \quad \text{and} \quad \hat{\mu}(\mathcal{L},\beta) = \sup_{l\in\mathcal{L}} \hat{\mu}(l,\beta). \tag{2.12}$$

Generalizing the approach in [46], we obtain the following result.

Theorem 2.19. *Let $\mathcal{L}$ be a set of non-negative, non-decreasing and continuous functions. Let $\mathcal{I}$ be a general multi-commodity instance with $(l_a)_{a\in A} \in \mathcal{L}^A$. Let x be δ-inducible for some $(0,\beta)$-deviation δ and let z be an arbitrary feasible flow. If $\hat{\mu}(\mathcal{L},\beta) < 1$, then*

$$C(x) \leq \frac{1+\beta}{1-\hat{\mu}(\mathcal{L},\beta)} C(z).$$

Moreover, this bound is tight if $\mathcal{L}$ contains all constant functions and is closed under scalar multiplication, i.e., for every $l \in \mathcal{L}$ and $\gamma \geq 0$, $\gamma l \in \mathcal{L}$.

Proof. Since x is a deviated Nash flow with respect to $l+\delta$, the following variational inequality holds:

$$\sum_{a\in A} x_a(l_a(x_a)+\delta_a(x_a)) \leq \sum_{a\in A} z_a(l_a(x_a)+\delta_a(x_a)).$$

We have

$$\begin{aligned} C(x) = \sum_{a\in A} x_a l_a(x_a) &\leq \sum_{a\in A} z_a l_a(x_a) + (z_a - x_a)\delta_a(x_a) \\ &\leq \sum_{x_a>z_a} z_a l_a(x_a) + \sum_{z_a\geq x_a} z_a(l_a(x_a)+\delta_a(x_a)) \\ &\leq \sum_{x_a>z_a} z_a l_a(x_a) + (1+\beta)\sum_{z_a\geq x_a} z_a l_a(x_a) \\ &\leq \sum_{x_a>z_a} z_a l_a(x_a) + (1+\beta)\sum_{z_a\geq x_a} z_a l_a(z_a), \end{aligned}$$

where the third inequality holds because δ is a $(0,\beta)$-deviation and the last inequality holds because the latency functions are non-decreasing. We then obtain

$$\begin{aligned} C(x) &\leq \sum_{x_a>z_a} z_a l_a(x_a) + (1+\beta)\sum_{z_a\geq x_a} z_a l_a(z_a) \\ &= \sum_{x_a>z_a} z_a[l_a(x_a) - (1+\beta)l_a(z_a) + (1+\beta)l_a(z_a)] + (1+\beta)\sum_{z_a\geq x_a} z_a l_a(z_a) \end{aligned}$$

[12]This upper bound in particular holds for the deviation ratio and price of risk aversion as well.

$$\begin{aligned}
&= (1+\beta)C(z) + \sum_{x_a > z_a} z_a[l_a(x_a) - (1+\beta)l_a(z_a)] \\
&\leq (1+\beta)C(z) + \hat{\mu}(\mathcal{L},\beta) \sum_{x_a > z_a} x_a l_a(x_a) \\
&\leq (1+\beta)C(z) + \hat{\mu}(\mathcal{L},\beta)C(x).
\end{aligned}$$

Thus, for $\hat{\mu}(\mathcal{L},\beta) < 1$, we obtain $C(x) \leq (1+\beta)/(1-\hat{\mu}(\mathcal{L},\beta))C(z)$.

We will now prove the tightness of the obtained bound if $\mathcal{L}$ contains all constant functions and is closed under scalar multiplication. For arbitrary $c \in \mathcal{L}$ and demand r, consider the parallel-arc instance in Figure 3.2. Clearly, a devi-

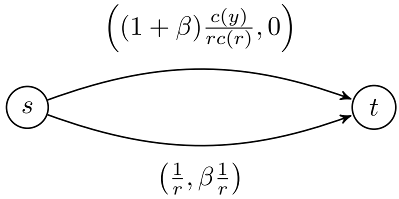

Figure 2.6: Example used in the proof of Theorem 2.19. The arcs are labeled by their respective (l_a, δ_a) functions. Note that $\delta \in \Delta(0,\beta)$.

ated Nash flow is given by $x = (x_1, x_2) = (r, 0)$, since then $l_1(x_1) + \delta_1(x_1) = l_2(x_2) + \delta_2(x_2) = (1+\beta)/r$. We have $C(x) = (1+\beta)$.

For a feasbile flow $z = (\epsilon, r-\epsilon)$. We have

$$C(z) = \frac{(1+\beta)\epsilon c(\epsilon) + (r-\epsilon)c(r)}{rc(r)} = \frac{rc(r) - \epsilon[c(r) - (1+\beta)c(\epsilon)]}{rc(r)}$$

which implies that, with z^* a socially optimal flow,

$$\frac{C(x)}{C(z^*)} \geq \frac{C(x)}{C(z)} = (1+\beta)\left(1 - \frac{\epsilon[c(r) - (1+\beta)c(\epsilon)]}{r \cdot c(r)}\right)^{-1}$$

In order to claim tightness we can choose $c \in \mathcal{L}$, and $r \geq \epsilon \geq 0$, arbitrary close to $\hat{\mu}(\mathcal{L},\beta)$. □

To see that our result improves the bound of $(1+\beta)/(1-\mu)$, note that

$$\hat{\mu}(\mathcal{L},\beta) \leq \hat{\mu}(\mathcal{L},0) \leq \mu.$$

We exemplify the increased strength of our general smoothness bound by deriving a closed form expression on the biased price of anarchy for affine latency functions.

Theorem 2.20. *Let $\mathcal{I}$ be a general multi-commodity instance with affine latency functions $(l_a)_{a\in A}$. Then*

$$BPoA(\mathcal{I},\beta) \leq \frac{(1+\beta)^2}{\frac{3}{4}+\beta}.$$

Note that the upper bound of $4(1+\beta)/3$ on the biased price of anarchy for affine latency functions given in [129, 119] is inferior to our bound.

Proof of Theorem 2.20. Let $\mathcal{L}$ be the set of all affine latency functions with non-negative coefficients. The claim follows from Theorem 2.19 by showing that $\hat{\mu}(\mathcal{L}, \beta) = \frac{1}{4(1+\beta)}$.

Let $l_a(y) = c_a y + d_a$ be an arbitrary affine latency function with $c_a, d_a \geq 0$. We need to show that

$$z_a[c_a x_a + d_a - (1+\beta)(c_a z_a + d_a)] \leq \frac{1}{4(1+\beta)} x_a [c_a x_a + d_a],$$

or, equivalently,

$$c_a[z_a x_a - (1+\beta) z_a^2] + d_a[z_a - z_a(1+\beta)] \leq c_a \left[\frac{1}{4(1+\beta)} x_a^2\right] + d_a \left[\frac{1}{4(1+\beta)} x_a\right].$$

It suffices to show that

$$z_a x_a - (1+\beta) z_a^2 \leq \frac{1}{4(1+\beta)} x_a^2 \quad \text{and} \quad z_a - z_a(1+\beta) \leq \frac{1}{4(1+\beta)} x_a.$$

The second inequality is always true, using the non-negativity of z_a, x_a and β. For the first inequality, we have

$$0 \leq \left(\frac{x_a}{2} - (1+\beta) z_a\right)^2 = (1+\beta)^2 z_a^2 + \frac{x_a^2}{4} - (1+\beta) x_a z_a,$$

which implies that

$$[1+\beta]\left(x_a z_a - (1+\beta) z_a^2\right) \leq \frac{x_a^2}{4}.$$

Dividing this inequality by $(1+\beta)$ gives the desired result. Further, we have tightness for $(x_a, z_a) = (1, \frac{1}{2(1+\beta)})$. $\square$

2.6 Heterogeneous populations

We provide tight bounds on the deviation ratio for general $(0, \beta)$-path deviations for instances on (single-commodity) series-parallel graphs and heterogeneous player populations. We first give the definition of a series-parallel graph.

Definition 2.21 (Series-parallel graph)**.** Let $G_i(V_i, A_i)$ with source $s_i \in V_i$ and target $t_i \in V_i$ for $i = 1, 2$ be two graphs. The *series-composition* of G_1 and G_2 is the graph $G = (V_1 \cup V_2, A_1 \cup A_2)$ in which we identify t_1 with s_2. The *parallel-composition* of G_1 and G_2 is the graph $G = (V_1 \cup V_2, A_1 \cup A_2)$ in which we identify s_1 with s_2, and also t_1 with t_2. A graph H is series-parallel if (i) it consists of a single edge; or (ii) it is a composition, either in series or parallel, of two series-parallel graphs.

We next present the main result of this section.

Theorem 2.22. *Let $\mathcal{I}$ be a single-commodity non-atomic network routing game on a series-parallel graph with heterogeneous players, demand distribution $r = (r_i)_{i\in[h]}$ with $\sum_{j\in[h]} r_i = 1$, and sensitivity distribution $\gamma = (\gamma_i)_{i\in[h]}$, with $\gamma_1 < \gamma_2 < \cdots < \gamma_h$. For $\beta \geq 0$ fixed, the β-deviation ratio is bounded by*

$$DR(\mathcal{I}, (0,\beta)) \leq 1 + \beta \cdot \max_{j\in[h]} \left\{ \gamma_j \left(\sum_{p=j}^{h} r_p \right) \right\}. \tag{2.13}$$

This bound is tight for all distributions r and γ.

For the homogeneous case, this bound reduces to $1+\beta$, which corresponds to the result in Corollary 2.6. To see this, note that for a series-parallel graph we always have $\eta_1 = 1$.[13]

Our bound on the deviation ratio also yields tight bounds on the price of risk aversion for series-parallel graphs and arbitrary heterogeneous risk-averse populations, both for the *mean-var objective* and *mean-std objective* as given in Section 2.7.1. We need the following technical lemma for the proof of the bound on the deviation ratio.

Lemma 2.23. *Let $0 \leq \tau_{k-1} \leq \cdots \leq \tau_1 \leq \tau_0$ and $c_i \geq 0$ for $i = 1, \ldots, k$ be given. We have $c_1\tau_0 + \sum_{i=1}^{k-1}(c_{i+1} - c_i)\tau_i \leq \tau_0 \cdot \max_{i=1,\ldots,k}\{c_i\}$.*

Proof. The statement is clearly true for $k = 1$. Now suppose the statement is true for some $k \in \mathbb{N}$. We will prove the statement for $k + 1$.

First suppose $c_{k+1} - c_k \leq 0$. Then we have

$$\begin{aligned} c_1\tau_0 + \sum_{i=1}^{k}(c_{i+1} - c_i)\tau_i &\leq c_1\tau_0 + \sum_{i=1}^{k-1}(c_{i+1} - c_i)\tau_i && \text{(using } \tau_k \geq 0\text{)} \\ &\leq \tau_0 \cdot \max_{i=1,\ldots,k}\{c_i\} && \text{(using induction hypothesis)} \\ &\leq \tau_0 \cdot \max_{i=1,\ldots,k+1}\{c_i\} && \text{(using non-negativity of } \tau_0\text{).} \end{aligned}$$

Secondly, suppose $c_{k+1} - c_k > 0$. Using $\tau_k \leq \tau_{k-1}$, we have

$$\begin{aligned} c_1\tau_0 + \sum_{i=1}^{k}(c_{i+1} - c_i)\tau_i &= c_1\tau_0 + \left[\sum_{i=1}^{k-1}(c_{i+1} - c_i)\tau_i\right] + (c_{k+1} - c_k)\tau_k \\ &\leq c_1\tau_0 + \left[\sum_{i=1}^{k-1}(c_{i+1} - c_i)\tau_i\right] + (c_{k+1} - c_k)\tau_{k-1} \\ &= c_1\tau_0 + \left[\sum_{i=1}^{k-2}(c_{i+1} - c_i)\tau_i\right] + (c_{k+1} - c_{k-1})\tau_{k-1} \end{aligned}$$

[13] If there would exist an alternating path with $\eta_1 > 1$, then the graph would have a minor corresponding to the (second) Braess graph G^2 on four nodes.

$$\begin{aligned} &\leq \tau_0 \cdot \max_{i=1,\ldots,k-2,k-1,k+1}\{c_i\} && \text{(induction hypothesis)} \\ &\leq \tau_0 \cdot \max_{i=1,\ldots,k+1}\{c_i\} && \text{(non-negativity of } \tau_0\text{).} \end{aligned}$$

Note that we apply the induction hypothesis with the set $\{c_1, \ldots, c_{k-1}, c_{k+1}\}$ of size k. □

We now continue with the proof of Theorem 2.22.

Proof of Theorem 2.22. Let $x = f^{\beta}$ be a β-deviated Nash flow with path deviations $(\delta_P)_{P\in\mathcal{S}} \in \Delta(\beta)$ and let $z = f^0$ be an original Nash flow. Let $X = \{a \in A : x_a > z_a\}$ and $Z = \{a \in A : z_a \geq x_a \text{ and } z_a > 0\}$ (arcs with $x_a = z_a = 0$ may be removed without loss of generality).

In order to analyze the ratio $C(x)/C(z)$ we first argue that we can assume without loss of generality that the latency function $l_a(y)$ is constant for values $y \geq x_a$ for all arcs $a \in Z$. To see this, note that we can replace the function $l_a(\cdot)$ with the function $\hat{l}_a$ defined by $\hat{l}_a(y) = l_a(x_a)$ for all $y \geq x_a$ and $\hat{l}_a(y) = l_a(y)$ for $y \leq x_a$. In particular, this implies that the flow x is still a β-deviated Nash flow for the same path deviations as before. This holds since for any path P the latency $l_P(x)$ remains unchanged if we replace the function l_a by $\hat{l}_a$.

By definition of arcs in Z, we have $x_a \leq z_a$ and therefore $\hat{l}_a(z_a) = l_a(x_a) \leq l_a(z_a)$. Let z' be an original Nash flow for the instance with l_a replaced by $\hat{l}_a$. Then we have $C(z') \leq C(z)$ using the fact that series-parallel graphs are immune to the Braess paradox, see Milchtaich [133, Lemma 4]. Note that, in particular, we find $C(x)/C(z) \leq C(x)/C(z')$. By repeating this argument, we may without loss of generality assume that all latency functions l_a are constant between x_a and z_a for $a \in Z$. Afterwards, we can even replace the function $\hat{l}_a$ by a function that has the constant value of $l_a(x_a)$ everywhere.

In the remainder of the proof, we will denote P_j as a flow-carrying arc for sensitivity class $j \in [h]$ that maximizes the path latency amongst all flow-carrying path for sensitivity class $j \in [h]$, i.e., $P_j = \text{argmax}_{P\in\mathcal{P}:x_{P,j}>0}\{l_P(x)\}$. Moreover, there also exists a path P_0 with the property that $z_a \geq x_a$ and $z_a > 0$ for all arcs $a \in P_0$ (see, e.g., Lemma 2 [133]).

For fixed $a < b \in \{1, \ldots, h\}$, the Nash conditions imply that (these steps are of a similar nature as Lemma 1 [78])

$$\begin{aligned} l_{P_a}(x) + \gamma_a \cdot \delta_{P_a}(x) &\leq l_{P_b}(x) + \gamma_a \cdot \delta_{P_b}(x) \\ l_{P_b}(x) + \gamma_b \cdot \delta_{P_b}(x) &\leq l_{P_a}(x) + \gamma_b \cdot \delta_{P_a}(x). \end{aligned}$$

Adding up these inequalities implies that $(\gamma_b - \gamma_a)\delta_{P_b}(x) \leq (\gamma_b - \gamma_a)\delta_{P_a}(x)$, which in turn yields that $\delta_{P_b}(x) \leq \delta_{P_a}(x)$ (using that $\gamma_a < \gamma_b$ if $a < b$). Furthermore, we also have

$$l_{P_1}(x) + \gamma_1\delta_{P_1}(x) \leq l_{P_0}(x) + \gamma_1\delta_{P_0}(x), \tag{2.14}$$

and $l_{P_0}(x) = l_{P_0}(z) \leq l_{P_1}(z) \leq l_{P_1}(x)$, which can be seen as follows. The equality follows from the fact that l_a is constant for all $a \in Z$ and, by choice, P_0 only consists of arcs in Z. The first inequality follows from the Nash conditions of the original Nash flow z, since there exists a flow-decomposition in which the path P_0 is used (since the flow on all arcs of P_0 is strictly positive in z). The second inequality follows from the fact that

$$\sum_{e \in P_1} l_e(z_e) = \sum_{e \in P_1 \cap X} l_e(z_e) + \sum_{e \in P_1 \cap Z} l_e(z_e) \leq \sum_{e \in P_1 \cap X} l_e(x_e) + \sum_{e \in P_1 \cap Z} l_e(x_e)$$

using that $z_e \leq x_e$ for $e \in X$ and the fact that latency functions for $e \in Z$ are constant. In particular, we find that $l_{P_0}(x) \leq l_{P_1}(x)$. Adding this inequality to (2.14), we obtain $\gamma_1 \delta_{P_1}(x) \leq \gamma_1 \delta_{P_0}(x)$ and therefore $\delta_{P_1}(x) \leq \delta_{P_0}(x)$. Thus $\delta_{P_h}(x) \leq \delta_{P_{h-1}}(x) \leq \cdots \leq \delta_{P_1}(x) \leq \delta_{P_0}(x)$. Moreover, by using induction it can be shown that

$$l_{P_j}(x) \leq l_{P_0}(x) + \gamma_1 \delta_{P_0}(x) + \left[\sum_{g=1}^{j-1} (\gamma_{g+1} - \gamma_g) \delta_{P_g}(x) \right] - \gamma_j \delta_{P_j}(x). \tag{2.15}$$

The case $j = 1$ is precisely (2.14). Now suppose it holds for some j, then we have

$$\begin{aligned} l_{P_{j+1}}(x) &\leq l_{P_j}(x) + \gamma_{j+1} \delta_{P_j}(x) - \gamma_{j+1} \delta_{P_{j+1}}(x) \quad \text{(Nash condition for } P_{j+1}) \\ &\leq l_{P_0}(x) + \gamma_1 \delta_{P_0}(x) + \left[\sum_{g=1}^{j-1} (\gamma_{g+1} - \gamma_g) \delta_{P_g}(x) \right] - \gamma_j \delta_{P_j}(x) \\ &\quad + \gamma_{j+1} \delta_{P_j}(x) - \gamma_{j+1} \delta_{P_{j+1}}(x) \quad \text{(induction hypothesis)} \\ &= l_{P_0}(x) + \gamma_1 \delta_{P_0}(x) + \left[\sum_{g=1}^{j} (\gamma_{g+1} - \gamma_g) \delta_{P_g}(x) \right] - \gamma_{j+1} \delta_{P_{j+1}}(x), \end{aligned}$$

which shows the result for $j + 1$. Using (2.15), we then have

$$\begin{aligned} C(x) &\leq \sum_{j=1}^{h} r_j l_{P_j}(x) \quad \text{(by choice of the paths } P_j) \\ &\leq \sum_{j=1}^{h} r_j \left(l_{P_0}(x) + \gamma_1 \delta_{P_0}(x) + \left[\sum_{g=1}^{j-1} (\gamma_{g+1} - \gamma_g) \delta_{P_g}(x) \right] - \gamma_j \delta_{P_j}(x) \right) \\ &= l_{P_0}(x) + \gamma_1 \delta_{P_0}(x) + \sum_{j=1}^{h} (r_{j+1} + \cdots + r_h)(\gamma_{j+1} - \gamma_j) \delta_{P_j}(x) - r_j \gamma_j \delta_{P_j}(x) \\ &\leq l_{P_0}(x) + \gamma_1 \delta_{P_0}(x) \\ &\quad + \sum_{j=1}^{h-1} \Big[(r_{j+1} + \cdots + r_h) \gamma_{j+1} - (r_j + r_{j+1} + \cdots + r_h) \gamma_j \Big] \delta_{P_j}(x). \end{aligned}$$

In the last inequality, we leave out the last negative term $-r_h \gamma_h \delta_{P_h}(x)$. Note that $\gamma_1 = (r_1 + \cdots + r_h)\gamma_1$ since we have normalized the demand to 1. We can then apply Lemma 2.23 with $\tau_i = \delta_{P_i}(x)$ for $i = 0, \dots, h-1$ and $c_i = \gamma_i \cdot \sum_{p=i}^{h} r_p$ for $i = 1, \dots, k$. Continuing the estimate, we get

$$C(x) \leq l_{P_0}(x) + \max_{j \in [h]} \left\{ \gamma_j \cdot \sum_{p=j}^{h} r_p \right\} \cdot \delta_{P_0}(x) \leq \left[1 + \beta \cdot \max_{j \in [h]} \left\{ \gamma_j \left(\sum_{p=j}^{h} r_p \right) \right\} \right] C(z)$$

where for the second inequality we use that $\delta_{P_0}(x) \leq \beta l_{P_0}(x)$, which holds by definition, and $l_{P_0}(x) = l_{P_0}(z) = C(z)$, which holds because z is an original Nash flow and all arcs in P_0 have strictly positive flow in z (and because of the fact that all arcs in P_0 have constant latency functions).

To prove tightness, fix $j \in [h]$ and consider the following instance on two arcs. We take $(l_1(y), \delta_1(y)) = (1, \beta)$ and $(l_2(y), \delta_2(y))$ with $\delta_2(y) = 0$ and $l_2(y)$ a strictly increasing function satisfying $l_2(0) = 1 + \epsilon$ and $l_2(r_j + r_{j+1} + \dots + r_h) = 1 + \gamma_j \beta$, where $\epsilon < \gamma_j \beta$. The (unique) original Nash flow is given by $z = (z_1, z_2) = (1, 0)$ with $C(z) = 1$. The (unique) β-deviated Nash flow x is given by $x = (x_1, x_2) = (r_1 + r_2 + \dots + r_{j-1}, r_j + r_{j+1} + \dots + r_h)$ with $C(x) = 1 + \beta \cdot \gamma_j (r_j + \dots + r_h)$. Since this construction holds for all $j \in [h]$, we find the desired lower bound. $\square$

Connection to approximate Nash flows. In this part we investigate further the relation between inducible Nash flows and approximate Nash flows. In case the population is *homogeneous*, i.e., when $\gamma_{ij} = 1$ for all pairs (i, j), the result in Theorem 2.22 actually extends to hold for the ϵ-stability ratio. This turns out to be no surprise, as flows induced by general path deviations can be shown to correspond to approximate Nash flows in a certain sense. That is, every β-approximate Nash flow can be induced by some $\delta \in \Delta(0, \beta)$. This is shown in Proposition 2.24.

Proposition 2.24. *Let $\mathcal{I}$ be a single-commodity non-atomic network routing game with homogeneous players. Then f is a β-approximate Nash flow if and only if f is a β-deviated Nash flow.*

Proof. Let f be a β-deviated Nash flow for some set of path deviations $\delta \in \Delta(0, \beta)$. For $P \in \mathcal{S}$, we have

$$l_P(f) \leq l_P + \delta_P(f) \leq l_{P'}(f) + \gamma \delta_{P'}(f) \leq (1 + \beta) l_{P'}(f)$$

where we use the non-negativity of $\delta_P(f)$ in the first inequality, the Nash condition for f in the second inequality, and the fact that $\delta_{P'}(f) \leq \beta l_{P'}(f)$ in the third inequality. By definition, it now holds that f is also a β-approximate Nash flow.

Reversely, let f be a β-approximate Nash flow. We show that there exist $(0, \beta)$-path deviations δ_P such that f is inducible with respect to these path deviations. Let $P_1, \dots, P_k$ be the set of flow-carrying paths under f, and assume without loss of generality that $l_{P_1}(f) \leq l_{P_2}(f) \leq \dots \leq l_{P_k}(f)$. We define $\delta_{P_i}(f) =$

$l_{P_k}(f) - l_{P_i}(f)$ for $i = 1, \ldots, k$. Using the Nash condition for the path P_k, we find

$$\delta_{P_i}(f) = l_{P_k}(f) - l_{P_i}(f) \leq (1+\beta) l_{P_i}(f) - l_{P_i}(f) = \beta l_{P_i}(f)$$

which shows that these path deviations are feasible. Moreover, we take $\delta_Q(f) = \beta l_Q(f)$ for all the paths $Q \in \mathcal{P} \setminus \{P_1, \ldots, P_k\}$ which are not flow-carrying under f. Now, let $i \in \{1, \ldots, k\}$ be fixed. Then for any $j \in \{1, \ldots, k\}$, we have

$$l_{P_i}(f) + \delta_{P_i}(f) = l_{P_k}(f) = l_{P_j}(f) + \delta_{P_j}(f)$$

and for any $Q \in \mathcal{P} \setminus \{P_1, \ldots, P_k\}$, we have

$$l_{P_i}(f) + \delta_{P_i}(f) = l_{P_k}(f) \leq (1+\beta) l_Q(f) = l_Q(f) + \delta_Q(f),$$

using the Nash condition for the path P_k and the definition of $\delta_Q(f)$. This shows that f is indeed a β-deviated Nash flow. □

We show that this correspondence is not true for heterogeneous populations by providing tight bounds on the ϵ-stability ratio for heterogeneous populations on instances with a series-parallel network topology. For sake of comparison we write $\epsilon = \beta\gamma$ for some given $\beta \geq 0$.

Theorem 2.25. *Let $\mathcal{I}$ be a single-commodity non-atomic network routing game on a series-parallel graph with heterogeneous players, demand distribution $r = (r_i)_{i \in [h]}$ with $\sum_{j \in [h]} r_i = 1$, and sensitivity distribution $\gamma = (\gamma_i)_{i \in [h]}$, with $\gamma_1 < \gamma_2 < \cdots < \gamma_h$. For $\beta \geq 0$ fixed, write $\epsilon = \beta\gamma$. Then the ϵ-stability ratio is bounded by*

$$\epsilon\text{-}SR(\mathcal{I}) \leq 1 + \beta \sum_{j=1}^{h} r_j \gamma_j. \tag{2.16}$$

This bound is tight for all distributions r and γ.

Proof. For $j \in [k]$, let $\bar{P}_j$ be a path maximizing $l_P(x)$ over all flow-carrying paths $P \in \mathcal{P}$ of type j. Moreover, there exists a path π such that $x_a \leq z_a$ and $z_a > 0$ for all $a \in \pi$ (see, e.g., Milchtaich [133]). We then have (this is also reminiscent of an argument by Lianeas et al. [120]):

$$l_{\bar{P}_j}(x) \leq (1 + \beta\gamma_j) l_\pi(x) = (1 + \beta\gamma_j) \sum_{a \in \pi} l_a(x_a).$$

Note that, by definition of the alternating path π, we have $x_a \leq z_a$ for all $a \in \pi$. Continuing with the estimate, we find $l_{\bar{P}_j}(x) \leq (1 + \beta\gamma_j) \sum_{a \in \pi} l_a(z_a)$ and thus

$$C(x) \leq \sum_{j \in [h]} r_j l_{\bar{P}_j}(x) \leq \sum_{j \in [h]} r_j (1 + \beta\gamma_j) \sum_{a \in \pi} l_a(z_a) = C(z) \Bigg(\sum_{j \in [h]} r_j (1 + \beta\gamma_j) \Bigg).$$

Since $\sum_{j\in[h]} r_j = 1$, we get the desired result. Note that we use $C(z) = \sum_{a\in\pi} l_a(z_a)$, which is true because there exists a flow-decomposition of z in which π is flow-carrying (here we use $z_a > 0$ for all $a \in \pi$).

Tightness follows by considering an instance with arc set $\{0, 1, \ldots, h\}$ where the zeroth arc has latency $l_0(y) = 1$ and the arcs $j \in \{1, \ldots, h\}$ have latency $l_j(y) = 1 + \beta\gamma_j$. An original Nash flow is given by $f^0 = (z_0, z_1, \ldots, z_h) = (1, 0, \ldots, 0)$, and an ϵ-approximate Nash flow is given by $f^\epsilon = (x_0, x_1, \ldots, x_h) = (0, r_1, r_2, \ldots, r_h)$. $\square$

It is not hard to see that the bound on the β-deviation ratio is always smaller or equal than the bound on the ϵ-stability ratio, under the correspondence $\epsilon = \beta\gamma$.[14]

Remark 2.26. We remark that in general other relations between approximate and inducible Nash flows could still hold. We only consider the natural correspondence $\epsilon = \beta\gamma$ arising from (2.6). For example, is it possible that for some given ϵ under a heterogeneous population, every ϵ-approximate Nash flow is inducible a $(0, \beta')$-deviation for some $\beta' = \beta'(\epsilon)$ large enough?

2.7 Applications

We use our results on the deviation ratio and the biased price of anarchy obtained in the previous sections to derive several new results below.

2.7.1 Price of risk aversion

Nikolova and Stier-Moses [139] (see also [119, 138]) consider non-atomic network routing games with uncertain latencies. Here the deviations correspond to variances $(v_a)_{a\in A}$ of some random variable ζ_a (with expectation zero). The *perceived latency* of a path $P \in \mathcal{P}$ with respect to a flow f is then defined as

$$q_P^\gamma(f) = l_P(f) + \gamma v_P(f),$$

where $\gamma \geq 0$ is a parameter representing the *risk-aversion* of the players. They consider two different objectives as to how the deviation $v_P(f)$ of a path P is defined:

1. *mean-var objective*: $v_P(f) = \sum_{a\in P} v_a(f_a)$
2. *mean-std objective*: $v_P(f) = (\sum_{a\in P} v_a(f_a))^{\frac{1}{2}}$.

Note that for the mean-var objective there is an equivalent arc-based definition, where the perceived latency of every arc $a \in A$ is defined as $q_a^\gamma(f_a) = l_a(f_a) + \gamma v_a(f_a)$. They define the *price of risk aversion* [139] as the worst-case ratio

[14] This follows from Markov's inequality: for a random variable Y and $t \geq 0$, we have $P(Y \geq t) \leq E(Y)/t$.

$C(x)/C(z)$, where x is a *risk-averse* Nash flow with respect to $q^\gamma = l + \gamma v$ and z is a *risk-neutral* Nash flow with respect to l.[15] In their analysis, it is assumed that the *variance-to-mean-ratio* of every arc $a \in A$ under the risk-averse flow x is bounded by some constant $\kappa \geq 0$, i.e., $v_a(x_a) \leq \kappa l_a(x_a)$ for all $a \in A$. Under this assumption, they prove that the price of risk aversion $\mathrm{PRA}(\mathcal{I}, \gamma, \kappa)$ of single-commodity instances $\mathcal{I}$ with non-negative and non-decreasing latency functions is at most $1 + \gamma\kappa\lceil (n-1)/2 \rceil$, where n is the number of nodes.

We now elaborate on the relation to our deviation ratio. The main technical difference is that in [139] the variance-to-mean ratio is only considered for the respective flow values x_a. Note however that if we write for every $a \in A$, $v_a(x_a) = \lambda_a l_a(x_a)$ for some $0 \leq \lambda_a \leq \kappa$, then the deviation function $\delta_a(y) = \gamma\lambda_a l_a(y)$ has the property that $x = f^\delta$ is δ-inducible with $\delta \in \Delta(0, \gamma\kappa)$. It follows that for every instance $\mathcal{I}$ and parameters γ, κ, $\mathrm{PRA}(\mathcal{I}, \gamma, \kappa) \leq \mathrm{DR}(\mathcal{I}, (0, \gamma\kappa))$.

We obtain the following bound on the price of risk aversion for multi-commodity networks with a common source.

Theorem 2.27. *The price of risk aversion for a common source multi-commodity instance $\mathcal{I}$ with non-negative and non-decreasing latency functions, variance-to-mean-ratio $\kappa > 0$ and risk-aversion parameter $\gamma \geq -1/\kappa$ is at most*

$$PRA(\mathcal{I}, \gamma, \kappa) \leq \begin{cases} 1 - \gamma\kappa/(1+\gamma\kappa)\lceil (n-1)/2 \rceil r & \textit{for } -1/\kappa < \gamma \leq 0 \\ 1 + \gamma\kappa\lceil (n-1)/2 \rceil r & \textit{for } \gamma \geq 0. \end{cases}$$

Moreover, these bounds are tight in all its parameters if $n = 2m+1$ and almost tight if $n = 2m$. In particular, for single-commodity instances we obtain tightness for all $n \in \mathbb{N}$.

Note that Theorem 2.27 generalizes the result in [139] to multi-commodity networks with a common source and to negative risk-aversion parameters. Further, it establishes that the bound is tight in all its parameters.

Proof of Theorem 2.27. Recall that the deviations $\delta_a = \gamma v_a$ can be interpreted as θ-deviations with

$$\theta_a^{\min} = \begin{cases} 0 & \text{if } \gamma \geq 0 \\ \gamma\kappa l_a & \text{if } -1/\kappa < \gamma \leq 0 \end{cases} \quad \text{and} \quad \theta_a^{\max} = \begin{cases} \gamma\kappa l_a & \text{if } \gamma \geq 0 \\ 0 & \text{if } -1/\kappa < \gamma \leq 0. \end{cases}$$

Here, the restriction $\gamma > -1/\kappa$ is necessary to satisfy Assumption 2.2. The theorem now follows directly from Theorem 2.5, Example 2.13 and Theorem 2.14. □

2.7.2 Stability of Nash flows under small perturbations

We next show that our results can be used to bound the relative error in social cost incurred by small latency perturbations.

[15] The existence of a risk-averse Nash flow is proven in [138].

We introduce some more notation. We say that $(\tilde{l}_a)_{a\in A}$ are *ϵ-perturbed latency functions* with respect to $(l_a)_{a\in A}$ if

$$\sup_{a\in A,\ x\geq 0}\left|\frac{l_a(x)-\tilde{l}_a(x)}{l_a(x)}\right|\leq \epsilon$$

for some small $\epsilon > 0$. We are interested in bounding the relative error in social cost due to ϵ-perturbations of the latency functions. More precisely, the *relative error in social cost* is defined as the ratio

$$\frac{C(\tilde{f})-C(f)}{C(f)},$$

where f is a Nash flow with respect to $(l_a)_{a\in A}$ and $\tilde{f}$ is a Nash flow with respect to ϵ-perturbed latency functions $(\tilde{l}_a)_{a\in A}$. To the best of our knowledge, this notion has not been studied in the literature before.

The theorem below establishes an upper bound on the relative error in social cost. In particular, for small ϵ-perturbations the theorem implies that the relative error is asymptotically $O(\epsilon r n)$.

Theorem 2.28. *Let $\mathcal{I}$ be a common source multi-commodity instance with non-negative and non-decreasing latency functions $(l_a)_{a\in A}$. Let f be a Nash flow with respect to $(l_a)_{a\in A}$ and let $\tilde{f}$ be a Nash flow with respect to ϵ-perturbed latency functions $(\tilde{l}_a)_{a\in A}$ for some $0 < \epsilon < 1$. Then the relative error in social cost satisfies*

$$\frac{C(\tilde{f})-C(f)}{C(f)}\leq \frac{2\epsilon}{1-\epsilon}\cdot\left\lceil\frac{n-1}{2}\right\rceil r.$$

Proof. Note that the ϵ-perturbation $l-\tilde{l}$ can be seen as a $(-\epsilon,\epsilon)$-deviation. Using Theorem 2.5, we obtain

$$\frac{C(\tilde{f})}{C(f)}\leq 1+\frac{2\epsilon}{1-\epsilon}\cdot\left\lceil\frac{n-1}{2}\right\rceil r.$$

The claim follows. □

2.8 Conclusion

We introduced a unifying model to study the impact of (bounded) worst-case latency deviations in non-atomic selfish routing games. We demonstrated that the deviation ratio is a useful measure to assess the cost deterioration caused by such deviations. Among potentially other applications, we showed that the deviation ratio provides bounds on the price of risk aversion and the relative error in social cost if the latency functions are subject to small perturbations. In turns out that, e.g., risk averse behavior can cause a serious deterioration in the quality of Nash flows, not only in single commodity instances [139]. In particular, the results for single-commodity instances in [139] can be generalized to common-source multi-commodity instances; for general multi-commodity instances the deterioration is even more severe.

Our bounds reveal a (mostly) linear dependence on model parameters for common-source multi-commodity instances (when all other parameters are kept fixed). Overall, these bounds, and already the work in [139], seem to provide an interesting complementary view to more traditional approaches for quantifying the inefficiency of (deviated) Nash flows. Most importantly, the deviation ratio is not independent of the network topology. We conjecture this remains to hold true, even when considering specific classes of latency functions as is done for the biased price of anarchy in Section 2.5.

For future research, studying the impact of (bounded) worst-case deviations on the input data of more general classes of games (e.g., in other congestion models as given in Section 1.3) is an interesting and challenging direction for future work. Moreover, it would also be interesting to obtain tight bounds on the deviation ratio for specific classes of latency functions. The biased price of anarchy always yields an upper bound on the deviation ratio, but in general these bounds are not tight. Can one give a tight bound, e.g., for the deviation ratio under affine latency functions? Some preliminary insights suggest that this question only seems interesting for small values of β. We next elaborate on this shortly. For large β, the bound we obtained in Theorem 2.20 is approximately equal to $1 + \beta$. Moreover, it is not hard to show a lower bound of $1 + \beta$ on the deviation ratio with affine latency functions. The latter can be established using the Pigou network topology consisting of two parallel arcs. However, for small values of β, one can obtain better lower bounds than $1+\beta$ for the deviation ratio when considering instances with affine latencies (using generalized Braess graph topologies).[16]

Another interesting open question is to determine (tight) upper bounds on the deviation ratio (or price of risk aversion) for general multi-commodity instances. We already know, because of the lower bound construction in Section 2.4, that an exponential dependence on the size of the network is the best we can hope for. Moreover, can one go beyond series-parallel network topologies when considering heterogeneous populations, either for additive path deviations or general path deviations?

[16] This is established in personal notes of the author. In particular, it can be shown that no $K > 0$ exists so that $1 + K\beta$ is an upper bound, on the deviation ratio for instances with affine latency functions, for *all* $\beta > 0$.

Chapter 3

On pure Nash equilibria in Rosenthal congestion games

3.1 Introduction

Congestion games constitute an important class of non-cooperative games which have been studied intensively since their introduction by Rosenthal [150]. In a *congestion game* there is a (finite) set of players that compete over a (finite) set of resources. Each resource is associated with a non-negative and non-decreasing cost (or delay) function which specifies its cost depending on the total number of players using it. Every player chooses a subset of resources from a set of available resource subsets (corresponding to the player's strategies) and experiences a cost equal to the sum of the costs of the chosen resources. The goal of each player is to minimize her individual cost. Congestion games are both theoretically appealing and practically relevant. For example, they find their applications in network routing, resource allocation and scheduling problems.

In a seminal paper, Rosenthal [150] establishes the existence of pure Nash equilibria in (unweighted) congestion games. He proves this result through the use of an *exact potential function* which assigns a value to each strategy profile such that the difference in potential value of any two strategy profiles corresponding to a unilateral deviation of a player is equal to the difference in cost experienced by that player. Rosenthal proves that every congestion game admits an exact potential function, also known as *Rosenthal's potential.* As a consequence, every best response sequence must converge to a pure Nash equilibrium because the game is finite. Further, this shows that the set of pure Nash equilibria corresponds to the set of *local*[1] *minima* of Rosenthal's potential. Especially this correspondence has helped to shed light on several important aspects of congestion games in recent years.

Computational complexity of pure Nash equilibria. One of the most predominant

[1]Here, we mean the natural locality induced by unilateral player deviations.

aspects that has been studied intensively in recent years is the computational complexity of finding a pure Nash equilibrium. In a seminal paper, Fabrikant et al. [75] show that the problem of finding a pure Nash equilibrium is PLS-complete, both for symmetric congestion games, where all players have the same strategy set, and non-symmetric network congestion games, see Example 1.2. In particular, this suggests that a polynomial time algorithm for finding a pure Nash equilibrium is unlikely to exist for these games. In their proof they construct instances of non-symmetric network congestion games where any best response sequence has exponential length. Ackermann et al. [1] strengthen this result by exhibiting instances of symmetric network congestion games for which every best response sequence (from certain initial configurations) has exponential length. On the positive side, they prove that best response dynamics converge in polynomial time for non-symmetric matroid congestion games, where the available resource subsets of the players correspond to bases of a given matroid (see Section 3.3.3). The authors also show that, in some sense, this is the only class of congestion games for which this property holds true.

Most previous works in this context focus on the analysis of decentralized dynamics to reach a pure Nash equilibrium (see, e.g., [1, 36, 31, 74, 75, 80, 106]); said differently, these works focus on finding a local minimum of Rosenthal's potential. Much less is known about the problem of computing a pure Nash equilibrium that corresponds to a *global* minimum. Fabrikant et al. [75] show that a pure Nash equilibrium can be computed in polynomial time for symmetric network congestion games. The authors observe that in this case a global minimum of Rosenthal's potential can be computed by a reduction to a min-cost flow problem (if all cost functions are non-decreasing). Note that this is in stark contrast with the fact that best response dynamics might need exponential time in this case [1].

Only very recently, Del Pia et al. [54] make further progress along these lines. The authors consider congestion games where the strategy sets of the players are given implicitly by a *polyhedral description* (see also [27]). More precisely, for each player i the incidence vectors of the strategies are defined as the binary vectors in a polytope $P_i = \{x : A_i x \leq b_i\}$, where A_i is an integral matrix and b_i is an integral vector. They (mostly) focus on the case where the matrix A_i is *totally unimodular* (see below for formal definitions) and thus the describing polytope P_i is integral (i.e., all its extreme points are integral); they term these games *totally unimodular (TU) congestion games.* For symmetric TU congestion games (when all A_i, b_i are identical), they devise an aggregation/decomposition framework that reduces the problem of finding a global minimum of Rosenthal's potential to an integer linear programming problem. Using this framework, they show that pure Nash equilibria can be computed efficiently for symmetric TU congestion games. The authors also show that this problem is PLS-complete for various non-symmetric TU congestion games. Further, they show that their framework can be adapted to the case of non-symmetric matroid congestion games.

Inefficiency of pure Nash equilibria. Another important aspect that has has been the subject of intensive research in recent years is the inefficiency of pure Nash equilibria in congestion games (see, e.g., [4, 8, 21, 34, 32, 53, 52, 80, 84, 77, 117, 123, 155]). Here the goal is to assess the *social cost*, defined as the sum of the costs of the players, of a pure Nash equilibrium relative to an optimal outcome (as explained in Section 1.2.3). Koutsoupias and Papadimitriou [117] introduce the *price of anarchy* as the ratio between the worst social cost of a Nash equilibrium and the social cost of an optimum. Anshelevich et al. [7] define the *price of stability* as the ratio between the best social cost of a Nash equilibrium and the social cost of an optimum.

Fotakis [80] reveals an intriguing connection between the price of stability of symmetric network congestion games and the price of anarchy of their *non-atomic* counterparts. More specifically, he shows that for symmetric network congestion games the ratio between the social cost of a global minimum of Rosenthal's potential and the social cost of a social optimum is at most $\rho(\mathcal{D})$, where $\rho(\mathcal{D})$ is a tight bound on the price of anarchy for *non-atomic* network congestion games with latency functions in class $\mathcal{D}$ introduced by Correa et al. [45], see also Section 2.2.1.1. In particular, this implies that the price of stability of symmetric network congestion games with cost functions in $\mathcal{D}$ is at most $\rho(\mathcal{D})$. For example, this parameter equals 4/3 for the class of affine functions and $(27 + 6\sqrt{3})/23 \approx 1.63$ for quadratic functions. These type of bounds fall within Roughgarden's *smoothness framework* [155]. Further, Fotakis [80] also shows that for symmetric network congestion games on *extension-parallel graphs*, every Nash equilibrium is a Rosenthal minimizer and thus the upper bound of $\rho(\mathcal{D})$ even holds for the price of anarchy of these games.

More recently, Feldman et al. [77] study the price of anarchy in *large games*. Basically, they show that the smoothness parameter introduced in [155] also provides an upper bound on the price of anarchy of many games (or mechanisms) when the number of involved players grows large.

Extensions and variations of Rosenthal's model. Moreover, in recent years, several extensions of Rosenthal's congestion game were proposed to incorporate aspects which are not captured by the standard model. For example, these extensions include risk sensitivity of players in uncertainty settings [146], altruistic player behavior [23, 28] and congestion games with taxes [22]. We elaborate in more detail on these extensions in Section 3.4.3. These games were studied intensively with the goal to obtain a precise understanding of the price of anarchy.

We introduce a new model, which we term *perception-parameterized congestion games*, that capture various extensions of Rosenthal's congestion game model introduced recently. The key idea here is to parameterize both the perceived cost of each player and the social cost function. Intuitively, each player perceives the load induced by the other players by an extent of $\rho \geq 0$, while the system designer estimates that each player perceives the load of all others by an extent of $\sigma \geq 0$.

We illustrate our model by means of a simple example; formal definitions of our perception-parameterized congestion games are given in Section 3.4. Suppose we are given a set of m resources and that every player has to choose precisely one of these resources. The cost of a resource e is given by a cost function c_e that maps the *load* on e to a real value. In the classical setting, the load of a resource e is defined as the total number of players x_e using e. That is, the cost that player i experiences when choosing resource e is $c_e(x_e)$. In contrast, in our setting players have different perceptions of the load induced by the other players. More precisely, the *perceived load* of player i choosing resource e is $1 + \rho(x_e - 1)$, where $\rho \geq 0$ is some parameter.[2] Consequently, the perceived cost of player i for choosing e is $c_e(1 + \rho(x_e - 1))$. Note that as ρ increases players care more about the presence of other players. In addition, we introduce a similar parameter $\sigma \geq 0$ for the social cost objective, i.e., the social cost is defined as $\sum_e c_e(1 + \sigma(x_e - 1))x_e$. Intuitively, this can be seen as the system designer's estimate of how each player perceives the load of the other players.

3.1.1 Our contributions

In order to explain our contributions for the classical Rosenthal model [150], we first introduce some terminology. We consider *polytopal congestion games* in which the incidence vectors of the strategies of player i are given by the binary vectors in a polytope $P_i = \{x : Ax \leq b_i\}$, where A is an integral matrix and b_i is an integral vector. Given the polytopes of all players, a strategy profile naturally corresponds to an integral vector in the *aggregation polytope* $P_N = \sum_i P_i$. We identify two general properties of the aggregation polytope P_N which are sufficient for our results to go through, namely the *integer decomposition property (IDP)* and the *box-totally dual integrality property (box-TDI)* (formal definitions are given below). The integer decomposition property is needed to decompose a load profile in P_N to a respective strategy profile of the players. Intuitively, the box-TDI property ensures that the intersection of a polytope with an arbitrary integer box is an integral polytope. This property is mostly needed for technical reasons.

Our main contributions for polytopal congestion games are as follows:

1. We show that the price of stability of polytopal congestion games satisfying IDP and box-TDI is bounded by $\rho(\mathcal{D})$ (Section 3.3.1). To this aim, we introduce a novel structural property (which we term the *symmetric difference decomposition property*) and show that it is satisfied by our games. By exploiting this property, we can generalize the bound of Fotakis [80] for symmetric network congestion games to the (much) larger class of polytopal congestion games satisfying IDP and box-TDI. We also prove that our bound is tight.
2. We derive an efficient algorithm for computing a feasible load profile minimizing Rosenthal's potential for polytopal congestion games satisfying IDP and box-TDI (Section 3.3.2). The time complexity of this algorithm is polynomial in the number of players and resources, the encoding length of

[2] In general, the perception parameter ρ might be player-specific.

$\sum_i b_i$ and the complexity of a separation oracle for the aggregation polytope. This generalizes the framework of [54] for symmetric TU congestion games and non-symmetric matroid congestion games, both being special cases of our polytopal congestion games.

3. We give several examples of polytopal congestion games satisfying IDP and box-TDI (Section 3.3.3). These examples include symmetric TU congestion games, common source network congestion games, non-symmetric matroid congestion games and certain symmetric matroid intersection congestion games (in particular, r-arborescences and strongly base-orderable matroids).
4. We show that our techniques can be used to extend some results on the computation and inefficiency of *strong equilibria* of the 'bottleneck variant' of our polytopal congestion games (Section 3.3.4). In particular, we show that strong equilibria can be computed in polynomial time for *polytopal bottleneck congestion games* satisfying IDP and box-TDI (see below for definitions). This generalizes a result by Harks et al. [99].

For perception-parameterized congestion games with affine cost functions, we give unifying price of anarchy bounds, as well as novel price of stability bounds. Applications can be found in Section 3.4.3; see also Table 3.1 for a comparison with existing work.

1. We prove that the price of anarchy can be upper bounded by
$$\max\left\{\rho+1, \frac{2\rho(1+\sigma)+1}{\rho+1}\right\} \tag{3.1}$$
for a certain range of pairs ρ and $\sigma \geq 1/2$ (see Theorems 3.25 and 3.26 for details on these ranges). We prove that this bound is tight in general. For the special case of symmetric network congestion games we show that the bound of $(2\rho(1+\sigma)+1)/(\rho+1)$ for the price of anarchy is asymptotically tight, for the range of ρ and σ where the maximum attains this value (see Figure 3.1).
2. We show that the price of stability can be upper bounded by
$$\frac{\sqrt{\sigma(\sigma+2)}+\sigma}{\sqrt{\sigma(\sigma+2)}+\rho-\sigma} \tag{3.2}$$
for a certain range of ρ and σ (Theorem 3.30). We also show that this bound is asymptotically tight in general. The tightness does not hold for symmetric network congestion games, in contrast to the price of anarchy bound. This follows from [80], where (as also mentioned in the introduction) it is shown that the price of stability for symmetric network congestion games with affine cost functions is $4/3$ (when $\rho = \sigma = 1$). This is strictly smaller than the bound of $1 + 1/\sqrt{3}$ for general games with affine cost functions [21].

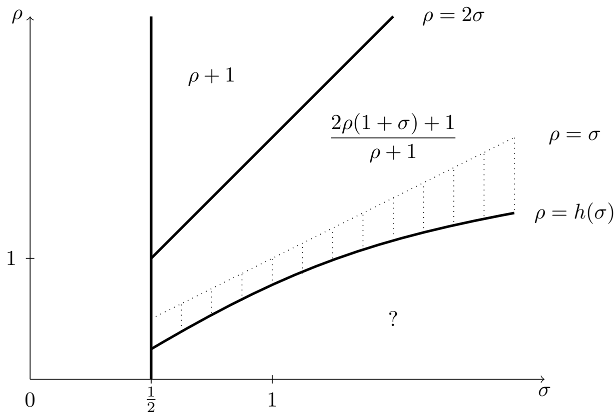

Figure 3.1: The bound $\rho+1$ holds for $\rho \geq 2\sigma \geq 1$. The bound $(2\rho(1+\sigma)+1)/(1+\rho)$ holds for $\sigma \leq \rho \leq 2\sigma$. Basically, this bound also holds for $h(\sigma) \leq \rho \leq \sigma$, but our proof of Theorem 3.26 only works for a discretized range of σ (hence the vertical dotted lines in this area). The function h is given in Theorem 3.26.

Remark 3.1. The inefficiency bounds that we present here for our so-called *perception parameterized congestion games* can be cast in the form of a smoothness argument (see Section 1.2.3). We do not do this explicitly, as this is closely related to, e.g., the work in [28] where this is done for congestion games with *altruistic players*.

3.1.2 Related work

We provide some additional related work that is not given in the introduction.

The inefficiency of pure Nash equilibria in congestion games is fairly well understood, in particular for polynomial cost functions. Awerbuch, Azar and Epstein [8], as well as Christodoulou and Koutsoupias [34], show that the price of anarchy of congestion games with affine cost functions is upper bounded by 5/2, and that this bound is tight in general. Correa et al. [44] show that tightness remains true for the special class of symmetric network congestion games. Aland et al. [4] provide a tight bound on the price of anarchy of congestion games where the cost functions are polynomials of degree at most d. Their bound roughly grows as $d^{\Theta(d)}$.[3] For the price of stability in congestion games with affine cost functions, Caragiannis et al. [21] give an upper bound of $1 + 1/\sqrt{3}$, which is tight by a lower bound construction of Christodoulou and Koutsoupias [33]. Christodoulou and Gairing [32] provide tight bounds for the price of stability for congestion games where the cost functions are polynomials of degree at most d. Their bound grows roughly as $d + 1$, which is much better than the bound

[3] For real-valued functions $f : \mathbb{R} \to \mathbb{R}$, we write $f = \Theta(g)$ if there exist constants c_1 and c_2 such that $c_1 g(x) \leq f(x) \leq c_2 g(x)$ for all $x \in \mathbb{R}_{\geq 0}$.

for the price of anarchy in games with polynomial cost functions as mentioned earlier.

Our inefficiency results for perception-parameterized congestion games with affine cost functions unify many price of anarchy results in the literature concerning extensions and variations of Rosenthal's model. An overview (also including our novel price of stability results) is given in Table 3.1. We informally discuss

Model	**Parameters**	**PoA**	**Ref.**	**PoS**	**Ref.**
Classical	$\rho = \sigma = 1$	$\frac{5}{2}$	[34]	1.577	[22]
Altruism (1)	$\sigma = 1,$ $1 \leq \rho \leq 2$	$\frac{4\rho+1}{1+\rho}$	[23, 28]	$\frac{\sqrt{3}+1}{\sqrt{3}+\rho-1}$	[*]
Altruism (2)	$\sigma = 1,$ $2 \leq \rho \leq \infty$	$\rho + 1$	[28]	–	–
Risk neutral	$\sigma = \rho = \frac{1}{2}$	$\frac{5}{3}$	[146]	1.447	[*]
Wald's minimax	$\sigma = \frac{1}{2}$, $\rho = 1$	2	[20, 146]	1	[*]
Universal taxes	$\sigma = 1,$ $\rho = h(1)$	2.155	[22]	2.013	[*]
Uniform affine CG	–	∞	[*]	2	[*]

Table 3.1: An overview of our (tight) price of anarchy and price of stability results for certain values of ρ and σ. Here $h(1) \approx 0.625$ (see Theorem 3.26 for a formal definition). The respective references where these bounds were established first are given in the column "Ref."; an asterisk indicates that this result is new.

two models that are mentioned in Table 3.1 in order to provide some background. Formal descriptions of all models, and an explanation of why they correspond to the respective values of ρ and σ as given in Table 3.1, are given in Section 3.3.3.

We first discuss the incorporation of *altruistic behavior* [23, 28]. Instead of players being completely selfish, it is assumed they also care about the overall player population to some extent. This can be modeled by defining the *altruistic player cost* to be a convex combination of the player cost, in the classical selfish setting, and the social cost. That is, for $\alpha \in [0, 1]$, an *α-altruistic player* wants to minimize the cost function

$$C_i^\alpha(s) = (1 - \alpha)C_i(s) + \alpha C(s),$$

where C_i is the original cost of player i, and $C(s)$ the social cost. Roughly speaking, the parameter α models to what extent players take into account the social cost in their altruistic player cost.

The second model we consider is that in [146], where congestion games with a (randomized) scheduling policy are studied. After every player has chosen a subset of resources to which they send a (unit weight) job each, a scheduler determines for every resource, using a randomized scheduling policy, in which order the players' jobs are processed. The randomization in the scheduling policy gives rise to uncertainty in the player costs. How do players deal with this uncertainty? Various *player attitudes* towards this uncertainty, and their effect on the inefficiency of pure Nash equilibria, are studied. Roughly speaking, interpreted in our setting, the parameter ρ models how *risk-taking* or *risk-averse* players are. That is, are they feeling lucky and expect to be scheduled relatively early on all the resources to which they have sent jobs (corresponding to a low value of ρ); or are they pessimistic and expect to be scheduled relatively late (corresponding to a high value of ρ)? The authors of [146] essentially provide price of anarchy bounds for two specific values of ρ (with $\sigma = 1/2$).[4]

3.1.3 Outline

In Section 3.2 we first give all the necessary preliminaries regarding Rosenthal's congestion game model, as well as some polytopal preliminaries. In Section 3.3 we then present our results for polytopal congestion games. In particular, in Section 3.3.4 we discuss the implications for bottleneck congestion games. In Section 3.4 we provide a description of our perception-parameterized congestion games, and our unifying inefficiency bounds.

3.2 Preliminaries

A congestion game Γ is given by a tuple $(N, E, (\mathcal{S}_i)_{i\in N}, (c_e)_{e\in E})$, where $N = [n]$ is a finite set of players, $E = [m]$ is a finite set of resources (or facilities), $\mathcal{S}_i \subseteq 2^E$ is a set of strategies of player $i \in N$, and $c_e : \mathbb{R}_{\geq 0} \to \mathbb{R}$ is a cost function of resource $e \in E$. Unless stated otherwise, the cost functions are assumed to be non-negative and non-decreasing.

For a strategy profile $s = (s_1, \dots, s_n) \in \times_i \mathcal{S}_i$, we define $x_e(s)$ as the number of players using resource e, i.e., $x_e(s) = |\{i \in N : e \in s_i\}|$. We call $x(s) = (x_e(s))_{e\in E}$ the *load profile corresponding to strategy profile* s. More generally, we say that $y \in \mathbb{N}^m$ is a *fesible load profile* for the tuple $(N, E, (\mathcal{S}_i))$ if there is some strategy profile s such that $y = x(s)$.

The cost of player $i \in N$ under a strategy profile $s = (s_1, \dots, s_n) \in \times_i \mathcal{S}_i$ is given by $C_i(s) = \sum_{e\in s_i} c_e(x_e(s))$. If $\mathcal{S}_i = \mathcal{S}_j$ for all $i, j \in N$, the game is called *symmetric*. The *social cost* $C(s)$ of a strategy profile refers to the sum of the players' individual costs, i.e.,

$$C(s) = \sum_{i\in N} C_i(s) = \sum_{e\in E} x_e(s) c_e(x_e(s)).$$

We say that $\Phi : \times_i \mathcal{S}_i \to \mathbb{R}$ is an *exact potential function* for a congestion game Γ if for every strategy profile $s \in \times_i \mathcal{S}_i$, for every player $i \in N$ and every

[4]The parameter ρ (or an analogue for that matter) does not appear in [146], but is used here to illustrate how the results in [146] can be interpreted, more general, in our model.

unilateral deviation $s'_i \in \mathcal{S}_i$ of i it holds:

$$\Phi(s) - \Phi(s_{-i}, s'_i) = C_i(s) - C_i(s_{-i}, s'_i).$$

Rosenthal [150] shows that

$$\Phi(s) = \sum_{e \in E} \sum_{k=1}^{x_e(s)} c_e(k) \tag{3.3}$$

is an exact potential function. Subsequently, we refer to this potential function simply as *Rosenthal's potential*. Further, a strategy profile minimizing Rosenthal's potential is said to be a *Rosenthal minimizer*.

3.2.1 Inefficiency measures and smoothness parameter

A strategy profile s is a *pure Nash equilibrium* if for every player $i \in N$ it holds that $C_i(s) \leq C_i(s'_i, s_{-i})$ for all $s'_i \in \mathcal{S}_i$. Further, a strategy profile s is a *strong equilibrium* if for every group of players $I \subseteq N$ and every deviation $s'_I \in \times_{i \in I} \mathcal{S}_i$ of the players in I, it holds that $C_i(s) \leq C_i(s'_I, s_{-I})$ for some $i \in I$.

The price of anarchy (PoA) and the price of stability (PoS) of a game Γ are defined as

$$\mathrm{PoA}(\Gamma) = \frac{\max_{s \in \mathrm{NE}(\Gamma)} C(s)}{\min_{s^* \in \times_i \mathcal{S}_i} C(s^*)} \quad \text{and} \quad \mathrm{PoS}(\Gamma) = \frac{\min_{s \in \mathrm{NE}(\Gamma)} C(s)}{\min_{s^* \in \times_i \mathcal{S}_i} C(s^*)},$$

where $\mathrm{NE}(\Gamma)$ denotes the set of all pure Nash equilibria of Γ. For a collection of games $\mathcal{H}$ we define

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

These notions naturally generalize to the solution concept of strong equilibria.

Correa et al. [45] show that for *non-atomic* network congestion games (see also Section 2.2) with cost functions in class $\mathcal{D}$ the price of anarchy of an instance is at most

$$\rho(\mathcal{D}) := (1 - \beta(\mathcal{D}))^{-1}, \quad \text{where} \quad \beta(\mathcal{D}) = \sup_{d \in \mathcal{D}} \sup_{x \geq y > 0} \frac{y(d(x) - d(y))}{x d(x)}. \tag{3.4}$$

The value of $\rho(\mathcal{D})$ is well-understood for many important classes of cost functions. For example, let

$$\mathcal{D}_d = \{g : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} : g(\mu x) \geq \mu^d g(x) \ \forall \mu \in [0, 1]\}.$$

In particular, $\mathcal{D}_d$ contains all polynomial cost functions with non-negative coefficients and maximum degree d. We have

$$\rho(\mathcal{D}_d) = \left(1 - \frac{d}{(d+1)^{(d+1)/d}}\right)^{-1}.$$

The parameter $\rho(\mathcal{D})$ plays a crucial role in bounding the price of stability of our congestion games.

We say that a class of cost functions $\mathcal{D}$ is *closed under dilations* if for every $d \in \mathcal{D}$ and for every $\gamma \in \mathbb{R}_{\geq 0}$, $d_\gamma \in \mathcal{D}$, where $d_\gamma(x) = d(\gamma x)$ for all $x \geq 0$.

3.2.2 Polytopes

We review some basic definitions and results from polyhedral combinatorics which are used in this chapter (see, e.g., [160] for a more detailed exposition).

A *polytope* $P \subset \mathbb{R}^m$ is the convex hull of a finite set $\{q_1, \dots, q_s\} \subset \mathbb{Q}^m$, or, alternatively, $P = \{x : Ax \leq b\}$ is a bounded set described by a system of rational inequalities.[5] For a non-zero vector c with $\delta = \max\{c^\mathsf{T} x : Ax \leq b\}$, the affine hyperplane $\{x : c^\mathsf{T} x = \delta\}$ is called a *supporting hyperplane* of P. A subset F of P is called a *face* if F is the intersection of P with some supporting hyperplane of P, or $F = P$. The *minimal faces* of P, i.e., faces not contained in another face, are the *vertices (or extreme points)* of P. Moreover, an *edge* of P is a one-dimensional face of P (which is the line-segment between two vertices). We say that P is *integral* if all its extreme points are integral vectors. P is said to be *box-integral* if the intersection of P with any integral box, i.e., $P \cap \{x : c \leq x \leq d\}$ for arbitrary integral c and d, yields an integral polytope.

A matrix $A \in \{0, \pm 1\}^{r \times m}$ is *totally unimodular (TUM)* if the determinant of each square submatrix of A is in $\{0, \pm 1\}$. If A is totally unimodular and $b \in \mathbb{Z}^m$ is an integer vector, then the polyhedron $P = \{x : Ax \leq b\}$ is integral [160, Theorem 19.1].

The work [65] introduced the powerful notion of total dual integrality. A rational system $Ax \leq b$ with $A \in \mathbb{Q}^{r \times m}$ and $b \in \mathbb{Q}^r$ is *totally dual integral (TDI)* if for every integral $c \in \mathbb{Z}^m$, the dual of minimizing $c^\mathsf{T} x$ over $Ax \leq b$, i.e.,

$$\max\{y^\mathsf{T} b : y \geq 0,\ y^\mathsf{T} A = c^\mathsf{T}\}, \tag{3.5}$$

has an integer optimum solution y, if it is finite. If $Ax \leq b$ is a TDI-system and b is integral, then the polyhedron $P = \{x : Ax \leq b\}$ is integral [160, Corollary 22.1c]. Note that TDI is a weaker sufficient condition for the integrality of P than TUM.

The system $Ax \leq b$ is *box-totally dual integral (box-TDI)* if the system $Ax \leq b$, $l \leq x \leq u$ is TDI for all rational vectors l and u. We say that a polytope P is *box-TDI*, if it can be described by a box-TDI system. If P has some box-TDI describing system, then every TDI-system describing P is also box-TDI [160, Theorem 22.8]. We will use the following properties of box-TDI descriptions:

Proposition 3.2. *[160, Section 22.5] The following statements are equivalent:*
(i) The system $Ax \leq b$, $x \geq 0$ is box-TDI.
(ii) The system $Ax + \mu = b$, $\mu \geq 0$, $x \geq 0$ is box-TDI.
(iii) The system $Ax \leq \alpha b$, $x \geq 0$ is box-TDI for all $\alpha \geq 0$.

[5] An inequality $a^\mathsf{T} x \leq b$ is rational if $a \in \mathbb{Q}^m$ and $b \in \mathbb{Q}$.

(iv) *The system* $a\zeta_0 + Ax \leq b$ *is box-TDI, where* a *is a column of* A *and* ζ_0 *is a new variable.*

Moreover, if a polytope P *is box-integral, then every edge of* P *is in the direction of a* $\{0, \pm 1\}$*-vector.*[6]

Finally, consider a finite collection of integral polytopes $(P_i)_{i \in N}$, where $P_i \subseteq \mathbb{R}^m$ for every $i \in N = [n]$, with a common constraint matrix but possibly different right-hand side vectors, i.e., there exists a matrix A such that

$$P_i = \{x : Ax \leq b_i\} \subseteq \mathbb{R}^m.$$

We define the *aggregation polytope* P_N induced by $(P_i)_{i \in N}$ as

$$P_N = \left\{ y : Ay \leq \sum_{i \in N} b_i \right\} \subseteq \mathbb{R}^m.$$

The aggregation polytope is said to have the *integer decomposition property (IDP)* if every integral $z \in P_N$ can be written as

$$z = \sum_{i=1}^{n} z^i, \quad \text{where} \quad z^i \in P_i \cap \mathbb{Z}^m \text{ for all } i = 1, \ldots, n.$$

Note that in the symmetric case $b_i = b_j$ for all $i, j \in N$ this definition reduces to the integer decomposition property for a polytope $P_N = nP$ as introduced in [10]. Moreover, if P_N has the IDP, then indeed $P_N = \sum_i P_i$, where the latter summation is the Minkowski sum of polytopes.

Optimization over polytopes

We discuss some classical results regarding the problem of optimizing a linear function over a polytope, i.e., we consider the problem

$$\min \ c^\mathsf{T} x \quad \text{s.t.} \quad x \in P \tag{3.6}$$

for some $c \in \mathbb{Q}^m$ and polytope $P \subset \mathbb{R}^m$. We first introduce some additional (computational) notions [97].

The *encoding length of an integer* $z \in \mathbb{Z}$, i.e., the space needed to represent z in binary representation, is

$$\langle z \rangle = 1 + \lceil \log_2(z + 1) \rceil.$$

The *encoding length of a rational number* $p/q \in \mathbb{Q}$ is $1 + \lceil \log_2(p+1) \rceil + \lceil \log_2(q+1) \rceil$. The *encoding length of a vector* $a \in \mathbb{Q}^m$ is $\langle a \rangle = \sum_{i=1}^m \langle a_i \rangle$, and the *encoding length of an inequality* $a^\mathsf{T} x \leq b$ is $\langle a \rangle + \langle b \rangle$, for $a \in \mathbb{Q}^m$ and $b \in \mathbb{Q}$.

[6]That is, the edge is of the form $\{a + \lambda b : 0 \leq \lambda \leq 1\}$ for vertices $a, a + b$ of P where b is a $\{0, \pm 1\}$-vector.

For a positive integer ϕ, we say that polytope P has *facet-complexity at most* ϕ if there exists a system of inequalities with rational coefficients describing P such that every inequality has encoding length at most ϕ. A triple $(P; m, \phi)$ is called a *well-described polytope* if the polytope $P \subset \mathbb{R}^m$ has facet-complexity at most ϕ.

Finally, a *(strong) separation oracle for* P is an algorithm that, given a vector $y \in \mathbb{Q}^m$, decides whether $y \in P$ or not, and in the latter case returns a vector $a \in \mathbb{Q}^m$ such that $a^\mathsf{T} x < a^\mathsf{T} y$ for all $x \in P$. If a separation oracle is used as a subroutine in an algorithm, this is referred to as a *call* to the oracle.

The following theorem summarizes a fundamental result in [97, 82]. We give a formulation in terms of polytopes based on Theorem 6.6.5 in [97].

Theorem 3.3. *There exists an algorithm that, for any well-described polytope* $(P; m, \phi)$ *specified by a strong separation oracle, and for given* $c \in \mathbb{Q}^m$*,*
(i) solves (3.6)*, and*
(ii) finds an optimum vertex solution of (3.6) *if one exists.*
The number of elementary arithmetic operations[7] *and calls of the separation oracle executed by the algorithm is bounded by a polynomial in* ϕ*. All arithmetic operations are performed on numbers whose encoding length is bounded by a polynomial in* $\phi + \langle c \rangle$*.*

We give two remarks related to Theorem 3.3.

Remark 3.4. For notational convenience, we use $\mathsf{poly}(\cdot)$ to denote a function that is polynomial in all its arguments. The algorithm of Theorem 3.3 is said to run in *strongly polynomial time* (for a class of problems) if the facet-complexity ϕ can be upper bounded by a polynomial in m, i.e., $\phi = \mathsf{poly}(m)$.

Remark 3.5. We do not always explicitly mention that all arithmetic operations are performed on numbers whose encoding length is bounded by a polynomial in $\phi + \langle c \rangle$. In all subsequent computational statements relying on Theorem 3.3, we implicitly assume that this property holds.

3.2.3 Matroids

We introduce some general terminology and facts for matroids; a more extensive treaty of matroids can be found, e.g., in [161]. Let $E = [m]$ be a finite set of elements and $\mathcal{I} \subseteq 2^E$ be a collection of subsets of E (called *independent sets*). The pair $\mathcal{M} = (E, \mathcal{I})$ is a *matroid* if the following three properties hold:
(i) $\emptyset \in \mathcal{I}$,
(ii) if $A \in \mathcal{I}$ and $B \subseteq A$, then $B \in \mathcal{I}$,
(iii) if $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists an $a \in A \backslash B$ such that $B + a \in \mathcal{I}$.
An independent set $B \in \mathcal{I}$ of maximum size is called a *basis*. We use $\mathcal{B}$ to denote the set of all bases of $\mathcal{M}$. The matroid $\mathcal{M}$ also has a rank function $r : 2^E \to [m]$

[7] Here by *elementary* arithmetic operations we mean addition, substraction, multiplication, division and comparison.

which maps every subset $A \subseteq E$ to the cardinality of the largest independent set contained in A.

The *base matroid polytope* is given by

$$P_{\mathcal{M}} = \{x : x(A) \leq r(A)\ \forall A \subset E,\ x(E) = r(E),\ x \geq 0\},$$

where $x(A) = \sum_{a \in A} x_a$ for all $A \subseteq E$. It is the convex hull of the incidence vectors of the bases in $\mathcal{B}$ [161]. If in the description above the equality $x(E) = r(E)$ is replaced by $x(E) \leq r(E)$, we obtain the *independent set polytope* which is the convex hull of the incidence vectors of the independent sets.

We assume that the matroid is given by an *independence oracle* that takes as input a subset $A \subseteq 2^E$ and returns whether or not $A \in \mathcal{I}$. Given an independence oracle, we can determine in time polynomial in $|E|$ and the complexity of the oracle, whether a set is a basis and what the rank of a set is. Further, there exists a separation oracle for $P_{\mathcal{M}}$ that runs in time polynomial in $|E|$ and the complexity of an independence oracle. This follows from the fact that the *most violated inequality problem* can be solved in time polynomial in $|E|$ and the complexity of an independence oracle. The most violated inequality problem takes as input a vector $x \in \mathbb{Q}^m$ and returns whether or not $x \in P$, and if not, it returns a subset A for which $r(A) - x(A)$ is minimized, see, e.g., [161, Section 40.3].

Given two matroids $\mathcal{M}_1$ and $\mathcal{M}_2$ on a common ground set E, the polytope

$$P_{\mathcal{M}_1,\mathcal{M}_2} = \{x : x(A) \leq r_i(A)\,\forall A \subset E,\, x(E) = r_i(E) \text{ for } i = 1,2,\ x \geq 0\} \quad (3.7)$$

is the convex hull of the common bases of matroids $\mathcal{M}_1$ and $\mathcal{M}_2$, see, e.g., [161, Corollary 41.12d]. It follows directly that $P_{\mathcal{M}_1,\mathcal{M}_2}$ also has a separation oracle which runs in time polynomial in $|E|$ and the complexity of the independence oracles for $\mathcal{M}_1$ and $\mathcal{M}_2$.

3.3 Polytopal congestion games

We consider *polytopal congestion games* $\Gamma = (N, E, (\mathcal{S}_i)_{i \in N}, (c_e)_{e \in E})$ with $N = [n]$ and $E = [m]$, where the set of strategies $\mathcal{S}_i$ of each player $i \in N$ is given implicitly by a polytopal representation. More precisely, let $\mathcal{X}_i$ be the finite set of all incidence vectors of the strategies of player i, i.e., for every $i \in N$,

$$\mathcal{X}_i = \{\chi_i \in \{0,1\}^m : \chi_{ie} = 1 \text{ iff } e \in s_i \text{ for } s_i \in \mathcal{S}_i\}.$$

The polytope P_i representing the strategies of player i is defined as the convex hull of $\mathcal{X}_i$, i.e., $P_i = \text{conv}(\mathcal{X}_i) \subseteq [0,1]^m$. We assume that P_i is given by

$$P_i = \{x : Ax \leq b_i\} \subseteq [0,1]^m,$$

where $A \in \mathbb{Z}^{r \times m}$ is an integral $r \times m$-matrix and $b_i \in \mathbb{Z}^r$ is an integral vector. Note that $\mathcal{X}_i = P_i \cap \{0,1\}^m$. For notational convenience, we subsequently use $\mathcal{S}_i$ also to refer to the set of incidence vectors $\mathcal{X}_i$; no confusion shall arise.

As defined above, the aggregation polytope induced by $(P_i)_{i\in N}$ is

$$P_N = \left\{ y : Ay \leq \sum_{i\in N} b_i \right\} \subseteq [0,n]^m.$$

We say that $(N, E, (\mathcal{S}_i)_{i\in N})$ is the *polytopal tuple* given by $(P_i)_{i\in N}$, where $\mathcal{S}_i = P_i \cap \{0,1\}^m$. If $b_i = b_j = b$ for all $i, j \in N$, the tuple is called *symmetric* and denoted by $(N, E, \mathcal{S})$ where $\mathcal{S} = P \cap \{0,1\}^m$, with $P = \{x : Ax \leq b\}$. If additionally we equip the tuple with cost functions $(c_e)_{e\in E}$, we call $\Gamma = (N, E, (\mathcal{S}_i)_{i\in N}, (c_e)_{e\in E})$ the *polytopal congestion game* given by $(P_i)_{i\in N}$. For notational convenience, we often omit the explicit reference of the domain of the indices.

Two crucial properties: IDP and box-TDI

Let $\Gamma = (N, E, (\mathcal{S}_i), (c_e))$ be a polytopal congestion game with aggregation polytope P_N. We identify two crucial properties that the aggregation polytope P_N has to satisfy for our results to go through:

(i) P_N has the integer decomposition property (IDP).
(ii) P_N is box-totally dual integral (box-TDI).

If both properties are satisfied, we also say that Γ is a *polytopal congestion game satisfying IDP and box-TDI.*

Remark 3.6. Note that for a *symmetric* polytopal congestion games $\Gamma = (N, E, \mathcal{S}, (c_e))$ given by a common polytope P, we have $P_N = nP = \{y : y/n \in P\}$. From Proposition 3.2(iii), it follows that the aggregation polytope P_N has a box-TDI description if and only if P has a box-TDI description. In particular, whenever we require below that a symmetric polytopal congestion game is box-TDI, then all we need is it that the common polytope P is box-TDI.

The IDP is crucial to establish a correspondence between feasible load profiles for $(N, E, (\mathcal{S}_i))$ and the integral vectors in P_N.

Proposition 3.7. *If the aggregation polytope P_N of a polytopal tuple $(N, E, (\mathcal{S}_i))$ has the IDP, then the feasible load profiles of the tuple correspond precisely to the integral vectors in P_N.*

Proof. Let $s = (s_1, \ldots, s_n) \in \times_i \mathcal{S}_i$ be a strategy profile and let x be the load profile corresponding to s. It follows directly that $x \in P_N$ by definition of P_N. Moreover, because of the IDP any integral vector z in P_N can be decomposed as $z = \sum_{i=1}^n z^i$ where $z^i \in P_i \cap \mathbb{Z}^m$ for all $i = 1, \ldots, n$. This implies that for every i the vector z^i is the incidence vector of some strategy of player i and thus z is a feasible load profile. $\square$

The main reason as to why box-TDI is useful, is that it serves as a sufficient condition to show that the polytope it describes is box-integral.

Proposition 3.8. *If the system $Ax \leq b$ describing a polytope P is box-TDI and b is integral, then P is box-integral.*

Proof. By assumption, the describing system $Ax \leq b$ of P is box-TDI. Thus the system $Ax \leq b,\ l \leq y \leq u$ is TDI for all rational vectors l and u. In particular, $Ax \leq b,\ c \leq y \leq d$ is TDI for arbitrary integral vectors c and d. Because b, c and d are integral, we can conclude that the polytope $P \cap \{y : c \leq y \leq d\}$ is integral (see, e.g., [160, Corollary 22.1c]). □

It seems that most 0/1-polytopes for which the integer decomposition property is known in the literature, also have a box-TDI describing system. We are not aware of any result showing that this is true in general, but it would imply that box-TDI, as an assumption, is redundant in all our statements below.

3.3.1 Price of stability

The following is the main result of this section. Recall that $\rho(\mathcal{D})$ is defined as in (3.4) and refers to the price of anarchy of *non-atomic* network congestion games with cost functions in class $\mathcal{D}$.

Theorem 3.9. *Let $\Gamma = (N, E, (\mathcal{S}_i), (c_e))$ be a polytopal congestion game satisfying IDP and box-TDI with cost functions in class $\mathcal{D}$. Then $PoS(\Gamma) \leq \rho(\mathcal{D})$. Further, this bound is (asymptotically) tight even for symmetric singleton congestion games, if $\mathcal{D}$ contains all non-negative constant functions and is closed under dilations.*

Proof overview. The remainder of this section is devoted to the proof of Theorem 3.9. We first introduce a novel structural property, which we term the *symmetric difference decomposition property.* We then show that the IDP and box-TDI properties of the aggregation polytope are sufficient to establish that the polytopal congestion game satisfies the symmetric difference decomposition property. This in turn allows us to adapt the proof of Fotakis [80] to bound the price of stability of these games.

3.3.1.1 Symmetric difference decomposition property

Our novel property is defined as follows:

Definition 3.10. A tuple $(N, E, (\mathcal{S}_i))$ satisfies the *symmetric difference decomposition property (SDD)* if for all feasible load profiles f and g, there exist vectors $a^1, \ldots, a^q \in \{0, \pm 1\}^m$ such that $g - f = \sum_{k=1}^{q} a^k$, and for all $k = 1, \ldots, q$,
(i) the load profile $f + a^k$ is feasible, and
(ii) a^k satisfies

$$a_e^k = -1 \;\Rightarrow\; f_e - g_e > 0 \qquad \text{and} \qquad a_e^k = 1 \;\Rightarrow\; f_e - g_e < 0. \tag{3.8}$$

As an example, let us consider symmetric network congestion games, where the common strategy set of all players is the set of all directed simple s,t-paths in some directed graph $G = (V, A)$ with $s, t \in V$. Here each feasible load profile corresponds to an integral feasible s,t-flow of value $n = |N|$. The symmetric difference of two flows f and g can be written as the sum of unit circuit flows on cycles.[8] The incidence vectors of these unit circuit flows correspond to the vectors a^k in Definition 3.10.

The following theorem establishes the symmetric difference decomposition property for polytopal congestion games satisfying IDP and box-TDI.

Theorem 3.11. *If the aggregation polytope P_N of a polytopal tuple $(N, E, (\mathcal{S}_i))$ satisfies IDP and box-TDI, then the tuple has the symmetric difference decomposition property.*

Proof. We start by adding slack-variables to the system $Ay \leq \sum_{i\in N} b_i$ describing P_N. Note that by Proposition 3.2(ii) box-TDI is preserved under adding slack variables. As a result, we obtain the polytope

$$Q_N = \left\{(y, \mu) : Ay + \mu = \sum_{i=1}^{n} b_i,\ \mu \geq 0,\ y \geq 0\right\}$$

for which its describing system is box-TDI. Also, Q_N is integral.

Let f and g be two feasible load profiles with $f \neq g$. By Proposition 3.7, we have $f, g \in P_N$. Therefore, there are non-negative integral slack vectors τ, σ such that $f' = (f, \tau)$, $g' = (g, \sigma) \in Q_N$. Observe that τ and σ are integral because of the integrality of A, $\sum_i b_i$ and f and g.

Note that the pairs $f' = (f, \tau)$ and $g' = (g, \sigma)$ are vectors in $\mathbb{Z}^{m+r}$ since A is an $r \times m$-matrix. Let $c, d \in \mathbb{Z}^{m+r}$ be vectors defined by $c_j = \min\{f'_j, g'_j\}$ and $d_j = \max\{f'_j, g'_j\}$ for $j = 1, \ldots, r+m$, and let B be the integral box defined by $B = \{z : c \leq z \leq d\} \subseteq \mathbb{R}^{m+r}$. We first prove the following claim.

Claim 3.12. *The polytope $Q_N \cap B$ is integral and every edge of $Q_N \cap B$ is in the direction of a $\{0, \pm 1\}$-vector.*

Proof. The integrality follows from box-TDI of the integral system Q_N. For the second part of the claim, we first show that $Q_N \cap B$ is box-integral. Note that Q_N is box-integral by Proposition 3.8. Let $B' = \{x : \gamma \leq x \leq \delta\} \in \mathbb{Z}^{m+r}$ be an arbitrary integral box. Note that $(Q_N \cap B) \cap B' = Q_N \cap (B \cap B')$ and that $B \cap B'$ is again an integral box, since B is integral as well (because f' and g' are integral). It follows that $Q_N \cap (B \cap B')$ is an integral polytope. Thus, $(Q_N \cap B) \cap B'$ is integral which proves that $Q_N \cap B$ is box-integral. The claim now follows from Proposition 3.2. $\square$

[8] A *unit circuit* flow is a $\{0, \pm 1\}$-flow that satisfies flow-conservation at every node, including s and t.

Note that f', $g' \in Q_N \cap B$. Further, both f' and g' are extreme points of this polytope because they are extreme points of the box B. We now fix some edge of $Q_N \cap B$ containing f'. Such an edge must exist because $Q_N \cap B$ contains at least two elements (since $f' \neq g'$). Let $(a^1)' = (a^1, \mu^1)$ be the non-zero $\{0, \pm 1\}$-vector describing the direction of the edge.[9] Since $Q_N \cap B$ is an integral polytope, we can show that $f' + (a^1)' \in Q_N \cap B$. To see this, let $h(\lambda) = f' + \lambda \cdot (a^1)'$ be a parameterization of the edge for some range $0 \leq \lambda \leq \lambda^*$, where $h' = h(\lambda^*)$ is the other extreme point of the edge $(a^1)'$. Since f' is integral and $(a^1)'$ a $\{0, \pm 1\}$-vector, it must be that λ^* is a strictly positive integer. Thus, $f' + (a^1)' \in Q_N \cap B$, as claimed.

It follows that $A(f + a^1) + (\tau + \mu^1) = \sum_i b_i$. We have $Aa^1 + \mu^1 = 0$ because $Af + \tau = \sum_i b_i$. Moreover, by construction of the box B it follows that for $j = 1, \dots, r + m$,

$$(a^1)'_j = -1 \ \Rightarrow \ f'_j - g'_j > 0 \qquad \text{and} \qquad (a^1)'_j = 1 \ \Rightarrow \ g'_j - f'_j > 0. \tag{3.9}$$

Exploiting that $Aa^1 + \mu^1 = 0$, it now also follows that $g' - (a^1)'_j \in Q_N \cap B$. To see this, note that

$$A(g - a^1) + (\sigma - \mu^1) = Ag + \sigma - (Aa^1 + \mu^1) = \textstyle\sum_i b_i.$$

Moreover, we also have $g' - (a^1)' \geq 0$ by construction, since if $(a^1)'_j = 1$ for some j then $g'_j > f'_j \geq 0$, so in particular $g'_j - 1 \geq 0$ (because of the integrality of g'_j).

We can now apply the same argument to the vectors f' and $g' - (a^1)'$ in order to obtain a vector $(a^2)'$ satisfying (3.9) and for which $f' + (a^2)'$, $g' - (a^1)' - (a^2)' \in Q_N$. Repeating this procedure we find vectors $(a^1)', \dots, (a^q)'$ satisfying (3.9), and such that $g' - f' = \sum_{k=1}^q (a^k)'$ with $f' + (a^k)' \in Q_N$ for $k = 1, \dots, q$.[10]

We argue that this process terminates. For the ℓ-th step of this procedure, we have by construction of the $(a^k)'$

$$T(\ell) = \left|\left| \left(g' - \sum_{k=1}^{\ell} (a^k)' \right) - f' \right|\right|_1 < \left|\left| \left(g' - \sum_{k=1}^{\ell-1} (a^k)' \right) - f' \right|\right|_1 = T(\ell - 1)$$

where $||\cdot||_1$ is the ℓ_1-norm. Since f', g' and the a^k are all integral this guarantees that the expression $T(\ell)$ decreases by at least one in every step.

We conclude the proof by showing that f and g can be decomposed according to Definition 3.10. We have $(a^k)' = (a^k, \mu^k)$ as defined before. It then follows that $a^1, \dots, a^q$ are vectors satisfying (3.8) such that $g - f = \sum_{k=1}^q a^k$ with $f + a^k \in P_N$ for $k = 1, \dots, q$. Note that a^k might be the zero-vector, if $(a^k)'$ only contained non-zero elements in the part of the vector corresponding to slack variables. These a^k can be left out.

It remains to show that $f + a^k$ corresponds to a feasible strategy profile for $k = 1, \dots, q$. This follows directly from the fact that P_N has the IDP. The decomposition yields the strategies of the players. □

[9] Here, μ^1 corresponds to the slack variables, and a^1 to the original variables.

[10] This construction is essentially a *conformal circuit decomposition* (see, e.g., [142]).

3.3.1.2 Upper bound on the price of stability

We prove the upper bound on the price of stability stated in Theorem 3.9. We first prove the following lemma, whose proof relies on the symmetric difference decomposition property.

Lemma 3.13. *Let $(N, E, (\mathcal{S}_i))$ be a polytopal tuple that satisfies the symmetric difference decomposition property and let $(c_e)_{e\in E}$ be arbitrary cost functions. Let f be a feasible load profile that minimizes Rosenthal's potential $\Phi(\cdot)$. Then for every feasible load profile g*

$$\Delta(f,g) := \sum_{e\in E\colon f_e>g_e} (f_e - g_e)c_e(f_e) - \sum_{e\in E\colon f_e<g_e} (g_e - f_e)c_e(f_e+1) \leq 0. \quad (3.10)$$

Proof. Let f be a global minimizer of Rosenthal's potential and let g be an arbitrary feasible load profile. Then by the SDD property, there exist vectors $a^1, \ldots, a^q$ such that $g - f = \sum_{k=1}^{q} a^k$ for some q. Moreover, for all $k = 1, \ldots, q$

$$\Phi(f) - \Phi(f + a^k) = \sum_{e\colon\ a_e^k=-1} c_e(f_e) - \sum_{e\colon\ a_e^k=1} c_e(f_e+1) \leq 0,$$

where the inequality holds because f minimizes Rosenthal's potential Φ. By adding up these inequalities for all $k = 1, \ldots, q$, we obtain that $\Delta(f, g) \leq 0$. To see this, note that if $e \in E$ with $f_e > g_e$ then there are precisely $f_e - g_e$ vectors a^k with $a_e^k = -1$; similarly, if $e \in E$ with $g_e > f_e$ then there are precisely $g_e - f_e$ vectors a^k with $a_e^k = 1$. □

We can now prove the upper bound on the price of stability.

Proof of Theorem 3.9, upper bound. The upper bound proof follows a similar line of arguments as the proof of Lemma 3 in [80]. We repeat the arguments here for the sake of completeness.

Let f be a minimizer of Rosenthal's potential and let g an arbitrary feasible load profile. Note that f is a pure Nash equilibrium.

Consider a resource $e \in E$ with $f_e > g_e$. We have

$$\begin{aligned} f_e c_e(f_e) &= g_e c_e(f_e) + (f_e - g_e)c_e(f_e) \\ &\leq g_e c_e(g_e) + \beta(\mathcal{D}) f_e c_e(f_e) + (f_e - g_e)c_e(f_e), \end{aligned} \quad (3.11)$$

where the inequality follows from the definition of $\beta(\mathcal{D})$ in (3.4), exploiting that $f_e > g_e \geq 0$ and $c_e \in \mathcal{D}$.

Next, consider a resource $e \in E$ with $f_e < g_e$. We have

$$\begin{aligned} f_e c_e(f_e) &= g_e c_e(g_e) - g_e c_e(g_e) + f_e c_e(f_e) \\ &\leq g_e c_e(g_e) - (g_e - f_e)c_e(f_e+1), \end{aligned} \quad (3.12)$$

where the inequality follows because c_e is non-decreasing and $f_e + 1 \leq g_e$ by assumption.

Combining these inequalities, we obtain

$$C(f) \leq C(g) + \sum_{e \in E: f_e > g_e} \beta(\mathcal{D}) f_e c_e(f_e) + \Delta(f,g) \leq C(g) + \beta(\mathcal{D}) C(f)$$

where the first inequality follows from (3.11) and (3.12) and the definition of $\Delta(f,g)$ in (3.10), and the last inequality holds because $\Delta(f,g) \leq 0$ by Lemma 3.13. By rearranging terms, we obtain $C(f)/C(g) \leq (1-\beta(\mathcal{D}))^{-1} = \rho(\mathcal{D})$, which proves the claim. □

3.3.1.3 Lower bound on the price of stability

We complete the proof of Theorem 3.9 by showing that the stated bound is asymptotically tight.

Proof of Theorem 3.9, lower bound. Our lower bound construction is similar to the one used by Correa et al. [46] to show tightness of the price of anarchy bound $\rho(\mathcal{D})$ for non-atomic network congestion games with cost functions in class $\mathcal{D}$. But we need some adjustments to make it work for atomic (unsplittable) congestion games.

Let $d \in \mathcal{D}$ and $a \geq b > 0$ be chosen arbitrarily. We show that there exists an instance whose price of stability is arbitrarily close to

$$\left(1 - \frac{b(d(a) - d(b))}{ad(a)}\right)^{-1}. \tag{3.13}$$

Because of the continuity of d, we can take $a, b \in \mathbb{Q}$ without loss of generality. Let $M \in \mathbb{N}$ such that $Ma, Mb \in \mathbb{N}$. Consider the instance depicted in Figure 3.2 with Ma players. Note that this is a symmetric singleton congestion game instance. Further, note that $c_1 \in \mathcal{D}$ because it is a constant function and $c_2 \in \mathcal{D}$ because $\mathcal{D}$ is closed under dilations. A Nash equilibrium is given by the flow $f = (f_1, f_2) = (0, Ma)$. A feasible (not necessarily socially optimal) flow is given by $g = (g_1, g_2) = (M(a-b), Mb)$. We have

$$\frac{C(f)}{C(g)} = \frac{Ma \cdot d(a)}{M(a-b)d(a) + Mb \cdot d(b)} = \left(1 - \frac{b(d(a) - d(b))}{ad(a)}\right)^{-1}.$$

In order to get a lower bound on the price of stability, we make f the unique Nash flow of this game. This can be done by adding a small enough $\epsilon > 0$ to the cost function of arc a_1, i.e., we take $c_1(x) = d(a) + \epsilon$. Doing the same analysis and sending $\epsilon \to 0$, then shows that we can get arbitrarily close to the expression in (3.13). □

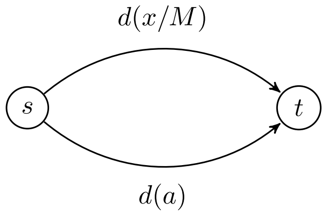

Figure 3.2: The bottom and top arc a_1 and a_2, respectively, have cost functions $c_1(x) = d(a)$ and $c_2(x) = d(x/M)$.

3.3.2 Minimizing Rosenthal's potential

In this section, we consider the problem of computing a minimizer of Rosenthal's potential function for a polytopal congestion game $\Gamma = (N, E, (\mathcal{S}_i), (c_e))$.

Throughout this section, we assume that the aggregation polytope P_N is a well-described polytope $(P_N; m, \phi)$ as described in Section 3.2.2 for which we have a separation oracle. Further, all operations are performed on numbers whose encoding length is bounded by a polynomial in $\phi + \langle c \rangle$ (see Remark 3.5).

The following is the main result of this section.

Theorem 3.14. *Let $\Gamma = (N, E, (\mathcal{S}_i), (c_e))$ be a polytopal congestion game whose aggregation polytope P_N satisfies IDP and box-TDI. Then a feasible load profile minimizing Rosenthal's potential can be computed using at most $\mathsf{poly}(n, m, \phi)$ arithmetic operations and separation oracle calls.*

Remark 3.15. If the facet-complexity ϕ can be upper bounded by a polynomial in n and m, and if P_N has a separation oracle running in strongly polynomial time, then a feasible load profile minimizing Rosenthal's potential can be computed in strongly polynomial time (see also Remark 3.4).

Note that by applying Theorem 3.14 we obtain a feasible *load profile*. We can turn such a load profile into a feasible *strategy profile* (corresponding to a pure Nash equilibrium) if the integer decomposition can be done in (strongly) polynomial time. To the best of our knowledge, there is no universal algorithm that can perform integer decomposition of an arbitrary polytope satisfying the IDP using at most $\mathsf{poly}(n, m, \phi)$ arithmetic operations and separation oracle calls (even in the symmetric case when all right-hand side vectors b_i are the same). So whether this decomposition can be done efficiently has to be investigated on a case-by-case basis.

However, under a slightly stronger integer decomposition property such a decomposition can always be done in polynomial time. Here we focus on symmetric congestion games for clarity; but these arguments can be extended to the non-symmetric case as well.

We say that a polytope P satisfies the *middle integral decomposition property (MIDP)* [125] if for $n \in \mathbb{N}$ and $w \in \mathbb{Z}^m$, the polytope $P \cap (w - (n-1)P)$ is integral. If this property is satisfied, the decomposition algorithm of Baum and Trotter [10] can be used to perform the integer decomposition (details are given in the proof of Theorem 3.16 below).

Theorem 3.16. *Let $\Gamma = (N, E, \mathcal{S}, (c_e))$ be a symmetric polytopal congestion game for which the common polytope P satisfies MIDP and box-TDI. Then a feasible strategy profile minimizing Rosenthal's potential can be computed using at most $\mathsf{poly}(n, m, \phi)$ arithmetic operations and separation oracle calls.*

We remark that all results in this section also hold for computing a social optimum of congestion games with *weakly convex* cost functions, since this problem can be reduced to computing a global optimum of Rosenthal's potential (we refer to [54] for more details).

Aggregation/decomposition framework

Del Pia et al. [54] introduce an *aggregation/decomposition framework* for computing a global minimum of Rosenthal's potential, which also constitutes the basis of our approach. It consists of two phases: In the *aggregation phase*, we find a feasible load profile f^* minimizing Rosenthal's potential. In the *decomposition phase*, f^* is then decomposed into a feasible strategy profile. The authors provide an aggregation approach (detailed below) that works for totally unimodular matrices A and one common vector $b = b_i$ for all $i \in N$. Here we extend this result to aggregation polytopes P_N that have a box-TDI description.

Recall from Proposition 3.7 that if the aggregation polytope P_N of a polytopal congestion game has the IDP, then the feasible load profiles correspond to the integral vectors of $P_N = \{y : Ay \leq \sum_{i \in N} b_i\}$. As a result, the problem we need to solve in the aggregation phase is equivalent to

$$(Z) \qquad \min \sum_{e=1}^{m} \sum_{k=1}^{y_e} c_e(k) \quad \text{s.t.} \quad Ay \leq \sum_{i \in N} b_i,\ y \in \mathbb{N}^m.$$

Note that this is an integer program with a non-linear objective. However, it is not hard to see that the objective function is convex separable.[11] It is known that under certain circumstances such integer programs can be transformed into equivalent linear programs. In particular, if the underlying system describes a (single-commodity) network flow problem, then this problem can be reduced to a minimum cost network flow problem (in fact, this trick dates back to Dantzig [51] and Ford and Fulkerson [79]). Similarly, if the constraint matrix A is TUM and $b = \sum_i b_i$ is integral, there exists an equivalent linear programming formulation that is integral (see Meyer [130]). This idea is (implicitly) also exploited in the works by Fabrikant et al. [75] and Del Pia et al. [54] to show that a Rosenthal minimizer can be computed for symmetric network congestion games and for

[11] Observe that the function $h_e(x) := \sum_{k=1}^{x} c_e(k)$ is convex because the cost functions are non-negative and non-decreasing.

TUM congestion games, respectively. Here we use the same idea for our (more general) setting.

We introduce binary variables $h_e^k \in \{0,1\}$ for $k = 1, \ldots, n$ and $e = 1, \ldots, m$. The interpretation is that $h_e^k = 1$ if at least k players are using resource $e \in E$, and $h_e^k = 0$ otherwise. Exploiting that the cost functions are non-negative and non-decreasing, the non-linear aggregation problem (Z) is equivalent to the problem (R) stated below:

$$
\begin{aligned}
(R) \qquad \min \quad & \sum_{e=1}^{m} \sum_{k=1}^{n} c_e(k) h_e^k \\
\text{s.t.} \quad & [A, A, \ldots, A](h_1^1, \ldots, h_m^1, \ldots, h_1^n, \ldots, h_m^n)^\intercal \leq \textstyle\sum_{i \in N} b_i && (3.14) \\
& h_e^k \in \{0,1\} \quad \forall k = 1, \ldots, n, \; e = 1, \ldots, m && (3.15)
\end{aligned}
$$

The equivalence of (Z) and (R) follows from the following observations: If $f = (f_e) \in P_N \cap Z^m$ is optimal for (Z), we define for every $e \in E$, $h_e^k = 1$ for $k = 1, \ldots, f_e$ and $h_e^k = 0$ for $k = f_e + 1, \ldots, n$. The resulting solution $h = (h_e^k)$ is feasible for (R). Similarly, if $h = (h_e^k)$ is an optimal solution for (R), then the vector f defined by $f_e = \sum_{k=1}^{n} h_e^k$ is feasible for (Z). Note that here we implicitly exploit that the cost functions are non-decreasing.

We show that the integer program (R) can be solved efficiently. In particular, this proves Theorem 3.14.

Lemma 3.17. *If P_N has a box-TDI description, then (R) can be solved using at most $\mathsf{poly}(n, m, \phi)$ arithmetic operations and separation oracle calls.*

Proof. Define $A' = [A, A, \ldots, A] \in \mathbb{Z}^{r \times mn}$ and $h = (h_e^k) \in \mathbb{Q}^{mn}$. The relaxation of the integral system (3.14) and (3.15) can then be written as the system $A'h \leq \sum_i b_i$, $0 \leq h \leq 1$. Let $Q_N = \{h : A'h \leq \sum_i b_i, \; 0 \leq h \leq 1\}$ be the polytope described by this system.

We first show that Q_N is integral. By assumption the description of $P_N = \{f : Af \leq \sum_i b_i\}$ is box-TDI. In particular, by applying Proposition 3.2(iv) repeatedly, we obtain that the system

$$[A, A, \ldots, A](h_1^1, \ldots, h_m^1, \ldots, h_1^n, \ldots, h_m^n)^\intercal \leq \textstyle\sum_i b_i$$

is box-TDI as well. In particular, this implies that the system $A'h \leq \sum_i b_i$, $0 \leq h \leq 1$ is TDI because the intersection of a box-TDI system with an arbitrary box yields a TDI system. Because $\sum_i b_i$ and the restrictions on h are integral vectors, we conclude that Q_N is indeed integral.

We now show how to construct a separation oracle for Q_N from a separation oracle for P_N. For

$$h = (h_1^1, \ldots, h_m^1, \ldots, h_1^n, \ldots, h_m^n) \in \mathbb{Q}^{mn},$$

let the aggregated vector $f \in \mathbb{Q}^m$ be defined as $f_e = \sum_{k=1}^{n} h_e^k$ for $e = 1, \ldots, m$. Then $h \in Q_N$ if and only if $f \in P_N$. We now give a separation oracle for Q_N.

Let $y = (y_e^k) \in \mathbb{Q}^{mn}$ be an arbitrary rational vector and let f be defined as above with respect to y. We use the separation oracle of P_N to check if $f \in P_N$ or not. If $f \in P_N$, then also $y \in Q_N$ and we are done. Otherwise if $f \notin P_N$ the oracle returns a vector $a \in \mathbb{Q}^m$ such that $a^\mathsf{T} x < a^\mathsf{T} f$ for all $x \in P_N$. In particular this means that $(a^\mathsf{T}, a^\mathsf{T}, \ldots, a^\mathsf{T})z < (a^\mathsf{T}, a^\mathsf{T}, \ldots, a^\mathsf{T})y$ for all $z = (z_e^k) \in Q_N$. Thus, we obtain a separation oracle for Q_N.

We conclude with an analysis of the running time. It is not hard to see that Q_N has a facet complexity that is at most a polynomial (in m and n) factor larger than ϕ. The claim that we only need a number of arithmetic operations and calls to a separation oracle for P_N, that is polynomial in n, m, and ϕ, now follows immediately from Theorem 3.3. This concludes the proof. □

Finally, we consider the case where the aggregation polytope satisfies the middle integral decomposition property (as introduced above) and prove Theorem 3.16.

Proof of Theorem 3.16. By Theorem 3.14, we can compute a feasible load profile f minimizing Rosenthal's potential function in the stated running time. Then f can be decomposed into n integer solutions in P by using the following decomposition algorithm by Baum and Trotter [10]: We start by computing an integral vector $x^1 \in P \cap (f - (n-1)P)$. By the middle integral decomposition property, we know that $P \cap (f - (n-1)P)$ is integral, and therefore we can find an integral (extreme) point in time polynomial in n, m, ϕ and the complexity of a separation oracle of P (using similar arguments as in the proof of Theorem 3.14). Using the same arguments, we can then find a vector $x_2 \in P \cap ((f - x^1) - (n-2)P)$. By repeating this procedure, we find the desired decomposition in the stated running time. □

3.3.3 Applications

We now give several examples of polytopal congestion games for which the aggregation polytope has the (middle) integer decomposition property, is box-TDI and admits an efficient separation oracle. As a consequence, our results on the price of stability (Theorem 3.9) and the computation of Rosenthal's potential minimizer (Theorem 3.14 and Theorem 3.16) apply.

Remark 3.18. In all applications considered below, the facet complexity ϕ of the well-described (aggregation) polyhedra are polynomially bounded in n and m. In particular, all the matrices A considered are in fact $\{0, \pm 1\}$-valued matrices, and the right-hand side vectors b_i are always integral valued.

Common source network congestion games. In a *common source network congestion game* we are given a directed graph $G = (V, A)$ and a source $s \in V$. The strategy set of player $i \in N$ is the set of all directed s, t_i-paths for some $t_i \in V$. Ackermann et al. [1] already showed that one can compute a global optimum of Rosenthal's potential function for these games. We outline how this case can be cast in our framework. The strategies of player i can be described by a polytope $P_i = \{x : Ax = b_i, 0 \leq x \leq 1\}$, where A is the arc-incidence matrix of the network G, and b is the vector with $(b_i)_s = 1$, $(b_i)_{t_i} = -1$ and zero otherwise.[12] The aggregation polytope is then $P_N = \{y : Ay = \sum_{i \in N} b_i, 0 \leq y \leq n\}$. Any feasible load profile minimizing Rosenthal's potential can be decomposed efficiently into a feasible strategy profile, using a similar argument as in [1]. Further, the describing system of P_N is totally unimodular and thus box-TDI.[13]

Symmetric totally unimodular congestion games. *Symmetric totally unimodular congestion games* [54] capture a wide range of combinatorial congestion games. Here the common strategy set of the players is described by a polytope $P = \{x : Ax \leq b\}$ with a totally unimodular $r \times m$-matrix A and an integral vector b. In particular, such a system satisfies the IDP and is box-TDI: The integer decomposition property was shown in [10]. We argue that the system is box-TDI. The constraint matrix describing the intersection of P with $\{x : c \leq x \leq d\}$ for $c, d \in \mathbb{Q}^m$ is again totally unimodular [173]. Any totally unimodular system is TDI (see, e.g., [160, Section 22.1]), and therefore the system $Ax \leq b, c \leq x \leq d$ is TDI. We conclude that the system $Ax \leq b$ is box-TDI. If (as in [54]) the parameter r is considered as part of the input size as well, then there is a trivial (strongly) polynomial separation oracle that simply checks all inequalities of the system $Ax \leq b$. For all combinatorial applications considered in [54] (i.e., network, matching, edge cover, vertex cover and stable set congestion games on bipartite graphs, and their respective extensions to the maximum (or minimum) cardinality versions) the parameter r is actually polynomially bounded in n and m, so then this assumption is justified.

Non-symmetric matroid congestion games. In a *non-symmetric matroid congestion game* $\Gamma = (N, E, (\mathcal{S}_i), (c_e))$, the strategy set of player i is given by the bases $\mathcal{B}_i$ of a matroid $\mathcal{M}_i = (E, \mathcal{I}_i)$ for $i \in N$.[14] The incidence vectors of the bases of $\mathcal{B}_i$ can be described by the base matroid polytope

$$P_i = \{x : x(A) \leq r_i(A), A \subset E,\ x(E) = r_i(E),\ x \geq 0\}$$

as introduced in the preliminaries. That is, for every player we have a polytope of the form $P_i = \{x : Ax \leq b_i, x \geq 0\}$ where b_i is the rank function r_i of the matroid $\mathcal{M}_i$. In particular, it follows that the aggregation polytope is given by

$$P_N = \{y : y(A) \leq \textstyle\sum_i r_i(A), A \subset E,\ y(E) = \sum_i r_i(E),\ y \geq 0\}.$$

[12] Technically, this polytope can also contain paths with a finite number of disjoint cycles, but these can always be removed in the end.

[13] Note that common source network congestion games are not symmetric and are thus not captured by the class of totally unimodular congestion games considered below.

[14] Our framework also captures the *independent set congestion games* studied in [54]. However, we mainly focus on non-negative cost functions here (because of the inefficiency measures) and then these games are trivial.

The polytope P_N has a box-TDI description, which follows from [161, Theorem 46.2].[15] The integer decomposition property is also satisfied (see, e.g., [161, Corollary 46.2c]). Using similar arguments as for r-arborescences (see below), we derive a strongly polynomial time algorithm to compute a minimum of Rosenthal's potential.

We also prove a result that is of independent interest: For non-symmetric matroid congestion games, we can derive a local search algorithm to compute a global minimum of Rosenthal's potential in strongly polynomial time. This local search algorithm can be seen as a natural generalization of best response dynamics (which are known to arrive at a local optimum in polynomial time [1]). Our algorithm is based on a combinatorial approach to compute the symmetric difference decomposition for these games (which is of a specific form). The details are given in Appendix A.

Symmetric matroid intersection congestion games. In *symmetric matroid intersection congestion games* $\Gamma = (N, E, \mathcal{S}, (c_e))$ the (symmetric) strategy set $\mathcal{S}$ of all players is given by the common bases of two matroids $\mathcal{M}_1 = (E, \mathcal{I}_1)$ and $\mathcal{M}_2 = (E, \mathcal{I}_2)$ over a common element set E. The polytope P of the players corresponds to the common base polytope $P_{\mathcal{M}_1,\mathcal{M}_2}$ as defined in (3.7), i.e.,

$$P = \{x : x(A) \leq r_i(A) \ \forall A \subset E,\ x(E) = r_i(E) \text{ for } i = 1, 2,\ x \geq 0\}.$$

The describing system of P is box-TDI (see, e.g., [161, Corollary 41.12e]). Further, as noted in the preliminaries there is a separation oracle for P (and thus P_N) which runs in time polynomial in $|E|$ and the complexity of the independence oracles for $\mathcal{M}_1$ and $\mathcal{M}_2$. However, it is not precisely known for which cases of matroid intersection the integer decomposition property holds.

Example 3.19 (r-Arborescences)**.** Let $D = (V, A)$ be a directed graph. An *r-arboresence* in D is a directed spanning tree rooted in $r \in V$. The set of all r-arboresences can be seen as the set of common bases of two matroids. The first matroid $\mathcal{M}_1$ is the graphic matroid on the undirected graph $D' = (V, A')$, where A' is the set formed by replacing every directed arc in A with its undirected version, i.e., $A' = \{\{u, v\} : (u, v) \in A\}$. The second matroid $\mathcal{M}_2$ is the partition matroid in which independent sets are given by sets of arcs for which there is at most one incoming arc at every node $v \neq r$ (we assume there are no incoming arcs at r). Thus, the common base polytope $P_{\mathcal{M}_1,\mathcal{M}_2}$ describes the arborescences of D and we let $P = P_{\mathcal{M}_1,\mathcal{M}_2}$.

We argue that there is a strongly polynomial time algorithm for computing a minimum of Rosenthal's potential. First note that the describing system of $P_{\mathcal{M}_1,\mathcal{M}_2}$ is box-TDI (see [161, Corollary 41.12e]). Also, $P_{\mathcal{M}_1,\mathcal{M}_2}$ satisfies the integer decomposition property, which follows from Edmonds' Disjoint Arborescences

[15] To see this, we use the fact that the rank function is submodular and that the sum of submodular functions is again submodular. We can then apply Theorem 46.2 in [161].

Theorem [64]. By Theorem 3.14, we can compute a minimum of Rosenthal's potential in time polynomial in n, m, $\sum_i \langle b_i \rangle$ and the complexity of a separation oracle for $P_{\mathcal{M}_1,\mathcal{M}_2}$. The elements of the vector b are bounded by $|E|$, by the definition of the rank functions. Moreover, it is not hard to see that there exist independence oracles for both $\mathcal{M}_1$ and $\mathcal{M}_2$ that run in time polynomial in m. These oracles can be used for separation oracles as described in the preliminaries. It is not hard to see that if both base matroid polytopes have a polynomial time separation oracle, then the intersection of these polytopes has one too. This shows that there is an algorithm for computing an optimal feasible load profile in time polynomial in n and m. Integer decomposition can also be done in time polynomial in n and m [83].

Example 3.20 (Intersection of strongly base-orderable matroids)**.** A matroid $\mathcal{M} = (E, \mathcal{I})$ is strongly base-orderable if for every pair of bases $B_1, B_2 \in \mathcal{B}$ there exists a bijection $\tau : B_1 \to B_2$ such that for every $X \subseteq B_1$, we have $B_1 - X + \tau(X) \in \mathcal{B}$. As in the previous example, a box-TDI description follows from [161, Corollary 41.12e]. It is also known that the independent set polytope of the intersection of strongly base-orderable matroids has the integer decomposition property [125, Theorem 5.1].[16]

3.3.4 Bottleneck congestion games

In this section we provide various results for the computation and inefficiency of equilibria in bottleneck congestion games, based on our polytopal point of view.

A *bottleneck congestion game* $\Gamma = (N, E, (\mathcal{S})_{i \in N}, (c_e)_{e \in E})$ is defined similarly to a congestion game, with the only difference that the objective of a player is to minimize the *maximum* (rather than the aggregated) congestion over all resources that she occupies. Formally, the cost of player $i \in N$ under strategy profile $s = (s_1, \dots, s_n)$ is given by $C_i(s) = \max_{e \in s_i} c_e(x_e(s))$.

Harks et al. [99] give a *dual greedy algorithm* to compute a strong equilibrium, which uses a *strategy packing oracle* as a subroutine (see Appendix B for details). They give efficient packing oracles for symmetric network congestion games, non-symmetric matroid congestion games and (a slight generalization of) r-arborescences. In particular, this leads to polynomial time algorithms for computing a strong equilibrium in these cases.

We adapt their algorithm to compute a load profile of a strong equilibrium for bottleneck polytopal congestion games satisfying the IDP and box-TDI property.

Theorem 3.21. *Let $\Gamma = (N, E, (\mathcal{S}_i), (c_e))$ be a polytopal bottleneck congestion game whose aggregation polytope P_N satisfies IDP and box-TDI. Then a load profile of a strong equilibrium can be computed using at most* $\mathsf{poly}(n, m, \phi)$ *arithmetic operations and separation oracle calls.*

[16] This also implies that the common base polytope has the integer decomposition property, since the integer decomposition property is preserved if we restrict ourselves to a face of a polytope with the integer decomposition property.

ALGORITHM 1: Load profile-dual greedy algorithm.

Input : Bottleneck congestion game $\Gamma = (N, E, (\mathcal{S}_i), (c_e))$, load profile oracle $\mathfrak{O}$
Output : Load profile of strong equilibrium of Γ

set $N' = N$, $u_e = n$ for all $e \in E$, $T = \emptyset$, $L = E$ **and** $a = \mathfrak{O}(T \cup L, P_N, (u_e)_{e \in E})$
while $\{e \in L : u_e > 0\} \neq \emptyset$ **do**
  choose $e' \in \operatorname{argmax}\{c_e(u_e) : e \in L,\, u_e > 0\}$
  $u_{e'} := u_{e'} - 1$
  if $\mathfrak{O}(T \cup L, P_N, (u_e)_{e \in E}) =$ NO **then**
    $u_{e'} := u_{e'} + 1$
    $L = L \setminus \{e'\},\ T = T \cup \{e'\}$
  end
  $a = \mathfrak{O}(T \cup L, P_N, (u_e)_{e \in E})$
end
return $(u_e)_{e \in E}$

We first need to adapt the definition of the strategy packing oracle of [99] (see Appendix B) to load profiles. Below, we assume that P_N is an aggregation polytope of a polytopal congestion game.

Load profile oracle $\mathfrak{O}(E = T \cup L, P_N, (u_e)_{e \in E})$:

Input: A finite set of resources $E = T \cup L$ with upper bounds $(u_e)_{e \in E}$ and an aggregation polytope P_N.

Output: YES, if there exists a feasible load profile $f \in P_N$ such that $f_e = u_e$ for all $e \in T$ and $f_e \leq u_e$ for all $e \in L$; NO otherwise.

Our adaptation of the dual greedy algorithm is given in Algorithm 1. Although the ideas are similar to the ones in [99], our algorithm only works with load profiles. In particular, we do not have to explicitly compute decompositions of feasible load profiles in intermediate steps of the algorithm, which (significantly) improves the running time. Intuitively, our algorithm works as follows. We start with capacities of n on every resource. In every step we pick a resource $e' \in L$ with maximum cost among all resources that are called *loose*, and check whether there is a feasible load profile if we reduce the capacity on e' by one. If this is not possible, we remove e' from L and add e' to the set T of so-called *tight* resources. Note that after the algorithm has terminated, all resources are in the set T.

The following lemma shows that the load profile output by our algorithm corresponds to a strong equilibrium. Its proof is similar to the correctness proof of the dual greedy algorithm by Harks et al. [99].

Lemma 3.22. *Algorithm 1 computes a load profile of a strong equilibrium in time polynomial in n, m and the complexity of the load profile oracle.*

Proof. Clearly, Algorithm 1 can be executed in time polynomial in n, m and the complexity of the load profile oracle. Let f be the load profile output by the algorithm. Assume without loss of generality that the resources in $E = [m]$ are added to the set T in the order $(1, 2, \dots, m)$. Let $s = (s_1, \dots, s_n)$ be a strategy profile corresponding to the load profile f. We define $\sigma_e = \{i \in N : e \in s_i\}$ as the set of players using resource e. Moreover, we set $N_1 = \sigma_1$, and define

$$N_j = \sigma_j \setminus (N_1 \cup \cdots \cup N_{j-1}) \quad \text{for } j = 1, \dots, m.$$

We show that players in N_1 will never participate in a coalitional deviation in which every player strictly improves.

Let $D \subset N$ be a coalition of players that can profitably deviate to strategy profile $t = (t_D, s_{-D})$. Remember that $x_e(t)$ denotes the number of players using resource e in strategy profile t. Let $(u_e)_{e \in E}$ be the capacity vector for which the load profile oracle returns NO for the first time in line 5. We consider two cases:

Case 1: $x_e(t) \leq u_e$ for all $e \in E$. In this case, there must be $|N_1|$ players using resource 1; otherwise, the oracle would have returned YES because the load profile of t would have been feasible for the capacities (u_e). Further, for all players $i \in N_1$, we have $C_i(s) = c_e(x_e(s))$, which is in particular the highest player cost in the strategy profile s. In particular, this implies that if a player in N_1 would strictly improve, then she cannot use resource 1 in t. This means that another player is now using resource 1 in t, but that player can never have strictly better cost than it had in s.

Case 2: $x_e(t) > u_e$ for some $e \in E$. Using similar arguments, we can show that $x_1(t) < u_1$ if some player in N_1 is also part of D (since $c_1(x_1(s))$ is the maximum cost resource in s). Since the algorithm iteratively reduces the capacities of resources with maximum cost, we must have that $c_e(x_e(t)) \geq c_1(u_1)$. Further, since $x_e(t) > u_e$, at least one player in D must be using resource e in t. But this player cannot have strictly improved then.

We can now use induction to show that no player in N_j will ever participate in a coalitional deviation. Assume that the players in $N_1 \cup \cdots \cup N_{j-1}$ will never participate in a coalitional deviation. By using similar arguments as above, we can show that the players in N_j will also never participate in a coalitional deviation. □

Proof of Theorem 3.21. By Lemma 3.22, Algorithm 1 computes a load profile of a strong equilibrium in time polynomial in n, m and the complexity of a load profile oracle.

Based on a separation oracle of P_N, we now show that there is an efficient load profile oracle. Given that P_N has a box-TDI description, it follows that the polytope

$$\{y : Ay \leq \textstyle\sum_{i \in N} b_i\} \cap \{y_e = u_e : e \in T\} \cap \{0 \leq y_e \leq u_e : e \in L\}$$

is integral. We can then use a separation oracle for P_N to find an integral vector in this polytope using at most $\mathsf{poly}(n, m, \phi)$ arithmetic operations and separation oracle calls. This concludes the proof. □

Once we have obtained the feasible load profile, we can use an integer decomposition algorithm to find the corresponding strategies of the players. If the integer decomposition can be done within the same time bounds as stated in Theorem 3.21, we obtain a (strongly) polynomial algorithm for computing a strong equilibrium in a polytopal bottleneck congestion game. In particular, this applies to all applications mentioned in Section 3.3.3.

We end this section by showing that for *matroid bottleneck congestion games*, we can use our techniques from the previous sections to derive an upper bound on the *strong price of stability (SPoS)*.

The proof of the following theorem exploits that Algorithm 1 in fact computes a global optimum of Rosenthal's potential in the case of matroid bottleneck congestion games.

Theorem 3.23. *Let* $\Gamma = (N, E, (\mathcal{S}_i), (c_e))$ *be a non-symmetric matroid bottleneck congestion game with cost functions in class* $\mathcal{D}$*. Then* SPoS$(\Gamma) \leq \rho(\mathcal{D})$.

Proof. Let f be the load profile returned by the algorithm. f is a strong equilibrium by Lemma 3.22. We prove that f is also a global optimum of Rosenthal's potential. We then obtain a bound of $\rho(\mathcal{D})$ on the strong price of stability by using similar arguments as in the proof of Theorem 3.9.

Suppose for contradiction that f is not a global optimum. Then there exist resources a and b such that $c_a(f_a) > c_b(f_b + 1)$ and for which the load profile f', defined by $f'_a = f_a - 1, f'_b = f_b + 1$ and $f'_e = f_e$ for all $e \in E \setminus \{a, b\}$, is feasible. This claim follows from similar arguments as given in the proof of Theorem A.4 (Appendix A).

Now, consider the point in execution of the algorithm where the capacity of resource a was fixed at f_a. Since $c_a(f_a) > c_b(f_b+1)$, we must have had $f_b+1 \leq u_b$ at that point (since the algorithm iteratively reduces the capacity of resources with maximum cost). But this contradicts the fact that the load profile oracle returned NO at this point; to see this note that f' would have been a feasible load profile for the capacity vector in which u_a was reduced by 1 (as in line 4 of the algorithm). $\square$

3.4 Perception-parameterized congestion games

In this section we introduce our unifying model of *perception-parameterized congestion games* with affine cost functions. We first extend some of the definitions given in Section 3.2. For a fixed parameter $\rho \geq 0$, we define the cost of player $i \in N$ by

$$C_i^\rho(s) = \sum_{e \in s_i} c_e(1 + \rho(x_e - 1)) = \sum_{e \in s_i} a_e[1 + \rho(x_e - 1)] + b_e \tag{3.16}$$

for a given strategy profile $s = (s_1, \ldots, s_n)$. For a fixed parameter $\sigma \geq 0$, the social cost of a strategy profile s is given by

$$C^{\sigma}(s) = \sum_{i \in N} C_i^{\sigma}(s) = \sum_{e \in E} x_e(a_e[1 + \sigma(x_e - 1)] + b_e). \tag{3.17}$$

We refer to the case $\rho = \sigma = 1$ as the *classical congestion game setting*, as it corresponds to the standard setting of Rosenthal [150]. We next introduce some notation that naturally generalizes that in Section 3.2.

A strategy profile s is a *pure Nash equilibrium* if for all players $i \in N$ it holds that $C_i^{\rho}(s) \leq C_i^{\rho}(s_i', s_{-i})$ for all $s_i' \in \mathcal{S}_i$, where (s_i', s_{-i}) denotes the strategy profile in which player i plays s_i' and all the other players their strategy in s.

The *price of anarchy (PoA)* and the *price of stability (PoS)* of a game Γ are defined as

$$\mathrm{PoA}(\Gamma, \rho, \sigma) = \frac{\max_{s \in \mathrm{NE}} C^{\sigma}(s)}{\min_{s^* \in \times_i \mathcal{S}_i} C^{\sigma}(s^*)} \quad \text{and} \quad \mathrm{PoS}(\Gamma, \rho, \sigma) = \frac{\min_{s \in \mathrm{NE}} C^{\sigma}(s)}{\min_{s^* \in \times_i \mathcal{S}_i} C^{\sigma}(s^*)},$$

where $\mathrm{NE} = \mathrm{NE}(\rho)$ denotes the set of pure Nash equilibria with respect to the player costs as defined in (3.16). These definitions are natural generalizations of those given in Section 3.2 for the case $\rho = \sigma = 1$. For a collection of games $\mathcal{H}$,

$$\mathrm{PoA}(\mathcal{H}, \rho, \sigma) = \sup_{\Gamma \in \mathcal{H}} \mathrm{PoA}(\Gamma, \rho, \sigma) \quad \text{and} \quad \mathrm{PoS}(\mathcal{H}, \rho, \sigma) = \sup_{\Gamma \in \mathcal{H}} \mathrm{PoS}(\Gamma, \rho, \sigma).$$

Unless stated otherwise, our results refer to the class of perception-parameterized congestion games with affine cost functions; we therefore drop the parameter $\mathcal{H}$ below.

For every fixed parameter $\rho \geq 0$ the Rosenthal potential (3.3) with cost functions as defined in (3.16), i.e.,

$$\bar{c}_e(x) = c_e(1 + \rho(x - 1)) = a_e[1 + \rho(x - 1)] + b_e,$$

is an exact potential function for the corresponding perception-parameterized congestion game. As a consequence, pure Nash equilibria always exist for these games.

In light of the above bounds, we obtain an (almost) complete picture of the inefficiency of equilibria (parameterized by ρ and σ). For example, see Figure 3.3 for an illustration of the price of anarchy for $\sigma = 1$. Note that the price of anarchy *decreases* from $\frac{5}{2}$ for $\rho = 1$ to 2.155 for $\rho = h(1) \approx 0.625$. The price of anarchy for $\rho = h(1)$ was first established by Caragiannis et al. [22]. Note that our bounds imply that the price of anarchy is in fact minimized at $\rho = h(1)$ (see also Figure 3.3). In particular, this shows that the bound of $(4\rho + 1)/(\rho + 1)$ proven in [28] for $1 \leq \rho \leq 2$ continues to hold for $h(1) \leq \rho \leq 1$. This nicely bridges the results in [22] and [28].

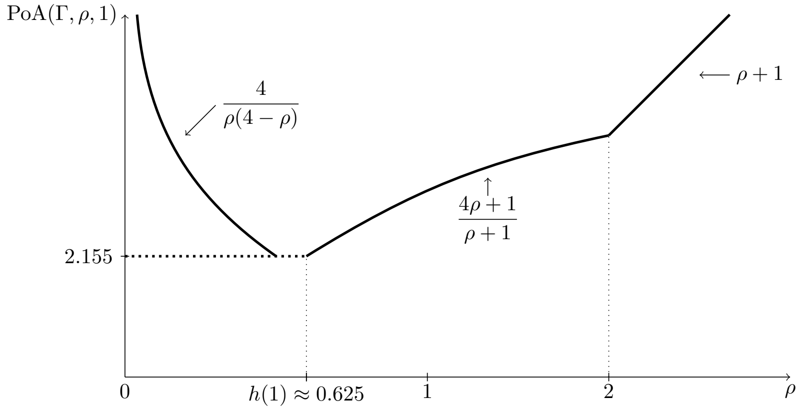

Figure 3.3: Lower bounds on the price of anarchy for $\sigma = 1$. The bounds $(4\rho+1)/(\rho+1)$ and $\rho+1$ are also tight upper bounds. The dotted horizontal line indicates the lower bound following from [23, Theorem 3.7]. The bound $4/(\rho(4-\rho))$ is a lower bound for symmetric singleton congestion games given in [115]. A tight bound for $0 < \rho \leq h(1)$ remains an open problem.

3.4.1 Price of anarchy

In this section, we present our bounds on the price of anarchy; see Figure 3.1 for an illustration. We first start with the simpler proof of the $\rho + 1$ bound (Section 3.4.1.1) and then turn to the more involved proof of the $(2\rho(1+\sigma) + 1)/(\rho+1)$ bound (Section 3.4.1.2). Both bounds are shown to be tight for affine congestion games. For the latter bound, we prove that it is asymptotically tight even for the special case of symmetric network congestion games (Section 3.4.1.3).

We need the following technical lemma.

Lemma 3.24. *Let $\rho, \sigma \geq 0$ be fixed. If there exist $\alpha := \alpha(\rho, \sigma) \geq 0$ and $\beta := \beta(\rho, \sigma) > 0$ such that for all non-negative integers x and y the inequality*

$$(1 + \rho x)y - \rho(x-1)x - x \leq -\beta(1 + \sigma(x-1))x + \alpha(1 + \sigma(y-1))y \tag{3.18}$$

holds, then $PoA(\rho, \sigma) \leq \alpha/\beta$.

Proof. Without loss of generality, we may assume that $a_e = 1$ and $b_e = 0$ for all resources $e \in E$ (see, e.g., [28, Lemma 4.3]). Let s be a Nash equilibrium with respect to the cost functions $C_i^\rho(s)$ and let s^* be a minimizer of $C^\sigma(\cdot)$. Further, let x and x^* be the load profiles for s and s^*, respectively.

We have

$$\begin{aligned}\sum_i C_i^\rho(s) &= \sum_e \rho(x_e - 1)x_e + \sum_e x_e \\ &= \sum_e \rho[1 - \sigma + \sigma](x_e - 1)x_e + \rho x_e - \rho x_e + \sum_e x_e\end{aligned}$$

$$= \rho \sum_e [1 + \sigma(x_e - 1)]x_e + \rho \sum_e (1-\sigma)(x_e - 1)x_e - x_e + \sum_e x_e$$
$$= \rho C^\sigma(s) + \rho \sum_e (1-\sigma)(x_e - 1)x_e + (1-\rho)\sum_e x_e.$$

By rearranging terms, we obtain

$$\begin{aligned}
\rho C^\sigma(s) &= \sum_i C_i^\rho(s) + \rho(\sigma - 1)\sum_e (x_e - 1)x_e + (\rho - 1)\sum_e x_e \\
&\leq \sum_i C_i^\rho(s_i^*, s_{-i}) + \rho(\sigma - 1)\sum_e (x_e - 1)x_e + (\rho - 1)\sum_e x_e \\
&\leq \sum_e [1 + \rho(x_e - 1 + 1)]x_e^* + \rho(\sigma - 1)\sum_e (x_e - 1)x_e + (\rho - 1)\sum_e x_e \\
&= \sum_e [1 + \rho x_e]x_e^* + \rho(\sigma - 1)(x_e - 1)x_e + (\rho - 1)x_e \\
&= \sum_e [1 + \rho x_e]x_e^* + \rho[1 + \sigma(x_e - 1)]x_e - \rho(x_e - 1)x_e - x_e \\
&= \sum_e [1 + \rho x_e]x_e^* - \rho(x_e - 1)x_e - x_e + \rho C^\sigma(s) \\
&\leq -\beta C^\sigma(s) + \alpha C^\sigma(s^*) + \rho C^\sigma(s).
\end{aligned}$$

Here the first inequality holds because s is a Nash equilibrium, the second inequality follows from the definition (3.16) and the last inequality holds because of (3.18). We conclude that $\beta C^\sigma(s) \leq \alpha C^\sigma(s^*)$, which proves the claim. □

We remark that our upper bounds on the price of anarchy of perception-parameterized congestion games can alternatively be proven by adapting the smoothness framework of Roughgarden [155] appropriately; similarly as in [28].

3.4.1.1 First price of anarchy bound

We establish the following tight bound on the price of anarchy.

Theorem 3.25. *We have $PoA(\rho, \sigma) \leq \rho + 1$ for $1 \leq 2\sigma \leq \rho$ and this bound is tight.*

Note that the bound itself does not depend on σ, only the range of ρ and σ for which it holds. For the altruism model of Caragiannis et al. [23] (corresponding to $\sigma = 1$ and $\rho \geq 2$) this bound is known the be tight for non-symmetric singleton congestion games (i.e., all strategies consist of a single resource). Here we only prove tightness for general congestion games, but our construction is significantly simpler.

Proof of Theorem 3.25. By Lemma 3.24 it is sufficient to show that inequality (3.18) holds with $\beta = 1$ and $\alpha = 1 + \rho$. Thus, we have to show that

$$(1 + \rho x)y - \rho(x - 1)x - x \leq -(1 + \sigma(x - 1))x + (1 + \rho)(1 + \sigma(y - 1))y.$$

By rearranging terms, we obtain

$$[y + \sigma y(y-1) - xy + x(x-1)]\,\rho + \sigma\,[y(y-1) - x(x-1)] \geq 0. \tag{3.19}$$

We first show that $[y + \sigma y(y-1) - xy + x(x-1)] \geq 0$ for all $\sigma \geq \frac{1}{2}$. If suffices to show this claim for $\sigma = \frac{1}{2}$, since $y(y-1) \geq 0$ for all $y \in \mathbb{N}$. We have

$$y + \frac{1}{2}y(y-1) - xy + x(x-1) = \frac{1}{2}\left[\left(x - y - \frac{1}{2}\right)^2 - \frac{1}{4} + x(x-1)\right]$$

and this last expression is clearly non-negative for all $x, y \in \mathbb{N}$ (since the quadratic term is always at least $\frac{1}{4}$).

It now suffices to show (3.19) for $\rho = 2\sigma$, since we have shown that the expression is a non-decreasing affine function of ρ, for every fixed $\sigma \geq \frac{1}{2}$. Substituting $\rho = 2\sigma$ and dividing (3.19) by σ, we get the equivalent statement

$$2\,[y + \sigma y(y-1) - xy + x(x-1)] + [y(y-1) - x(x-1)] \geq 0 \tag{3.20}$$

which we will show to be non-negative for all non-negative integers x and y and $\sigma \geq \frac{1}{2}$. Again, it suffices to show the statement for $\sigma = \frac{1}{2}$. The statement in (3.20) is then equivalent to

$$\left(x - y - \frac{1}{2}\right)^2 - \frac{1}{4} + y\,(y-1)$$

which is clearly non-negative for all $x, y \in \mathbb{N}$.

To see that the bound is tight, consider the following game on four resources with two players: Player A has strategies $\{\{1\}, \{2,4\}\}$ and player B has strategies $\{\{2\}, \{1,3\}\}$. Resources $e = 1, 2$ have cost function $c_e(x) = x$ and resources $e = 3, 4$ have cost function $c_e(x) = \rho x$. The optimum $s^* = (\{1\}, \{2\})$ has cost 2, whereas the Nash equilibrium $s = (\{2,4\}, \{1,3\}$ has cost $2(1+\rho)$. □

3.4.1.2 Second price of anarchy bound

We next prove our $(2\rho(1+\sigma)+1)/(\rho+1)$ bound on the price of anarchy. We first establish the upper bound for different ranges of parameters ρ and σ.

Theorem 3.26. *We have*

$$PoA(\rho, \sigma) \leq \frac{2\rho(1+\sigma)+1}{\rho+1} \tag{3.21}$$

for

1. $\frac{1}{2} \leq \sigma \leq \rho \leq 2\sigma$, *or*
2. $\sigma = 1$ *and* $h(\sigma) \leq \rho \leq 2\sigma$, *where* $h(\sigma) = g(1 + \sigma + \sqrt{\sigma(\sigma+2)}, \sigma)$ *is the maximum of the function*

$$g(a, \sigma) = \frac{\sigma(a^2-1)}{(1+\sigma)a^2 - (2\sigma+1)a + 2\sigma(\sigma+1)}.$$

Further, there exists a function $\Delta = \Delta(\sigma)$ (specified in the proof below) satisfying that for every fixed $\sigma_0 \geq \frac{1}{2}$: if $\Delta(\sigma_0) \geq 0$, then the bound in (3.21) *also holds for all $h(\sigma_0) \leq \rho \leq 2\sigma_0$.*

We need the following technical lemma in the proof of Theorem 3.26:

Lemma 3.27. *Let $\sigma \geq \sigma^* := \frac{1}{2}$ be fixed. Then for every $(x, y) \in \mathbb{N}^2 \setminus \{(1, 0)\}$, we have*

$$f_1(x, y, \sigma) := 2y(y-1)\sigma^2 + [x^2 + 2y^2 - 2xy - x]\sigma + [x^2 - xy + 2(y-x)] \geq 0.$$

Proof. Note that $2y(y-1) \geq 0$ for all $y \in \mathbb{N}$. Furthermore,

$$x^2 + 2y^2 - 2xy - x = (x-y)^2 + y^2 - x \geq (x-y)^2 + (y-x) \geq 0$$

for all $(x, y) \in \mathbb{N}^2$ because $a^2 - a \geq 0$ for all $a \in \mathbb{N}$. Thus $f_1(x, y, \sigma)$ is non-decreasing and it suffices to prove the statement for $\sigma^* = \frac{1}{2}$.

We need to prove

$$\begin{aligned} &\frac{1}{2}y(y-1) + \frac{1}{2}[x^2 + 2y^2 - 2xy - x] + [x^2 - xy + 2(y-x)] \geq 0 \\ \Leftrightarrow \quad & y(y-1) + x^2 + 2y^2 - 2xy - x + 2x^2 - 2xy + 4(y-x) \geq 0. \end{aligned}$$

By simplifying we obtain

$$3x^2 + 3y^2 - 4xy + 3y - 5x \geq 0 \quad \Leftrightarrow \quad 2(x-y)^2 + x(x-5) + y(y+3) \geq 0.$$

The last inequality clearly holds for all pairs (x, y) with $x \geq 5$. For $x = 4$, we find $2(4-y)^2 - 4 + y(y+3) \geq 0$ which is true for $y \geq 1$, and for $y = 0$ it can be verified by inspection. For $x = 3$, we find $2(3-y)^2 - 6 + y(y+3) \geq 0$ which is true for $y \geq 2$, and for $y \in \{0, 1\}$ it can be verified by inspection. For $x = 2$, we find $2(2-y)^2 - 6 + y(y+3) \geq 0$ which again holds for $y \geq 2$, and for $y \in \{0, 1\}$ it can be verified by inspection. For $x = 1$, we find $2(1-y)^2 - 4 + y(y+3) \geq 0$ which holds for $y \geq 1$. For $y = 0$ the inequality does not hold, but this is the case $(x, y) = (1, 0)$ which we explicitly excluded in the claim. For $x = 0$, the inequality holds. □

We now give the formal proof of Theorem 3.26.

Proof of Theorem 3.26. We first show that inequality (3.18) of Lemma 3.24 holds for the functions $\alpha(\rho, \sigma) = (2\rho(1+\sigma)+1)/(1+2\sigma)$ and $\beta(\rho, \sigma) = (1+\rho)/(1+2\sigma)$. That is,

$$(1+\rho x)y - \rho(x-1)x - x \leq -\frac{1+\rho}{1+2\sigma}(1+\sigma(x-1))x + \frac{2\rho(1+\sigma)+1}{1+2\sigma}(1+\sigma(y-1))y. \tag{3.22}$$

Multiplying both sides by $(1+2\sigma)$, we obtain

$$\begin{aligned} &(1+2\sigma)\left[(1+\rho x)y - \rho(x-1)x - x\right] \leq \\ &\qquad -(1+\rho)(1+\sigma(x-1))x + (2\rho(1+\sigma)+1)(1+\sigma(y-1))y, \end{aligned}$$

which we rewrite as $f_1(x,y,\sigma)\rho + f_2(x,y,\sigma) \geq 0$, where

$$\begin{aligned} f_1(x,y,\sigma) &= -(1+\sigma(x-1))x + 2(1+\sigma)(1+\sigma(y-1))y \\ &\quad + (1+2\sigma)((x-1)x - xy) \\ &= 2y(y-1)\sigma^2 + (-(x-1)x + 2(y-1)y + 2y + 2x(x-1) - 2xy)\sigma \\ &\quad + (-x + 2y + (x-1)x - xy) \\ &= 2y(y-1)\sigma^2 + [x^2 + 2y^2 - 2xy - x]\sigma + [x^2 - xy + 2(y-x)] \end{aligned}$$

and

$$\begin{aligned} f_2(x,y,\sigma) &= -(1+\sigma(x-1))x + (1+\sigma(y-1))y + (1+2\sigma)(x-y) \\ &= \sigma y(y-1) - \sigma x(x-1) + 2\sigma(x-y) \\ &= \big(y^2 - x^2 + 3(x-y)\big)\sigma. \end{aligned}$$

We first consider the case $(x,y) = (1,0)$. In this case, we do no have $f_1(x,y,\sigma) \geq 0$. Substituting the values for x and y, we obtain $-\rho + 2\sigma \geq 0$, which is true if and only if $\rho \leq 2\sigma$.

Case i). Suppose $\frac{1}{2} \leq \sigma \leq \rho \leq 2\sigma$. For the pair $(x,y) = (1,0)$, the inequality is true if and only if $\rho \leq 2\sigma$. For all other pairs, we have $f_1(x,y,\sigma) \geq 0$, and hence

$$f_1(x,y,\sigma)\rho + f_2(x,y,\sigma) \geq f_1(x,y,\sigma)\sigma + f_2(x,y,\sigma)$$

meaning that is suffices to show that $f_1(x,y,\sigma)\sigma + f_2(x,y,\sigma) \geq 0$. Dividing by σ, this is equivalent to

$$\begin{aligned} & 2y(y-1)\sigma^2 + [x^2 + 2y^2 - 2xy - x]\sigma + [x^2 - xy + 2(y-x)] \\ & \qquad\qquad + \big(y^2 - x^2 + 3(x-y)\big) \geq 0 \\ \Leftrightarrow \quad & 2y(y-1)\sigma^2 + [x^2 + 2y^2 - 2xy - x]\sigma + [y^2 - xy + (x-y)] \geq 0. \end{aligned}$$

Again, we see that the terms before σ^2 and σ are non-negative for all $x, y \in \mathbb{N}$ (see proof of Lemma 3.27). Thus, if the inequality holds for some σ^*, then it holds for all $\sigma \geq \sigma^*$. We take $\sigma^* = \frac{1}{2}$. Multiplying the resulting inequality by 2, we obtain

$$\begin{aligned} y(y-1) + [x^2 + 2y^2 - 2xy - x] + 2[y^2 - xy + (x-y)] \geq 0 \\ \Leftrightarrow \quad x^2 + 5y^2 - 4xy - 3y + x \geq 0 \\ \Leftrightarrow \quad (x-2y)^2 + y(y-3) + x \geq 0. \end{aligned}$$

The latter inequality holds for all $y \notin \{1,2\}$. For $y = 1$, we find $(x-2)^2 - 2 + x \geq 0$. The inequality clearly holds for all $x \geq 2$, and for $x \in \{0,1\}$ it can be verified by inspection. For $y = 2$, we find $(x-4)^2 - 2 + x \geq 0$. This is again clearly true for $x \geq 2$, and can be checked by inspection for $x \in \{0,1\}$.

Case ii). We now prove the second claim of the theorem. If $(x,y) \in \mathbb{N}^2 \setminus \{(1,0)\}$, then $f_1(x,y,\sigma) \geq 0$ by Lemma 3.27, meaning that $f_1(x,y,\sigma)\rho +$

$f_2(x, y, \sigma)$ is non-decreasing in ρ. From the proof of Lemma 3.27, it follows that $f_1(x, y, \sigma) = 0$ if and only if $(x, y) \in \{(1, 1), (2, 1)\}$ (which can be seen by checking all the cases). Note that this observation is independent of σ. For $(x, y) \in \{(1, 1), (2, 1)\}$ it also holds that $f_2(x, y, \sigma) = 0$, which implies that $f_1(x, y, \sigma)\rho + f_2(x, y, \sigma) = 0$ for every ρ. Therefore, we can focus on pairs (x, y) for which $f_1(x, y, \sigma) > 0$. It follows that any ρ^* for which

$$\rho^* \geq \sup_{x,y \in \mathbb{N}: f_1(x,y,\sigma) > 0} -\frac{f_2(x, y, \sigma)}{f_1(x, y, \sigma)}$$

yields the inequality for all $\rho \geq \rho^*$. It is not hard to see that this supremum is indeed finite, for every fixed σ. It can be proved that $f_1(x, y, \sigma)\rho' + f_2(x, y, \sigma) \geq 0$ holds for some large constant ρ', which then serves as an upper bound on the supremum. For the pair $(x, y) = (0, 1)$, we find $-f_2/f_1 = \sigma/(1 + \sigma)$, but we will see later that the supremum on the other pairs obtained is larger than $\sigma/(1+\sigma)$.

Note that by now, we can focus on pairs in $\{(x, y) : x \geq 1, y \geq 2\}$, since for all other pairs we have either proven the inequality or given $-f_2/f_1$. That is, we are interested in

$$\sup_{\{(x,y):x\geq 1,y\geq 2\}} -\frac{f_2(x, y, \sigma)}{f_1(x, y, \sigma)}. \tag{3.23}$$

Note that $f_2(x, y, \sigma) = \big(y^2 - x^2 + 3(x - y)\big)\sigma = (x + y - 3)(y - x) \geq 0$ if $y \geq x$ (using that $x + y \geq 3$ for $(x, y) \in \{(x, y) : x \geq 1, y \geq 2\}$). Hence, if $y \geq x$ we have $-f_2/f_1 \leq 0$, so these pairs are not relevant for the supremum (if it follows that the upper bound on the supremum for all other pairs is positive, which we will indeed see later). Therefore, we can focus on pairs with $y < x$.

We substitute $x = ay$ for some (rational) $a > 1$. Note that

$$\sup_{a\in\mathbb{R}_{>1}} \sup_{y\geq 2} -\frac{f_2(ay, y, \sigma)}{f_1(ay, y, \sigma)} \tag{3.24}$$

provides an upper bound on (3.23). Using the identities above, we have

$$\begin{aligned}
f_1(ay, y, \sigma) &= [(1 + \sigma)a^2 - (2\sigma + 1)a + 2\sigma(\sigma + 1)]y^2 - [(2 + \sigma)a + 2\sigma^2 - 2]y \\
-f_2(ay, y, \sigma) &= [(a^2 - 1)\sigma]y^2 + [3(1 - a)\sigma]y
\end{aligned}$$

We determine an upper bound on the expression

$$\begin{aligned}
-\frac{f_2(ay, y, \sigma)}{f_1(ay, y, \sigma)} &= \frac{[(a^2 - 1)\sigma]y^2 + [3(1 - a)\sigma]y}{[(1 + \sigma)a^2 - (2\sigma + 1)a + 2\sigma(\sigma + 1)]y^2 - [(2 + \sigma)a + 2\sigma^2 - 2]y} \\
&= \frac{[(a^2 - 1)\sigma]y + [3(1 - a)\sigma]}{[(1 + \sigma)a^2 - (2\sigma + 1)a + 2\sigma(\sigma + 1)]y - [(2 + \sigma)a + 2\sigma^2 - 2]} \\
&= \frac{\alpha y + \beta}{\gamma y - \delta}
\end{aligned} \tag{3.25}$$

for $y \geq 2$. Elementary calculus shows that the derivative with respect to y of (3.25) is given by $-(\alpha\delta + \gamma\beta)/(\gamma y - \delta)^2$, which means the expression in (3.25) is non-decreasing or non-increasing in y. We have

$$\begin{aligned}
\alpha\delta + \gamma\beta &= (a^2-1)\sigma[(2+\sigma)a + 2\sigma^2 - 2] + 3(1-a)[(1+\sigma)a^2 \\
&\quad - (2\sigma+1)a + 2\sigma(1+\sigma)] \\
&= (1-a)\sigma\Big[-(1+a)((2+\sigma)a + 2\sigma^2 - 2) + 3((1+\sigma)a^2 \\
&\quad - (2\sigma+1)a + 2\sigma(1+\sigma))\Big] \\
&= (1-a)\sigma\Big[(3(1+\sigma)-(2+\sigma))a^2 + (2-(2+\sigma)-2\sigma^2 \\
&\quad - 3(2\sigma+1))a + 2 - 2\sigma^2 + 6\sigma(1+\sigma)\Big] \\
&= (1-a)\sigma\left[(2\sigma+1)a^2 - (2\sigma^2+7\sigma+3)a + (4\sigma^2+6\sigma+2)\right] \\
&= (1-a)\sigma\left[(2\sigma+1)a^2 - (2\sigma+1)(\sigma+3)a + (2\sigma+1)(2\sigma+2)\right] \\
&= (1-a)\sigma(1+2\sigma)\left[a^2 - (\sigma+3)a + (2\sigma+2)\right] \\
&= (1-a)\sigma(1+2\sigma)\left[\left(a - \frac{\sigma+3}{2}\right)^2 - \frac{1}{4}(1-\sigma)^2\right]. \qquad (3.26)
\end{aligned}$$

Before we can proceed, we need to prove the following claim.

Claim 1. *The function $x_2 = (\alpha x_1 + \beta)/(\gamma x_1 - \delta)$ has a vertical asymptote at $x_1^* = \delta/\gamma < 2$.*

Proof. It is not hard to verify that x_2 has a vertical asymptote at $x_1^* = \delta/\gamma$. Note that since $a > 1$ we have $\delta > 0$ for all $\sigma \geq 0$. If $\gamma < 0$ then $x_1^* < 0$. If $\gamma > 0$, we claim that $x_1^* < 2$. This is equivalent to showing that

$$(2+\sigma)a + 2\sigma^2 - 2 < 2(1+\sigma)a^2 - 2(2\sigma+1)a + 4\sigma(\sigma+1),$$

which holds if and only if

$$\begin{aligned}
2(1+\sigma)a^2 - (5\sigma+4)a + 2(1+\sigma)^2 = 2(1+\sigma)\Bigg(&\left[a - \frac{5\sigma+4}{4(1+\sigma)}\right]^2 \\
&- \frac{1}{4}\left[\frac{5\sigma+4}{2(1+\sigma)}\right]^2 + (1+\sigma)\Bigg) > 0.
\end{aligned}$$

If now suffices to show that

$$-\frac{1}{4}\left[\frac{5\sigma+4}{2(1+\sigma)}\right]^2 + (1+\sigma) > 0.$$

But this is true for all $\sigma > 0$ and thus the claim follows. □

We can now conclude the proof of Theorem 3.26 by distinguishing three cases:

Case $\sigma = 1$**.** It follows that the expression in (3.26) is non-positive for all $a > 1$, which implies that $-(\alpha\delta + \gamma\beta)/(\gamma y - \delta)^2 \geq 0$ and hence $-f_2/f_1$ is non-decreasing in $y \geq 2$ for every $a > 1$ (using Claim 1). We obtain

$$\lim_{y\to\infty} -\frac{f_2(ay, y, \sigma)}{f_1(ay, y, \sigma)} = \frac{\sigma(a^2-1)}{(1+\sigma)a^2 - (2\sigma+1)a + 2\sigma(\sigma+1)} =: h_1(a, \sigma)$$

and maximizing this function over $a \in \mathbb{R}_{>1}$, we find the optimum

$$a^*(\sigma) = 1 + \sigma + \sqrt{\sigma(\sigma+2)}. \tag{3.27}$$

Case $\frac{1}{2} \leq \sigma < 1$**.** More generally, for any $\sigma < 1$ it holds that $\alpha\delta + \gamma\beta \leq 0$ if and only if $a \notin (1+\sigma, 2)$. In particular for every $a \notin (1+\sigma, 2)$, we can then show that

$$\sup_{y\geq 2} -\frac{f_2(ay, y, \sigma)}{f_1(ay, y, \sigma)} \leq \lim_{y\to\infty} -\frac{f_2(a^*y, y, \sigma)}{f_1(a^*y, y, \sigma)} \tag{3.28}$$

with a^* as in (3.27) using the same argument as in the case $\sigma = 1$. Claim 1 implies that if the expression (3.25) is non-increasing in y, which is the case when $a \in (1+\sigma, 2)$, then the maximum value is attained in $y = 2$. That is, we are interested in the expression $-f_2(2a, 2, \sigma)/f_1(2a, 2, \sigma)$, and in particular, we want to show that the supremum over $a \in (1+\sigma, 2)$ does not exceed the right hand side of (3.28), i.e., the supremum over all $a \notin (1+\sigma, 2)$.

Given the discussion above, it suffices to study

$$\begin{aligned} -\frac{f_2(2a, 2, \sigma)}{f_1(2a, 2, \sigma)} &= \frac{[(a^2-1)\sigma]2 + [3(1-a)\sigma]}{[(1+\sigma)a^2 - (2\sigma+1)a + 2\sigma(\sigma+1)]2 - [(2+\sigma)a + 2\sigma^2 - 2]} \\ &= \frac{\sigma(2a^2 - 3a + 1)}{2(1+\sigma)a^2 - (5\sigma+4)a + 2(1+\sigma)^2} =: h_2(a, \sigma) \end{aligned} \tag{3.29}$$

for $a \in (1+\sigma, 2)$. For $a > 1$, this expression is maximized for

$$b^*(\sigma) = 1 + \sigma + \sqrt{\sigma\left(\sigma + \frac{1}{2}\right)}, \tag{3.30}$$

which in particular gives an upper bound for $a \in (1+\sigma, 2)$.

It now suffices to show that

$$\Delta(\sigma) := h_1(a^*(\sigma), \sigma) - h_2(b^*(\sigma), \sigma) \geq 0,$$

since this implies that the supremum over $a > 1$ in (3.24) is attained at some $a \notin (1+\sigma, 2)$. While unfortunately we lack an analytical proof of this inequality, it can be verified numerically (see [115]).

Case $\sigma > 1$**.** Here we can use a similar reasoning as in the previous case. The only difference is that now the expression in (3.25) is non-increasing for $a \in (2, 1+\sigma)$, but this does not affect the above arguments because we maximize over all $a > 1$ when obtaining $b^*(\sigma)$. □

Numerical experiments suggest that $\Delta(\sigma)$ is non-negative for all $\sigma \geq \frac{1}{2}$. In [115] we describe a procedure to verify this for $\sigma \in [\frac{1}{2}, \bar{\sigma}]$ for any given $\bar{\sigma}$. We emphasize that for a fixed σ with $\Delta(\sigma) \geq 0$, the proof that the inequality holds for all $h(\sigma) \leq \rho \leq 2\sigma$ is exact in the parameter ρ. The first two cases of Theorem 3.26 capture all price of anarchy results from the literature.

We now show that the bound in Theorem 3.26 is tight for arbitrary $\rho, \sigma \geq 0$. To this aim, we generalize the lower bound construction of Christodoulou and Koutsoupias [34] for classical congestion games with $\rho = \sigma = 1$. This construction is also adapted in the risk-uncertainty model by Nikolova et al. [146] and the altruism model by Chen et al. [28].

Theorem 3.28. *For $\rho, \sigma > 0$ fixed, there exists a linear congestion game such that*

$$PoA(\rho, \sigma) \geq \frac{2\rho(1+\sigma)+1}{\rho+1}.$$

Proof. We construct a congestion game of $n \geq 3$ players and $|E| = 2n$ resources. The set E is divided in the sets $E_1 = \{h_1, \ldots, h_n\}$ and $E_2 = \{g_1, \ldots, g_n\}$. Player i has two pure strategies: $\{h_i, g_i\}$ and $\{h_{i+1}, g_{i-1}, g_{i+1}\}$, where the indices appear as i mod n. The cost functions of the elements in E_1 are $c_e(x) = x$, whereas the cost functions of the elements in E_2 are $c_e(x) = \rho x$.

Regardless which strategy player i plays, he always uses at least one resource from both E_1 and E_2, implying that $C_i^\sigma(s) \geq \rho + 1$. This implies that

$$C^\sigma(t) = \sum_{i \in N} C_i^\sigma(s) \geq (\rho+1)n \tag{3.31}$$

for every strategy profile t, and in particular for a social optimum s^*.

We will now show that the strategy profile s where every agent i plays its second strategy $\{h_{i+1}, g_{i-1}, g_{i+1}\}$ is a Nash equilibrium. We have

$$C_i^\rho(s) = 2\rho[1 + \rho(2-1)] + 1 = 2\rho^2 + 2\rho + 1.$$

If some agent i deviates to its first strategy s_i', we have

$$C_i^\rho(s_i', s_{-i}) = \rho[1 + \rho(3-1)] + (1 + \rho(2-1)) = 2\rho^2 + 2\rho + 1,$$

since there are then three agents using g_i and two agents using h_i. This shows that s is a Nash equilibrium. The social cost of this strategy s is

$$C^\sigma(s) = n(1 + 2\rho[1 + \sigma(2-1)]) = (1 + 2\rho(1+\sigma))\, n. \tag{3.32}$$

Combining (3.32) with (3.31) then gives the desired result. $\square$

3.4.1.3 Lower bound for symmetric network congestion games

In this section, we show that the bound of Theorem 3.26 is asymptotically tight even for the special case of linear symmetric network congestion games. This improves a result for the risk-uncertainty model by Piliouras et al. [146], who prove asymptotic tightness for symmetric linear congestion games for their respective values of ρ and σ only. It also improves a result for the altruism model by Chen et al. [28], who show tightness only for general congestion games.

For the classical congestion game setting with $\rho = \sigma = 1$, Christodoulou and Koutsoupias [34] showed that for symmetric affine congestion games the bound of $\frac{5}{2}$ on the price of anarchy is asymptotically tight. More recently, Correa et al. [44] proved that the bound of $\frac{5}{2}$ is tight for symmetric network affine congestion games. Our lower bound proof is a generalization of their construction.

Theorem 3.29. *For $\rho, \sigma > 0$ fixed, there exists a symmetric network affine congestion game such that for every $\epsilon > 0$,*

$$PoA(\rho, \sigma) \geq \frac{2\rho(1+\sigma)+1}{\rho+1} - \epsilon.$$

Proof. We construct a symmetric network linear congestion game with n players. We first describe the graph topology used in the proof of Theorem 5 in [44] (using similar notation and terminology). The graph G consists of n principal disjoint paths $P_1, \dots, P_n$ from s to t (horizontal paths in Figure 3.4 with P_1 and P_n being the topmost and bottommost paths, respectively), each consisting of $2n-1$ arcs (and hence $2n$ nodes). With $e_{i,j}$ the j-th arc on path i is denoted for $i = 1, \dots, n$ and $j = 1, \dots, 2n-1$. Also, $v_{i,j}$ denotes the j-th node on path i for $i = 1, \dots, n$ and $j = 1, \dots, 2n$. There are also $n(n-1)$ connecting arcs: for every path i there is an arc $(v_{i,2k+1}, v_{i+1,2k})$ for $k = 1, \dots, n$, where $i+1$ is taken modulo n (the diagonal arcs in Figure 3.4). For $j \geq 1$ fixed, we say that the arcs $e_{i,j}$ for $i = 1, \dots, n$ form the $(j-1)$-th layer of G (see Figure 3.4).

The cost functions are as follows. All arcs leaving s (the arcs $e_{i,1}$ for $i = 1, \dots, n$) and all arcs entering t (the arcs $e_{i,2n}$ for $i = 1, \dots, n$) have cost $c_e(x) = (1+\rho)x$. For all $i = 1, \dots, n$, the arcs $e_{i,2k-1}$ for $k = 1, \dots, n-1$ have cost function $c_e(x) = \rho x$, whereas the arcs $e_{i,2k}$ for $k = 1, \dots, n-2$ have cost function $c_e(x) = x$. All other arcs (the diagonal connecting arcs) have cost zero.

The feasible strategy profile t in which player i uses principal path P_i, for all $i = 1, \dots, n$ has social cost $C^\sigma(t) = n(2(1+\rho) + (n-1)\rho + (n-2)) = n((1+\rho)n + \rho)$. A Nash equilibrium is given by the strategy profile in which every player k uses the following path: she starts with arcs $e_{k,1}$ and $e_{k,2}$, then uses all arcs of the form $e_{k+j,2j}, e_{k+j,2j+1}, e_{k+j,2j+2}$ for $j = 1, \dots, n-1$, and ends with arcs $e_{k+n-1,2n-2}, e_{k+n-1,2n-1}$ (and uses all connecting arcs in between).[17] Note that all the (principal) arcs of layer j have load 1 is j is even, and load 2 if j is odd. The social cost of this profile is given by $C^\sigma(s) = n(2(1+\rho) + (n -$

[17] This is similar to the construction in [44, Theorem 5].

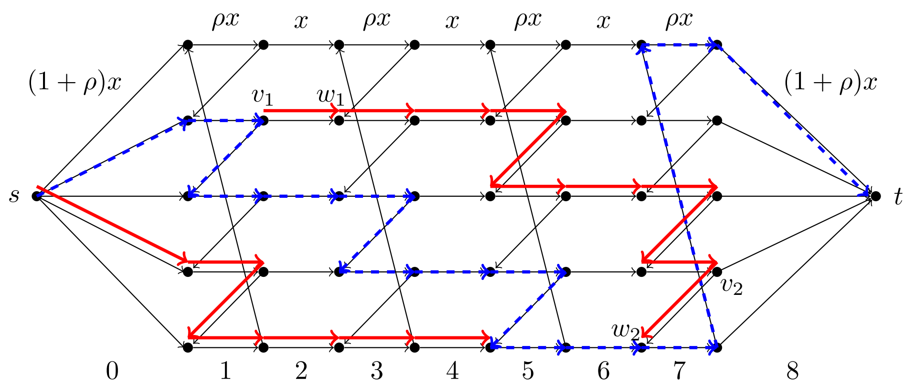

Figure 3.4: Illustration of the instance for $n = 5$. The dashed (blue) path indicates the strategy of player 2 in the Nash equilibrium. For every principal path, the first and last arc have cost $(1+\rho)x$, and in between the costs alternate between ρx and x (starting and ending with ρx). The diagonal connecting arcs have cost zero. The numbers at the bottom indicate the layers. The bold (red) subpaths indicate the two deviation situations that are analyzed to prove that s is indeed a Nash equilibrium.

$1) \cdot 2 \cdot \rho(1+\sigma(2-1)) + n - 2) = n((1+2\rho(1+\sigma))n - 2\rho\sigma)$. It follows that $C^\sigma(s)/C^\sigma(t) \uparrow (1+2\rho(1+\sigma))/(1+\rho)$ as $n \to \infty$. We now show that the above mentioned strategy profile s is indeed a Nash equilibrium.

Fix some player, say player 2, as in Figure 3.4, and suppose that this player deviates to some path Q. Let j be the first layer in which P_2 and Q overlap. Note that j must be odd. The cost $C_2^\rho(s)$ of player 2, on the subpath of P_2 leading to the first overlapping arc with Q, is at most

$$\begin{aligned}(1+\rho) + \frac{j-1}{2} \cdot [2 \cdot [\rho(1+\rho(2-1))] + 1] + \rho(1+\rho(2-1))\\ = (1+\rho)^2 + \frac{j-1}{2}(1+2\rho(1+\rho))\end{aligned}$$

The subpath of Q leading to the first overlapping arc with P_2 has $C_i^\rho(Q, s_{-i})$ as follows. She uses at least one arc in every odd layer (before the overlapping layer) with a load of 3 and one arc of every even layer (before the overlapping arc) with load 2, meaning that the cost of player i on the subpath of Q is at least

$$\begin{aligned}(1+\rho)(1+\rho(2-1)) + \frac{j-1}{2} \cdot [(\rho(1+\rho(3-1))) + (1+\rho(2-1))]\\ = (1+\rho)^2 + \frac{j-1}{2} \cdot (2\rho+1)(1+\rho)\end{aligned}$$

Since $1+2\rho(1+\rho) < (2\rho+1)(1+\rho)$ for all $\rho \geq 0$, if follows that the cost of player i on the subpath of P_2 is no worse than that of the subpath of Q, when player 2 deviates from P_2 to Q. If follows that it suffices to show that P_2 is an equilibrium strategy in s with respect to deviations Q that overlap on the first arc $e_{2,1}$ with P_2. A similar argument shows that it also suffices to look at deviations Q for which Q and P_2 overlap on the last arc $e_{2,2n-1}$ of P_2.

Now suppose that P_2 and Q do not overlap on some internal part of P_2. Note that the first arc of Q that is not contained in P_2, say (v_1, w_1) must be in an even layer, and also that the last arc, say (v_2, w_2) (which is a connecting arc) is in an odd layer (note that $v_1 \neq s$ and $w_2 \neq t$ w.l.o.g. by what is said in the previous paragraph). It is not hard to see that the subpath of Q from v_1 to w_2 contains the same number of even-layered arcs as the subpath of P_2, and the same number of odd-layered arcs as the subpath of P_2. However, the load on all the odd-layered arcs on the subpath of deviation Q is 3, whereas the load on odd-layered arcs in the subpath of P_2 between v_1 and w_2 (in strategy s) is 2. Similarly, the load on every even-layered arc on the subpath of deviation Q is 2, whereas the load on ever even-layered arc in the subpath of P_2 is 1. Hence the subpath of deviation Q between v_1 and w_2 can never be profitable. $\square$

3.4.2 Price of stability

In this section, we present our bound on the price of stability for pure Nash equilibria in general affine congestion games. We first establish our upper bound and show that it is tight afterwards.

Theorem 3.30. *We have*

$$PoS(\rho, \sigma) \leq \frac{\sqrt{\sigma(\sigma+2)} + \sigma}{\sqrt{\sigma(\sigma+2)} + \rho - \sigma} \quad \textit{for} \quad \frac{2\sigma}{1 + \sigma + \sqrt{\sigma(\sigma+2)}} \leq \rho \leq 2\sigma$$

and $\sigma > 0$. This bound is asymptotically tight.

We need the following technical lemma.

Lemma 3.31. *Let $\sigma \geq 0$ be fixed. For all non-negative integers x and y we have*

$$\left(x - y + \frac{1}{2}\right)^2 - \frac{1}{4} + 2\sigma x(x-1) + (\sqrt{\sigma(\sigma+2)} + \sigma)[y(y-1) - x(x-1)] \geq 0.$$

Proof. The inequality is clearly true for all $y \geq x$ so we focus on the case $y < x$. By rewriting the inequality, we obtain

$$\begin{aligned}(1 + \sigma + \sqrt{\sigma(\sigma+2)})y^2 - 2xy + (1 + \sigma - \sqrt{\sigma(\sigma+2)})x^2 \\ - (1 + \sigma + \sqrt{\sigma(\sigma+2)})y + (1 - \sigma + \sqrt{\sigma(\sigma+2)})x \geq 0.\end{aligned}$$

By multiplying both sides with $1 + \sigma - \sqrt{\sigma(\sigma+2)}$ (which is non-negative for all $\sigma \geq 0$) and exploiting that $(1 + \sigma + \sqrt{\sigma(\sigma+2)})(1 + \sigma - \sqrt{\sigma(\sigma+2)}) = 1$, we obtain

$$y^2 - 2\left(1 + \sigma - \sqrt{\sigma(\sigma+2)}\right)xy + \left(1 + \sigma - \sqrt{\sigma(\sigma+2)}\right)^2 x^2$$

$$-y + (1+\sigma-\sqrt{\sigma(\sigma+2)})(1-\sigma+\sqrt{\sigma(\sigma+2)})x \geq 0.$$

This is equivalent to

$$\left(\left(1+\sigma-\sqrt{\sigma(\sigma+2)}\right)x - y + \frac{1}{2}\right)^2 + \left(1+\sigma-\sqrt{\sigma(\sigma+2)}\right)\left(\left[1+\sigma-\sqrt{\sigma(\sigma+2)}\right]-1\right)x - \frac{1}{4} \geq 0.$$

Define $c := c(\sigma) = 1+\sigma-\sqrt{\sigma(\sigma+2)}$. Note that $c(\sigma)$ is a bijective function from $\mathbb{R}$ to $[0,1)$. Substituting c in the above inequality, we obtain for $0 \leq c < 1$ the equivalent formulation

$$\left(cx - y + \frac{1}{2}\right)^2 + c(1-c)x - \frac{1}{4} \geq 0. \tag{3.33}$$

For $x = 0$, the inequality reduces to $(\frac{1}{2}-y)^2 - \frac{1}{4} \geq 0$ which is true for all $y \in \mathbb{N}$. For $x = 1$, we get the equivalent formulation $(y-1)(y-2c) \geq 0$, which is clearly true for $y = 1$. For $y = 0$, it follows from the fact that $c \geq 0$. For $y \geq 2$ it follows from the fact that $y - 2c \geq 0$ for all $y \geq 2$, since $0 \leq c < 1$. This completes the case $x = 1$.

For $x \geq 2$, we rewrite the expression (3.33) to

$$x(x-1)c^2 + 2x(1-y)c + y(y-1) \geq 0. \tag{3.34}$$

If $y = 0$, the expression in (3.34) is clearly non-negative for all $x \geq 2$ and $0 \leq c < 1$. For $y \geq 1$, note that $g(c) = x(x-1)c^2 + 2x(1-y)c + y(y-1)$ is a quadratic and convex function for all fixed x and y. Therefore, in particular, for any x and y fixed, it suffices to show that the inequality holds for the minimizer of g, which is $c^* = (y-1)/(x-1)$ (which can be found by differentiating with respect to c). Note that $0 \leq c^* < 1$ by our assumption that $y \geq 1$ and $y < x$ (made at the beginning of the proof). Substituting implies that it suffices to show that

$$\frac{x(x-1)(y-1)^2}{(x-1)^2} + \frac{2x(1-y)(y-1)}{x-1} + y(y-1) \geq 0.$$

Multiplying the expression with $(x-1)$ implies that it now suffices to show that

$$x(y-1)^2 - 2x(y-1)^2 + y(y-1)(x-1) \geq 0$$

for all $1 \leq y < x$. This is always true since

$$\begin{aligned} x(y-1)^2 - 2x(y-1)^2 + y(y-1)(x-1) &= -x(y-1)^2 + y(y-1)(x-1) \\ &= (y-1)\left[-x(y-1) + y(x-1)\right] \\ &= (y-1)(x-y) \\ &\geq 0 \end{aligned}$$

whenever $1 \leq y < x$. This completes the proof. □

Our proof is similar to the approach used by Christodoulou, Koutsoupias and Spirakis [35] to upper bound the price of stability of ρ-approximate equilibria. However, for general σ the analysis is more involved. The main technical contribution is to establish the inequality in Lemma 3.31. The proof of the asymptotic tightness is also based on a construction given in [35] to obtain a (non-tight) lower bound on the price of stability of approximate equilibria.

Proof of Theorem 3.30. Without loss of generality, we may assume that $a_e = 1$ and $b_e = 0$ for all resources $e \in E$. Using this, we obtain that the cost of player i with respect to strategy profile s is

$$C_i^\rho(s) = \sum_{e \in s_i} (1 + \rho(x_e - 1)) = \sum_{e \in s_i} (x_e + (\rho - 1)(x_e - 1)).$$

By adapting Rosenthal's potential function (3.3), we obtain that

$$\Phi^\rho(s) := \sum_{e \in E} \frac{x_e(x_e+1)}{2} + (\rho - 1) \sum_{e \in E} \frac{(x_e - 1)x_e}{2}$$

is an exact potential for $C_i^\rho(s)$. The idea of the proof is to combine the Nash inequalities and the fact that the global minimum of $\Phi^\rho(\cdot)$ is a Nash equilibrium.

Let s denote the global minimum of Φ^ρ and s^* a socially optimal solution. Further, let x and x^* be the load profiles for s and s^*, respectively. Similar to the proof of Lemma 3.24, by exploiting that s is a Nash equilibrium we obtain

$$\sum_{e \in E} x_e(1 + \rho(x_e - 1)) = \sum_{i \in N} C_i^\rho(s) \leq \sum_{i \in N} C_i^\rho(s_i^*, s_{-i}) \leq \sum_{e \in E} (1 + \rho x_e)x_e^*.$$

The fact that s is a global optimum of $\Phi^\rho(\cdot)$ yields $\Phi^\rho(s) \leq \Phi^\rho(s^*)$, which reduces to

$$\sum_{e \in E} \rho x_e^2 + (2 - \rho)x_e \leq \sum_{e \in E} \rho(x_e^*)^2 + (2 - \rho)x_e^*.$$

If we can find $\gamma, \delta \geq 0$ and some $K \geq 1$, for which

$$\begin{aligned} \big(0 \leq \big) \quad & \gamma \left[\rho(x_e^*)^2 + (2 - \rho)x_e^* - \rho x_e^2 - (2 - \rho)x_e\right] \\ & + \ \delta\left[(1 + \rho x_e)x_e^* - x_e(1 + \rho(x_e - 1))\right] \\ & \leq K \cdot x_e^*[1 + \sigma(x_e^* - 1)] - x_e[1 + \sigma(x_e - 1)], \end{aligned} \tag{3.35}$$

then this implies that $C^\sigma(s)/C^\sigma(s^*) \leq K$. We take $\delta = (K-1)/\rho$ and $\gamma = ((\rho - 1)K + 1)/(2\rho)$. It is not hard to see that $\delta \geq 0$ always holds. However, for γ we have to be more careful. We will later verify for which combinations of ρ and σ the parameter γ is indeed non-negative. Rewriting the expression in (3.35) yields that we have to find K satisfying $K \geq f_2(x_e, x_e^*, \sigma)/f_1(x_e, x_e^*, \rho, \sigma)$, where

$$\begin{aligned} f_2(x_e, x_e^*, \sigma) &:= (x_e^*)^2 - 2x_e x_e^* + (1 + 2\sigma)x_e^2 - x_e^* + (1 - 2\sigma)x_e \\ f_1(x_e, x_e^*, \rho, \sigma) &:= (1 - \rho + 2\sigma)(x_e^*)^2 - 2x_e x_e^* + (1 + \rho)x_e^2 \end{aligned}$$

$$+ (\rho - 1 - 2\sigma)x_e^* - (\rho - 1)x_e.$$

Note that this reasoning is correct only if $f_1(x_e, x_e^*, \rho, \sigma) \geq 0$. This is true because

$$f_1(x_e, x_e^*, \rho, \sigma) = \left(x_e - x_e^* + \frac{1}{2}\right)^2 - \frac{1}{4} + (2\sigma - \rho)x_e^*(x_e^* - 1) + \rho x_e(x_e - 1)$$

is non-negative for all $x_e, x_e^* \in \mathbb{N}$, $\sigma \geq 0$ and $0 \leq \rho \leq 2\sigma$. Furthermore, the expression is zero if and only if $(x_e, x_e^*) \in \{(0,1),(1,1)\}$. But for these pairs the nominator is also zero, and hence the expression in (3.35) is satisfied for these pairs. We can write

$$f_2(x_e, x_e^*, \sigma) = \left(x_e - x_e^* + \frac{1}{2}\right)^2 - \frac{1}{4} + 2\sigma x_e(x_e - 1)$$

and therefore $f_2/f_1 = \frac{A}{A+(2\sigma-\rho)B}$, where

$$A = \left(x_e - x_e^* + \frac{1}{2}\right)^2 - \frac{1}{4} + 2\sigma x_e(x_e - 1) \quad \text{and} \quad B = x_e^*(x_e^* - 1) - x_e(x_e - 1).$$

Note that if $\rho = 2\sigma$, we have $f_2/f_1 = 1$, and hence we can take $K = 1$. Otherwise,

$$\frac{A}{A + (2\sigma - \rho)B} \leq \frac{\sqrt{\sigma(\sigma+2)} + \sigma}{\sqrt{\sigma(\sigma+2)} + \rho - \sigma} =: K \quad \Leftrightarrow \quad A + (\sqrt{\sigma(\sigma+2)} + \sigma)B \geq 0.$$

The inequality on the right is true by Lemma 3.31.

To finish the proof, we determine the pairs (ρ, σ) for which the parameter γ is non-negative. This holds if and only if

$$(\rho - 1)K + 1 = (\rho - 1)\frac{\sqrt{\sigma(\sigma+2)} + \sigma}{\sqrt{\sigma(\sigma+2)} + \rho - \sigma} + 1 \geq 0.$$

Rewriting this inequality yields the bound on ρ in the statement of the theorem.

□

Note that Theorem 3.30 does not provide the bound of 2 for uniform affine congestion games stated in Table 3.1. The reason is that the bound in Theorem 3.30 with $\rho = \sigma$ is only valid for $\sigma \geq \frac{1}{4}$ (because otherwise the lower bound on ρ is not satisfied). However, for $0 \leq \sigma \leq \frac{1}{4}$ the corresponding cost functions $c_e(x) = \sigma x + (1 - \sigma)$ have non-negative constants and thus the price of stability for classical congestion games applies (see [22]). As a consequence, we obtain

$$\text{PoS}(\mathcal{A}') = \max\left\{1.577, \sup_{\sigma \geq 1/4}\left\{1 + \sqrt{\sigma/(\sigma+2)}\right\}\right\} = 2.$$

We next provide a lower bound on the price of stability for arbitrary non-negative pairs (ρ, σ). The proof is similar to a construction of Christodoulou et al. [35]

used to give a lower bound on the price of stability for ρ-approximate equilibria. The key difference is to tune some parameters in the proof with respect to the Nash definition based on the cost function $C_i^\rho(\cdot)$ rather than the ρ-approximate Nash equilibrium definition.

Theorem 3.32. *For $\rho, \sigma > 0$ fixed with $\rho < 2\sigma$, there exists a linear congestion game such that for every $\epsilon > 0$*

$$PoS(\rho, \sigma) \geq \frac{\sqrt{\sigma(\sigma+2)} + \sigma}{\sqrt{\sigma(\sigma+2)} + \rho - \sigma} - \epsilon.$$

Proof. We describe the construction of Theorem 9 [35] (using similar notation and terminology). We have a game of $n = n_1 + n_2$ players divided into two sets G_1 and G_2 with size resp. n_1 and n_2. Each player $i \in G_1$ has two strategies: A_i and P_i. The players in G_2 have a unique strategy D. The strategy profile $A = (A_1, \dots, A_{n_1}, D, \dots, D)$ will be the unique Nash equilibrium, and the strategy profile $P = (P_1, \dots, P_{n_1}, D, \dots, D)$ will be the social optimum.

We have three types of resources:

- n_1 resources α_i, $i = 1, \dots n_1$, with cost function $c_{\alpha_i}(x) = \alpha x$. The resource α_i only belongs to strategy P_i.
- $n_1(n_1 - 1)$ resources[18] β_{ij}, $i, j = 1, \dots, n_1$ with $i \neq j$, with cost function $c_{\beta_{ij}}(x) = \beta x$. The resource β_{ij} belongs only to strategies A_i and P_j.
- One resource γ with cost function $c_\gamma(x) = x$, that belongs to A_i for $i = 1, \dots, n_1$ and to D.

The idea is to set the parameters α and β in such a way that A becomes the unique Nash equilibrium. For any strategy profile s, there are k players playing strategy A_i and $n_1 - k$ players playing strategy P_i in the set G_1, for some $0 \leq k \leq n_1$. By symmetry, it then suffices to look at profiles $S_k = (A_1, \dots, A_k, P_{k+1}, \dots, P_{n_1}, D, \dots, D)$ for $0 \leq k \leq n_1$. Furthermore, the first k players playing A_i all have the same cost, and also, the $n_1 - k$ players playing P_i have the same cost. We can therefore focus on the costs of player 1, denoted by $C_A^\rho(k)$, and that of player n_1, denoted by $C_P^\rho(k)$. We have

$$\begin{aligned} C_A^\rho(k) &= \beta(k-1) + \beta(1 + \rho(2-1))(n_1 - k) + 1 + \rho(n_2 + k - 1) \\ &= (\beta - \beta(1+\rho) + \rho)k + (-\beta + \beta(1+\rho)n_1 + 1 + \rho(n_2 - 1)) \\ &= \rho(1-\beta) \cdot k + (1 - \beta - \rho) + \beta(1+\rho)n_1 + \rho n_2 \end{aligned}$$

and

$$C_P^\rho(k) = \alpha + \beta(n_1 - 1 - k) + \beta(1 + \rho(2-1))k$$

[18]The proof of Theorem 9 [35] contains a typo here: it says there are $n(n-1)$ resources of this type, instead of $n_1(n_1 - 1)$.

$$= \beta\rho \cdot k + \alpha + \beta(n_1 - 1). \tag{3.36}$$

We can set the parameters α and β such that $C_A^\rho(k) = C_P^\rho(k-1)$, meaning that S_k is a Nash equilibrium for every k (we will create a unique Nash equilibrium in a moment), that is we take

$$\rho(1-\beta) = \beta\rho \quad \text{and} \quad (1-\beta-\rho) + \beta(1+\rho)n_1 + \rho n_2 = \alpha + \beta(n_1 - 1) - \beta\rho.$$

Note that the $-\beta\rho$ term on the far right of the second equation comes from the fact that we evaluate $C_P^\rho(\cdot)$ in $k-1$ (remember that k denotes the number of players playing strategy A_i, so if a player would switch to P_i this number decreases by 1). Solving the left equation leads to $\beta = 1/2$. Inserting this in the right equation, and solving for α, gives

$$\alpha = \rho\left(\frac{n_1}{2} + n_2 - \frac{1}{2}\right) + 1.$$

We emphasize that $\alpha, \beta > 0$ for all $\rho \geq 0$. In order to make A the unique Nash equilibrium, we can slightly increase α such that we get $C_A^\rho(k) < C_P^\rho(k-1)$ for all k (which means that A_i is a dominant strategy for player i). Note that this increase in α can be arbitrary small. We have

$$\frac{C^\sigma(A)}{C^\sigma(P)} = \frac{n_1\left[1 + \sigma(n_1+n_2-1) + \frac{1}{2}(n_1-1)\right] + n_2\left[1+\sigma(n_1+n_2-1)\right]}{n_1\left[\rho(\frac{n_1+1}{2} + n_2 - 1) + 1 + \frac{1}{2}(n_1-1)\right] + n_2\left[1 + \sigma(n_2-1)\right]}.$$

Inserting $n_2 = a \cdot n_1$ for some rational $a > 0$, and sending $n_1 \to \infty$ gives a lower bound of

$$f(a) = \frac{2\sigma(1+a)^2 + 1}{\rho(1+2a) + 1 + 2\sigma a^2}$$

on the price of stability. Optimizing over $a > 0$ (this only works if $\rho < 2\sigma$) gives

$$a^* = -\frac{1}{2} + \sqrt{\frac{1}{4} + \frac{1}{2\sigma}}$$

and $f(a^*)$ then yields the bound in the statement of the theorem. □

3.4.3 Applications

We review various models that fall within our model of perception parameterized congestion games introduced above (for certain values of ρ and σ). A comparison with existing work is given in Table 3.1 in Section 3.1.2.

Altruism [23, 28]. We can rewrite the cost of player i as

$$C_i^\rho(s) = \sum_{e \in s_i} (a_e x_e + b_e) + (\rho - 1)a_e(x_e - 1).$$

The term $(\rho - 1)a_e(x_e - 1)$ can be interpreted as a "dynamic" (meaning load-dependent) tax that all players using resource e have to pay. For $1 \leq \rho \leq \infty$ and $\sigma = 1$, this model is equivalent to the altruistic player setting proposed by Caragiannis et al. [23]. Chen et al. [28] also study this model of altruism for $1 \leq \rho \leq 2$ and $\sigma = 1$.[19]

Universal taxes [22]. We can rewrite the cost of player i as

$$C_i^\rho(s) = \sum_{e \in s_i} \rho a_e x_e + (1-\rho)a_e + b_e.$$

Dividing by ρ gives that s is a Nash equilibrium with respect to C_i^ρ if and only if s is a Nash equilibrium with respect to

$$T_i^\rho(s) = \frac{C_i^\rho}{\rho} = \sum_{e \in s_i} \left(a_e x_e + \frac{b_e}{\rho} \right) + \sum_{e \in s_i} \frac{1-\rho}{\rho} a_e.$$

That is, s is a Nash equilibrium in a classical congestion game in which players take into account constant resource taxes of the form $\tau(\rho) \cdot a_e$, where $\tau(\rho) = (1-\rho)/\rho$. Caragiannis, Kaklamanis and Kanellopoulos [22] study this type of taxes, which they call *universal tax functions*, for ρ satisfying $\tau(\rho) = \frac{3}{2}\sqrt{3} - 2$. They consider these taxes to be *refundable*, i.e., they are not taken into account in the social cost, which is equivalent to the case $\sigma = 1$. Note that the function $\tau : (0, 1] \to [0, \infty)$ with $\tau(\rho) = (1-\rho)/\rho$ is bijective. That is, there is a one-to-one correspondence between universal taxes with $\tau(\rho) \in [0, \infty)$ and spiteful behavior with $\rho \in (0, 1]$; this relation is also mentioned by Caragiannis et al. [23].

Risk sensitivity under uncertainty [146]. Piliouras, Nikolova and Shamma [146] consider congestion games in which there is a (non-deterministic) order of the players on every resource. A player is only affected by players in front of her. That is, the load on resource e for player i in a strict ordering r, where $r_e(i)$ denotes the position of player i, is given by $x_e(i) = |\{j \in N : r_e(j) \leq r_e(i)\}|$. The cost of player i is then $C_i(s) = \sum_{e \in s_i} c_e(x_e(i))$. Note that $x_e(i)$ is a random variable if the ordering is non-deterministic. The social cost of the model is defined by the sum of all player costs

$$C^{\frac{1}{2}}(s) = \sum_{e \in E} \frac{1}{2} a_e x_e (x_e + 1) + b_e$$

which is independent of the ordering r.[20] Note that the social cost corresponds to the case $\sigma = \frac{1}{2}$ in our framework. Piliouras et al. [146] study various risk attitudes

[19] The equivalence between the altruism model in [23] and our model is immediate; the equivalence between the altruism model in [28] and the model in [23] (and thus also our model) is proven in [28].

[20] In every ordering there is always one player first, one player second, and so on.

towards the ordering r that is assumed to have a uniform distribution over all possible orderings. In particular, they consider players who are *risk-neutral* and players who apply *Wald's minimax principle*. In the risk-neutral setting the cost of a player is defined as the expected cost under the ordering r, which corresponds to the case $\rho = \frac{1}{2}$ in (3.16). Intuitively, this can be interpreted as that players expect to be scheduled in the middle on average. In contrast, when players apply Wald's minimax principle they adopt a worst-case perspective, i.e., each player assumes that she is scheduled last on all the resources; this corresponds to the case $\rho = 1$.

Approximate Nash equilibria [35]. Suppose that s is a Nash equilibrium under the cost functions defined in (3.16). Then, in particular, we have

$$C_i^1(s) \leq C_i^\rho(s) \leq C_i^\rho(s_i', s_{-i}) \leq \rho C_i^1(s_i', s_{-i})$$

for any player i and $s_i' \in \mathcal{S}_i$ and $\rho \geq 1$. That is, we have $C_i^1(s) \leq \rho C_i^1(s_i', s_{-i})$ which means that the profile s is a ρ-*approximate equilibrium*, as studied by Christodoulou, Koutsoupias and Spirakis [35]. In particular, this implies that any upper bound on the price of anarchy, or price of stability, in our framework yields an upper bound on the price of stability for ρ-approximate equilibria for the same class of games.[21]

Uniform affine congestion games. Let $\mathcal{A}'$ denote the class of all congestion games Γ for which all resources have uniform costs $c(x) = ax + b$, where $a = a(\Gamma)$ and $b = b(\Gamma)$ satisfy $a \geq 0$ and $a+b > 0$. Note that we allow b to be negative here. The class of affine congestion games with non-negative coefficients as defined above is contained in $\mathcal{A}'$ since every such game can always be transformed[22] into a game Γ' with $a_e = 1$ and $b_e = 0$ for all resources $e \in E'$, where E' is the resource set of Γ'. Without loss of generality, we can assume that $a+b = 1$ (since the cost functions can be scaled by $1/(a+b)$). The cost functions of $\Gamma \in \mathcal{A}'$ can then equivalently be written as $c(x) = \rho x + (1-\rho)$ for $\rho \geq 0$. This is precisely the definition of $C_i^\rho(s)$ (with $a_e = 1$ and $b_e = 0$). In particular, if we take $\sigma = \rho$ (and thus $C^\rho(s) = \sum_{i \in N} C_i^\rho(s)$), we have

$$\mathrm{PoA}(\mathcal{A}') = \sup_{\rho \geq 0} \mathrm{PoA}(\mathcal{A}, \rho, \rho) \qquad \text{and} \qquad \mathrm{PoS}(\mathcal{A}') = \sup_{\rho \geq 0} \mathrm{PoS}(\mathcal{A}, \rho, \rho),$$

where $\mathcal{A}$ denotes the class of affine congestion games with non-negative coefficients.

[21] In particular, we essentially show in Section 3.4.2 that the analysis carried out in [35], for the price of stability of approximate equilibria, actually gives a tight bound on the price of stability of the altruism model in [28].

[22] This transformation can be done in such a way that both the PoA and the PoS of the game do not change; see, e.g., [28, Lemma 4.3] for a proof.

3.5 Conclusion

We identified two structural properties of polytopal congestion games which are sufficient to efficiently compute a global minimizer of Rosenthal's potential: IDP and box-TDI. Further, we proved that the computed Nash equilibria obtain a social cost approximation guarantee of $\rho(\mathcal{D})$ if the cost functions belong to class $\mathcal{D}$. As we showed, this also establishes a tight bound on the price of stability for polytopal congestion games satisfying IDP and box-TDI. Intuitively, these games thus inherit the social cost approximation guarantee of non-atomic network routing games [45]. In our inefficiency proofs, we crucially exploited the symmetric difference decomposition property of polytopes; we believe that this new notion might be useful also in other contexts. Finally, we provided several examples of classes of congestion games that can be cast into our framework and showed that some of the results also extend to bottleneck congestion games.

For future work it would be interesting to see whether out techniques extend to other classes of games, e.g., to special cases of weighted congestion games. Note that, although having an *exact* potential function turned out to be convenient, our approach per se is not limited by this requirement. In fact, it would be interesting to see how different (ordinal) potential functions impact the inefficiency guarantee of the respective global potential function minimizers. Another interesting direction for future work is whether or not the condition of box-TDI can be dropped. More generally, is it true that every aggregation polytope with the integer decomposition property is also box-TDI? Furthermore, can one show that the integer decomposition property is in some sense necessary for the price of stability results to go through? For the computation of an arbitrary pure Nash equilibrium, this is not true, as one can efficiently compute a pure Nash equilibrium in so-called *max-cut games* with unit weights. It is interesting to note that computing a global minimum of Rosenthal's potential is an NP-complete problem for this case. Hence, it is not true that whenever we can compute some pure Nash equilibrium, we can also always compute a Rosenthal potential minimizer. Is the integer decomposition property in some sense necessary for the computation of a Rosenthal minimizer?

We feel that in general the power of polyhedral techniques to compute good Nash equilibria in games is not well-understood and worth being investigated more intensively. In particular, research in this direction opens up an intriguing connection between the fields of polyhedral combinatorics and computational game theory.

Secondly, we introduced a new model of affine congestion games by parameterizing both the cost functions of the players and the social cost function. Our model encompasses several extensions of Rosenthal's (classical) congestion games which were previously studied in the literature. We derived bounds on the price of anarchy and the price of stability which are tight for a large range of parameters ρ and σ. Our work reveals that tight bounds on the inefficiency of these extensions can be derived in a unifying manner. The study of such parameterized games seems particularly valuable if tight bounds can be derived.

A first natural extension of our model is to go beyond affine cost functions. Some of the connections between perception-parameterized congestion games and other models revealed in this chapter, continue to hold for more general cost functions (although not always as clean as for the affine case). Another natural direction for future research is to consider parameterized versions of other fundamental games such as cost sharing games, utility games, network design games or auctions.

For non-atomic network routing games [175] several extensions which were recently studied in the literature can also be unified; in particular, there are close connections between the extensions considered in [17, 29, 35, 81, 129, 128]. Similar to the viewpoint adopted here, these extensions can be viewed as network routing games where the cost functions of the players are suitably parameterized. In fact, many of these models incorporate (implicitly or explicitly) some *scaled marginal tolls* into the cost functions of the players. Further, this also connects to the notion of approximate Nash equilibria (as in [35]). Several of these works find similar inefficiency bounds which can also be derived in a unifying manner by using these scaled marginal tolls.

In this chapter, we focused on the homogeneous player case because this is the setting addressed in most previous studies which we unify here. An interesting direction for future research is to consider heterogeneous players. In this context, Chen et al. [28] derived a price of anarchy bound for their altruistic congestion games which depends on the extreme values of the parameter ρ used by players (with $\sigma = 1$).

Chapter 4

New results for the switch Markov chain

4.1 Introduction

A classical result due to Erdős and Gallai [86] characterizes when a sequence of non-negative integers $d_1 \geq \cdots \geq d_n$ can be realized as the degree sequence of a simple undirected (labelled) graph on n vertices, i.e., if there exists a *graphical realization* of the sequence. The Erdős-Gallai theorem states that the sequence can be realized if and only if: the sum $d_1 + \cdots + d_n$ is even and

$$\sum_{i=1}^{k} d_i \leq k(k-1) + \sum_{i=k+1}^{n} \min\{d_i, k\} \tag{4.1}$$

for every $1 \leq k \leq n$.

Example 4.1. Consider the sequence $d = (2, 2, 2, 2)$. It is not hard to see that this sequence satisfies the conditions in (4.1). There are precisely three (labelled) graphical realizations corresponding to this degree sequence, as illustrated in Figure 4.1.

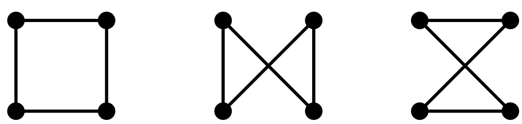

Figure 4.1: All graphical realizations for the sequence $d = (2, 2, 2, 2)$.

Havel [102] and, independently, Hakimi [98], provide a simple algorithm to compute a graphical realization if it exists. Their algorithm relies on the observation that the sequence $d = (d_1, \ldots, d_n)$, with $d_1 \geq \cdots \geq d_n$, is realizable if and only if the sequence $d_{-1} = (d_2 - 1, \ldots, d_{d_1+1} - 1, \ldots, d_n)$ is non-negative and realizable. The natural greedy approach, in which we connect d_1 to $d_2, \ldots, d_{d_1+1}$ and then repeat this procedure starting with the sequence d_{-1} (reordered from largest to smallest if necessary), outputs a graphical realization after at most $n-1$ steps of this procedure if one exists.

Similar results are known for bipartite graphs. The following characterization is due to Gale [85] and Ryser [157]. Given two sequences of non-negative integers $r = (r_1, \ldots, r_m)$ and $c = (c_1, \ldots, c_n)$ with $r_1 \geq r_2 \geq \cdots \geq r_m$, there is a bipartite graphical realization[1] if and only if $\sum_{i=1}^{n} c_i = \sum_{i=1}^{m} r_i$ and

$$\sum_{i=1}^{k} c_i \leq \sum_{i=1}^{m} \min\{r_i, k\} \tag{4.2}$$

for $1 \leq k \leq n$.

With the existence and construction problem being well-understood, there has also been a great interest in the last decades in uniformly sampling graphs with a given degree sequence, as it finds many applications in, e.g., hypothesis testing in network structures [141]. The first explicit sampler appears as the *configuration model* in the work of Bollobás [16]. Roughly speaking, every node i is given d_i *half-edges (or stubs)* which are paired up uniformly at random, with every pair of stubs being paired up becoming an edge between the nodes corresponding to the stubs. In general, this procedure outputs a *loopy multigraph* (possibly containing parallel edges and self-loops) with degree sequence d. The probability that the output of the configuration model is a simple graph is exponentially small in general, and has therefore no direct algorithmic value. Nevertheless, various approaches (see Section 4.1.2) for sampling a graph with given degrees rely on probabilistic procedures to turn the (possibly) non-simple output of the configuration model into a simple graph.

Another prominent line of work for sampling graphs with given degrees is the *Markov Chain Monte Carlo (MCMC)* method; see Section 1.4.2. Here one studies a random walk on the set of all graphical realizations induced by a probabilistic algorithmic procedure that specifies how to make (small) random changes to the current graphical realization. The probabilities, arising from the algorithmic procedure, with which graphical realizations are turned into each other define a Markov chain on the set of all graphical realizations. The idea, roughly, is that after a sufficient number of changes, the so-called *mixing time of the Markov chain*, the resulting graphical realization corresponds to a sample from an almost uniform distribution over all graphical realizations of the given degree sequence. The goal is to show that the chain mixes *rapidly*, meaning that one only needs to simulate the Markov chain a polynomial (in the size of the graph) number of steps in order to obtain an approximately uniform sample.

[1] That is, there exists a bipartite graph $G = (A \cup B, E)$ where $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$ so that $d_{a_i} = c_i$ for $i \in A$ and $d_{b_j} = r_j$ for $j \in B$.

One of the most well-known probabilistic procedures for making these small changes uses local operations called *switches* (also known as *swaps* or *transpositions*); see Figure 4.2 for an example. The notion of a switch naturally gives

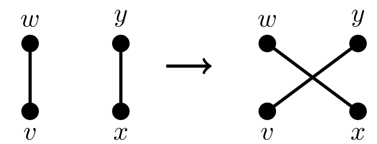

Figure 4.2: Example of a switch in which edges $\{v, w\}, \{x, y\}$ are replaced by $\{v, y\}, \{x, w\}$. Note that the degree sequence is preserved when applying a switch operation.

rise to the *switch algorithm*: start with some initial graphical realization G_0 with degree sequence d, that we can compute in polynomial time using the Havel-Hakimi algorithm explained earlier, and repeatedly apply random switches. This can, e.g., be done by selecting a tuple of four nodes (x, y, v, w) uniformly at random. If, as in Figure 4.2, the edges $\{x, y\}$ and $\{v, w\}$ are present in G_0, and $\{x, w\}$ and $\{v, y\}$ are not, we switch the edges $\{x, y\}$ and $\{v, w\}$. Does this algorithm have all the desired properties? That is, do we have the guarantee that if one applies sufficiently many switches that the resulting graph is close to a uniform sample from the set of all graphical realizations?

In order to establish correctness of this approach we consider the *switch Markov chain* on the set of all graphical realizations induced by this algorithmic procedure, where the transition probabilities of the Markov chain are given by the probabilities with which the switches are applied. We should first check that this is an aperiodic, irreducible Markov chain with the uniform distribution as stationary distribution, i.e., that it actually does the job. Aperiodicity is easy to check, as well as the fact that the chain is reversible with respect to the uniform distribution. Irreducibility is less trivial, but still well-understood. In particular, Taylor [167] has shown that every two graphical realizations of a degree sequence d can be transformed into one another by a finite sequence of switches, which implies that the switch chain is irreducible. It then follows that the switch algorithm is a *fully polynomial almost uniform sampler* (see Section 1.4) if the switch Markov chain is rapidly mixing. The switch Markov chain has been shown to be rapidly mixing for various degree sequences [41, 95, 96, 131, 73, 71], but it is still open whether it is rapidly mixing for all degree sequences.

For the problem of sampling bipartite graphs with a given degree sequence, we also study a second algorithm that is closely related to the switch algorithm, the so-called *curveball algorithm* [174, 166]. The curveball algorithm is a variation on the switch algorithm in which essentially multiple switches are performed simultaneously, with the goal of speeding up switch-based algorithms. These type of operations are called *binomial trades*; see Example 4.14 in Section 4.2.4. Roughly speaking, the neighborhoods of two given vertices are completely randomized in this procedure (while preserving the degree sequence), as opposed to just applying one switch operation.

Joint degree matrix problem. Apart from the problem of sampling graphs with given degrees, we also study the *joint degree matrix (JDM)* problem, where, in addition to the degree sequence d, we are also given a so-called (symmetric) *joint degree distribution* $c = (c_{ij})_{i,j\in[d_{\max}]}$ with c_{ij} specifying the total number of edges between nodes of degree i and j. Motivation for including such extra information is, e.g., given by Mahadevan et al. [124], who argued that the joint degree distribution is a much more reliable metric for a synthetic graph[2] to resemble a real network topology, compared to just using the degree sequence. The joint degree matrix model of Amanatidis, Green, and Mihail [5] formalizes this approach.

Example 4.2. We let $n = 11$, and consider the pair (c, d) given by

$$c = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 4 \\ 0 & 0 & 4 & 8 \end{pmatrix} \quad \text{and} \quad d = (3,3,3,3,3,3,4,4,4,4,4).$$

This means that there are six nodes of degree three, five nodes of degree four, and there are in total four edges between the nodes of degree three and four. In Figure 4.3 below we give a possible graphical realization of this tuple (c, d).

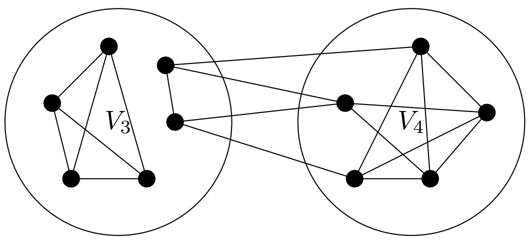

Figure 4.3: An example of a graphical realization for c and d as given in Example 4.2.

Although there are polynomial-time algorithms that produce a graphical realization of a given joint degree distribution [5, 163, 49, 92], it is not known how to uniformly sample such a realization efficiently. In particular, bounding the mixing time of the natural restriction of the switch Markov chain (in which only switches that preserve the joint degree distribution are allowed) for this setting has been an open problem since the introduction of the model [5, 163, 73].

In Section 4.2 we work with a slightly different definition of the joint degree matrix model where we leave out the zeros from the matrix c. In particular,

[2]That is, a graph not obtained from empirical data, but generated by a random graph model.

in this work we will focus on sampling graphical realizations for the case of *two degree classes*. That is, we are given a partition $V = V_1 \cup V_2$ of the node set $V = \{1, \ldots, n\}$ and we are interested in sampling simple graphs for which all nodes in V_1 have degree β_1, all nodes in V_2 have degree β_2, and where there are precisely γ edges between nodes in V_1 and V_2. This essentially corresponds to the case in which the joint degree matrix c is non-zero only for the submatrix induced by the rows and columns corresponding to β_1 and β_2.

Remark 4.3. We note that for the joint degree matrix model, it is actually not necessary to specify the degree sequence d as it is uniquely defined (up to relabelling of the nodes) by the joint degree distribution c. However, our analysis in Section 4.4 will actually be carried out in a more general model, called the *partition adjacency matrix* model [48], for which this is not the case.

4.1.1 Our contributions

We present three new results related to the switch Markov chain.

1. In Section 4.3, we present a new proof idea for showing rapid mixing of the switch Markov chain that unifies and extends all ranges of degrees for which the switch chain is known to be rapidly mixing. In particular, we show that the switch chain is rapidly mixing for so-called *strongly stable* degree sequences (Theorem 4.16). We introduce strong stability as a stricter version of the notion of P-stability [109], which roughly means that the number of graphical realizations that a degree sequence has, does not vary too much if the degree sequence is slightly perturbed. The strong stability condition is satisfied by the degree sequences in the works [113, 41, 95, 131, 73, 71] and by characterizations of P-stability [107]. In particular, our results resolve an open question posed by Greenhill [95] (see Corollary 4.17). We should note that the unification of the existing results mentioned so far is qualitative rather than quantitative, in the sense that our simpler, indirect approach provides weaker polynomial bounds for the mixing time. For examples of explicit mixing time bounds we refer the reader to [41, 42, 96].

   The proofs of the results in [131, 95, 73, 72] for the analysis of the switch Markov chain in undirected (or bipartite) graphs are all using conceptually similar ideas to the ones introduced by Cooper, Dyer and Greenhill [41] for the analysis of the switch chain for regular undirected graphs, and are based on the multi-commodity flow method of Sinclair [162]. Sinclair's method roughly speaking states that if one can define a good multi-commodity flow (of which the demands depend on the stationary distribution) in the state space graph of the Markov chain in which no edge gets too congested, then the Markov chain mixes rapidly. The individual parts of this method for the known switch chain analyses can become quite technical and require somewhat long proofs.

In this work we take a different approach for proving that the switch chain is rapidly mixing. First we analyze an easier auxiliary Markov chain introduced by Jerrum and Sinclair [109]; such a chain can be used to sample graphical realizations that *almost* have a given fixed degree sequence. We show that there exists an efficient multi-commodity flow for the auxiliary chain when the given instance is *strongly stable*, and then show how it can be transformed into an efficient multi-commodity flow for the switch chain. In this last step we compare two Markov chains with different state spaces, as the auxiliary chain samples from a strictly larger set of graphs than the switch chain. For this part of the proof we rely on embedding arguments similar to those by Feder, Guetz, Mihail and Saberi [76].

2. In Section 4.4, building on the ideas in Section 4.3, we study the problem of sampling undirected simple graphs with a given joint degree distribution using the switch Markov chain. We show that the switch chain restricted on the space of the graphical realizations of a given joint degree distribution with two degree classes is *always* rapidly mixing (Theorem 4.24). Despite being for the case of two classes, this is the very first rapid mixing result for the problem. We again first analyze an auxiliary chain, the so-called *hinge flip chain*. We study this chain in the more general *partition adjacency matrix model* for two classes. Establishing the rapid mixing of the hinge flip chain in this case presents significant challenges. To attack this problem, we rely on ideas introduced by Bhatnagar, Randall, Vazirani and Vigoda [15] in the context of sampling *exact matchings* (Remark 1.13). At the core of this approach lies the *mountain-climbing problem* [105, 176].

3. In Section 4.5 we address the mixing time of the curveball chain. We give some mathematically rigorous evidence for the claim that the curveball chain might speed up switch-based approaches, by providing a spectral gap comparison between the switch and curveball chain. The spectral gap is a quantity that essentially determines the mixing time of a Markov chain (see Section 4.2). In order to establish our results, we introduce a general comparison framework inspired by, and based on, the notion of a heat-bath Markov chain, using the definition by Dyer, Greenhill and Ullrich [61]. This framework essentially compares a given Markov chain with a *locally refined* version, which we will call its *heat-bath variant*. We introduce a novel decomposition of the state space graph of the switch and curveball chains, based on Johnson graphs, in order to apply this framework.

Recent improvements. Erdős et al. [67] show, in a recent preprint, that the proof templates used in [41, 95, 131, 73, 72] can be adjusted to show rapid mixing for P-stable degree sequences. This is an improvement over our result for strongly stable degree sequences. In particular, it allows the authors to claim rapid mixing of the switch Markov chain for certain power-law degree sequences. These sequences are claimed to be P-stable in [89], based on results in [88], but it is not known if these power-law degree sequences are also strongly stable (or provably not strongly stable in general).

Omitted contributions. A similar approach as sketched in the first contribution above can be used to show that the switch chain is rapidly mixing for various strongly stable bipartite degree sequences. This is shown in [6]. We choose to omit these results here as they are all proved along similar lines, and do not add much in terms of technical contributions.[3]

4.1.2 Related work

Jerrum and Sinclair [109] provide a fully polynomial almost uniform sampler (FPAUS) for generating graphical realizations of degree sequences coming from any P-*stable* family of sequences (see Section 4.2.1). Jerrum, Sinclair and Vigoda [110] give the first FPAUS for sampling bipartite graphs with any given degree sequence. This is a corollary of their breakthrough work [110] on approximating the permanent of a non-negative matrix. Bezáková, Bhatnagar and Vigoda provide a more direct and improved sampler of that in [110]. In the latter work the problem is reduced to that of sampling perfect matchings in a bipartite graph. It is open whether or not there exists an FPAUS for general undirected degree sequences. More generally, it is still open if there is an FPAUS for sampling perfect matchings in undirected graphs. Recently Štefankovič, Wilmes and Vigoda [164] showed that the approach in [110] does not go through for general undirected graphs.

One drawback of the sampler of Jerrum and Sinclair [109] is that it works with auxiliary states. Kannan, Tetali and Vempala [113] introduce the switch chain as a simpler and more direct sampler that does not have to work with auxiliary states. They addressed the mixing time of such a switch-based Markov chain for the (near)-regular bipartite case. Cooper, Dyer and Greenhill [41] then gave a rapid mixing proof for regular undirected graphs, and later Greenhill [95] extended this result to certain ranges of irregular degree sequences; see also Greenhill and Sfragara [96]. Miklós, Erdős and Soukup [131] proved rapid mixing for the *half-regular* bipartite case, and Erdős, Miklós and Toroczkai [73] for the *almost half-regular* case. Very recently, Erdős, Mezei and Miklós [71] presented a range of bipartite degree sequences unifying and generalizing the results in [131, 73].

Switch-based Markov chain Monte Carlo approaches have also been studied for other graph sampling problems. Feder et al. [76], as well as Cooper et al. [43], study the mixing time of a Markov chain using a switch-like probabilistic procedure (called a *flip*) for sampling connected graphs. For sampling perfect matchings, switch-based Markov chains have also been studied, see, e.g., the recent work of Dyer, Jerrum and Müller [62] and references therein.

The joint degree matrix model was first studied by Patrinos and Hakimi [145],

[3]Moreover, some of the results we obtain in [6] for bipartite degree sequences have been improved in the preprint [67] mentioned above.

albeit with a different formulation and name, and was reintroduced in Amanatidis et al. [5]. While it has been shown that the switch chain restricted on the space of the graphical realizations of any given joint degree distribution is irreducible [5, 49], almost no progress has been made towards bounding its mixing time. Stanton and Pinar [163] performed experiments based on the autocorrelation of each edge, suggesting that the switch chain mixes quickly. The only relevant result is that of Erdős et al. [73] showing rapid mixing for a related Markov chain over the severely restricted subset of so-called *balanced* joint degree matrix realizations; this special case, however, lacks several of the technical challenges that arise in the original problem.

The curveball chain was first described by Verhelst [174] and a slightly different version was later independently formulated by Strona et al. [166]. The term 'curveball' was introduced in [166]. The curveball chain has also been formulated for (un)directed graphs, see Carstens, Berger and Strona [25]. Our comparison analysis for the curveball chain is a special case of the classical comparison framework developed largely by Diaconis and Saloff-Coste and is based on so-called Dirichlet form comparisons of Markov chains, see, e.g., [55, 56], and also Quastel [147]. See also the expository paper by Dyer, Goldberg, Jerrum and Martin [60]. As the stationary distributions are the uniform distribution for all our Markov chains, we can use a more direct, but equivalent, framework based on positive semidefiniteness. Finally, the transition matrix of the curveball Markov chain is a special case of a heat-bath Markov chain under the definition of Dyer, Greenhill and Ullrich [61]. Our work partially builds on [61] in the sense that we compare a Markov chain, with a similar decomposition property as in the definition of a heat-bath chain, to its heat-bath variant.

Non-MCMC algorithms for sampling graphs with given degrees. There also exist graph sampling algorithms not relying on the MCMC method as mentioned in the introduction, many inspired by the configuration model [16]. Although this model in general outputs a loopy multigraph, there exist degree sequences for which the configuration model yields a simple graph with positive probability. For d-regular degree sequences, in which all nodes have degree $d \in \mathbb{N}$, the configuration model outputs a simple graph with probability roughly $e^{(d-1)^2/4}$. This is actually an *exact* uniform sample from the set of all simple d-regular graphs (not just close to a uniform sample). The probability for obtaining a simple graph is a strictly positive constant if and only if $d = O\left(\sqrt{\log(n)}\right)$.

Steger and Wormald [165] analyze a natural variant of the configuration model in which repeatedly only feasible edges are added uniformly at random. An edge is feasible if it does not create a loop or parallel edge. They show this procedure gives an output distribution over all simple d-regular graphs which is asymptotically uniform, for $d = o(n^{1/28})$. This range was extended to $d = o(n^{1/3-\epsilon})$ by Kim and Vu [114], and later by Bayati, Kim and Saberi [11] to $d = o(n^{1/2-\epsilon})$. The latter work [11] also studies irregular degree sequences.

McKay and Wormald [126] considered a different extension of the configuration model, in which first a loopy multigraph is generated, after which loops and parallel edges are carefully *switched out*. This procedure provides an exact uniform sample, in particular for regular graphs with $d = o(n^{1/3})$, and runs in expected polynomial time. Based on this algorithm, Gao and Wormald [89] provided an algorithm that gives an exact uniform sample for $d = o(\sqrt{n})$, and later obtained similar results for *power-law degree sequences* [90]. Very recently, Gao and Greenhill [87] obtained results along this line in which there is a set of forbidden edges, that cannot be used in any graphical realization.

Counting the number of graphs with given degrees. Jerrum and Sinclair [109] show the existence of a fully polynomial randomized approximation scheme (FPRAS) for counting the number of graphical realizations in case the degree sequence comes from a class of P-stable degree sequences as a corollary of the sampling result mentioned earlier. Similarly, the sampler for general bipartite degree sequences of Jerrum, Sinclair and Vigoda [110] can be turned into an FPRAS for counting the number of bipartite graphical realizations. Designing an FPRAS is in fact the main point of interest in [110]. We note that, to the best of our knowledge, it is actually not known whether or not counting the number of graphs with a given degree sequence is #P-complete.

Apart from algorithmic approaches for approximating $|\mathcal{G}(d)|$ in polynomial time, there is also a great interest in obtaining asymptotic formulas for $|\mathcal{G}(d)|$; see, e.g., the recent breakthrough work of Liebenau and Wormald [121] and references therein. In particular, it is shown in [121] that the number of d-regular[4] graphs on n nodes is approximately

$$\frac{\binom{n-1}{d}^n \binom{\binom{n}{2}}{m}}{\binom{n(n-1)}{2m}} e^{\frac{1}{4}}$$

if n is large, where $m = nd/2$ and $1 \leq d \leq n-2$. For other asymptotic results we refer the reader to, e.g., the work of Barvinok and Hartigan [9] who obtain results for so-called *tame* degree sequences.

4.1.3 Outline

In Section 4.2 we give all the necessary Markov chain preliminaries and we formally describe the auxiliary chain of Jerrum and Sinclair [109], the switch chain, the restricted switch chain, and the curveball chain. Section 4.3 presents our new proof approach for the switch Markov chain and our rapid mixing results for strongly stable degree sequences. These ideas are then extended in Section 4.4 to show rapid mixing of the switch Markov chain for joint degree matrix instances with two degree classes. In both Sections 4.3 and 4.4 we consider the sampling of general undirected graphs. In Section 4.5 we switch[5] to bipartite graphs and pro-

[4] A graph is d-regular is all nodes have degree $d \in \mathbb{N}$.
[5] No pun intended.

vide our comparison argument for the curveball chain, showing that it is rapidly mixing whenever the switch chain is rapidly mixing.

4.2 Preliminaries

We begin with the necessary preliminaries regarding Markov chains and the multicommodity flow method of Sinclair [162]. For Markov chain definitions not given here, see for example [118] or Section 1.4.2.

Let $\mathcal{M} = (\Omega, P)$ be an ergodic, time-reversible Markov chain over state space Ω with transition matrix P and stationary distribution π. We write $P^t(x, \cdot)$ for the distribution over Ω at time step t given that the initial state is $x \in \Omega$. The *total variation distance at time t with initial state x* is

$$\Delta_x(t) = d_{TV}(P^t(x,\cdot),\pi) = \max_{S\subseteq\Omega} \left|P^t(x,S) - \pi(S)\right| = \frac{1}{2}\sum_{y\in\Omega}\left|P^t(x,y) - \pi(y)\right|,$$

and the *mixing time* $\tau(\epsilon)$ is defined as

$$\tau(\epsilon) = \max_{x\in\Omega}\left\{ \min\{t : \Delta_x(t') \leq \epsilon \text{ for all } t' \geq t\}\right\}.$$

Informally, $\tau(\epsilon)$ is the number of steps until the Markov chain is ϵ-close to its stationary distribution independent of the initial state $x \in \Omega$. A Markov chain is said to be *rapidly mixing* if the mixing time can be upper bounded by a function polynomial in $\ln(|\Omega|/\epsilon)$.

It is well-known that, since the Markov chain is time-reversible, the matrix P only has real eigenvalues $1 = \lambda_0 > \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{|\Omega|-1} > -1$. We may replace the transition matrix P of the Markov chain by $(P + I)/2$, to make the chain *lazy*, and hence guarantee that all its eigenvalues are non-negative. It then follows that the second-largest eigenvalue of P is λ_1. In this work we always consider the lazy versions of the Markov chains involved unless specified otherwise.[6] It follows directly from Proposition 1 in [162] that

$$\tau(\epsilon) \leq \frac{1}{1-\lambda_1}\big(\ln(1/\pi_*) + \ln(1/\epsilon)\big),$$

where $\pi_* = \min_{x\in\Omega}\pi(x)$. For the special case where π is the uniform distribution, the above bound becomes

$$\tau(\epsilon) \leq \frac{1}{1-\lambda_1}(\ln(|\Omega|) + \ln(1/\epsilon)).$$

The quantity $(1 - \lambda_1)^{-1}$ can be upper bounded using the *multicommodity flow method* of Sinclair [162].

[6] In particular, in Section 4.5 we will be interested in non-lazy versions of the switch Markov chain.

We define the state space graph of the chain $\mathcal{M}$ as the directed graph $\mathbb{G}$ with node set Ω that contains exactly the arcs $(x, y) \in \Omega \times \Omega$ for which $P(x, y) > 0$ and $x \neq y$. Let $\mathcal{P} = \cup_{x \neq y} \mathcal{P}_{xy}$, where $\mathcal{P}_{xy}$ is the set of simple paths between x and y in the state space graph $\mathbb{G}$. A *flow* f in Ω is a function $\mathcal{P} \to [0, \infty)$ satisfying $\sum_{p \in \mathcal{P}_{xy}} f(p) = \pi(x)\pi(y)$ for all $x, y \in \Omega, x \neq y$. The flow f can be extended to a function on oriented edges of $\mathbb{G}$ by setting $f(e) = \sum_{p \in \mathcal{P}: e \in p} f(p)$, so that $f(e)$ is the total flow routed through $e \in E(\mathbb{G})$. Let $\ell(f) = \max_{p \in \mathcal{P}: f(p) > 0} |p|$ be the length of a longest flow carrying path, and let $\rho(e) = f(e)/Q(e)$ be the *load* of the edge e, where $Q(e) = \pi(x)P(x, y)$ for $e = (x, y)$. The maximum load of the flow is $\rho(f) = \max_{e \in E(\mathbb{G})} \rho(e)$. Sinclair ([162], Corollary $6'$) shows that

$$(1 - \lambda_1)^{-1} \leq \rho(f)\ell(f).$$

We use the following standard technique for bounding the maximum load of a flow in case the chain $\mathcal{M}$ has uniform stationary distribution π. Suppose θ is the smallest positive transition probability of the Markov chain between two distinct states. If b is such that $f(e) \leq b/|\Omega|$ for all $e \in E(\mathbb{G})$, then it follows that $\rho(f) \leq b/\theta$. Thus, we have

$$\tau(\epsilon) \leq \frac{\ell(f) \cdot b}{\theta} \ln(|\Omega|/\epsilon)\,.$$

Now, if $\ell(f), b$ and $1/\theta$ can be bounded by a function polynomial in $\log(|\Omega|)$, it follows that the Markov chain $\mathcal{M}$ is rapidly mixing. In this case, we say that f is an *efficient* flow. Note that in this approach the transition probabilities do not play a role as long as $1/\theta$ is polynomially bounded.

4.2.1 Graphical degree sequences and the switch chain

A sequence of non-negative integers $d = (d_1, \dots, d_n)$ is called a *graphical degree sequence* if there exists a simple, undirected, labeled graph on n nodes having degrees $d_1, \dots, d_n$; such a graph is called a *graphical realization of* d. For a given degree sequence d, $\mathcal{G}(d)$ denotes the set of all graphical realizations of d. Throughout this work we only consider sequences d with positive components, and for which $\mathcal{G}(d) \neq \emptyset$. Let $\mathcal{G}'(d) = \cup_{d'} \mathcal{G}(d')$ with d' ranging over the set

$$\left\{ d' : d'_j \leq d_j \text{ for all } j, \text{ and } \sum_{i=1}^{n} |d_i - d'_i| \leq 2 \right\}.$$

That is, we have (i) $d' = d$, or (ii) there exist distinct κ, λ such that $d'_i = d_i - 1$ if $i \in \{\kappa, \lambda\}$ and $d'_i = d_i$ otherwise, or (iii) there exists a κ so that $d'_i = d_i - 2$ if $i = \kappa$ and $d'_i = d_i$ otherwise. In the case (ii) we say that d' has two nodes with degree deficit one, and in the case (iii) we say that d' has one node with degree deficit two. A family $\mathcal{D}$ of graphical degree sequences is called *P-stable* [109] if there exists a polynomial $q(n)$ such that for all $d \in \mathcal{D}$ we have $|\mathcal{G}'(d)|/|\mathcal{G}(d)| \leq q(n)$, where n is the number of components of d.

Jerrum and Sinclair [109] define the following Markov chain on $\mathcal{G}'(d)$, which will henceforth be referred to as the *JS chain.*[7]

Let $G \in \mathcal{G}'(d)$ be the current state of the JS chain. Choose an ordered pair of vertices (i, j) uniformly at random:

- if $G \in \mathcal{G}(d)$ and (i, j) is an edge of G, delete (i, j) from G (*Type 0 transition*),
- if $G \notin \mathcal{G}(d)$ and the degree of i in G is less than d_i, and (i, j) is not an edge of G, add (i, j) to G; if this causes the degree of j to exceed d_j, select an edge (j, k) uniformly at random and delete it (*Type 1 transition*).

In case the degree of j does not exceed d_j in the second case, we call this a *Type 2 transition.*

The graphs $G, G' \in \mathcal{G}'(d)$ are *JS adjacent* if G can be obtained from G' with positive probability in one transition of the JS chain and vice versa. The properties of the JS chain, stated in Theorem 4.4 below, are easy to check [109].

Theorem 4.4. *The JS chain is irreducible, aperiodic and symmetric, and, hence, has uniform stationary distribution over $\mathcal{G}'(d)$. Moreover, $P(G, G')^{-1} \leq 2n^3$ for all JS adjacent $G, G' \in \mathcal{G}'(d)$, and also the maximum in- and out-degrees of the state space graph of the JS chain are bounded by n^3.*

We say that two graphs G, G' are *within distance r in the JS chain* if there exists a path of at most length r from G to G' in the state space graph of the JS chain. By $\text{dist}(G, d)$ we denote the minimum distance of G to an element in $\mathcal{G}(d)$. The following parameter will play a central role in this work. Let

$$k_{JS}(d) = \max_{G \in \mathcal{G}'(d)} \text{dist}(G, d)\,. \tag{4.3}$$

Based on the parameter $k_{JS}(d)$, we define the notion of *strong stability.* The simple observation in Proposition 4.6 justifies the terminology. For other settings, e.g., for sampling bipartite graphs [6] or joint degree matrix realizations (Section 4.4), the definition of k_{JS} can be adjusted accordingly.

Definition 4.5 (Strong stability)**.** A family of graphical degree sequences $\mathcal{D}$ is called *strongly stable* if there exists a constant ℓ such that $k_{JS}(d) \leq \ell$ for all $d \in \mathcal{D}$.

[7] A slightly different definition of stability is given by Jerrum, McKay and Sinclair [107]. Based on this variant, one could define the corresponding variant of the JS chain. Nevertheless, the definitions of stability in [107] and [109] (and their corresponding definitions of strong stability) are equivalent. To avoid confusion, here we only use the definitions in [109] in this section.

Proposition 4.6. *If $\mathcal{D}$ is strongly stable, then it is P-stable.*

Proof. Suppose $\mathcal{D}$ is strongly stable with respect to the constant ℓ. Let $d \in \mathcal{D}$ be a degree sequence with n components. For every $G \in \mathcal{G}'(d) \setminus \mathcal{G}(d)$ choose some $\varphi(G) \in \mathcal{G}(d)$ within distance $k = k_{JS}(d)$ of G. As the in-degree of any node in the state space graph of the JS chain is bounded by n^3, the number of paths with length at most k that end up at any particular graph in $\mathcal{G}(d)$ is upper bounded by $(n^3)^k$. Therefore, $|\mathcal{G}'(d)|/|\mathcal{G}(d)| \leq n^{3k} \leq n^{3\ell}$, meaning that $\mathcal{D}$ is P-stable, since ℓ is constant. $\square$

Finally, the lazy version of the *switch chain on* $\mathcal{G}(d)$ is defined as follows; see, e.g., [41].

Let $G \in \mathcal{G}(d)$ be the current state of the switch chain:

- With probability 1/2, do nothing.
- Otherwise, perform a *switch* operation: select two edges $\{a, b\}$ and $\{x, y\}$ uniformly at random, and select a perfect matching M on nodes $\{x, y, a, b\}$ uniformly at random (there are three possible options). If $M \cap E(G) = \emptyset$, then delete $\{a, b\}, \{x, y\}$ from $E(G)$ and add the edges of M.

The graphs $G, G' \in \mathcal{G}(d)$ are *switch adjacent* if G can be obtained from G' with positive probability in one transition of this chain and vice versa. Below we summarize some properties of the switch chain; see, e.g., [96] and references therein. The bound on the transition probabilities follows from a simple counting argument.

Theorem 4.7. *The switch chain is irreducible, aperiodic and symmetric, and, thus, has uniform stationary distribution over $\mathcal{G}(d)$. Also, we have $P(G, G')^{-1} \leq 6n^4$ for all switch adjacent $G, G' \in \mathcal{G}(d)$, and the maximum in- and out-degrees of the state space graph of the switch chain are bounded by n^4.*

4.2.2 JDM model and the restricted switch chain

Here in addition to the degrees, we would also like to specify the number of edges between nodes of degree i and nodes of degree j for every pair (i, j). Let $V = \{1, \dots, n\}$ be a set of nodes. An instance of the *joint degree matrix (JDM)* model is given by a partition $V_1 \cup V_2 \cup \dots \cup V_q$ of V into pairwise disjoint (degree) classes, a symmetric *joint degree matrix* $c = (c_{ij})_{i,j \in [q]}$ of non-negative integers, and a sequence $d = (d_1, \dots, d_q)$ of non-negative integers.[8] We say that the tuple $((V_i)_{i \in q}, c, d)$ (or just (c, d) when it is clear what the partition is) is *graphical*, if

[8] This is shorthand notation. More formally, we could write $\hat{d} = \left(d_1^1, \dots, d_1^{|V_1|}, \dots, d_q^1, \dots, d_q^{|V_q|}\right)$ corresponding to the definition of a graphical degree sequence. In such a case, $d_i^j = d_i$ for $i \in V$ and $j \in \{1, \dots, |V_i|\}$.

there exists a simple, undirected, labeled graph $G = (V, E)$ on the nodes in V such that all nodes in V_i have degree d_i and there are precisely c_{ij} edges between nodes in V_i and V_j. Such a G is called a graphical realization of the tuple. We let $\mathcal{G}((V_i)_{i\in q}, c, d)$, or just $\mathcal{G}(c, d)$, denote the set of all graphical realizations of $((V_i)_{i\in q}, c, d)$. We focus on the case of $q = 2$, i.e., when two degree classes are given.

While switches maintain the degree sequence, this is no longer true for the joint degree constraints. However, some switches do respect these constraints as well, e.g., if w, y in Figure 4.2 are in the same degree class. Thus, we are interested in the following (lazy) *restricted switch Markov chain* for sampling graphical realizations of $\mathcal{G}(c, d)$.

Let $G \in \mathcal{G}(c, d)$ be the current state of the (restricted) switch chain:

- With probability 1/2, do nothing.
- Otherwise, perform a *switch* operation: select two edges $\{a, b\}$ and $\{x, y\}$ uniformly at random, and select a perfect matching M on nodes $\{x, y, a, b\}$ uniformly at random. If $M \cap E(G) = \emptyset$ and $G + M - (\{a, b\} \cup \{x, y\}) \in \mathcal{G}(c, d)$, then delete $\{a, b\}, \{x, y\}$ from $E(G)$ and add the edges of M.

Theorem 4.8 below, that summarizes some properties of the restricted switch chain, follows from [5, 49].

Theorem 4.8. *This restricted switch chain is irreducible, aperiodic and symmetric. Like the switch chain defined above, $P(G, G')^{-1} \leq n^4$ for all adjacent $G, G' \in \mathcal{G}'(c, d)$, and also the maximum in- and out-degrees of the state space graph are less than n^4.*

4.2.3 PAM model and the hinge flip chain

We give a description of the *partition adjacency matrix (PAM)* model [48], that forms a generalization of the joint degree matrix model described in Section 4.2.2. Let $V = \{1, \ldots, n\}$ be a given set. An instance of the partition adjacency matrix model is given by a partition $V_1 \cup V_2 \cup \cdots \cup V_q$ of V into pairwise disjoint classes. Moreover, we are given a symmetric *partition adjacency matrix* $c = (c_{ij})_{i,j\in[q]}$ of non-negative integers, and a sequence $d = (d_1, \ldots, d_n)$ of non-negative integers. We say that the tuple $((V_i)_{i\in q}, c, d)$ is graphical if there exists a simple, undirected, labelled graph $G = (V, E)$ on the nodes in V with node $i \in V$ having degree d_i, and so that there are precisely c_{ij} edges between endpoints in V_i and V_j. This is denoted by $E[V_i, V_j] = c_{ij}$. The graph G is called a graphical realization of the tuple. We let $\mathcal{G}((V_i)_{i\in q}, c, d)$ denote the set of all graphical realizations of the tuple $((V_i)_{i\in q}, c, d)$. We often write $\mathcal{G}(c, d)$ instead of $\mathcal{G}((V_i)_{i\in[q]}, c, d)$ when it is clear what the partition is (similar to the notation in Section 4.2.2).

In this work we focus on the case of a partition into two classes V_1 and V_2, and, without loss of generality, assume that $1 \leq c_{12} \leq |V_1| \cdot |V_2| - 1$.[9] For the case of two classes an initial state can be computed in polynomial time [68].[10] We let $\mathcal{G}'(c,d) = \cup_{(c',d')}\mathcal{G}'(c',d')$ with (c',d') ranging over tuples satisfying

(i) $\sum_{i=1}^{n} d_i - d_i' = 0$,

(ii) $\sum_{i=1}^{n} |d_i - d_i'| \in \{0, 2, 4\}$,

(iii) $c_{12}' \in \{c_{12} - 1, c_{12}, c_{12} + 1\}$.

We call elements in $\mathcal{G}'(c,d) \setminus \mathcal{G}(c,d)$ *perturbed (auxiliary) states.* For any $G \in \mathcal{G}'(c,d)$ the *perturbation at node* $v \in V$ is defined as $\alpha_v = d_v' - d_v$ where d' is the degree sequence of G. We say that the node v has a degree surplus if $\alpha_v > 0$ and a degree deficit if $\alpha_v < 0$. Moreover, the total degree surplus is defined as $\sum_{v:\alpha_v>0} \alpha_v$, and the total degree deficit as $-\sum_{v:\alpha_v<0} \alpha_v$. Note that

$$\sum_{v:\alpha_v>0} \alpha_v - \sum_{v:\alpha_v<0} \alpha_v = \sum_{i=1}^{n} |d_i - d_i'|.$$

Finally, we say that a tuple (c',d') is *edge-balanced* if $c' = c$ (but possibly $d' \neq d$). From the conditions defining $\mathcal{G}'(c,d)$, we may infer the following properties.

Proposition 4.9. *For any $G \in \mathcal{G}'(c,d)$, for some tuple (c',d') satisfying (i)-(iii) above, it holds that*

(a) the perturbation at node v satisfies $\alpha_v \in \{-2,-1,0,1,2\}$ for any $v \in V$,

(b) $\max_{i,j=1,2} |c_{ij} - c_{ij}'| \leq 1$, and $\sum_{1 \leq i < j \leq 2} |c_{ij} - c_{ij}'| \in \{0, 2\}$.

Proof. If there is some node with degree surplus greater or equal than three, then the total degree deficit is also at least three, which follows from the first condition defining $\mathcal{G}'(c,d)$. This means that $\sum_{i=1}^{n} |d_i - d_i'| \geq 6$, which violates the second condition defining $\mathcal{G}'(c,d)$. A similar argument holds in case there is some node with degree deficit greater or equal than three. To see that the second property is true, assume without loss of generality, because of (iii), that $c_{11}' \geq c_{11} + 2$ (similar arguments hold for c_{22}'). Because of the fact that $c_{12}' \in \{c_{12} - 1, c_{12}, c_{12} + 1\}$, by the third condition defining $\mathcal{G}'(c,d)$, it must be that the total degree surplus of the nodes in V_1 is at least three. This gives a contradiction for similar reasons as before. An analogous argument holds in case $c_{11}' \leq c_{11} - 2$. Finally, the last property is a direct consequence of $\max_{i,j=1,2} |c_{ij} - c_{ij}'| \leq 1$ and the fact that $\sum_{i,j=1,2} |c_{ij} - c_{ij}'|$ is an even number, because of the first property defining $\mathcal{G}'(c,d)$. $\square$

[9] It is not hard to see that the cases $c_{12} \in \{0, |V_1| \cdot |V_2|\}$ reduce to the single class case.

[10] For general instances, it is not known if an initial state can be computed in time polynomial in n. It is conjectured to be NP-hard in general [68]; see also [50].

Remark 4.10. As we focus on the case in which V is partitioned into two classes V_1 and V_2 here, we will sometimes use shorthand notation in this section. Given a sequence d, the number $\gamma = c_{12}$ uniquely determines the matrix c, and the set $\mathcal{G}(c, d)$ is then denoted by $\mathcal{G}(V_1, V_2, \gamma, d)$. As before, we often leave out V_1 and V_2 from the tuple (for sake of readability). That is, we then write $\mathcal{G}'(\gamma, d)$ instead of $\mathcal{G}'(V_1, V_2, c, d)$.

We define the *hinge flip Markov chain* $\mathcal{M}(\gamma, d)$ on $\mathcal{G}'(\gamma, d)$ as follows.

Let $G \in \mathcal{G}'(\gamma, d)$ be the current state of the hinge flip chain:

- With probability 1/2, do nothing.
- Otherwise, perform a *hinge flip* operation: select an ordered triple i, j, k of nodes uniformly at random. If $\{i, j\} \in E(G)$, $\{j, k\} \notin E(G)$, and $G - \{i, j\} + \{j, k\} \in \mathcal{G}'(\gamma, d)$, then delete $\{i, j\}$ and add $\{j, k\}$.

Note that we can check if $G - \{i, j\} + \{j, k\} \in \mathcal{G}'(\gamma, d)$ in time polynomial in n based on the state G.

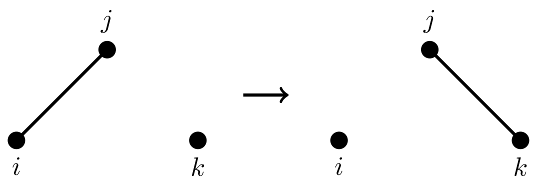

Figure 4.4: Example of a hinge flip operation for the ordered triple i, j, k.

Graphs $G, G' \in \mathcal{G}'(\gamma, d)$ are said to be *adjacent* in $\mathcal{M}$ if G can be obtained from G' with positive probability in one transition of the chain $\mathcal{M}$. We say that two graphs G, G' are within distance r in $\mathcal{M}$ if there exists a path of at most length r from G to G' in the state space graph of $\mathcal{M}$. By $\text{dist}(G, \gamma, d)$ we denote the minimum distance of G from an element in $\mathcal{G}(\gamma, d)$. The following parameter is the analogue of (4.3) for the current setting and will be used in a similar manner to define the appropriate variant of strong stability. We define

$$k(\gamma, d) = \max_{G \in \mathcal{G}'(\gamma, d)} \text{dist}(G, \gamma, d). \tag{4.4}$$

In the PAM model with two degree classes, a family $\mathcal{D}$ of graphical tuples (γ, d) is called *strongly stable*[11] if there exists a constant k such that $k(\gamma, d) \leq k$ for all $(\gamma, d) \in \mathcal{D}$.

[11] When restricted to the single class case, this notion is essentially equivalent to that defined after (4.3).

Theorem 4.11. *Let $\mathcal{D}$ be a family of graphical tuples that is strongly stable with respect to some constant k. Then for every $(\gamma, d) \in \mathcal{D}$, the chain $\mathcal{M}(\gamma, d)$ is irreducible, aperiodic and symmetric, and, hence, has uniform stationary distribution over $\mathcal{G}'(\gamma, d)$. Moreover, $P(G, G')^{-1} \leq n^3$ for all adjacent $G, G' \in \mathcal{G}'(\gamma, d)$, and also the maximum in- and out-degrees of the state space graph of the chain $\mathcal{M}(\gamma, d)$ are bounded by n^3.*[12]

Proof. The only claim that requires a detailed argument, and uses the assumption of strong stability, is that of the irreducibility of the chain. By definition of strong stability, we always know that every perturbed state is connected to some element in $\mathcal{G}(\gamma, d)$ so it suffices to show that there is a path between any two states in $\mathcal{G}(\gamma, d)$. This follows from the analysis in Section 4.4. Aperiodicity follows from the holding probability in the description of the chain $\mathcal{M}$, and symmetry is straightforward. The bound on $P(G, G')^{-1}$ follows directly from the description of the chain, as do the bounds on the in- and out-degrees of the state space graph. $\square$

Remark 4.12. In general, the space of all graphical realizations satisfying a given partition adjacency matrix constraint with two classes is not connected under switches [68]. However, it is shown in [68] that it is connected under so-called *double switches*, in which one is, roughly speaking, allowed to perform two switches simultaneously.

4.2.4 Bipartite degree sequences and the curveball chain

In this section we provide terminology related to bipartite degree sequences used in Section 4.5. We will consider bipartite graphs with given degrees $r = (r_1, \ldots, r_m)$ and $c = (c_1, \ldots, c_n)$. An equivalent way of looking at such graphs is to consider their adjacency matrices, which are binary matrices with row sums r and column sums c. The vectors r and c are referred to as the *marginals.* The latter viewpoint is adopted as it is more convenient to illustrate our ideas in Section 4.5. We actually consider a slightly more general model (than that in Section 4.2.1) in which we also have a set of *forbidden entries* that have to be zero in any binary matrix with the given marginals.[13] Formal definitions are given next.

We are given $n, m \in \mathbb{N}$, fixed row sums $r = (r_1, \ldots, r_m)$, column sums $c = (c_1, \ldots, c_n)$, and a set of forbidden entries $\mathcal{F} \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$. We define $\Omega = \Omega(r, c, \mathcal{F})$ as the set of all binary $m \times n$-matrices A, with entries in $\{0, 1\}$, satisfying these row and column sums, and for which $A(a, b) = 0$ if $(a, b) \in \mathcal{F}$. Deciding whether or not Ω is non-empty, and computing an element

[12] It might be the case that the chain is always irreducible, even if $\mathcal{D}$ is not strongly stable, but this is not relevant at this point. The assumption of strong stability allows for a shortcut in the proof of irreducibility.

[13] This would correspond to introducing a set of forbidden edges in the setting of Section 4.2.1.

from it in case it is non-empty, can be done in time polynomial in m and n.[14]

In Section 4.5 we consider a switch Markov chain with different transition probabilities than that given in Section 4.2.1. We will consider the implementation of Kannan, Tetali and Vempala [113], henceforth referred to as the *KTV switch chain*. It proceeds as follows.[15]

Let $A \in \Omega(r, c, \mathcal{F})$ be the current state of the KTV switch chain:

- Select two rows $1 \leq i < j \leq m$ and two columns $1 \leq k < \ell \leq n$ uniformly at random.
- Perform a *switch* operation: If $A(i,k) = A(j,\ell) = 1,\ A(i,\ell) = A(j,k) = 0$, and $A(i,\ell), A(j,k) \notin \mathcal{F}$, replace the entries $A(i,k),\ A(j,\ell)$ by zeros and the entries $A(i,\ell),\ A(j,k)$ by ones. Similarly, if $A(j,k) = A(i,\ell) = 1, A(j,\ell) = A(i,k) = 0$, and $A(j,\ell), A(i,k) \notin \mathcal{F}$, replace the entries $A(j,k),\ A(i,\ell)$ by zeros and the entries $A(j,\ell),\ A(i,k)$ by ones.

Matrices $A, B \in \Omega$ are *switch adjacent for row i and j* if $A = B$ or if $A - B$ contains exactly four non-zero elements that occur on rows i and j and columns k and ℓ. Two matrices are *switch adjacent* if they are switch adjacent for some rows i and j.

Remark 4.13. We always assume the (KTV) switch chain is irreducible for given marginals r and c, and forbidden entry set $\mathcal{F}$ (it is clearly always aperiodic, symmetric and finite). Irreducibility is for instance guaranteed in case there are no forbidden entries [148]; or in case $n = m \geq 4$, with $\mathcal{F}$ is the set of diagonal entries and regular marginals $c_i = r_i = d$ for some given $d \geq 1$ [93]. A characterization for irreducibility in the case where $\mathcal{F}$ is the set of diagonal entries is given in [14].

We next continue with some additional terminology in order to define the curveball chain. For $A \in \Omega$, we let A_{ij} be the $2 \times n$-submatrix formed by rows i and j, for $1 \leq i < j \leq m$. We define

$$U_{ij}(A) = \{k \in \{1, \dots, n\} : A(i,k) = 1,\ A(j,k) = 0 \text{ and } (j,k) \notin \mathcal{F}\}, \quad (4.5)$$

with $u_{ij}(A) = |U_{ij}(A)|$, and similarly

$$L_{ij}(A) = \{k \in \{1, \dots, n\} : A(i,k) = 0,\ A(j,k) = 1 \text{ and } (i,k) \notin \mathcal{F}\}, \quad (4.6)$$

with $\ell_{ij}(A) = |L_{ij}(A)|$. Note that $L_{ij} \cup U_{ij}$ are precisely the columns k for which A_{ij} has different values on its rows and for which (i,k) and (j,k) are both not

[14] Deciding non-emptiness of Ω can be reduced to deciding if a certain auxiliary graph (a variation on Tutte's construction) contains a perfect matching [170]. The latter can be done using Edmond's blossom algorithm [63]. This is also mentioned in [70].

[15] This is *not* a lazy Markov chain (staying at the current $x \in \Omega$ with probability $P(x,x) \geq \frac{1}{2}$).

forbidden. We will often write u_{ij} and ℓ_{ij} instead of $u_{ij}(A)$ and $\ell_{ij}(A)$ for brevity. With the given notation, we can proceed with the definition of the curveball chain.

Let $A \in \Omega(r, c, \mathcal{F})$ be the current state of the curveball chain:

- Select two rows $1 \leq i < j \leq m$ uniformly at random. Let S be the submatrix formed by the rows i and j and columns in $U_{ij} \cup L_{ij}$.
- Perform a *binomial trade* operation: Replace the submatrix S by a uniform randomly chosen $2 \times (u_{ij} + \ell_{ij})$-matrix S_u with marginals $r_u = (u_{ij}, \ell_{ij})$ and $c_u = (1, \ldots, 1)$.

Note that S_u can be computed by uniformly at random choosing u_{ij} column indices in $U_{ij} \cup L_{ij}$, and that the sets U_{ij} and L_{ij} can be found easily.

Two matrices A and B are called *trade adjacent for rows* i *and* j if $A = B$ or if B can be obtained from A using one binomial trade operation on rows i and j. Two matrices are *trade adjacent* if they are trade adjacent for some row pair. Roughly speaking, instead of only performing one switch on rows i and j, we completely randomize rows i and j in the curveball chain.

Example 4.14 (Binomial trade)**.** Suppose that the matrix A_{ij} (the matrix A restricted to rows i and j) is given by

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix},$$

and that $A(i, 4) \in \mathcal{F}$ is the only forbidden entry appearing on either row i or j. We have $U_{ij}(A) = \{2, 6\}$ and $L_{ij}(A) = \{5\}$. We consider the submatrix

$$S = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

formed by the second, fifth and sixth column. We now replace the submatrix S by a randomly chosen submatrix

$$S_u \in \left\{ \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right\}.$$

4.2.5 Johnson graphs

One class of graphs that are of particular interest in Section 4.5, are the so-called Johnson graphs. For given integers $1 \leq q \leq p$, the undirected Johnson graph $J(p, q)$ contains as nodes all subsets of size q of $\{1, \ldots, p\}$, and two subsets $u, v \subseteq \{1, \ldots, p\}$ are adjacent if and only if $|u \cap v| = q - 1$. We refer the reader to [104, 19] for the following facts. The Johnson graph $J(p, q)$ is a $q(p - q)$-regular graph and the eigenvalues of its adjacency matrix are given by

$$(q - i)(p - q - i) - i \qquad \text{with multiplicity} \qquad \binom{p}{i} - \binom{p}{i-1}$$

for $i = 0, \ldots, q$, with the convention that $\binom{p}{-1} = 0$. The following observation is included for ease of reference. It will often be used to lower bound the smallest eigenvalue of a Johnson graph.

Proposition 4.15. *Let $p, q \in \mathbb{N}$ be given. The continuous function $f : \mathbb{R} \to \mathbb{R}$ defined by*

$$f(x) = [(q-x)(p-q-x) - x] - q(p-q) = x(x-(p+1))$$

is minimized for $x^* = (p+1)/2$, with $f(x^*) = -(p+1)^2/4$.

4.3 Switch chain for strongly stable sequences

The result in Theorem 4.16 below is our main result regarding the mixing time of the switch chain for strongly stable degree sequences. Its proof is divided in two parts. First, in Section 4.3.1, by giving an efficient multicommodity flow, we show that for any d in a family of strongly stable degree sequences the JS chain is rapidly mixing on $\mathcal{G}'(d)$. Then, in Section 4.3.2, we show that such an efficient flow for the JS chain on $\mathcal{G}'(d)$ can be transformed into an efficient flow for the switch chain on $\mathcal{G}(d)$. This yields the following theorem.

Theorem 4.16. *Let $\mathcal{D}$ be a strongly stable family of degree sequences with respect to some constant k. Then there exists a polynomial $q(n)$ such that, for any $0 < \epsilon < 1$, the mixing time τ_{sw} of the switch chain for a graphical sequence $d = (d_1, \ldots, d_n) \in \mathcal{D}$ satisfies*

$$\tau_{\text{sw}}(\epsilon) \leq q(n)^k \ln(1/\epsilon)\,.$$

We next discuss a direct corollary of Theorem 4.16 which was posed as an open question in [95]. Its proof is esessentially the same as that of a similar result for a slightly different, but equivalent, notion of stability in [107]. It is given here for self-containment.

Corollary 4.17. *Let $\mathcal{D} = \mathcal{D}(\delta, \Delta)$ be the set of all graphical degree sequences $d = (d_1, \ldots, d_n)$ satisfying*

$$(\Delta - \delta + 1)^2 \leq 4\delta(n - \Delta - 1) \tag{4.7}$$

where δ and Δ are the minimum and maximum component of d, respectively. For any $d \in \mathcal{D}$, we have $k_{JS}(d) \leq 6$. Hence, the switch chain is rapidly mixing for sequences in $\mathcal{D}$.

Proof. We first introduce some notation, using the same terminology as in [107]. Let $G = (V, E)$ be an undirected graph. For distinct $u, v \in V$ we say that u, v are co-adjacent if $\{u, v\} \notin E$, and $\{u, v\}$ is called a co-edge. An alternating path of length q in G is a sequence of (not necessarily distinct) nodes $v_0, v_1, \ldots, v_q$ such that $\{v_i, v_{i+1}\}$ is an edge when i is even, and a co-edge if i is odd. The path is called a cycle if $v_0 = v_q$. As in the proof of Theorem 2 [107], we need the following lemma from [107].

Lemma 4.18 ([107]). *Let H be an undirected n-vertex graph with distinguished vertices s and t (not necessarily distinct), and suppose the set of vertices adjacent to s is equal to the set of vertices adjacent to t. Suppose that $\delta_{\min}$ and $\delta_{\max}$ are natural numbers such that the degrees of all vertices other than s and t lie in the range $[\delta_{\min}, \delta_{\max}]$, and such that s and t themselves have degree at least $\delta_{\min} + 1$. If $(\delta_{\max} - \delta_{\min} + 1)^2 \leq 4\delta_{\min}(n - \delta_{\max} - 1)$, then there exists an edge-disjoint alternating path in G which starts at s, ends at t, and has length $1, 3, 5$ or 7.*

Let $G \in \mathcal{G}'(d) \setminus \mathcal{G}(d)$. First consider the case where for the degree sequence d' of G there exist x, y so that

$$d'_i = \begin{cases} d_i - 1 & \text{if } i = x, y, \\ d_i & \text{otherwise.} \end{cases}$$

If $\{x, y\}$ is a co-edge, then clearly $\text{dist}(G, d) = 1$, as we can then simply add the edge $\{x, y\}$ to obtain a graph in $\mathcal{G}(d)$. Therefore, assume that $\{x, y\}$ is an edge in G. It follows that both nodes x and y have degree at most $n - 2$ in $G - \{x, y\}$, and therefore there exist two nodes a and b so that $\{x, a\}$ and $\{y, b\}$ are co-edges. If nodes a and b have the same set of neighbors in $G + \{x, a\} + \{y, b\}$, we can directly apply Lemma 4.18 to the graph $G + \{x, a\} + \{y, b\}$ to obtain an odd alternating path from a to b of length at most 7. Otherwise, without loss of generality, we may assume that there exists a node c which is a neighbor of a but not of b (again in $G + \{x, a\} + \{y, b\}$). We can then remove $\{a, c\}$ and add $\{c, b\}$ in order to get a degree surplus of two at node b, and then we can apply Lemma 4.18 with $s = t = b$ in the graph $G + \{x, a\} + \{y, b\} - \{a, c\} + \{c, b\}$ (if $c = x$ this part of the proof can be skipped by choosing $a = b$ in the beginning of the argument, as both $\{x, b\}$ and $\{y, b\}$ are then co-edges in the graph G we start with). In any case, it follows that there exists an alternating (between edges and non-edges of G) circuit of even length containing the edge $\{x, y\}$ in G of length at most 12. This implies that $\text{dist}(G, d) \leq 6$. That is, in at most six moves in the JS chain we can now reach an element in $\mathcal{G}(d)$.

Next, suppose that for the degree sequence d' of G there exists some x so that

$$d'_i = \begin{cases} d_i - 2 & \text{if } i = x, \\ d_i & \text{otherwise.} \end{cases}$$

It is clear that the degree of x is at most $n - 2$ in G. Let $\{x, a\}$ and $\{x, b\}$ be two co-edges. Applying similar steps as in the previous case to the graph $G + \{x, a\} + \{x, b\}$ it follows that G has an alternating path of length at most 9 starting in a and ending in b. It then again follows that $\text{dist}(G, d) \leq 6$. $\square$

Explicit families satisfying these conditions are given in [107]. For instance, all sequences $d = (d_1, \ldots, d_n)$ with (i) $\delta(d) \geq 1$ and $\Delta(d) \leq 2\sqrt{n} - 2$, or (ii) $\delta(d) \geq \frac{1}{4}n$ and $\Delta(d) \leq \frac{3}{4}n - 1$ satisfy (4.7). The bound in Corollary 4.17 is in a sense best possible with respect to the graph parameters involved. Namely, there exist non-stable degree sequence families the members of which only slightly violate (4.7); see the discussion in [107] for details.

4.3.1 Flow for the Jerrum-Sinclair chain

Jerrum and Sinclair [109] claim, without proof, that the JS chain is rapidly mixing for (some) families of stable degree sequences. For completeness, we prove in Theorem 4.19 that the chain is rapidly mixing for any family of strongly stable degree sequences.

Theorem 4.19 ([109]). *Let $\mathcal{D}$ be a strongly stable family of degree sequences with respect to some constant k. Then there exist polynomials $p(n)$ and $r(n)$ such that for any $d = (d_1, \ldots, d_n) \in \mathcal{D}$ there exists an efficient multicommodity flow f for the JS chain on $\mathcal{G}'(d)$ satisfying $\max_e f(e) \leq p(n)/|\mathcal{G}'(d)|$ and $\ell(f) \leq r(n)$.*

Our proof of Theorem 4.19, given below, uses conceptually similar arguments to the ones used in [41] for the analysis of the switch chain on regular undirected graphs. However, the analysis done here for the JS chain is, in our opinion, easier and cleaner than the corresponding analysis for the switch chain. In particular, the so-called *circuit processing procedure* is simpler in our setting, as it only involves altering edges in the symmetric difference of two graphical realizations in a straightforward fashion. In the switch chain analyses [41, 96, 131, 73, 71] one also has to temporarily alter edges that are not in the symmetric difference and this significantly complicates things. Moreover, for the analysis of the JS chain, we can rely on arguments used (in a somewhat different context) by Jerrum and Sinclair [108] for the analysis of a Markov chain for sampling (near) perfect matchings of a given graph. This usage of arguments in [108] was suggested by Jerrum and Sinclair [109] for showing that the JS chain is rapidly mixing for stable degree sequences.

We will use the following idea from [108]—used in a different setting—in order to restrict ourselves to establishing flow between states in $\mathcal{G}(d)$, rather than between all states in $\mathcal{G}'(d)$. Assume that d is is a degree sequence with n components that is a member of a strongly stable family of degree sequences (with respect to some k).

Lemma 4.20. *Let f' be a flow that routes $1/|\mathcal{G}'(d)|^2$ units of flow between any pair of states in $\mathcal{G}(d)$ in the JS chain, so that $f'(e) \leq b/|\mathcal{G}'(d)|$ for all e in the state space graph of the JS chain. Then f' can be extended to a flow f that routes $1/|\mathcal{G}'(d)|^2$ units of flow between any pair of states in $\mathcal{G}'(d)$ with the property that for all e in the state space graph of the JS chain*

$$f(e) \leq q(n)\frac{b}{|\mathcal{G}'(d)|},$$

where $q(\cdot)$ is a polynomial whose degree only depends on $k_{JS}(d)$. Moreover, $\ell(f) \leq \ell(f') + 2k_{JS}(d)$.[16]

[16] We omit the proof of Lemma 4.20 as the lemma is actually not needed for proving Theorem 4.16. Careful consideration of the proof of Theorem 4.22 shows that we can only focus on flow between states in $\mathcal{G}(d)$, since the flow h given in the proof of Theorem 4.22 only has positive

We now continue with the proof of Theorem 4.19. It consists of four parts following, in a conceptual sense, the proof template in [41] developed for proving rapid mixing of the switch chain for regular graphs. Certain parts use similar ideas as in [108] where a Markov chain for sampling (near)-perfect matchings is studied. Whenever we refer to [108], the reader is referred to Section 3 of [108].

Proof of Theorem 4.19. We only need to define a flow f' as in Lemma 4.20 so that $b \leq p_1(n)$ and $\ell(f') \leq p_2(n)$ for some polynomials $p_1(\cdot), p_2(\cdot)$ whose degrees may only depend on $k = k_{JS}(d)$. Actually, we are going to show that we may use $p_1(n) = p_2(n) = n^2$. Then the theorem follows from the lemma and the fact that $\ln(|\mathcal{G}'(d)|)$ is upper bounded by a polynomial in n. The latter follows from Equation (1) of McKay and Wormald [127] that implies that

$$|\mathcal{G}(d')| \leq n^{n^2}$$

for any degree sequence d' with n components (see also [96]). So, by the definition of $|\mathcal{G}'(d)|$ we have

$$|\mathcal{G}'(d)| \leq \left(\frac{n(n-1)}{2} + n + 1\right) n^{n^2},$$

and thus $\ln(|\mathcal{G}'(d)|) \leq 3n^3$.

Before we define f', we first introduce some basic terminology similar to that in [41]. Let V be a set of labeled vertices, let $\prec_E$ be a fixed total order on the set $\{\{v, w\} : v, w \in V\}$ of edges, and let $\prec_\mathcal{C}$ be a total order on all *circuits* on the complete graph K_V, i.e., $\prec_\mathcal{C}$ is a total order on the closed walks in K_V that visit every edge at most once. We fix for every circuit one of its vertices where the walk begins and ends.

For given $G, G \in \mathcal{G}(d)$, let $H = G \triangle G'$ be their symmetric difference. We refer to the edges in $G \backslash G'$ as *blue*, and the edges in $G' \backslash G$ as *red*. A *pairing* of red and blue edges in H is a bijective mapping that, for each node $v \in V$, maps every red edge adjacent to v, to a blue edge adjacent to v. The set of all pairings is denoted by $\Psi(G, G')$, and, with θ_v the number of red edges adjacent to v (which is the same as the number of blue edges adjacent to v), we have $|\Psi(G, G')| = \Pi_{v \in V} \theta_v!$.

4.3.1.1 Canonical paths and circuit processing

Similar to the approach in [41], the goal is to construct for each pairing $\psi \in \Psi(G, G')$ a canonical path from G to G' that carries a $|\Psi(G, G')|^{-1}$ fraction of the total flow from G to G' in f'. For notational convenience, for the remaining of the proof we write uv instead of $\{u, v\}$ to denote an edge. For a given pairing ψ

flow between states corresponding to elements in $\mathcal{G}(d)$. That is, when defining the flow h, we essentially forget about all flow in f between any pair of states where at least one state is an auxiliary state, i.e., an element of $\mathcal{G}'(d) \backslash \mathcal{G}(d)$. Said differently, in Theorem 4.22 we could start with the assumption that f routes $1/|\mathcal{G}'(d)|^2$ units of flow between any pair of states in $\mathcal{G}(d)$ in the state space graph of the JS chain, and then the transformation still works. However, the formulations of Theorems 4.19 and 4.22 are more natural for describing a comparison between the JS and switch chains.

and the total order $\prec_E$ given above, we first decompose H into the edge-disjoint union of circuits in a canonical way. We start with the lexicographically smallest edge w_0w_1 in E_H and follow the pairing ψ until we reach the edge w_kw_0 that was paired with w_0w_1. This defines the circuit C_1. If $C_1 = E_H$, we are done. Otherwise, we pick the lexicographically smallest edge in $H \setminus C_1$ and repeat this procedure. We continue generating circuits until $E_H = C_1 \cup \cdots \cup C_s$. Note that all circuits have even length and alternate between red and blue edges, and that they are pairwise edge-disjoint. We form a path

$$G = Z_0, Z_1, \ldots, Z_M = G'$$

from G to G' in the state space graph of the JS chain, by processing the circuits C_i in turn according to the total order $\prec_C$. The *processing of a circuit* C is the procedure during which all blue edges on C are deleted, and all red edges of C are added to the current graphical realization, using the three types of transitions in the JS chain mentioned at the beginning of this section. All other edges of the current graphical realization remain unchanged. In general, this can be done similarly to the circuit processing procedure in [108].[17]

Circuit processing [108]. Let $C = vx_1x_2\ldots x_qv$ be a circuit with start node v. We may assume, without loss of generality, that vx_1 is the lexicographically smallest blue edge adjacent to the starting node v. We first perform a type 0 transition in which we remove the blue edge vx_1. Then we perform a sequence of $\frac{q-1}{2}$ type 1 transitions in which we add the red edge x_ix_{i+1} and remove the blue edge $x_{i-1}x_i$ for $i = 1, 3, \ldots, q$. Finally we perform a type 2 transition in which we add the red edge vx_q. In particular, this means that the elements on the canonical path right before and after the processing of a circuit belong to $\mathcal{G}(d)$. It is easy to see that all the intermediate elements that we visit during the processing of the circuit C belong to $\mathcal{G}'(d) \setminus \mathcal{G}(d)$, i.e., every element has either precisely two nodes with degree deficit one, or one node with degree deficit two. This is illustrated in Figures 4.6, 4.7 and 4.8 for the circuit in Figure 4.5.

For the next part, we define the notion of an encoding that can be used to bound the congestion of an edge in the state space graph of the JS chain using an injective mapping argument.

4.3.1.2 Encoding

Let $t = (Z, Z')$ be a given transition of the Markov chain. Suppose two graphs G and G' use the transition t over some canonical path for some pairing $\psi \in \Psi(G, G')$. Let $H = G \triangle G'$. We define the encoding

$$L_t(G, G') = \begin{cases} (H \triangle (Z \cup Z')) - e_{H,t} & \text{if } t \text{ is a Type 1 transition,} \\ H \triangle (Z \cup Z') & \text{otherwise,} \end{cases}$$

[17] This is the main difference between the switch chain analyses [41, 96, 131, 70, 73, 71] and our analysis. The processing of a circuit is much more complicated if performed directly in the switch chain.

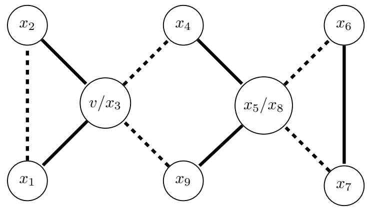

Figure 4.5: The circuit $C = vx_1x_2x_3x_4x_5x_6x_7x_8x_9v$ with $v = x_3$ and $x_5 = x_8$. The blue edges are represented by the solid edges, and the red edges by the dashed edges.

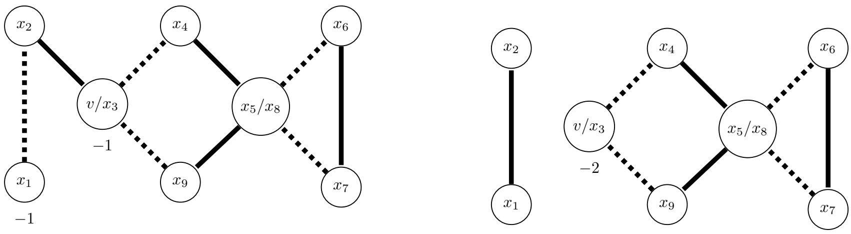

Figure 4.6: The edge vx_1 is removed using a Type 0 transition (left). The edge x_1x_2 is added and $x_2x_3 = x_2v$ is removed using a Type 1 transition (right). We have also indicated the non-zero degree deficits.

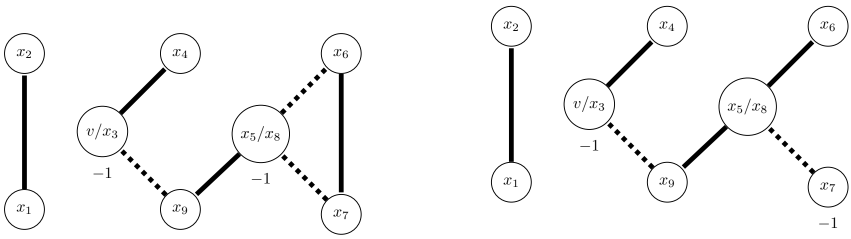

Figure 4.7: The edge x_3x_4 is added and x_4x_5 is removed using a Type 1 transition (left). The edge x_5x_6 is added and x_6x_7 is removed using a Type 1 transition (right).

where $e_{H,t}$ is the first blue edge on the circuit that is currently being processed on the canonical path from G to G' (for the given pairing ψ). This encoding is of a similar nature as the encoding used in [108]. An example is given in Figures 4.9, 4.10 and 4.11. We also refer the reader to Figure 1 in [108] for a detailed example.[18] The following lemma is crucial for the analysis.

[18]Although the perfect matching setting might seem different at first glance, it is actually closely related to our setting, with the only difference that the symmetric difference of two perfect matchings is the union of *node-disjoint cycles*, whereas in our setting the symmetric difference of two graphical realizations is the union of *edge-disjoint circuits*. This is roughly

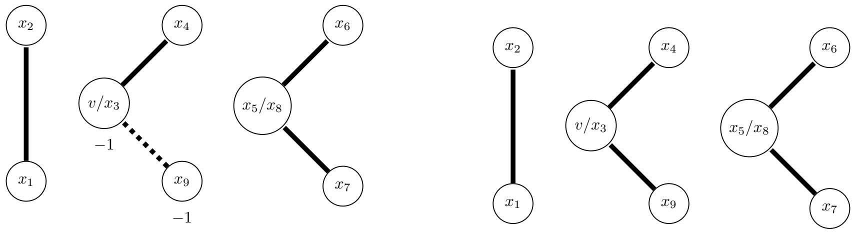

Figure 4.8: The edge $x_7x_8 = x_5x_8$ is added and $x_5x_9 = x_8x_9$ is removed using a Type 1 transition (left). The edge vx_9 is added using a Type 2 transition (right).

Lemma 4.21. *Given $t = (Z, Z')$, L, and ψ, we can uniquely recover G and G'. That is, if L is such that $L_t = L_t(G, G')$ for some pair (G, G'), then (G, G') is the unique pair for which this is the case, given t, L, ψ.*

Proof. We give the proof for when t is a Type 1 transition. The cases of the two other types are similar, and arguably somewhat easier. The proof uses the arguments in [108] interpreted in our setting. First note that $L\triangle(Z \cup Z')$ is a graph in which there are precisely two nodes with odd degree. In particular, the edge $e_{H,t}$ is the unique edge (having as endpoints these odd degree nodes) that has to be added to $L\triangle(Z \cup Z')$ to obtain $H = G\triangle G'$. That is, we have $(L\triangle(Z \cup Z')) + e_{H,t} = H$. The pairing ψ then yields a unique circuit decomposition of $E(H)$ as explained at the beginning of the proof. From the transition t it can be inferred which circuit is currently being processed, and, moreover, we can infer which edges of that circuit belong to G and which to G'. Furthermore, the global ordering $\prec_{\mathcal{C}}$ on all circuits can then be used to determine for every other circuit whether it has been processed already or not. For every such circuit, we can then infer which edges on it belong to G and which to G' by comparing with Z (or Z'). Therefore, G and G' can be uniquely recovered from t, L and ψ. $\square$

4.3.1.3 Bounding the congestion

We complete the proof by using an injective mapping argument to bound the congestion of the flow f' on the edges of the state space graph of the JS chain. The arguments used are a combination of ideas from [108] and the proof of Lemma 2.5 in [41] (see also Lemma 1 in [42]).[19] We again focus on Type 1 transitions t as the proofs for the other two types are similar but simpler.

For a tuple (G, G', ψ), let $p_\psi(G, G')$ denote the canonical path from G to G' for pairing ψ. Let

$$\mathcal{L}_t = \{L_t(G, G') \,|\, (G, G', \psi) \in \mathcal{F}_t\}$$

why the notion of pairings is needed, as they allow us to uniquely determine the circuits. That is, the edge-disjoint circuits determined by the pairing are the analogue of the node-disjoint cycles in the perfect matching setting in [108].

[19] Lemma 2.5 in [41] contained a flaw for which the corrigendum [42] was published.

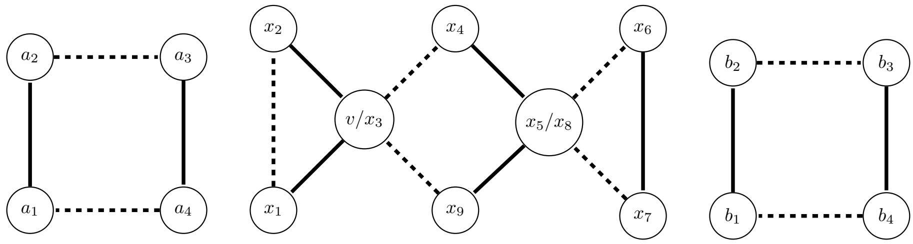

Figure 4.9: Symmetric difference $H = G \triangle G'$ where the solid edges represent the edges G and the dashed edges the edges of G'. From left to right the circuit are numbered C_1, C_2 and C_3, and assume that this is also the order in which they are processed.

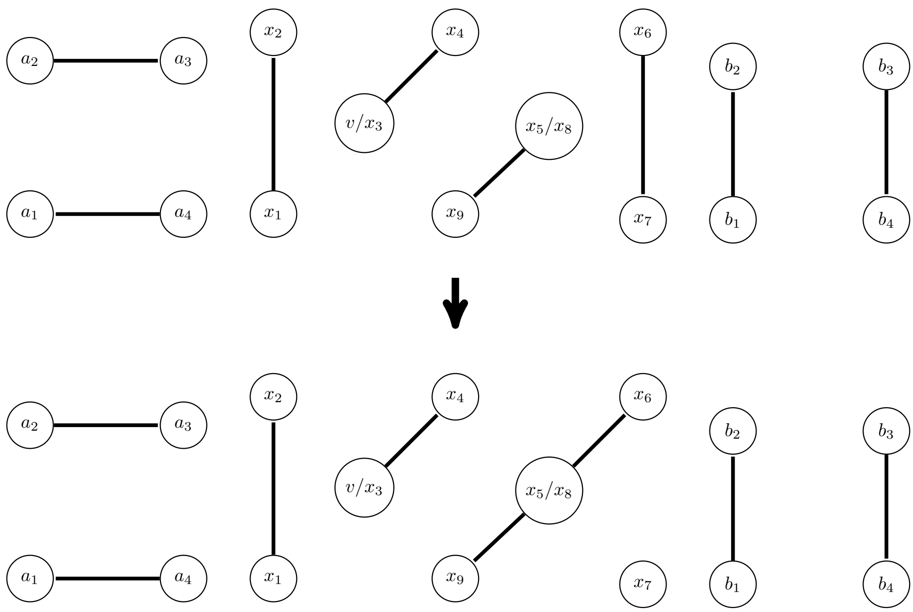

Figure 4.10: The transition $t = (Z, Z')$ that removes the edge x_6x_7 and adds the edge x_5x_6 as part of the processing of C_2. Note that C_1 has already been processed. The edges in $(E(G) \cup E(G')) \setminus E(H)$ are left out.

be the set of all (distinct) encodings L_t, where

$$\mathcal{F}_t = \big\{(G, G', \psi) : t \in p_\psi(G, G')\big\}$$

is the set of all tuples (G, G', ψ) such that the canonical path from G to G' under pairing ψ uses the transition t. A crucial observation is that every encoding $L_t(G, G')$ itself is an element of $\mathcal{G}'(d)$ (see Figure 4.11 for an example). This

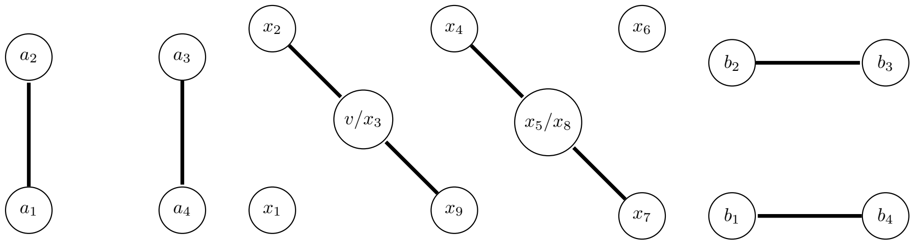

Figure 4.11: The encoding $L = L_t(G, G')$, where again the edges in $(E(G) \cup E(G')) \setminus E(H)$ are left out. Note that in this case $e_{H,t} = vx_1$ and that L is itself an element of $\mathcal{G}'(d)$.

implies that

$$|\mathcal{L}_t| \leq |\mathcal{G}'(d)|. \tag{4.8}$$

Moreover, with $H = G \triangle G'$ and $L = L_t(G, G')$, the pairing ψ has the property that it pairs up the edges of $E(H) \setminus E(L)$ and $E(H) \cap E(L)$ in such a way that for every node v (with the exception of at most two nodes) each edge in $E(H) \setminus E(L)$ that is incident to v is paired up with an edge in $E(H) \cap E(L)$ that is incident to v. However, there are either two nodes for which the incident edges in $E(H) \setminus E(L)$ exceed by 2 the incident edges in $E(H) \cap E(L)$, or one node for which the incident edges in $E(H) \setminus E(L)$ exceed by 4 the incident edges in $E(H) \cap E(L)$. These are exactly the two nodes with degree deficit 1 or the one node with degree deficit 2 in L; for the example in Figure 4.11 these are nodes x_1 and x_6. There ψ pairs up each edge of $E(H) \cap E(L)$ to an edge of $E(H) \setminus E(L)$ but also two edges of $E(H) \setminus E(L)$ with each other; or in the case of one node with degree deficit 2 ψ pairs up each edge of $E(H) \cap E(L)$ to an edge of $E(H) \setminus E(L)$ but also makes two pairs out of the remaining 4 edges in $E(H) \setminus E(L)$. Let $\Psi'(L)$ be the set of all pairings with this property.[20] Note that not every such pairing has to correspond to a tuple (G, G', ψ) for which $t \in p_\psi(G, G')$.

By simply counting, we can upper bound $|\Psi'(L)|$ in terms of $|\Psi(H)|$. We show the calculation for the case where L has two nodes with degree deficit 1. The case of one node with degree deficit 2 is very similar and the same upper bound works there as well. Suppose that u, w are the two nodes of L with degree deficit 1. Then

$$\begin{aligned}
|\Psi'(L)| &= \left(\Pi_{v \in V \setminus \{u,w\}} \theta_v!\right) \cdot \frac{(\theta_u + 1)!}{2} \cdot \frac{(\theta_w + 1)!}{2} \\
&= |\Psi(H)| \cdot \frac{(\theta_u + 1)(\theta_w + 1)}{4} \\
&\leq n^2 \cdot |\Psi(H)| \,.
\end{aligned} \tag{4.9}$$

[20]Remember that we do not need to know G and G' in order to determine the set H. It can be found based on L and the transition $t = (Z, Z')$, as described in the proof of Lemma 4.21.

Putting everything together, we have

$$\begin{aligned}
|\mathcal{G}'(d)|^2 f'(e) &= \sum_{(G,G')} \sum_{\psi \in \Psi(G,G')} \mathbf{1}(e \in p_\psi(H)) |\Psi(H)|^{-1} \\
&\leq \sum_{L \in \mathcal{L}_t} \sum_{\psi' \in \Psi'(L)} |\Psi(H)|^{-1} \qquad \text{(using Lemma 4.21)} \\
&\leq n^2 \sum_{L \in \mathcal{L}_t} 1 \qquad \text{(using (4.9))} \\
&\leq n^2 \cdot |\mathcal{G}'(d)|. \qquad \text{(using (4.8))} \qquad (4.10)
\end{aligned}$$

The usage of Lemma 4.21 for the first inequality works as follows. Every tuple $(G, G', \psi) \in \mathcal{F}_t$ with encoding $L_t(G, G')$ generates a unique tuple in $\{L_t(G, G')\} \times \Psi'(L_t(G, G'))$. But since, by Lemma 4.21, we can uniquely recover G and G' from L, t and ψ, we have that $\sum_{L \in \mathcal{L}_t} |\{L\} \times \Psi'(L)| = \sum_{L \in \mathcal{L}_t} \sum_{\psi' \in \Psi'(L)} 1$ is an upper bound on the number of canonical paths that use t.

By rearranging (4.9) we get the upper bound for f' required in Lemma 4.20. What is left to show is that $\ell(f')$ is not too large. This, however, is determined by the way we defined the canonical paths. It is easy to see that for any canonical path between any two graphs $G, G' \in \mathcal{G}(d)$ has length at most $\frac{3}{4}|E(G \triangle G')|$ and, therefore, $\ell(f') \leq n^2$.

This finishes the proof of Theorem 4.19. □

4.3.2 Flow transformation

Next we show that, when d comes from a family of strongly stable degree sequences, an efficient multicommodity flow for the JS chain on $\mathcal{G}'(d)$ can be transformed into an efficient multicommodity flow for the switch chain on $\mathcal{G}(d)$. In combination with Theorem 4.19 this implies that if $\mathcal{D}$ is strongly stable, then for any sequence in $\mathcal{D}$ there exists an efficient flow for the switch chain. For the sake of simplicity, we did not attempt to optimize the bounds in the proof of Theorem 4.22.

Theorem 4.22. *Let $\mathcal{D}$ be a strongly stable family of degree sequences with respect to some constant k, and let $p(n)$ and $r(n)$ be polynomials such that for any $d = (d_1, \ldots, d_n) \in \mathcal{D}$ there exists an efficient multicommodity flow f_d for the JS chain on $\mathcal{G}'(d)$ with the property that $\max_e f(e) \leq p(n)/|\mathcal{G}'(d)|$ and $\ell(f) \leq r(n)$.*

Then there exists a polynomial $t(n)$ such that for all $d = (d_1, \ldots, d_n) \in \mathcal{D}$ there is a feasible multicommodity flow g_d for the switch chain on $\mathcal{G}(d)$ with (i) *$\ell(g_d) \leq 2k \cdot \ell(f_d)$, and* (ii) *for every edge e of the state space graph of the switch chain, we have*

$$g_d(e) \leq t(n)^k \cdot \frac{p(n)}{|\mathcal{G}(d)|}. \qquad (4.11)$$

Proof. Let $d \in \mathcal{D}$. For simplicity we will write f and g instead of f_d and g_d respectively. We let $\mathcal{P}_{xy}$ refer to the set of simple paths between x and y in the state space graph of the *JS chain* (not those in the state space graph of the switch chain). We first introduce some additional notation.

For every pair $(x, y) \in \mathcal{G}'(d) \times \mathcal{G}'(d)$ with $x \neq y$, and for any $p \in \mathcal{P}_{xy}$, we write $\alpha(p) = f(p)|\mathcal{G}'(d)|^2$. Recall that since the stationary distribution of the JS chain is uniform on $\mathcal{G}'(d)$ we have $\sum_{p \in \mathcal{P}_{xy}} f(p) = |\mathcal{G}'(d)|^{-2}$. Thus, $\sum_{p \in \mathcal{P}_{xy}} \alpha(p) = 1$. Moreover, we define $\alpha(e) = \sum_{p \in \mathcal{P}_{xy}: e \in p} \alpha(p) = f(e)|\mathcal{G}'(d)|^2$.

Now, for every $G \in \mathcal{G}'(d) \setminus \mathcal{G}(d)$ choose some $\varphi(G) \in \mathcal{G}(d)$ that is within distance k of G in the JS chain, and take $\varphi(G) = G$ for $G \in \mathcal{G}(d)$. Based on the arguments in the proof of Proposition 4.6, it follows that for any $H \in \mathcal{G}(d)$,

$$|\varphi^{-1}(H)| \leq n^{3k}, \tag{4.12}$$

using that the maximum in-degree of any element in the state space graph of the JS chain is upper bounded by n^3. In particular, this implies that

$$\frac{|\mathcal{G}'(d)|}{|\mathcal{G}(d)|} \leq n^{3k}. \tag{4.13}$$

Let the flow h be defined as follows for any given pair (x, y). If $(x, y) \in \mathcal{G}(d) \times \mathcal{G}(d)$, take $h(p) = \alpha(p)/|\mathcal{G}(d)|^2$ for all $p \in \mathcal{P}_{xy}$. If either x or y is not contained in $\mathcal{G}(d)$, take $h(p) = 0$ for every $p \in \mathcal{P}_{xy}$. Note that h is a multicommodity flow that routes $1/|\mathcal{G}(d)|^2$ units of flow between any pair $(x, y) \in \mathcal{G}(d) \times \mathcal{G}(d)$, and zero units of flow between any other pair of states in $\mathcal{G}'(d)$.

Note that

$$h(e) \leq \frac{|\mathcal{G}'(d)|^2}{|\mathcal{G}(d)|^2} \cdot f(e) \leq \frac{|\mathcal{G}'(d)|^2}{|\mathcal{G}(d)|^2} \frac{p(n)}{|\mathcal{G}'(d)|} = \frac{p(n)}{|\mathcal{G}(d)|} \frac{|\mathcal{G}'(d)|}{|\mathcal{G}(d)|} \leq n^{3k} \cdot \frac{p(n)}{|\mathcal{G}(d)|}, \tag{4.14}$$

using the definition of h in the first inequality, the assumption on f in the second inequality, and the upper bound of (4.13) in the last one.

Next, we merge the "auxiliary states" in $\mathcal{G}'(d) \setminus \mathcal{G}(d)$, i.e., the states not reached by the switch chain, with the elements of $\mathcal{G}(d)$. Informally speaking, for every $H \in \mathcal{G}(d)$ we merge all the nodes in $\varphi^{-1}(H)$ into a *supernode*. Self-loops created in this process are removed, and parallel arcs between states are merged into one arc that gets all the flow of the parallel arcs. Formally, we consider the graph Γ where $V(\Gamma) = \mathcal{G}(d)$ and $e = (H, H') \in E(\Gamma)$ if and only if H and H' are switch adjacent or if there exist $G \in \varphi^{-1}(H)$ and $G' \in \varphi^{-1}(H')$ such that G and G' are JS adjacent. Moreover, for a given h-flow carrying path $(G_1, G_2, \ldots, G_q) = p \in \mathcal{P}_{xy}$, let $p'_\Gamma = (\varphi(G_1), \varphi(G_2), \ldots, \varphi(G_q))$ be the corresponding (possibly non-simple) directed path in Γ. Any self-loops and cycles can be removed from p'_Γ and let p_Γ be the resulting simple path in Γ. Over p_Γ we route $h_\Gamma(p_\Gamma) = h(p)$ units of flow. Note that h_Γ is a flow that routes $1/|\mathcal{G}(d)|^2$ units of flow between any pair of states $(x, y) \in \mathcal{G}(d) \times \mathcal{G}(d)$ in the graph Γ and that $\ell(h_\Gamma) \leq \ell(f)$. Furthermore, the flow h_Γ on an edge $(H, H') \in E(\Gamma)$ is then bounded by

$$h_\Gamma(H, H') \leq \sum_{\substack{(G, G') \in \varphi^{-1}(H) \times \varphi^{-1}(H') \\ G \text{ and } G' \text{ are JS adjacent}}} h(G, G'), \tag{4.15}$$

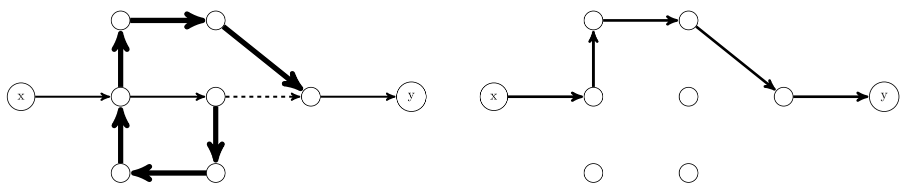

Figure 4.12: The dashed edge on the left represents an illegal edge, and the bold path represents a "short" detour. The shortcutted path on the right is the result of removing any loops and cycles.

where the inequality (instead of an equality) follows from the fact that when we map a path $p \in \mathcal{P}_{xy}$ to the corresponding path p_Γ, some edges of the intermediate path p'_Γ may be deleted. Using (4.12), it follows that $|\varphi^{-1}(H) \times \varphi^{-1}(H')| \leq n^{3k} \cdot n^{3k} = n^{6k}$ and therefore, in combination with (4.14) and (4.15), we have that

$$h_\Gamma(e) \leq n^{3k} \cdot n^{6k} \cdot \frac{p(n)}{|\mathcal{G}(d)|}\,. \tag{4.16}$$

Now recall how $E(\Gamma)$ was defined. An edge (H, H') might have been added because: *(i)* H and H' are switch adjacent (we call these edges of Γ *legal*), or *(ii)* H and H' are *not* switch adjacent but there exist $G \in \varphi^{-1}(H)$ and $G' \in \varphi^{-1}(H')$ that are JS adjacent (we call these edges of Γ *illegal*). The final step of the proof consists of showing that the flow on every illegal edge in $E(\Gamma)$ can be rerouted over a "short" path consisting only of legal edges. In particular, for every flow carrying path p using e, we are going to show that the flow $h_\Gamma(p)$ can rerouted over some legal detour, the length of which is bounded by a multiple of k. Doing this iteratively for every remaining illegal edge on p, we obtain a directed path p'' only using legal edges, i.e., edges of the state space graph of the switch chain. Of course, p'' might not be simple, so any self-loops and cycles can be removed, as before, to obtain the simple legal path p'. Figure 4.12 illustrates this procedure for a path with a single illegal edge. Note that deleting self-loops and cycles only decreases the amount of flow on an edge.

The crucial observation here is that if $(H, H') \in E(\Gamma)$, then $|E(H) \triangle E(H')| \leq 4k$. That is, even though H and H' might not be switch adjacent, they are not too far apart. To see this, first note that the symmetric difference of any two JS adjacent graphs has size at most 2. Moreover, if one of any two JS adjacent graphs is in $\mathcal{G}(d)$, then their symmetric difference has size 1. In particular, for any $G^* \in \mathcal{G}'(d)$, we have $|E(G^*) \triangle E(\varphi(G^*))| \leq 2k - 1$.

Clearly, if $(H, H') \in E(\Gamma)$ is legal, then $|E(H) \triangle E(H')| = 4 \leq 4k$. Assume $(H, H') \in E(\Gamma)$ is illegal. Then there exist JS adjacent $G \in \varphi^{-1}(H)$ and $G' \in \varphi^{-1}(H')$ and according to the above we have

$$\begin{aligned} |E(H) \triangle E(H')| \;&\leq\; |E(H) \triangle E(G)| + |E(G) \triangle E(G')| + |E(G') \triangle E(H')| \\ &\leq\; 2k - 1 + 2 + 2k - 1 \leq 4k\,. \end{aligned}$$

Moreover, this implies that we can go from H to H' in a 'small' number of moves in the switch chain. This easily follows from most results showing that the state space of the switch chain is connected, e.g., from [167].[21] Specifically, here we use the following result of Erdős, Király, and Miklós [69] which implies that we can go from H to H' in $2k$ switches.

Theorem 4.23 (follows from Theorem 3.6 in [69])**.** *Let $d = (d_1, \ldots, d_n)$ be a degree sequence. For any two graphs $H, H' \in \mathcal{G}(d)$, H can be transformed into H' using at most $\frac{1}{2}|E(H)\triangle E(H')|$ switches.*

For every *illegal* edge $e \in E(\Gamma)$, we choose such a (simple) path from H to H' with at most $2k$ transitions and reroute the flow of e over this path. Note that for any legal edge $e \in E(\Gamma)$, the number of illegal edge detours that use e for this rerouting procedure, is at most $(n^4)^{2k} \cdot (n^4)^{2k} = n^{16k}$, using the fact that in the state space graph of the switch chain the maximum degree of an element is at most n^4 and any illegal edge using e in its rerouting procedure must lie within distance $2k$ of e. Combining this with (4.16), we see that the resulting flow, g, satisfies

$$g(e) \leq \frac{p(n) \cdot n^{9k} + p(n) \cdot n^{16k}}{|\mathcal{G}(d)|}. \tag{4.17}$$

Note that $\ell(g) \leq 2k\ell(h_\Gamma)$. This holds because every illegal edge on a flow-carrying path gives rise to at most $2k$ additional edges as a result of rerouting the flow over legal edges, and the removal of loops and cycles from any resulting non-simple path can only decrease its length. Combining this inequality with $\ell(h_\Gamma) \leq \ell(f)$ (as we noted above), we get $\ell(g) \leq 2k \cdot \ell(f)$. This completes the proof of (4.11), as we have now constructed a feasible multicommodity flow g in the state space graph of the switch chain with the desired properties. □

4.4 Switch chain for 2-class JDM instances

In this section we use a similar high-level approach as that in Section 4.3 to show that the (restricted) switch chain[22] defined in Subsection 4.2.2 is *always* rapidly mixing for JDM instances with two degree classes. The formal statement is as follows.

Theorem 4.24. *Let $\mathcal{D}$ be the family of instances of the joint degree matrix model with two degree classes. Then the switch chain is rapidly mixing for instances in $\mathcal{D}$.*

The proof of Theorem 4.24 consists of three parts. In analogy to the JS chain we first analyze a simpler Markov chain, called the *hinge flip chain*, that adds

[21] To be precise, we can focus on the subgraph induced by the nodes with positive degree in the symmetric difference. Taylor's proof on the connectivity of the state space of the switch chain [167] implies that we can find $O(k^2)$ switches to get from H to H', only using edges in this induced subgraph.

[22] In this section the switch chain always refers to the restricted switch chain.

and removes (at most) one edge at a time (see Figure 4.4).[23] Very much like the JS chain, the hinge flip chain might slightly violate the degree constraints. Now, however, the joint degree constraints might be violated as well. The definition of *strong stability* is appropriately adjusted to account for both deviations from the original requirements, and in Section 4.4.2 we show that instances of the JDM model with two degree classes are indeed strongly stable under this definition. Finally, we use a similar embedding argument as in Theorem 4.22 to argue that the (restricted) switch chain is rapidly mixing. Next, we give a more detailed description of these three parts.

Proof overview. The first step of the proof is to show that the hinge flip chain defined on a *strict superset* of the state space mixes rapidly for strongly stable instances. This is done in Section 4.4.1. The auxiliary states have the property that the joint degree constraint may only be violated slightly, by an additive value of one to be precise. This makes the analysis much more challenging than the one for the JS chain presented in Section 4.3.1. In order to overcome the difficulties that arise due to the fact that the number of edges across the two degree classes should remain almost the same, we rely on ideas introduced by Bhatnagar et al. [15] for uniformly sampling exact matchings; see Remark 1.13.[24] In particular, in the circuit processing part of the proof, we process a circuit at multiple places *simultaneously* in case there is only one circuit in the canonical decomposition of a pairing; or we process multiple circuits *simultaneously* in case the decomposition yields multiple circuits. At the core of this approach lies a variant of the *mountain-climbing problem* [105, 176]. In our case the analysis is more involved than that of [15], and we therefore use different arguments in various parts of the proof.

It is interesting to note that the analysis of the hinge flip chain is not carried out in the JDM model but in the more general partition adjacency matrix model (Section 4.2.3). The difference from the JDM model is that in each class V_i the nodes need not have the same degree but rather follow a given degree sequence of size $|V_i|$. Given that small deviations from the prescribed degrees cannot be directly handled—by definition—by the JDM model, the PAM model is indeed a more natural choice for this step.

Next, in Section 4.4.2, we show that for any JDM instance, any graph in the state space of the hinge flip chain (i.e., graphs that satisfy or *almost* satisfy the joint degree requirements) can be transformed to a graphical realization of the original instance within 7 hinge flips at most. That is, the set of JDM instances is a strongly stable family of instances of the PAM model and thus the hinge flip chain mixes rapidly for JDM instances.

The final step is an embedding argument, along the lines of the argument of

[23] This is the Markov chain that naturally corresponds to the definition of P-stability given in [107], whereas the JS chain is the natural choice for the (equivalent) definition of P-stability given in [109].

[24] These are called bichromatic matchings in [15].

Section 4.3.2, for transforming the efficient flow for the hinge flip chain to an efficient flow for the switch chain. As an intermediate step we need an analogue of Theorem 4.23, but this directly follows from the proof of irreducibility of the (restricted) switch chain in [5]. This step is presented in Section 4.4.3.

4.4.1 Rapid mixing of the hinge flip chain

In this section we show that the hinge flip chain is rapidly mixing for strongly stable tuples (Theorem 4.25). We prove Theorem 4.25 based on ideas introduced in [15]. Throughout this section we always consider tuples (γ, d) coming from strongly stable families.

Theorem 4.25. *Let $\mathcal{D}$ be a strongly stable family of tuples (γ, d) with respect to some constant k. Then there exist polynomials $p(n)$ and $r(n)$ such that for any $(\gamma, d) \in \mathcal{D}$, with $d = (d_1, \ldots, d_n)$, there exists an efficient multicommodity flow f for the hinge flip chain $\mathcal{M}(\gamma, d)$ on $\mathcal{G}'(\gamma, d)$ satisfying $\max_e f(e) \leq p(n)/|\mathcal{G}'(d)|$ and $\ell(f) \leq r(n)$. Hence, the hinge flip chain $\mathcal{M}(\gamma, d)$ is rapidly mixing for families of strongly stable tuples.*

We will use the following lemma in order to simplify the proof of Theorem 4.25. It is the analogue of Lemma 4.20 in Section 4.3.1.

Lemma 4.26. *Let f' be a flow that routes $1/|\mathcal{G}'(\gamma, d)|^2$ units of flow between any pair of states in $\mathcal{G}(\gamma, d)$ in the chain $\mathcal{M}(\gamma, d)$, so that $f'(e) \leq b/|\mathcal{G}'(\gamma, d)|$ for all e in the state space graph of $\mathcal{M}(\gamma, d)$. Then f' can be extended to a flow f that routes $1/|\mathcal{G}'(\gamma, d)|^2$ units of flow between any pair of states in $\mathcal{G}'(\gamma, d)$ with the property that for all e*

$$f(e) \leq q(n) \frac{b}{|\mathcal{G}'(\gamma, d)|}, \tag{4.18}$$

where $q(\cdot)$ is a polynomial whose degree only depends on $k(\gamma, d)$ $(\leq k)$. Moreover, $\ell(f) \leq \ell(f') + 2k(\gamma, d)$.

Proof. We extend the flow f' to f as follows. For any $G \in \mathcal{G}'(\gamma, d) \setminus \mathcal{G}(\gamma, d)$ fix some $\phi(G) \in \mathcal{G}(\gamma, d)$ within distance k of G (which exists by assumption of strong stability), and fix some path in the state space graph from G to $\phi(G)$ of length at most k. Moreover, define $\phi(H) = H$ for all $H \in \mathcal{G}(\gamma, d)$. The flow between G and any given $G' \in \mathcal{G}'(\gamma, d)$ is now sent as follows.

First route $1/|\mathcal{G}'(\gamma, d)|^2$ units of flow from G to $\phi(G)$ over the fixed path from G to $\phi(G)$. Then use the flow-carrying paths used to send $1/|\mathcal{G}'(\gamma, d)|^2$ units of flow between $\phi(G)$ and $\phi(G')$ as in the flow f' (note that in general multiple paths might be used for this in the flow f'). Finally, use the reverse of the fixed path from G' to $\phi(G')$ to route $1/|\mathcal{G}'(\gamma, d)|^2$ from $\phi(G')$ to G'. For any $H \in \mathcal{G}(\gamma, d)$, we have $|\phi^{-1}(H)| \leq \text{poly}(n^k)$, as the in- and out-degrees of the nodes in the state space graph of $\mathcal{M}(\gamma, d)$ are polynomially bounded. It can then be shown that this extension of f', yielding the flow f, only gives an additional term of at most $\text{poly}(n^k) \frac{b}{|\mathcal{G}'(\gamma, d)|}$ to the congestion of every arc in the state space graph of the chain $\mathcal{M}(\gamma, d)$ in the flow f'. Hence, the extended flow f satisfies (4.18) for some appropriately chosen polynomial $q(n)$. □

Because of Lemma 4.26 it now suffices to show that there exists a flow f' that routes $1/|\mathcal{G}'(\gamma, d)|^2$ units of flow between any two pair of states in $\mathcal{G}(\gamma, d)$, in the state space graph of the chain $\mathcal{M}(\gamma, d)$, with the property that $f'(e) \leq p(n)/|\mathcal{G}'(\gamma, d)|$, and $\ell(f') \leq q(n)$ for some polynomials $p(\cdot), q(\cdot)$ whose degrees may only depend on $k(\gamma, d)$. Note that f' is not a feasible multi-commodity flow as defined in Section 4.2, but should rather be interpreted as an intermediate auxiliary flow. The proof of Theorem 4.25 will consist of multiple parts following, conceptually, the proof template in [41] developed for proving rapid mixing of the switch chain for regular graphs. The main difference is that for the so-called *canonical paths* between states we rely on ideas introduced in [15].

4.4.1.1 Canonical paths

We first introduce some basic terminology similar to that in [41]. Let V be a set of labeled vertices, let $\prec_E$ be a fixed total order on the set $\{\{v, w\} : v, w \in V\}$ of edges, and let $\prec_{\mathcal{C}}$ be a total order on all *circuits* on the complete graph K_V, i.e., $\prec_{\mathcal{C}}$ is a total order on the closed walks in K_V that visit every edge at most once. We fix for every circuit one of its vertices where the walk begins and ends.

For given $G, G \in \mathcal{G}(\gamma, d)$, let $H = G \triangle G'$ be their symmetric difference. We refer to the edges in $G \setminus G'$ as *blue*, and the edges in $G' \setminus G$ as *red*. A *pairing* of red and blue edges in H is a bijective mapping that, for each node $v \in V$, maps every red edge adjacent to v, to a blue edge adjacent to v. The set of all pairings is denoted by $\Psi(G, G')$, and, with θ_v the number of red edges adjacent to v (which is the same as the number of blue edges adjacent to v), we have $|\Psi(G, G')| = \Pi_{v \in V} \theta_v!$.

Remember that we are considering an instance of the PAM model with two classes V_1 and V_2. For a given graphical realization $G \in \mathcal{G}(\gamma, d)$ we say that $e \in E(G)$ is a *cut* edge if it has an endpoint in both V_1 and V_2. Otherwise we say that e is an *internal* edge, as both endpoints either lie both in the class V_1 or both in class V_2.

Similar to the approach in [41], the goal is to construct for each pairing $\psi \in \Psi(G, G')$ a canonical path from G to G' that carries a fraction $|\Psi(G, G')|^{-1}$ of the total flow from G to G' in f'. For a given pairing ψ and the total order $\prec_E$ given above, we first decompose H into the edge-disjoint union of circuits in a canonical way. We start with the lexicographically least edge w_0w_1 in E_H and follow the pairing ψ until we reach the edge w_kw_0 that was paired with w_0w_1. This defines the circuit C_1 (which is indeed a closed walk). If $C_1 = E_H$, we are done. Otherwise, we pick the lexicographically least edge in $H \setminus C_1$ and repeat this procedure. We continue generating circuits until $E_H = C_1 \cup \cdots \cup C_s$. Note that all circuits have even length and alternate between red and blue edges, and that they are pairwise edge-disjoint.

We form a path from G to G' in the state space graph of the chain $\mathcal{M}(\gamma, d)$ by changing the blue edges of G into the red edges of G' using hinge flip operations. For certain pairings this can be done in a straightforward way, but in general this is not the case. As a warm-up, we first consider a simple case (this case essentially describes how we would process the circuits in case there is only one class).

Warm-up example. If for every i, the circuit C_i exclusively consists of internal edges, only within V_1 or only within V_2, or exclusively of cut edges, then circuits can be processed according to the ordering $\prec_{\mathcal{C}}$ as follows. Let $C = x_0x_1x_2 \dots x_qx_0$ be a circuit, and assume w.l.o.g. that x_0x_1 is the lexicographically smallest blue edge adjacent to the starting node x_0 of the circuit. The processing of C now consists of performing a sequence of hinge flips on the ordered pairs (x_{i-1}, x_i, x_{i+1}) for $i = 1, \dots, q$ with the convention that $x_{q+1} = x_0$. This is illustrated in Figures 4.13, 4.14 and 4.15 for an example of C as illustrated in Figure 4.13 on the left.[25] We have also indicated the degree surplus and deficit at every step. By assumption, the edges of C either are all internal edges or all cut edges. Therefore, throughout the processing of C, we never violate the constraint that there should be γ edges between the classes V_1 and V_2, and, in particular, this implies that every intermediate state is an element of $\mathcal{G}'(\gamma, d)$.

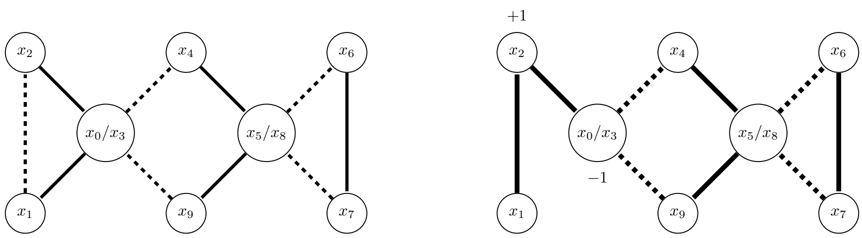

Figure 4.13: The circuit $C = x_0x_1x_2x_3x_4x_5x_6x_7x_8x_9x_0$ with $x_0 = x_3$ and $x_5 = x_8$. The blue edges are represented by the solid edges, and the red edges by the dashed edges (left). The edge x_0x_1 is removed and x_1x_2 is added (right).

In general, however, it might happen that circuits contain both cut and internal edges, in which case we cannot use the circuit processing procedure explained above, as the processing of a circuit might result in a graphical realization for which the number of edges between the classes V_1 and V_2 lies outside the set $\{\gamma - 1, \gamma, \gamma + 1\}$. The latter condition is necessary for the intermediate states in the circuit processing procedure to be elements of $\mathcal{G}'(\gamma, d)$, by definition of that set. In order to overcome the issue described above, we will use the ideas in [15], and process a circuit at multiple places *simultaneously* in case there is only one circuit in the canonical decomposition of a pairing, or, process multiple circuits

[25] This is similar to the procedure described in Figures 4.6, 4.7 and 4.8 for the JS chain (we give an example here as well for self-containment).

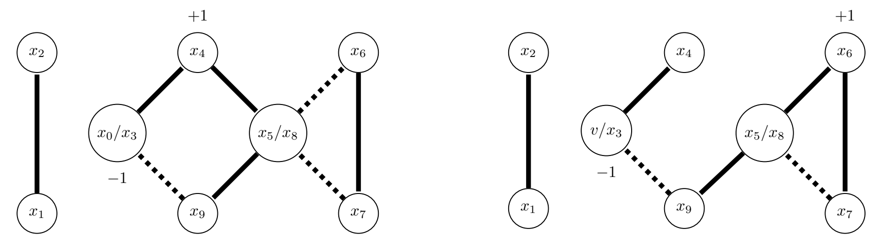

Figure 4.14: The edge x_2x_3 is removed and x_3x_4 is added (left). The edge x_4x_5 is removed and x_5x_6 is added (right).

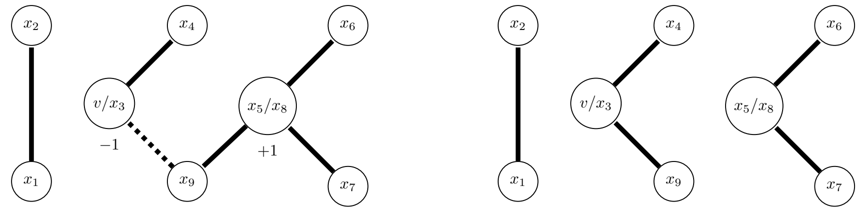

Figure 4.15: The edge x_6x_7 is removed and x_7x_8 is added (left). The edge x_8x_9 is removed and x_9x_0 is added (right).

simultaneously in case the decomposition yields multiple circuits. At the core of this approach lies (a variation of) the *mountain-climbing problem* [105, 176]. We begin with introducing this problem, and afterwards continue with the description of the circuit processing procedure, based on the solution to the mountain climbing problem.

Intermezzo: mountain climbing problem. We first introduce some notation and terminology. For non-negative integers a, b with $a + 1 < b$ we define an $\{a, b\}$-*mountain* as a function $P : \{a, a + 1, \dots, b\} \to \mathbb{Z}_{\geq 0}$ with the properties that (i) $P(a) = P(b) = 0$; (ii) $P(i) > 0$ for all $i \in \{a + 1, \dots, b - 1\}$; and (iii) $|P(i+1) - P(i)| = 1$ for all $i \in \{a, \dots, b-1\}$. A function $P : \{a, a+1, \dots, b\} \to \mathbb{Z}_{\leq 0}$ is called an $\{a, b\}$-*valley* if the function $-P$ is an $\{a, b\}$-mountain. We subdivide a mountain into a left side $\{a, \dots, t\}$ and right side $\{t, \dots, b\}$ where t is the smallest integer maximizing the function P. For a valley function P, the left and right side are determined by the smallest integer t minimizing the function P.

Definition 4.27. A *traversal* of the mountain P on $\{a, \dots, b\}$ is a sequence

$$(a, t) = (i_1, j_1), \dots, (i_k, j_k) = (t, b)$$

with the properties

(a) $|i_r - i_{r+1}| = |j_r - j_{r+1}| = 1$,

(b) $P(i_r) + P(j_r) = P(t)$,

(c) $a \leq i_r \leq t$ and $t \leq j_r \leq b$,

for all $1 \leq r \leq k-1$. We always assume that a traversal is minimal, in the sense that there is no subsequence of $(a,t) = (i_1, j_1), \ldots, (i_k, j_k) = (t, b)$ which is also a traversal.

Roughly speaking, we place one person at the far left end of the mountain, and one at the first top. These persons now simultaneously traverse the mountain in such a way that the sum of their heights is always equal, and they always stay on their respective sides of the mountain that they started. The goal of the person on the left it to ascend to the top, whereas the goal of the player at the top is to descend to the far right of the mountain.

Lemma 4.28 ([15])**.** *For any mountain or valley function P on $\{a, \ldots, b\}$ with first top t, there exists a traversal of P of length at most $O((t-a)(b-t))$, that can be found in time $O((t-a)(b-t))$.*

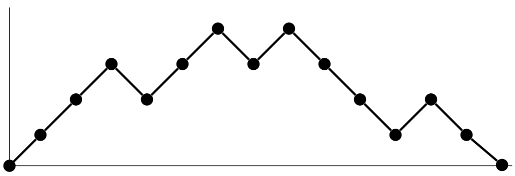

Figure 4.16: Example of a mountain function P on the integers in $\{0, \ldots, 14\}$ with the first top at $t = 6$. The left side of the mountain is given by $\{0, \ldots, 6\}$ and the right side by $\{6, \ldots, 14\}$. A traversal of P is given by the sequence $(0,6), (1,7), (0,8), (1,9), (2,10), (3,11), (4,12), (5,13), (6,14)$.

We finish this part with some additional notation that will be used later on. Let $P_j : \{a_j, \ldots, b_j\} \to \mathbb{Z}$ for $j = 1, \ldots, l$ be a collection of mountain and valley functions such that $a_1 = 0$, $b_j = a_{j+1}$ for $j = 1, \ldots, p-1$, and every P_j is either a mountain or a valley. We define the *landscape* Q of the functions $P_1, \ldots, P_l$ as the function $Q : \{0, 1, \ldots, b_l\} \to \mathbb{Z}$ given by $Q(i) = P_j(i)$ where $j = j(i)$ is such that $i \in \{a_j, \ldots, b_j\}$. Note that $Q(0) = Q(b_l) = 0$, and $|Q(i+1) - Q(i)| = 1$ for all $i \in \{0, \ldots, b_j - 1\}$. Moreover, for any function $R : \{0, \ldots, r\} \to \mathbb{Z}$ satisfying the latter two conditions, there is a unique collection of mountain and valley functions so that R is the landscape of those functions. We call functions satisfying these conditions *landscape functions.*

General case. We first partition every circuit into a collection of so-called sections, which in turn will be grouped into so-called segments. Let $C_1, \ldots, C_s$ be the canonical circuit decomposition of the symmetric difference $G \triangle G'$ for some pairing ψ, and assume w.l.o.g. that $C_i \prec_C C_j$ whenever $i < j$. We write $C_i =$

$x_0^i x_1^i \dots x_{q_i}^i x_0^i$ where $x_0^i x_1^i$ is the lexicographically smallest blue edge adjacent to the starting point x_0^i of the circuit C_i, and where q_i is such that C_i has $q_i + 1$ edges (and where $x_0 = x_{q_i+1}$). For any i, we define the function

$$l_i(r) = \begin{cases} -1 & \text{if } \{x_{r-2}^i, x_{r-1}^i\} \text{ is cut edge and } \{x_{r-1}^i, x_r^i\} \text{ is internal edge,} \\ 1 & \text{if } \{x_{r-2}^i, x_{r-1}^i\} \text{ is internal edge and } \{x_{r-1}^i x_r^i\} \text{ is cut edge,} \\ 0 & \text{otherwise,} \end{cases}$$

for $r = 2, 4, \dots, q_i + 1$. The function l_i indicate what happens to the number of cut edges of a graphical realization when we perform a hinge flip on a pair of consecutive edges $\{x_{r-2}^i, x_{r-1}^i\}$ and $\{x_{r-1}^i, x_r^i\}$ on the circuit C_i.

Decomposition into segments. We subdivide every circuit C_i into a sequence of (not necessarily closed) walks of even length, called *sections*. Let $Z_i = \{r : l_i(r) \neq 0\} = \{z_1, \dots, z_{u_i}\} \subseteq \{2, 4, \dots, q_i+1\}$ be the set of indices that represent a change in cut edges along the circuit, where we assume that $z_1 \leq z_2 \leq \dots \leq z_{u_i}$. We define $C_i^1 = x_0^i x_1^i \dots x_{z_1}^i$ and $C_i^j = x_{z_j}^i \dots x_{z_{j+1}}^i$ for $j = 2, \dots, u_i - 1$. If $l_i(q_i + 1) \neq 0$ this procedure partitions the circuit C_i completely, with $C_i^{u_i}$ being the last section. Otherwise, we define $C_i^{u_1+1} = x_{z_{u_i}}^i \dots x_0^i$ as the final section, which is the remainder of the circuit C_i. We define U_i as the total number of obtained sections, which is either u_i or $u_i + 1$. Note that when $Z_i = \emptyset$, the whole circuit will form one section $C_i = C_i^1$. Also note that a section always starts with a blue edge. We extend the function l_i to sections in the following way:

$$l_i\left(C_i^j\right) = \sum_{r=2,4,\dots,z_j} l_i(r) = \begin{cases} -1 & \text{if } l_i(z_j) = -1, \\ 1 & \text{if } l_i(z_j) = 1, \\ 0 & \text{otherwise,} \end{cases}$$

for $j = 1, \dots, U_i$. Note that $l(C_i^j) \in \{-1, 1\}$ for $j = 1, \dots, u_i$, and zero for $j = u_i + 1$ if this term is present. An example is given in Figure 4.17.

We continue by grouping the union of all sections into segments in a similar flavor. For sake of readability, we rename the sections

$$C_1^1, \dots, C_1^{U_1}, C_2^1, \dots, C_2^{U_2}, \dots, C_s^1, \dots, C_s^{U_s}$$

as $D_1, \dots, D_U$ in the obvious way, where $U = \sum_{i=1}^s U_i$, and we define $l(D_k) = l_i(C_i^j)$ if C_i^j was renamed D_k. We define $W = \{k : l(D_k) \neq 0\} = \{w_1, \dots, w_B\}$ as the set of sections representing a change in cut edges along a circuit, where we assume that $w_1 \leq \dots \leq w_B$. We define the *segment* $S_1 = (D_1, \dots, D_{w_1})$, and $S_i = (D_{w_{i-1}+1}, \dots, D_{w_i})$ for $i = 2, \dots, w_B - 1$. If $l(D_U) \neq 0$, i.e., when $w_B = U$, this procedure completely groups the collection of sections into segments. Otherwise, we redefine the last segment as $S_B = (D_{w_{B-1}+1}, \dots, D_U)$. We can extend the function l to segments in the following way:

$$l(S_i) = \sum_{j=w_{i-1}+1}^{w_i} l(D_j) = \begin{cases} -1 & \text{if } l(D_{w_i}) = -1, \\ 1 & \text{if } l(D_{w_i}) = 1, \end{cases}$$

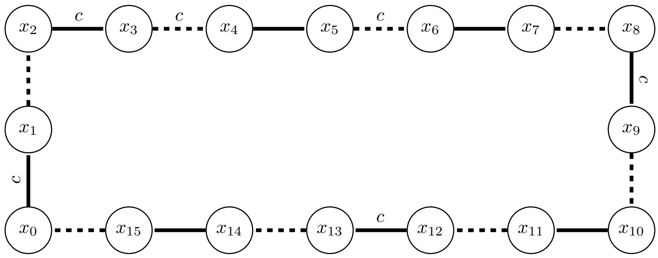

Figure 4.17: The circuit $C_1 = x_0x_1 \ldots x_{15}x_0$ with $q_1 = 15$. The blue edges are represented by the solid edges, and the red edges by the dashed edges. A label c on an edge indicates that it is a cut edge (all others are internal edges). We have $C_1^1 = x_0x_1x_2$ with $l_1(C_1^1) = -1$; $C_1^2 = x_2x_3x_4x_5x_6$ with $l_1(C_1^2) = 1$; $C_1^3 = x_6x_7x_8x_9x_{10}$ with $l_1(C_1^3) = -1$; $C_1^4 = x_{10}x_{11}x_{12}x_{13}x_{14}$ with $l_1(C_1^4) = -1$; and $C_1^5 = x_{14}x_{15}x_0$ with $l_1(C_1^5) = 0$ (note that $U_1 = 5$ in this example).

for $i = 1, \ldots, B-1$, and $l\,(S_B) = \sum_{j=w_{B-1}+1}^{U} l(D_j)$. Note that

$$l(S_i) \in \{-1, 1\} \quad \text{for } i = 1, \ldots, B, \tag{4.19}$$

unless in the special case that there is only one segment S_1 covering all circuits, then $l(S_1) = 0$. This happens, e.g., in the situation of the warm-up example.

An example of a decomposition into segments is given in Figures 4.20 and 4.21 later on. Roughly speaking, a segment is a maximal collection of edges that could be processed, using hinge flips operations as in the warm-up example, until the number of cut-edges changes. In particular, the first segment represents precisely the point up to where we could carry out the same processing steps as in the warm-up example until the number of cut edges will have changed for the first time. Note that a segment might contain sections from multiple circuits, in particular, it might consist of a final section of a circuit J_1, then some full circuits $J_2, \ldots, J_h$ (which all form a section on their own) and then the first section of some circuit J_{h+1}. The function l is then zero on the last section of J_1 and all circuits (sections) $J_2, \ldots, J_h$, and non-zero on the section of J_{h+1}.

Unwinding/rewinding of a segment. The *unwinding* of a section $D = x_f \ldots x_g$ consists of performing a number of hinge flip operations, that represent transitions in the Markov chain $\mathcal{M}'(\gamma, d)$. That is we perform a sequence of hinge flip operations replacing the (blue) edges $\{x_{r-2}, x_{r-1}\}$ by (red) edges $\{x_{r-1}, x_r\}$ for $r = f+2, \ldots, g$, in increasing order of r. Sometimes, we need to temporarily undo the unwinding of a section, in which case we perform a sequence of hinge flip operations replacing the (red) edges $\{x_{r-1}, x_r\}$ by (blue) edges $\{x_{r-2}, x_{r-1}\}$ for $r = f+2, \ldots, g$, in decreasing order of r this time. That is, we reverse the operations done during the unwinding. This is called *rewinding* a section. We say that a circuit is (currently) processed if all its sections have been unwound, and it is (currently) unprocessed if at least one section has not been unwound.

The unwinding of a segment $S_i = (D_{a_i}, \dots, D_{a_i+1})$ consists of unwinding the sections $D_{a_i}, \dots, D_{a_i+1}$ in increasing order. The rewinding of S_i consists of rewinding the section $D_{a_i}, \dots, D_{a_i+1}$ in decreasing order.

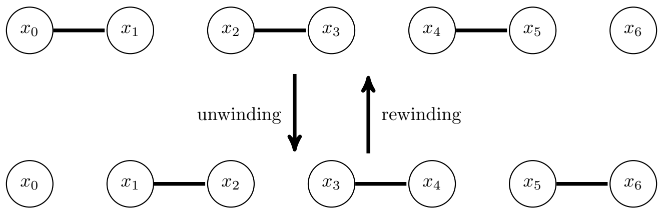

Figure 4.18: A section $D = x_0x_1 \dots x_6$. The blue edges are represented by the solid edges. The unwinding consists of performing first a hinge flip with $\{x_0, x_1\}$ to $\{x_1, x_2\}$; then $\{x_2, x_3\}$ to $\{x_3, x_4\}$; and finally $\{x_4, x_5\}$ to $\{x_5, x_6\}$. The rewinding consist of first a hinge flip with $\{x_5, x_6\}$ to $\{x_4, x_5\}$; then $\{x_3, x_4\}$ to $\{x_2, x_3\}$; and finally $\{x_1, x_2\}$ to $\{x_0, x_1\}$

Landscape processing. Remember that B is the number of segments obtained from the decomposition of circuits into segments. We define the function $P : \{0, 1, \dots, B\} \to \mathbb{Z}$ by $P(0) = 0$ and $P(i) = \sum_{j=1}^{i} l(S_j)$ for $i = 1, \dots, B$.

Lemma 4.29. *The function P is a landscape function.*

Proof. We have to check that $P(0) = P(B) = 0$ and that $|P(i+1) - P(i)| = 1$ for all $i = 0, \dots, B-1$, see the description of the mountain climbing problem. We have $P(0)$ by definition. Moreover, since both graphical realizations G and G' contain γ cut edges, it holds that $P(B) = \sum_{i=1}^{B} l(S_i) = 0$. Finally, using (4.19) and the definition of P, it follows that

$$|P(i+1) - P(i)| = \left| \sum_{j=1}^{i+1} l(S_j) - \sum_{j=1}^{i} l(S_j) \right| = |l(S_i)| = 1$$

for all $i = 1, \dots, B-1$. □

Based on the segments $S_1, \dots, S_B$, we define the canonical path from G to G' in the state space graph of the chain $\mathcal{G}'(\gamma, d)$ that replaces all the blue edges in $G \triangle G'$ with the red edges in $G \triangle G'$. By Lemma 4.29 we know P is a landscape function and therefore there is a unique decomposition into mountain and valley functions $P_1, \dots, P_p$ so that P is the landscape function for this collection, where every function is of the form $P_j : \{a_j, \dots, b_j\} \to \mathbb{Z}$ with $a_1 = 0, b_j = a_{j+1}$ for $j = 1, \dots, p-1$, and $b_p = B$.[26] The *processing of a mountain/valley P_j* means that all segments $S_{a_j+1}, \dots, S_{b_j}$ will be unwound (it might be that during this procedure

[26] The function P_1 can be found by determining the first $j > 0$ so that $P(j) = 0$. The sign of $P(1)$ determines if it is a mountain or a valley. The remaining mountains and valleys can be found similarly.

segments are temporarily rewound). This processing will rely on a traversal of the mountain, see Definition 4.27. We say that the segments $S_{a_j+1}, \ldots, S_{t_j}$ are on the *left side of the mountain*, and the segments $S_{t_j+1}, \ldots, S_{b_j}$ on the *right side of the mountain*, where t_j is the first top of the mountain. Let $P = P_j$ for some j and assume that P is a mountain function. For sake of notation, we write $a = a_j$ and $b = b_j$, and $t = t_j$.

Now, fix some traversal $(a, t) = (r_1, s_1), \ldots, (r_k, s_k) = (t, b)$ of P. For $c = 1, \ldots, k-1$ in increasing order, do the following:

1. if $r_{c+1} > r_c$ and $s_{c+1} > s_c$: first unwind segment $S_{r_{c+1}}$, then unwind segment $S_{s_{c+1}}$;
2. if $r_{c+1} > r_c$ and $s_{c+1} < s_c$: first unwind segment $S_{r_{c+1}}$, then rewind segment S_{s_c};
3. if $r_{c+1} < r_c$ and $s_{c+1} > s_c$: first rewind segment S_{r_c}, then unwind segment $S_{s_{c+1}}$;
4. if $r_{c+1} < r_c$ and $s_{c+1} < s_c$: first rewind segment S_{r_c}, then rewind segment S_{s_c}.

This describes the processing of a mountain based on a traversal. Note that after the processing of a mountain, indeed all its segments have been unwound (see also the example worked out in the Figures 4.20, 4.21, 4.22 and 4.23). If P is a valley function, we can use essentially the same procedure performed on $-P$. The *processing of a landscape* is done by processing the mountains/valleys $P_1, \ldots, P_p$ in increasing order.

This procedure generates a sequence $G = Z_1, Z_2, \ldots, Z_l = G'$ of graphical realizations transforming G into G' where any two consecutive realizations differ by a hinge flip operation. The following lemma shows that this sequence indeed defines a (canonical) path from G to G' in the state space graph of $\mathcal{M}(\gamma, d)$, for a given pairing ψ. This lemma is essentially the motivation for the definition of $\mathcal{G}'(\gamma, d)$.

Lemma 4.30. *Let $Z = Z_i$ be a graphical realization on the constructed path from G to G' for pairing ψ, with degree sequence d' and γ' cut edges. Then (γ', d') satisfies the properties (i), (ii) and (iii) defining $\mathcal{G}'(\gamma, d)$ (see Section 4.2.3).*

Moreover, there exists a polynomial $r(\cdot)$ such that the length of any constructed (canonical) path carrying flow is at most $r(n)$.

Proof. Since hinge flip operations never change the number of edges in a graph, property (i) is clearly satisfied. Since the operations (1)–(4) given above unwind and rewind at most two segments, and by construction of the trajectories describing the traversal, the property (ii) is also satisfied. Finally, the cases (1)–(4), in combination with the second property of a traversal as in Definition 4.27, guarantee that property (iii) is satisfied. To see that all canonical paths have polynomial length, note that the traversal has polynomial length, and also every individual segment has polynomial length. ☐

4.4.1.2 Encoding

We continue with defining the notion of an *encoding* that will be used in the next section to bound the congestion of an edge in the state space graph of $\mathcal{M}(\gamma, d)$. Let $\tau = (Z, Z')$ be a given transition of the Markov chain. Suppose that a canonical path from G to G' for some pairing $\psi \in \Psi(G, G')$, with canonical circuit decomposition $\{C_1, \dots, C_s\}$, uses the transition τ. We define $L_\tau(G, G') = (G \triangle G') \triangle Z$. An example is given in Figures 4.20, 4.21, 4.22 and 4.23.

Lemma 4.31. *Given $\tau = (Z, Z')$, ψ, L, if there is some pair (G, G') so that $L = L_\tau(G, G')$, then there are at most $\frac{1}{8}n^4$ such pairs.*

Proof. For any pair (G, G'), let P be the landscape function of this canonical path between G and G' using the transition τ, and $P_1, \dots, P_p$ its decomposition into mountain and valley functions. Let $T_{\tau,\psi}(G, G') \in \{C_1, \dots, C_s\}$ be the circuit containing the first node of the first segment of the right part of the mountain/valley P_j containing the transition τ. Without loss of generality, we assume that P_j is a mountain. Moreover, let Γ be the circuit containing the transition τ. If τ is used in the processing of a segment on the left side of the mountain P_j containing τ, let $\sigma_\psi(G, G')$ be the circuit containing the last node of the segment with highest index on the right side of the mountain that is currently unwound. If τ lies on the right side of the mountain, we let $\sigma_\psi(G, G')$ be the circuit containing the last node of the segment with highest index on the left side of the mountain that is currently unwound.

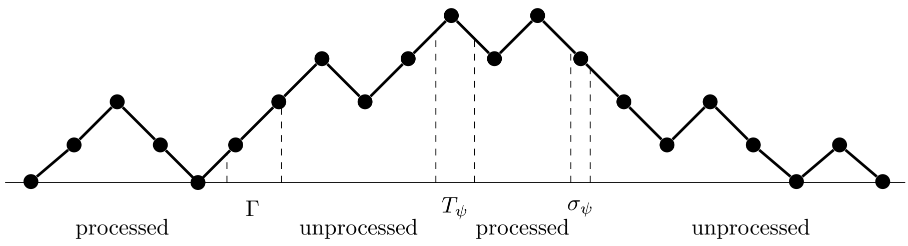

Figure 4.19: The dashed vertical lines sketch the ranges of the circuits T_ψ, σ_ψ and Γ. For every other circuit, contained in one of the four regions represented below the landscape, we know whether it has currently been processed or not.

We claim that, given $T_\psi, \sigma_\psi \in \{C_1, \dots, C_s\}$, it can be argued that there are at most 8 pairs (G, G') so that $T_\psi = T_\psi(G, G')$, $\sigma_\psi = \sigma_\psi(G, G')$. This can be seen as follows. Note that we can infer for all other circuits in $\{C_1, \dots, C_s\} \setminus \{T_\psi, \sigma_\psi, \Gamma\}$ which edges belong to G and which to G' using the (global) circuit ordering. To see this, assume that $\Gamma \preceq_C T_\psi \preceq_C \sigma_\psi$ (the only other case $\sigma_\psi \preceq_C T_\psi \preceq_C \Gamma$ is similar). Because the landscapes of the canonical paths always respect the circuit ordering, we know that all circuits in the canonical decomposition of ψ appearing before Γ have been unwound at this point. All circuits lying strictly between Γ and T_ψ are not unwound. The circuits strictly between T_ψ and σ_ψ again have been unwound, and finally, all circuits appearing after σ_ψ have not been unwound (see Figure 4.19). By comparison with Z, it is uniquely determined which edges on these circuits belong to G and which to G'. For the remaining three circuits T_ψ, σ_ψ and Γ there are for every circuit two possible configurations of the edges of G and G', since every circuit alternates between edges of G and G'.[27] Hence, there are at most $2^3 = 8$ possible pairs (G, G') with the desired properties given T_ψ and σ_ψ.

Finally, note that for any pairing ψ, there are at most $\frac{1}{4}\binom{n}{2}$ circuits in the canonical circuit decomposition $\{C_1, \dots, C_s\}$ of the pairing ψ, as every circuit has length at least four. Hence, for both T_ψ and σ_ψ there are at most $\frac{1}{4}\binom{n}{2}$ possible choices. Since Γ is uniquely determined by the transition τ, this implies that there are at most

$$8 \cdot \frac{1}{4}\binom{n}{2} \cdot \frac{1}{4}\binom{n}{2} \leq \frac{n^4}{8}$$

possible pairs (G, G') with $L = L_\tau(G, G')$.[28] □

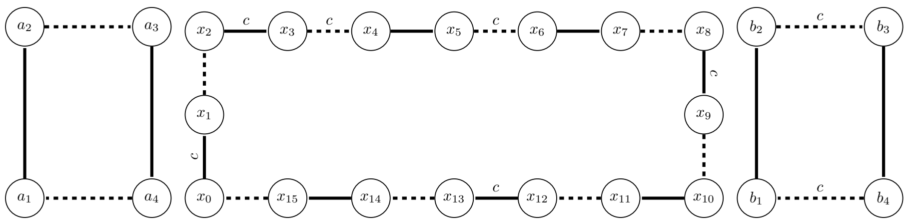

Figure 4.20: Symmetric difference $H = G \triangle G'$ where the solid edges represent the (blue) edges G and the dashed edges the (red) edges of G'. From left to right the circuit are numbered $C_1 = a_1a_2a_3a_4a_1$, $C_2 = x_0 \cdots x_{15}x_0$ and $C_3 = b_1b_2b_3b_4b_1$, and assume that this is also the order in which they are processed. Cut edges are indicated with the label c.

4.4.1.3 Bounding the congestion

For a tuple (G, G', ψ), let $p_\psi(G, G')$ denote the canonical path from G to G' for pairing ψ. Let

$$\mathcal{L}_\tau = \cup_{(G,G',\psi)\in\mathcal{F}_\tau} L_\tau(G, G')$$

be the union of all (distinct) encodings L_τ, where $\mathcal{F}_\tau = \{(G, G', \psi) : \tau \in p_\psi(G, G')\}$ is the set of all tuples (G, G', ψ) such that the canonical path from G to G' under pairing ψ uses the transition τ. A crucial observation is the following.

[27] Note that we cannot use the transition τ to infer which edges belong to G and G' on the circuit Γ, as we do not know (i.e., we do not encode) whether we are unwinding or rewinding the segment containing τ.

[28] A canonical path uses every transition at most once, which follows from the fact that we assumed that a traversal is always minimal, see Definition 4.27.

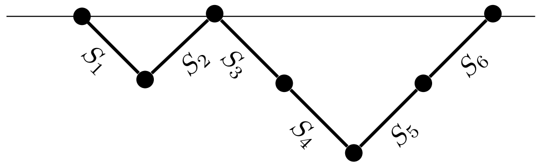

Figure 4.21: The landscape, consisting of two valleys, corresponding to the symmetric difference in Figure 4.20. The segments are given by $S_1 = (a_1a_2a_3a_4a_1, x_0x_1x_2)$, $S_2 = (x_2x_3x_4x_5x_6)$, $S_3 = (x_6x_7x_8x_9x_{10})$, $S_4 = (x_{10}x_{11}x_{12}x_{13}x_{14})$, $S_5 = (x_{14}x_{15}x_0, b_1b_2b_3)$, and $S_6 = (b_3b_4b_1)$.

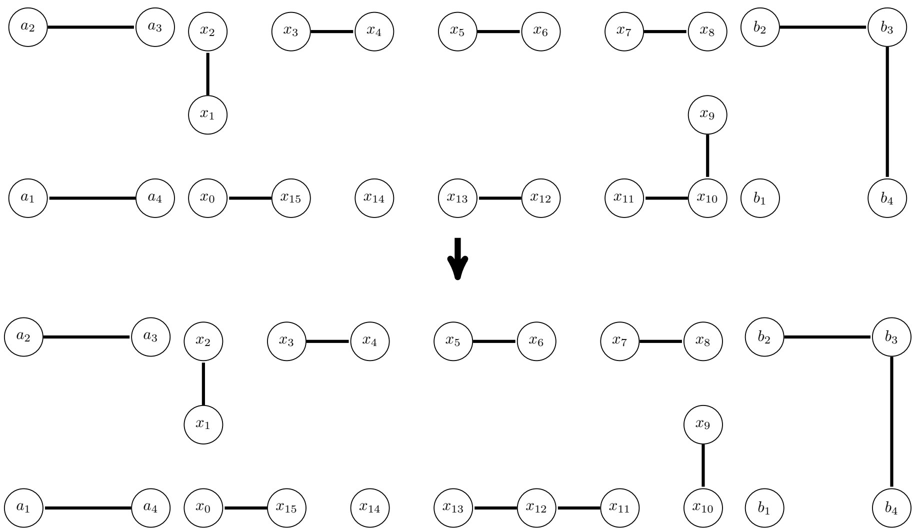

Figure 4.22: The transition $\tau = (Z, Z')$ that is the hinge flip operation that removes the edge $\{x_{10}, x_{11}\}$ and adds the edge $\{x_{11}, x_{12}\}$ as part of the unwinding of S_4. Note that the segments S_1 and as S_2, forming the first valley, have been processed already. Also, the first segment S_3 of the left part of the second valley, as well as the segment S_5 being the first segment of the right part of the second valley, have been processed already. The segment S_6 has not been processed yet. The edges in $(E(G) \cup E(G')) \setminus E(H)$ are left out.

Lemma 4.32. *If $L_\tau(G, G') = (G \triangle G') \triangle Z$ for transition $\tau = (Z, Z')$ used by a canonical path between G and G', then $L \in \mathcal{G}'(\gamma, d)$. This implies that*

$$|\mathcal{L}_\tau| \leq |\mathcal{G}'(\gamma, d)|. \tag{4.20}$$

Proof. We check that the properties (i), (ii) and (iii) defining the set $\mathcal{G}'(\gamma, d)$ (see Section 4.2.3) are satisfied by L. Note that $L \triangle Z = G \triangle G$. As every individual hinge flip operations adds and removes an arc from the symmetric difference, it follows that L and Z have the same number of edges. This proves property (i).

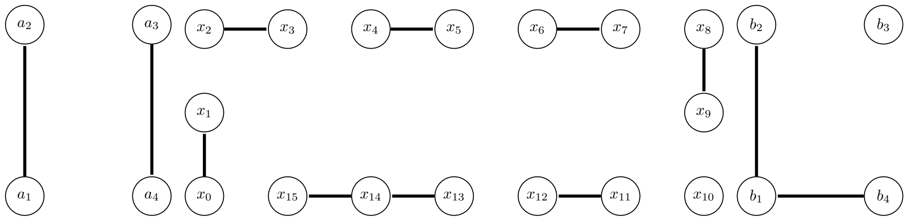

Figure 4.23: The encoding $L = L_t(G, G') = (G \triangle G') \triangle Z$ for the symmetric difference in Figure 4.20 and transition as in Figure 4.22, where again the edges in $(E(G) \cup E(G')) \setminus E(H)$ are left out.

Also, if Z has a perturbation of $\alpha_v \in \{-2, -1, 0, -1, -2\}$ (see Proposition 4.9) at node v, then L has a perturbation of $-\alpha_v$ at node v, which shows that property (ii) is satisfied for L. Finally, with $\beta \in \{-1, 0, 1\}$, if Z contains $\gamma - \beta$ cut edges, then L contains $\gamma + \beta$ cut edges (using implicitly that G and G' contain the same number of cut edges). This implies that property (iii) is satisfied. □

Moreover, with $H = G \triangle G'$ and $L = L_t(G, G')$, the pairing ψ has the property that it pairs up the edges of $E(H) \setminus E(L)$ and $E(H) \cap E(L)$ in such a way that for every node v each edge in $E(H) \setminus E(L)$ that is incident to v is paired up with an edge in $E(H) \cap E(L)$ that is incident to v, except for at most four pairs.[29] Let $\Psi'(L)$ be the set of all pairings with this property. Remember that we do not need to know G and G' in order to determine the set $H = L \triangle Z$. Note that not every pairing in $\Psi'(L)$ has to correspond to a tuple (G, G', ψ) for which $t \in p_\psi(G, G')$. Using a counting argument,[30] we can upper bound $|\Psi'(L)|$ in terms of $|\Psi(H)|$. In particular, there exists a polynomial $q(n)$ such that

$$|\Psi'(L)| \leq q(n) \cdot |\Psi(H)|.^{31} \tag{4.21}$$

Putting everything together, we have

$$\begin{aligned}
|\mathcal{G}'(\gamma, d)|^2 f'(\tau) &= \sum_{(G,G')} \sum_{\psi \in \Psi(G,G')} \mathbf{1}(e \in p_\psi(H)) |\Psi(H)|^{-1} \\
&\leq \frac{1}{8} n^4 \sum_{L \in \mathcal{L}_\tau} \sum_{\psi' \in \Psi'(L)} |\Psi(H)|^{-1} \qquad \text{(using Lemma 4.21)} \\
&\leq \frac{1}{8} n^4 \cdot q(n) \sum_{L \in \mathcal{L}_\tau} 1 \qquad \text{(using (4.21))} \\
&\leq \frac{1}{8} n^4 \cdot q(n) \cdot |\mathcal{G}'(\gamma, d)| \qquad \text{(using (4.20))} \qquad (4.22)
\end{aligned}$$

[29] These are the nodes x_{10}, x_{14}, b_1 and b_3 in Figure 4.23.
[30] This can be done similarly as the argument used in Section 4.3.1.3.
[31] A very rough choice is $q(n) = n^{20}$.

The usage of Lemma 4.31 for the first inequality works as follows. Every tuple $(G, G', \psi) \in \mathcal{F}_t$ with encoding $L_t(G, G')$ generates a unique tuple in $\{L_t(G, G')\} \times \Psi'(L_t(G, G'))$. But since, by Lemma 4.31, there are at most $\frac{1}{8}n^4$ pairs (G, G') with $L = L_\tau(G, G')$ for given L, τ and ψ, we have that $\frac{1}{8}n^4 \sum_{L \in \mathcal{L}_\tau} |\{L\} \times \Psi'(L)| = \frac{1}{8}n^4 \sum_{L \in \mathcal{L}_\tau} \sum_{\psi' \in \Psi'(L)} 1$ is an upper bound on the number of canonical paths that use τ.

By rearranging (4.22) we get the upper bound for f' required in Lemma 4.26. We already observed that the length of any canonical path is polynomially bounded as well. This then completes the proof of Theorem 4.25. □

4.4.2 Strong stability of 2-class JDM instances

In Section 4.4.1 we have shown that the hinge flip Markov chain for PAM instances with two classes is rapidly mixing on $\mathcal{G}'(\gamma, d)$ in case (γ, d) comes from a family of strongly stable tuples. In this section we show that JDM instances with two degree classes are strongly stable. When dealing with a family of instances, even when this is not explicitly mentioned, we only consider the tuples (c, d) for which there is at least one graphical realization.

Theorem 4.33. *Let $\mathcal{D}$ be the family of instances of the joint degree matrix model, i.e., where for every tuple (V_1, V_2, γ, d) it holds that $1 \leq \beta_1, \beta_2 \leq |V| - 1$, and $1 \leq \gamma \leq |V_1||V_2| - 1$, where β_1 and β_2 are the common degrees in the classes V_1 and V_2, respectively. The family $\mathcal{D}$ is strongly stable for $k = 7$, and, hence, the hinge flip chain is rapidly mixing for all tuples in $\mathcal{D}$.*

Proof. We first show that this family is strongly stable for $k = 7$. For convenience, we will work with the notation $\mathcal{G}'(c, d)$ instead of $\mathcal{G}'(\gamma, d)$. Remember that

$$c_{ii} = \left(\sum_{j \in V_i} d_j \right) - \gamma$$

for $i = 1, 2$ is the number of internal edges that V_i has in any graphical realization in $\mathcal{G}(\gamma, d)$, and that $\gamma = c_{12} = c_{21}$. For sake of readability, we define the notion of a cancellation hinge flip. For either $i = 1$ or $i = 2$, suppose nodes $v, w \in V_i$, are such that v has a degree deficit of at least one, and w a degree surplus of at least one. Then w has a neighbor $z \in V$ that is not a neighbor of v (using that v and w have the same degree β_i). The hinge flip operation that removes the edge $\{z, w\}$ and adds the edge $\{z, v\}$ is called a *cancellation flip on v and w*. Note that the number of internal edges in V_1 and V_2 as well as the number of cut edges does not change with such an operation.[32] Moreover, we say that an edge $\{a, b\}$ is a non-edge of a graphical realization G if $\{a, b\} \notin E(G)$.

[32] That is, either z lies in the other class, in which case the cancellation flip removes and adds a cut edge, or, z lies in the same class as v and w in which case an internal edge in V_i is removed and added.

Let $G \in \mathcal{G}'(c,d)$ for some tuple (c',d') as in the definition of $\mathcal{G}'(c,d)$ at the start of Section 4.4.1. We first show that with at most four hinge flip operations, we can obtain a perturbed auxiliary state $G^* \in \mathcal{G}'(c,d)$ for which its tuple (c^*,d^*) is edge-balanced. That is, it satisfies $c^* = c$. Remember that the value c'_{12} uniquely determines the matrix c', and, by assumption of $\mathcal{G}'(c,d)$, we have $c'_{12} \in \{c_{12}-1, c_{12}, c_{12}+1\}$. We can therefore distinguish the following cases.

- **Case 1:** $c'_{12} = c_{12}+1$. Then, by Proposition 4.9, either $c'_{11} = c_{11}-1$ and $c'_{22} = c_{22}$, or, $c'_{22} = c_{22}-1$ and $c'_{11} = c_{11}$. Assume without loss of generality that we are in the first case. Then it holds that

$$\sum_{j \in V_1} d'_j = \left(\sum_{j \in V_1} d_j\right) - 1 \quad \text{and} \quad \sum_{j \in V_2} d'_j = \left(\sum_{j \in V_2} d_j\right) + 1. \tag{4.23}$$

  Moreover, there must be at least one node $v_2 \in V_2$ with a degree surplus (of either one or two), and there is at least one non-edge $\{a,b\}$ with both endpoints in V_1. If v_2 is adjacent to either a or b, we can perform a hinge flip to make the realization G edge-balanced, so assume this is not the case. Also, if the total deficit of a and b is -2, there must be a node in V_1 with degree surplus, otherwise (4.23) is violated. Then we can perform a cancellation flip in V_1 to remove the deficit at either a or b. Hence, we may assume without loss of generality that a does not have a degree deficit.

  - **Case A:** v_2 **has a neighbor** $v_1 \in V_1$**.** If v_1 has a degree surplus we can perform a cancellation flip in V_1 to remove it, which must exist by (4.23). So assume v_1 has no degree surplus. As node a has no deficit, and is not connected to v_2, whereas v_1 is, there must be some neighbor p of a which is not a neighbor of v_1. This holds since v_1 and b have the same degree β_1 in the sequence d. Then the path $v_2 - v_1 - p - a - b$ alternates between edges and non-edges of G, and with two hinge flips we can obtain an edge-balanced realization in $\mathcal{G}'(\gamma,d)$.

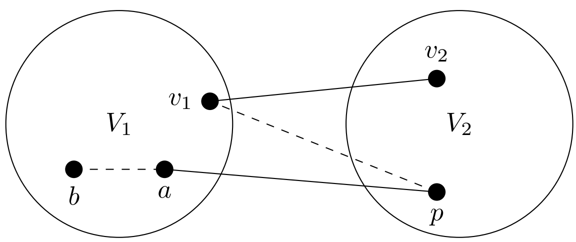

Figure 4.24: Sketch of first case with subcase A.

  - **Case B:** v_2 **has no neighbors in** V_1**.** We know that there is at least one cut edge $\{q,r\}$, with $q \in V_2$ and $r \in V_1$, in the realization G, since $c'_{12} = c_{12}+1$. If q has a degree surplus, we are in the situation of

Case A. Otherwise v_2 has a neighbor u which is not a neighbor of q, since q and v_2 have the same degree β_2 in the sequence d. We can then perform the hinge flip that removes $\{v_2, u\}$ and adds $\{u, q\}$. If q now has a degree surplus, we are in Case A. Otherwise, in case this hinge flip cancelled out a degree deficit at q, there must be at least one other node in V_2 with a degree surplus, because of (4.23). We can then perform the same step again, which will now result in a degree surplus at q. This is true since the node q cannot have a deficit of -2, since (4.23) would then imply that the the total degree surplus of nodes in V_2 is at least three, which violates the second property defining $\mathcal{G}'(c, d)$. That is, we can always reduce to the situation in Case A.

Summarizing, we can always find an edge-balanced realization G^* using at most four hinge flip operations in case $c'_{12} = c_{12} + 1$.

- **Case 2:** $c'_{12} = c_{12} - 1$. Using complementation, it can be seen that this case is similar to Case 1. That is, we consider the tuple $(\bar{c}, \bar{d})$ in which all nodes in V_1 have degree $|V| - \beta_1$, all nodes in V_2 have degree $|V| - \beta_2$, and where all feasible graphical realizations have $\bar{c}_{12} = |V_1||V_2| - c_{12}$ cut edges. The case $c_{12} = c_{12} - 1$ then corresponds to the case $\bar{c}'_{12} = \bar{c}_{12} + 1$.

- **Case 3:** $c'_{12} = c_{12}$. If also $c_{11} = c'_{11}$ we are done. Otherwise, suppose that $c_{11} = c'_{11} + 1$. Then it must be that $c_{22} = c'_{22} - 1$, as $c'_{12} = c_{12}$, and it holds that

$$\sum_{j \in V_1} d'_j = \left(\sum_{j \in V_1} d_j \right) + 2 \quad \text{and} \quad \sum_{j \in V_2} d'_j = \left(\sum_{j \in V_2} d_j \right) - 2. \tag{4.24}$$

Then there is at least one edge $\{a, b\}$ in the graphical realization with $a, b \in V_1$. Moreover, we may assume that a has a degree surplus. If not, then there is at least one other node u with a degree surplus because of (4.24). Performing a cancellation flip then gives the node a a degree surplus (it could not be that a had a degree deficit, as this would imply, in combination with (4.24), that the total degree surplus of nodes in V_1 is at least three).

Now, if there is a non-edge of the form $\{b, v_2\}$ for some $v_2 \in V_2$, we can perform a hinge flip operation removing $\{a, b\}$ and adding $\{b, v_2\}$ in order to end up in Case 1. Otherwise, assume that b is adjacent to all $v_2 \in V_2$. As b is also adjacent to a, and a has a degree surplus of at least one,[33] it follows that $\beta_1 \geq |V_2|$. Now, by the assumption that $c_{12} \leq |V_1||V_2| - 1$, there is at least one non-edge $\{p, q\}$ with $p \in V_1$ and $q \in V_2$. As p is not adjacent to q, but has degree at least $\beta_1 \geq |V_2|$, it must be that p is adjacent

[33] That is, b can have a degree surplus of at most one. A degree surplus of two at b would only give a bound of $\beta_1 \geq |V_2| - 1$.

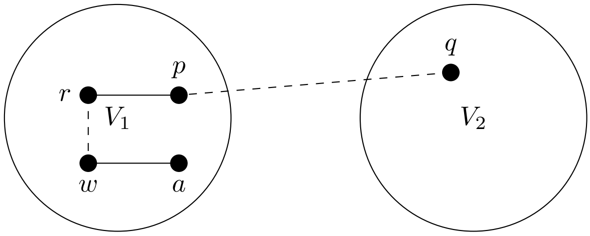

Figure 4.25: Sketch of last situation in Case 3.

to some $r \in V_1$. If r has a degree surplus, then we can perform a hinge flip that removes $\{p, r\}$ and adds $\{p, q\}$ in order to end up in the situation of Case 1. Otherwise, node a, which has a degree surplus, has some neighbor w which is not a neighbor of r. This implies the path $a - w - r - p - q$ alternates between edges and non-edges of G. Performing two hinge flips then brings us in the situation of Case 1.

We have shown that with at most five hinge flips we can always obtain some $G^* \in \mathcal{G}'(c, d)$ that is edge-balanced. This implies that

$$\sum_{j \in V_1} d_j^* = \sum_{j \in V_1} d_j \quad \text{and} \quad \sum_{j \in V_2} d_j^* = \sum_{j \in V_2} d_j. \tag{4.25}$$

Now, if $v \in V_1$ has a degree surplus, there must be some $w \in V_1$ that has a degree deficit, because of (4.25). We can then perform a cancellation flip to decrease the sum of the total degree deficit and degree surplus. A similar statement is true if $v \in V_2$ has a degree surplus. By performing this step at most twice, we obtain a realization $H \in \mathcal{G}(c, d)$. That is, with at most seven hinge flip operations in total we can transform G into a graphical realization in $\mathcal{G}(c, d)$. This shows that $\mathcal{D}$ is strongly stable for $k = 7$. $\square$

4.4.3 Rapid mixing of the switch chain

In this section we will use an embedding argument similar to that in the proof of Theorem 4.22 to show that the restricted switch chain is rapidly mixing in case both classes are regular, i.e., for instances that are essentially JDM instances with two degree classes.

While the restricted switch chain is known to be irreducible for the instances of the JDM model [5, 49], in general this is not true [68]. To the best of our knowledge, there is no clear understanding for which pairs c and d it is irreducible in general. Nevertheless, we present the following meta-result for the rapid mixing of the switch chain, which in particular applies in case the degrees are component-wise regular (Theorem 4.24).

Theorem 4.34. *Let $\mathcal{D}$ be a strongly stable family of tuples (γ, d) with respect to some constant k, and suppose there exists a function $p_0 : \mathbb{N} \to \mathbb{N}$ with the property that, for any fixed $x \in \mathbb{N}$: if $(\gamma, d) \in \mathcal{D}$, and $G, G' \in \mathcal{G}(\gamma, d)$ so that $|E(G) \triangle E(G')| \leq x$, the switch-distance satisfies $dist_{\mathcal{G}(\gamma,d)}(G, G') \leq p_0(x)$. Then the switch chain is rapidly mixing for all tuples in the family $\mathcal{D}$ with respect to the uniform stationary distribution over $\mathcal{G}(\gamma, d)$.*

Proof. First note that by definition of the function p_0 the switch chain is irreducible. Moreover, it is not hard to see that the switch chain is aperiodic and symmetric as well. This implies that it has a unique stationary distribution which is the uniform distribution over $\mathcal{G}(\gamma, d)$. Moreover, by assumption of strong stability, we know that the hinge flip chain $\mathcal{M}(\gamma, d)$ is rapidly mixing. In particular, from the proof of Theorem 4.25, we know there exists a flow f' that routes $1/|\mathcal{G}'(\gamma, d)|^2$ units of flow between any pair of states in $\mathcal{G}(\gamma, d)$ in the state space graph of the chain $\mathcal{M}(\gamma, d)$, with the property that $f'(e) \leq p(n)/|\mathcal{G}'(\gamma, d)|$, and $\ell(f') \leq q(n)$, for some polynomials $p(\cdot), q(\cdot)$ whose degrees may only depend on $k(\gamma, d)$.

One can then use exactly the same embedding argument as in the proof of Theorem 4.22. The existence of the function p_0, together with the notion of strong stability as defined in Section 4.2.3, are sufficient for reproducing all the arguments. □

We end this section with the proof of Theorem 4.24.

Proof of Theorem 4.24. Strong stability was shown in the previous section in Theorem 4.33. Moreover, from the proof of Lemma 7 in [5] it follows that for any two graphs $H, H' \in \mathcal{G}(\gamma, d)$, H can be transformed into H' using at most $\frac{3}{2}|E(H) \triangle E(H')|$ switches of the restricted switch chain. That is, we may take $p_0(x) = \frac{3}{2}x$. Then the statement follows from Theorem 4.34. □

4.5 Curveball chain

The main result of this section is to show that the spectral gaps of the KTV switch chain and the curveball chain are equivalent up to polynomial factors, see Theorem 4.35 below. The transition matrices P_{KTV} and P_C are specified later on in this section.

Theorem 4.35. *Let $r = (r_1, \ldots, r_n)$ and $c = (c_1, \ldots, c_m)$ be given marginals with $n \geq 3$, $\mathcal{F}$ a set of forbidden entries, and assume that $\Omega(r, c, \mathcal{F}) \neq \emptyset$. Let P_C and P_{KTV} be the transition matrices of respectively the curveball and KTV switch Markov chains. Then*

$$\frac{2}{n(n-1)} \cdot (1 - \lambda_*^{KTV})^{-1} \leq (1 - \lambda_*^C)^{-1} \leq \min\left\{1, \frac{(2r_{\max} + 1)^2}{2n(n-1)}\right\} \cdot (1 - \lambda_*^{KTV})^{-1},$$

where $\lambda_*^{KTV} (= \lambda_1^{KTV})$ is the second largest eigenvalue of P_{KTV}, and $\lambda_*^C (= \lambda_1^C)$ that of P_C.

In particular, Theorem 4.35 implies that the curveball chain is rapidly mixing whenever the switch chain is, and vice versa.

Proof overview. We first present a general comparison framework in Section 4.5.1, that compares a Markov chain with a locally refined version. We then prove Theorem 4.35 in Section 4.5.2 as the first application of this framework. We also provide a second application in Section 4.5.3, in the form of a comparison between the curveball chain and the *global curveball chain* where multiple binomial trades are performed in parallel [25].

4.5.1 Comparison framework

In this section we describe the comparison framework that will be used to compare the KTV switch and curveball Markov chains (Section 4.5.2), and to compare the curveball and the global curveball chain (Section 4.5.3). In general, we consider an ergodic (irreducible) Markov chain $\mathcal{M} = (\Omega, P)$ with stationary distribution π, being strictly positive for all $x \in \Omega$, that can be decomposed as[34]

$$P = \sum_{a\in\mathcal{L}} \rho(a) \sum_{R\in\mathcal{R}_a} P_R \tag{4.26}$$

which is given by a

1. finite index set $\mathcal{L}$, and probability distribution ρ over $\mathcal{L}$,
2. partition $\mathcal{R}_a = \cup R_{\ell,a}$ of Ω for $a \in \mathcal{L}$,

and where the restriction of a matrix P_R to the rows and columns of $R = R_{\ell,a}$ defines the transition matrix of an ergodic, time-reversible Markov chain on R (and is zero elsewhere), with stationary distribution $\tilde{\pi}_R(x) = \pi(x)/\pi(R)$ for $x \in R$. We use $1 = \lambda_0^R \geq \lambda_1^R \geq \cdots \geq \lambda_{|R|-1}^R$ to denote its eigenvalues. Note that these are also eigenvalues of P_R and that all other eigenvalues of P_R are zeros (as all rows and columns not corresponding to elements in R only contain zeros). The chain $\mathcal{M}$ proceeds by drawing an index a from the set $\mathcal{L}$, and then performs a transition in the Markov chain on the set R that the current state is in.

The *heat-bath variant* $\mathcal{M}_{heat}$ of the chain $\mathcal{M}$ is given by the transition matrix

$$P_{heat} = \sum_{a\in\mathcal{L}} \rho(a) \sum_{R\in\mathcal{R}_a} \mathbf{1} \cdot \sigma_R \tag{4.27}$$

where σ_R is the row-vector given by $\sigma_R(x) = \tilde{\pi}_R(x)$ if $x \in R$ and zero otherwise, and $\mathbf{1}$ the all-ones column vector. Intuitively, the chain $\mathcal{M}_{heat}$ proceeds by drawing an index a from $\mathcal{L}$, and then drawing a state $x \in R$ with probability $\tilde{\pi}_R(x)$. It can be shown that $\mathcal{M}_{heat}$ is an ergodic Markov chain whenever $\mathcal{M}$ is

[34]This description is almost the same as that of a heat-bath chain [61], and is introduced to illustrate the conceptual idea.

ergodic, as the state space graph of $\mathcal{M}$ is a subgraph of the state space graph of $\mathcal{M}_{heat}$. It is reversible by construction [61]. The eigenvalues of P_{heat} are always non-negative as was shown in [61].

Theorem 4.36. *Let $\mathcal{M} = (\Omega, P)$ be a Markov chain as in (4.26) with the property that $\lambda_0^R, \dots, \lambda_{|R|-1}^R \geq 0$ for all $a \in \mathcal{L}$ and $R \in \mathcal{R}_a$. Let $\mathcal{M}_{heat} = (\Omega, P_{heat})$ be its heat-bath variant as in (4.27) and let α and β be constants with $\alpha\beta > 0$. If*

$$\alpha - \beta(1 - \lambda_i^R) \geq 0 \tag{4.28}$$

for all $a \in \mathcal{L}$ and $R \in \mathcal{R}_a$ and $i \in \{1, \dots, |R| - 1\}$, then P only has non-negative eigenvalues and

$$\frac{1}{\alpha} \frac{1}{1 - \lambda_*^{heat}} \leq \frac{1}{\beta} \frac{1}{1 - \lambda_*}, \tag{4.29}$$

where λ_* (resp. λ_*^{heat}) is the second largest eigenvalue of P (resp. P_{heat}). For $\alpha = \beta = 1$, we find $(1 - \lambda_*^{heat})^{-1} \leq (1 - \lambda_*)^{-1}$.

The intuition behind Theorem 4.36 is that in order to compare the relaxation times of a Markov chain and its heat-bath variant, it suffices to compare them locally on the sets R. Note that α and β can both be negative, so that this statement can be used both to upper bound and lower bound the relaxation time of the heat-bath variant in terms of the original relaxation time.

We will use Propositions 4.37 and 4.38 in the proof of Theorem 4.36. Our proof makes use of positive semidefinite matrices; a symmetric real-valued matrix A is *positive semidefinite* if all its eigenvalues are non-negative, this is denoted by $A \succeq 0$.

Proposition 4.37 ([177])**.** *Let X, Y be symmetric $\ell \times \ell$ matrices. If $X - Y \succeq 0$, then $\lambda_i(X) \geq \lambda_i(Y)$ for $i = 1, \dots, \ell$, where $\lambda_i(C)$ is the i-th largest eigenvalue of $C = X, Y$.*

Proposition 4.38. *Let X be the $k \times k$ transition matrix of an ergodic reversible Markov chain with stationary distribution π, and eigenvalues $1 = \lambda_0 > \lambda_1 \geq \dots \geq \lambda_{k-1}$. Let $X^* = \lim_{t\to\infty} X^t$ be the matrix containing the row vector π on every row. Then the eigenvalues of $\alpha(I - X^*) - \beta(I - X)$ are*

$$\{0\} \cup \{\alpha - \beta(1 - \lambda_i) \mid i = 1, \dots, k - 1\}.$$

for given constants α and β.

Proof. Since X is the transition matrix of a reversible Markov chain, it holds that the matrix VXV^{-1} is symmetric[35], where $V = \mathrm{diag}(\pi_1^{1/2}, \pi_2^{1/2}, \dots, \pi_k^{1/2}) =$

[35] This is the same argument that is used to show that a reversible Markov chain only has real eigenvalues.

$\text{diag}(\sqrt{\pi})$. Using the fact that similar[36] matrices have the same set of eigenvalues we determine the eigenvalues of $\alpha(I - X^*) - \beta(I - X)$ by finding those of

$$V(\alpha(I - X^*) - \beta(I - X))V^{-1} = \alpha(I - \sqrt{\pi}^T\sqrt{\pi}) - \beta(I - VXV^{-1}).$$

Let $\mathbf{1} = (1, 1, 1, \dots, 1)^T$ denote the all-ones vector. We find

$$VXV^{-1}\sqrt{\pi}^T = VX\mathbf{1} = V\mathbf{1} = \sqrt{\pi}^T,$$

so that $\sqrt{\pi}^T$ is an eigenvector of VXV^{-1} with eigenvalue 1. It then follows that $\sqrt{\pi}^T$ is an eigenvector of $\alpha(I - \sqrt{\pi}^T\sqrt{\pi}) - \beta(I - VXV^{-1})$ with eigenvalue 0. Let $\sqrt{\pi}^T = w_0, w_1, \dots, w_{k-1}$ be a basis of orthogonal eigenvectors for VXV^{-1} corresponding to eigenvalues $1, \lambda_1, \dots, \lambda_{k-1}$ (note that X and VXV^{-1} have the same eigenvalues). It then follows that

$$[\alpha(I - \sqrt{\pi}^T\sqrt{\pi}) - \beta(I - VXV^{-1})]w_i = (\alpha - \beta(1 - \lambda_i))\, w_i$$

because of orthogonality. This completes the proof. □

Proof of Theorem 4.36. We first show that all eigenvalues of P are non-negative. Let D be the $|\Omega| \times |\Omega|$ diagonal matrix with $(D)_{xx} = \sqrt{\pi(x)}$. Note that the matrices $D^{-1}P_R D$ are positive semidefinite: they are symmetric because P_R defines a reversible Markov chain on R. The eigenvalues of $D^{-1}P_R D$ are equal to those of P_R, which are all non-negative by assumption. Any non-negative linear combination of positive semidefinite matrices is again positive semidefinite, hence $D^{-1}PD \succeq 0$. Thus, P has non-negative eigenvalues. A similar argument holds for P_{heat} and was shown in [61]. In particular, this implies that $\lambda_* = \lambda_1$ and $\lambda_*^{heat} = \lambda_1^{heat}$.

Let

$$Y_R := D^{-1}[\alpha(I_R - \mathbf{1} \cdot \sigma_R) - \beta(I_R - P_R)]D$$

where I_R is defined by $I_R(x, y) = 1$ if $x = y \in R$ and $I_R(x, y) = 0$ otherwise. The matrix Y_R is symmetric since the matrices $\mathbf{1} \cdot \sigma_R$ and P_R define reversible Markov chains on R. Furthermore its eigenvalues are $\{0\} \cup \{\alpha - \beta(1 - \lambda_i) \mid i = 1, \dots, k-1\}$ by Proposition 4.38 and the fact that similar matrices have the same set of eigenvalues. These eigenvalues are non-negative by assumption, hence Y_R is positive semidefinite. It then follows that the matrix

$$D^{-1}[\alpha(I - P_{heat}) - \beta(I - P)]D = \sum_{a \in \mathcal{L}} \rho(a) \sum_{R \in \mathcal{R}_a} D^{-1}[\alpha(I_R - \mathbf{1} \cdot \sigma_R) - \beta(I_R - P_R)]D$$

is also positive semidefinite. Using Proposition 4.37 and again the fact that similar matrices have the same set of eigenvalues, it follows that

$$\alpha(1 - \lambda_i^{heat}) \geq \beta(1 - \lambda_i)$$

which finishes the proof. □

[36] Two square matrices A and B are *similar* if there exists an invertible matrix T such that $A = T^{-1}BT$.

4.5.2 Comparing the switch and curveball chain

In order to prove Theorem 4.35, we give a novel decomposition of the state space of the KTV switch chain. We then show that the curveball chain is its heat-bath variant. In fact, we introduce a general γ-switch chain, as there are multiple switch-based chains in the literature, and show that the curveball chain is the heat-bath variant of this general switch chain. The KTV switch chain corresponds to a specific choice of γ.[37]

Definition 4.39 (γ-switch chain)**.** Let γ be such that

$$1 - u_{ij}(A)\ell_{ij}(A) \cdot \gamma > 0 \tag{4.30}$$

for all $A \in \Omega = \Omega(r, c, \mathcal{F})$ and $1 \leq i < j \leq m$. The transition matrix of the γ-switch chain on state space Ω is given by

$$P_\gamma(A, B) = \begin{cases} \binom{m}{2}^{-1} \cdot \gamma & \text{if } A \neq B \text{ are switch adjacent,} \\ \binom{m}{2}^{-1} \sum_{1 \leq i < j \leq m} (1 - u_{ij}(A)\ell_{ij}(A) \cdot \gamma) & \text{if } A = B, \\ 0 & \text{otherwise.} \end{cases}$$

Note that the transition probability for switch-adjacent matrices is the same everywhere in the state space, and does not depend on the matrices A and B. In particular, the transition matrix P_γ is symmetric and hence the chain is reversible with respect to the uniform distribution. The factor $2/(m(m-1))$ is included for notational convenience. The γ-switch chain can roughly be interpreted as follows. We first choose two distinct rows i and j uniformly at random, and then transition to a different matrix switch-adjacent for rows i and j, of which there are $u_{ij}\ell_{ij}$ possibilities and where every matrix has probability γ of being chosen; with probability $1 - u_{ij}\ell_{ij}\gamma$ we do nothing. Taking $\gamma = 2/(n(n-1))$ we obtain the KTV switch chain [113].

We continue with the description of the transition probabilities of the curveball Markov chain $\mathcal{M}_C = (\Omega, P_C)$, where $\Omega = \Omega(r, c, \mathcal{F})$. We have

$$P_C(A, B) = \begin{cases} \binom{m}{2}^{-1} \cdot \binom{u_{ij}+\ell_{ij}}{u_{ij}}^{-1} & \text{if } A \neq B \text{ are trade adjacent,} \\ \binom{m}{2}^{-1} \sum_{1 \leq i < j \leq m} \binom{u_{ij}+\ell_{ij}}{u_{ij}}^{-1} & \text{if } A = B, \\ 0 & \text{otherwise.} \end{cases}$$

State space decomposition. We next explain how the switch and curveball chain fit in the comparison framework by giving a suitable state space decomposition of the γ-switch chain. The index set $\mathcal{L} = \{(i, j) : 1 \leq i < j \leq m\}$ is the set of all pairs of distinct rows, and ρ is the uniform distribution over $\mathcal{L}$, that is, $\rho(i, j) = \binom{m}{2}^{-1}$ for all $(i, j) \in \mathcal{L}$. The partitions $\mathcal{R}_{(i,j)}$ for $(i, j) \in \mathcal{L}$ are based on the notion of a binomial neighborhood, as defined in [174].

[37] In [26] we compare the curveball chain with another switch-based chain for a different value of γ.

Definition 4.40 (Binomial neighborhood)**.** For a fixed binary matrix A and row-pair (i,j), the (i,j)-binomial neighborhood $\mathcal{N}_{ij}(A)$ of A is the set of matrices that can be reached by only applying switches on rows i and j. That is, $\mathcal{N}_{ij}(A)$ contains all matrices that are trade adjacent to A for rows i and j. Note that for $B \in \mathcal{N}_{ij}(A)$ we have $U_{ij}(A) \cup L_{ij}(A) = U_{ij}(B) \cup L_{ij}(B)$ and furthermore $u_{ij}(A) = u_{ij}(B)$ and $\ell_{ij}(A) = \ell_{ij}(B)$.

Next we will discuss the structure and properties of these binomial neighborhoods. This discussion will culminate into Theorem 4.43 describing the switch and curveball chains as being of the forms (4.26) and (4.27). Note that we have $A \in \mathcal{N}_{ij}(A)$; if $B \in \mathcal{N}_{ij}(A)$, then $A \in \mathcal{N}_{ij}(B)$ [174]; and, if $A \in \mathcal{N}_{ij}(B)$, $B \in \mathcal{N}_{ij}(C)$, then $A \in \mathcal{N}_{ij}(C)$. That is, the relation $\sim_{ij}$ defined by $a \sim_{ij} b$ if and only if $a \in \mathcal{N}_{ij}(b)$, is an equivalence relation on Ω. The equivalence classes of $\sim_{ij}$ define the sets $\mathcal{R}_{(i,j)}$.

Furthermore, two matrices $A, B \in \Omega$ can be part of *at most* one common binomial neighborhood. This follows directly from the observation that if $B \in \mathcal{N}_{ij}(A) \setminus \{A\}$, then A and B differ on precisely rows i and j, so switches using any other pair of rows $\{k,\ell\} \neq \{i,j\}$ can never transform A into B, see [174]. Finally, since $u_{ij}(A) = u_{ij}(B)$ and $\ell_{ij}(A) = \ell_{ij}(B)$ when A and B are part of the same binomial neighborhood, these numbers are only neighborhood-dependent, and not element-dependent within a fixed neighborhood. Observe that

$$|\mathcal{N}_{ij}| = \binom{u_{ij} + \ell_{ij}}{u_{ij}}.$$

A crucial observation now is that the undirected state space graph H of the γ-switch chain induced on a binomial neighborhood $\mathcal{N}_{ij}$ is isomorphic to a Johnson graph $J(u_{ij} + \ell_{ij}, u_{ij})$ whenever $u_{ij}, \ell_{ij} \geq 1$ (see Section 4.2.5 for notation and definition).[38] To see this, note that every element in the binomial neighborhood $\mathcal{N}_{ij}(A)$ can be represented by the set of indices $U_{ij}(A)$. The set $\{1, \dots, \ell_{ij} + u_{ij}\}$ here is then the set of indices of $U_{ij}(A) \cup L_{ij}(A)$. Indeed, matrices $A \neq B$ are switch-adjacent for rows i and j if $U_{ij}(A) \cap U_{ij}(B) = u_{ij} - 1$.

Example 4.41. Consider the binary matrix

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

and the 2×7-submatrix formed by rows 1 and 2, which is

$$A_{12} = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

For sake of simplicity, we (uniquely) describe every element of the $(1,2)$-binomial neighborhood $\mathcal{N}_{12}(A)$ by the first four columns (precisely those with column sums

[38] If either $u_{ij} = 0$ or $\ell_{ij} = 0$ it consists of a single binary matrix.

equal to one in the submatrix). For the switch chain, the induced subgraph of the undirected state space graph H on the $(1,2)$-binomial neighborhood of A, the Johnson graph $J(4,2)$ is given in Figure 4.26.

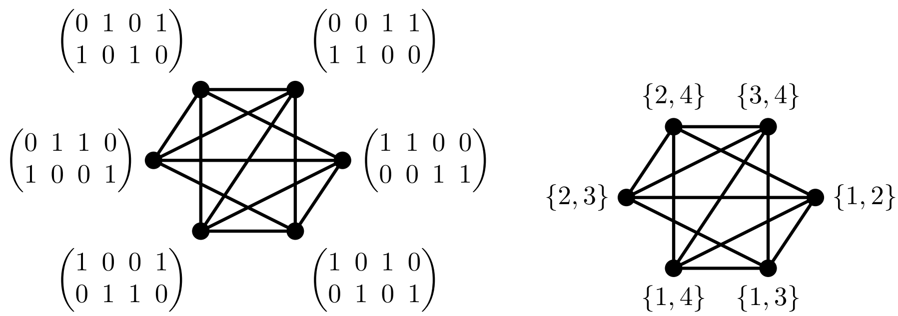

Figure 4.26: The induced subgraph H for the switch chain on the $(1,2)$-binomial neighborhood of A. On the left we have indexed the nodes by the submatrices of the first four columns, and on the right by label sets, indicating the positions of the 1's on the top row (i.e., row 1).

Remark 4.42. A fixed binomial neighborhood is reminiscent of the Bernoulli-Laplace Diffusion model, see, e.g., [57, 58] for an analysis of this model. In this model there are two bins with respectively k and $n-k$ balls, and in every transition two randomly chosen balls, one from each bin, are interchanged between the bins. Indeed, the state space graph is then a Johnson graph [58]. The transition probabilities are different, due to the non-zero holding probabilities in the switch algorithm, but the eigenvalues of this Markov chain are related to the eigenvalues of the switch Markov chain on a fixed binomial neighborhood, see also [57, 58].

Informally, the Markov chain resulting from always deterministically choosing rows i and j in the switch algorithm, is the disjoint union of smaller Markov chains each with a state space graph isomorphic to some Johnson graph. For a binomial neighborhood $\mathcal{N} = \mathcal{N}_{ij}(A)$ for given $i < j$ and $A \in \Omega$, the undirected graph $H_{\mathcal{N}} = (\Omega, E_{\mathcal{N}})$ is the graph where $E_{\mathcal{N}}$ forms the edge-set of the Johnson graph $J(u_{ij} + \ell_{ij}, u_{ij})$ on $\mathcal{N} \subseteq \Omega$, and where all binary matrices $B \in \Omega \setminus \mathcal{N}$ are isolated nodes. We use $M(H_{\mathcal{N}})$ to denote its adjacency matrix. The discussion above leads to the following theorem, where we define I_S as the identity matrix on S, and we define J_S as the all-ones matrix on S, that is $J_S(x,y) = 1$ if $x, y \in S$ and zero elsewhere.

Theorem 4.43. *The transition matrix P_γ of the γ-switch chain is of the form (4.26) namely*

$$P_\gamma = \sum_{1 \le i < j \le m} \binom{m}{2}^{-1} \sum_{\mathcal{N} \in \mathcal{R}_{(i,j)}} \left((1 - u_{ij}\ell_{ij} \cdot \gamma) \cdot I_{\mathcal{N}} + \gamma \cdot M(H_{\mathcal{N}}) \right). \tag{4.31}$$

The heat-bath variant of the γ-switch chain is given by the curveball chain, and can be written as

$$P_C = \sum_{1 \leq i < j \leq m} \binom{m}{2}^{-1} \sum_{\mathcal{N} \in \mathcal{R}_{(i,j)}} \binom{u_{ij} + \ell_{ij}}{u_{ij}}^{-1} J_{\mathcal{N}}. \tag{4.32}$$

Proof. The decomposition in (4.31) follows from the discussion above, and (4.30) guarantees that the matrix $(1 - u_{ij}\ell_{ij} \cdot \gamma) \cdot I_{\mathcal{N}} + \gamma \cdot M(H_{\mathcal{N}})$ indeed defines the transition matrix of a Markov chain for every $\mathcal{N}$. Moreover, remember that the γ-switch chain has uniform stationary distribution π over Ω. Indeed, for a binomial neighborhood $\mathcal{N} = \mathcal{N}_{ij}(A)$ for given $i < j$ and $A \in \Omega$, the vector $\sigma_{\mathcal{N}}$ as used in (4.27) is then given by

$$\sigma_{\mathcal{N}}(x) = \frac{\pi(x)}{\pi(\mathcal{N})} = \frac{1}{|\Omega|} \cdot \frac{|\Omega|}{|\mathcal{N}|} = \frac{1}{|\mathcal{N}|} = \binom{u_{ij} + \ell_{ij}}{u_{ij}}^{-1}$$

if $x \in \mathcal{N}$, and zero otherwise. This implies that $\mathbf{1} \cdot \sigma_{\mathcal{N}} = \binom{u_{ij}+\ell_{ij}}{u_{ij}}^{-1} J_{\mathcal{N}}$ as desired. $\square$

As a by-product of this decomposition, we can show that the transition matrix of the KTV switch chain [113] only has non-negative eigenvalues when $n \geq 3$. This is of independent interest as it shows that the KTV switch chain does not have to be made lazy in order to guarantee that all its eigenvalues are non-negative.[39][40]

Theorem 4.44. *The transition matrix of the KTV switch Markov chain only has non-negative eigenvalues when $n \geq 3$.*

Proof. The KTV switch chain is exactly the γ-switch chain with $\gamma = 2/(n(n-1))$. As the product $u_{ij}(A)\ell_{ij}(A)$ can be at most $n^2/4$ for any $A \in \Omega$ and $1 \leq i < j \leq m$, we see that γ satisfies (4.30) when $n \geq 3$. To show that P_{KTV} has all non-negative eigenvalues we show that the property assumed in Theorem 4.36 is satisfied by showing that the matrices

$$Y_{\mathcal{N}} = \left[1 - u_{ij}\ell_{ij} \cdot \binom{n}{2}^{-1}\right] I_{\mathcal{N}} + \binom{n}{2}^{-1} M(H_{\mathcal{N}})$$

have all non-negative eigenvalues. Theorem 4.36 then implies that P_{KTV} also only has non-negative eigenvalues. For any eigenvalue λ of the submatrix $Y_{\mathcal{N}}$, we have $\lambda = 1 + (\mu - u_{ij}\ell_{ij})\binom{n}{2}^{-1}$ where $\mu = \mu(\lambda)$ is an eigenvalue of the Johnson

[39] We refer the reader to a note of Greenhill [94] for more examples where 'laziness' can be avoided.

[40] Remember that a Markov chain is made lazy by replacing the transition matrix P by $(P + I)/2$. This is done in order to avoid technical issues in Sinclair's multi-commodity flow method [162] regarding negative eigenvalues.

graph $J(u_{ij}+\ell_{ij},u_{ij})$ on $\mathcal{N}$. In particular, using Proposition 4.15 with $p = u_{ij} + \ell_{ij}$ and $q = u_{ij}$, we get $(\mu - u_{ij}\ell_{ij}) \geq -\frac{1}{4}(u_{ij}+\ell_{ij}+1)^2 \geq -\frac{1}{4}(n+1)^2$ using that $0 \leq u_{ij}+\ell_{ij} \leq n$. Therefore, when $n \geq 5$, we have $\lambda \geq 1-(n+1)^2/(2n(n-1)) \geq 0$. The cases $n = 3, 4$ can be checked with some elementary arguments. This is left to the reader. □

We conclude this section with the proof of Theorem 4.35.

Proof of Theorem 4.35. Let $\mathcal{N} = \mathcal{N}_{ij}(A)$ for given $i < j$ and $A \in \Omega$. Note that the upper bound $(1-\lambda_*^{KTV})^{-1}$ follows from Theorem 4.36 with $\alpha = \beta = 1$ for which (4.29) holds as was shown in Theorem 4.44. We apply Theorem 4.36 for two pairs (α, β) to obtain the remaining upper and lower bound.

Case 1: $\alpha = 1$ and $\beta = (2n(n-1))/((2r_{\max}+1)^2)$. We show that condition (4.28) is satisfied. That is, we show that

$$\lambda = 1 - \beta\left(1 - \left(1 + (\mu - u_{ij}\cdot\ell_{ij})\binom{n}{2}^{-1}\right)\right) = 1 + \beta(\mu - u_{ij}\cdot\ell_{ij})\binom{n}{2}^{-1} \geq 0$$

for any $\mu = \mu(\lambda)$ that is an eigenvalue of the Johnson graph $J(u_{ij}+\ell_{ij},u_{ij})$. Again, using Proposition 4.15 in order to lower bound the quantity $(\mu - u_{ij}\cdot\ell_{ij})$, we find

$$\begin{aligned} 1 + \beta\cdot(\mu - u_{ij}\cdot\ell_{ij})\binom{n}{2}^{-1} &\geq 1 - \frac{\beta}{4}(u_{ij}+\ell_{ij}+1)^2\binom{n}{2}^{-1} \\ &\geq 1 - \frac{\beta}{4}(2r_{\max}+1)^2\binom{n}{2}^{-1} \\ &\geq 0, \end{aligned}$$

using the fact that $0 \leq u_{ij}+\ell_{ij} \leq 2r_{\max}$ and the choice of β. Hence we find the second part of the upper bound.

Case 2: $\alpha = -1$ and $\beta = -(n(n-1))/2$. We have to show that

$$\lambda = \binom{n}{2}\left(1 - \left(1 + (\mu - u_{ij}\cdot\ell_{ij})\binom{n}{2}^{-1}\right)\right) - 1 = u_{ij}\cdot\ell_{ij} - \mu - 1 \geq 0$$

for all $\mu = \mu(k) = (u_{ij}-k)(\ell_{ij}-k) - k$ where $k = 1, \ldots, u_{ij}$. Note that the eigenvalue $u_{ij}\cdot\ell_{ij}$ for the case $k = 0$ yields the largest eigenvalue $1 = \lambda_0^{\mathcal{N}}$ of $Y_{\mathcal{N}}$, and does not have to be considered here. The maximum over $k = 1, \ldots, u$ is then attained for $k = 1$, and we have $u_{ij}\cdot\ell_{ij}-\mu-1 \geq u_{ij}\cdot\ell_{ij}-((u_{ij}-1)(\ell_{ij}-1)-1)-1 = u_{ij}+\ell_{ij}-1 \geq 0$, since $u_{ij},\ell_{ij} \geq 1$. This gives us the lower bound and finishes the proof. □

4.5.3 Parallelism in the curveball chain

In this section we discuss an additional application of the comparison framework in Section 4.5.1. As a binary matrix is only adjusted on two rows at the time in the curveball algorithm, one might perform multiple binomial trades in parallel on distinct pairs of rows [25]. To be precise, in every step of the so-called k-*curveball algorithm*, we choose a set of $k \leq \lfloor m/2 \rfloor$ disjoint pairs of rows uniformly at random and perform a binomial trade on every pair (see Section 4.5). For $k = \lfloor m/2 \rfloor$ this corresponds to the *global curveball algorithm* described in [25]. We show that the k-*curveball chain*, resulting from the k-curveball algorithm, is a heat-bath variant of the curveball chain. We use the notation as given in Section 4.5.1

The index set $\mathcal{L} = \mathcal{L}(k)$ is the collection of all sets containing k pairwise disjoint sets of two rows, i.e.,

$$\mathcal{L}(k) = \Big\{ \{(1_c, 1_d), (2_c, 2_d), \dots, (k_c, k_d)\} \ : \ 1_c, 1_d, \dots, k_c, k_d \in [m], \\ \big|\{1_c, 1_d, 2_c, 2_d, \dots, k_c, k_d\}\big| = 2k \Big\}, \tag{4.33}$$

and ρ is the uniform distribution over $\mathcal{L}$. For a fixed collection $\kappa \in \mathcal{L}(k)$, we define the κ-neighborhood $\mathcal{N}_\kappa(A)$ of a binary matrix $A \in \Omega$ as the set of binary matrices $B \in \Omega$ that can be obtained from A by binomial trade-operations only involving the row-pairs in κ. Formally speaking, we have $B \in \mathcal{N}_\kappa(A)$ if and only if there exist binary matrices A_ℓ for $\ell = 0, \dots, k-1$, so that

$$A_{\ell+1} \in \mathcal{N}_{(\ell+1)_c,(\ell+1)_d}(A_\ell)$$

where $A = A_0$ and $B = A_k$. Note that the matrices A_ℓ might not all be pairwise distinct, as A and B could already coincide on certain pairs of rows in κ. Also note that $u_{i_c i_d}(A) = u_{i_c i_d}(B)$ and $l_{i_c i_d}(A) = l_{i_c i_d}(B)$ if $B \in \mathcal{N}_\kappa(A)$ for $i = 1, \dots, k$. It is not hard to see that such a neighborhood is isomorphic to a Cartesian product $W_1 \times W_2 \times \cdots \times W_k$ of finite sets[41] $W_1, \dots, W_k$ with

$$|W_i| = \binom{u_{i_c i_d} + l_{i_c i_d}}{u_{i_c i_d}}.$$

Moreover, the relation $\sim_\kappa$ defined by $a \sim_\kappa b$ if and only if $b \in \mathcal{N}_\kappa(a)$ defines an equivalence relation, and its equivalence classes give the set $\mathcal{R}_\kappa$. We now consider the following artificial formulation of the original curveball chain: we first select k pairs of distinct rows uniformly at random, and then we choose one of those pairs uniformly at random and apply a binomial trade on that pair. It should be clear that this generates the same Markov chain as when we directly select a pair of distinct rows uniformly at random. For $\mathcal{N}_\kappa \in \mathcal{R}_\kappa$ the matrix $P_{\mathcal{N}_\kappa}$ restricted to the rows and columns in $\mathcal{N}_\kappa$ is then the transition matrix of a Markov chain

[41] That is, the elements of W_i describe a matrix on row-pair (i_c, i_d).

over $W_1 \times \cdots \times W_k$, where in every step we choose an index $i \in [k]$ uniformly at random and make a transition in W_i based on the (uniform) transition matrix

$$Q_i = \binom{u_{i_c i_d} + l_{i_c i_d}}{u_{i_c i_d}}^{-1} J$$

where J is the all-ones matrix of appropriate size. More formally, the matrix $P_{\mathcal{N}_\kappa}$ restricted to the columns and rows in $\mathcal{N}_\kappa$ is given by

$$\frac{\sum_{i=1}^{k} \left[\otimes_{j=1}^{i-1} \mathcal{I}_j\right] \otimes Q_i \otimes \left[\otimes_{j=i+1}^{k} \mathcal{I}_j\right]}{k}, \tag{4.34}$$

forming a transition matrix on $\mathcal{N}_\kappa$, and is zero elsewhere. Here $\mathcal{I}_j$ is the identity matrix with the same size as Q_j and $\otimes$ the usual tensor product. The eigenvalues of the matrix in (4.34) are given by

$$\lambda_{\mathcal{N}_\kappa} = \left\{ \frac{1}{k} \sum_{i=1}^{k} \lambda_{j_i,i} : 0 \leq j_i \leq |W_i| - 1 \right\} \tag{4.35}$$

where $1 = \lambda_{0,i} \geq \lambda_{1,i} \geq \cdots \geq \lambda_{|W_i|-1,i}$ are the eigenvalues[42] of Q_i for $i = 1, \ldots, k$. It then follows that

$$P_C = \sum_{\kappa \in \mathcal{L}(k)} \frac{1}{|\mathcal{L}(k)|} \sum_{\mathcal{N}_\kappa \in \mathcal{R}_\kappa} P_{\mathcal{N}_\kappa}$$

which is of the form (4.26). For $k = 1$, we get back the description of the previous section. Now, its heat-bath variant is precisely the k-curveball Markov chain

$$P_{k,C} = \sum_{\kappa \in \mathcal{L}(k)} \frac{1}{|\mathcal{L}(k)|} \sum_{\mathcal{N}_\kappa \in \mathcal{R}_\kappa} \frac{1}{|\mathcal{N}_\kappa|} J_{\mathcal{N}_\kappa},$$

where

$$|\mathcal{N}_\kappa| = \prod_{i=1}^{k} \binom{u_{i_c i_d} + l_{i_c i_d}}{u_{i_c i_d}}^{-1}$$

as, roughly speaking, for a fixed neighborhood $\mathcal{N}_\kappa$, the k-curveball chain is precisely the uniform sampler over such a neighborhood.

Theorem 4.45. *We have* $(1 - \lambda_*^C)^{-1}/k \leq (1 - \lambda_*^{k,C})^{-1} \leq (1 - \lambda_*^C)^{-1}$ *where* $\lambda_*^{k,C}$ *is the second-largest eigenvalue of the* k*-curveball chain, and* λ_*^C *the second-largest eigenvalue of the original (*1*-)curveball chain.*

Proof. The upper bound follows from Theorem 4.36, with $\alpha = \beta = 1$, as the eigenvalues of all the Q_i are non-negative, and therefore (4.35) implies that the

[42] See, e.g., [73] for a similar argument regarding the transition matrix, and eigenvalues, of a Markov chain of this form. These statements follow directly from elementary arguments involving tensor products.

eigenvalues of the matrix in (4.34) are also non-negative. For the lower bound, we take $\alpha = -1$ and $\beta = -k$. That is, we have to show that $-1 + k(1-\mu) \geq 0$ with $\mu \in \lambda_{\mathcal{N}_\kappa} \setminus \{1\}$ as in (4.35). It is not hard to see that the second-largest eigenvalue in $\lambda_{\mathcal{N}_\kappa}$ is $(k-1)/k$, as the eigenvalues of every fixed Q_i are $1 = \lambda_{0,i} > \lambda_{1,i} = \cdots = \lambda_{|W_i|-1} = 0$. This implies that $-1 + k(1-\mu) \geq -1 + k(1-(k-1)/k) = 0$ for all $\mu \in \lambda_{\mathcal{N}_\kappa} \setminus \{1\}$. □

In general, the upper bound is tight for certain (degenerate) cases, that is, parallelism in the curveball chain does not necessarily guarantee an improvement in its relaxation time. E.g., take column marginals $c_i = 1$ for $i = 1, \ldots, n$, and row-marginals $r_1 = r_2 = n/2$ and $r_3 = r_4 = 0$, and consider $k = 2$.

4.6 Conclusion

We have provided mixing time analyses for three different Markov chains: the switch chain for strongly stable degree sequences; the restricted switch chain for joint degree matrix instances with two degree classes; and, finally, the curveball chain for bipartite degree sequences. The first two results rely on the multi-commodity flow method of Sinclair, whereas the latter relies on a Markov chain comparison argument.

We believe that our ideas introduced in Section 4.3 can be also used to simplify the switch chain analyses in settings where there is some given forbidden edge set,[43] the elements of which cannot be used in any (bipartite) graphical realization [93, 96, 70, 72]. This is an interesting direction for future work, as it captures the case of sampling directed graphs. Further, it is not clear whether there exist degree sequence families that are P-stable but not strongly stable. For instance, in a recent work by Gao and Wormald [90], who provide a very efficient non-MCMC approximate sampler for certain power-law degree sequences, it is argued that these power-law degree sequences are P-stable. Is it the case these sequences are strongly stable as well? A central open question is how to go beyond (strong) stability.

The problem of sampling graphical realizations of a given joint degree distribution with three or more degree classes is also open, either using the switch chain or any other method for that matter. Although our proof breaks down for more than two classes, we hope that our high level approach can facilitate progress on the problem.

For the curveball chain we believe similar ideas as in this work can be used to prove that the curveball chain is rapidly mixing for the sampling of undirected graphs with given degree sequences [25], whenever a switch-based chain is rapidly mixing for those degree sequences. It should be noted that the main conclusion of our results in Section 4.5 is not that the curveball approach is necessarily better than the switch-based approaches. In particular, the improvement in relaxation time in Theorem 4.35, when the maximum row sum is small compared

[43] This is similar to the setting we consider in Section 4.5.

to n, is mostly caused by the fact that the KTV switch chain is a bad choice of implementation here (as the holding probability of a state in the Markov chain is relatively large in this case). There exist other implementations of the switch chain that are more efficient than the KTV switch chain for certain marginals. For example, an implementation similar to the switch chain as in Section 4.2. Although we believe the curveball chain will outperform any switch-based chain for certain marginals, it not obvious for which marginals this is true. For example, it is not clear to us if this is true in the case of sampling regular directed graphs with in- and out-degree some small constant. However, for graphs with large regular degrees we expect the curveball chain to be better.

Moreover, one step of the curveball algorithm is computationally more expensive than one step of a switch-based algorithm, so although the relaxation time of the curveball chain might be better than a switch-based chain, this does not automatically imply that the overall running time of the curveball algorithm is better than that of a switch-based algorithm. Nevertheless, we believe that our results are a first theoretical step for speeding up switch-based Markov chains for sampling (bipartite) graphs with a given degree sequence.

Summary

In this thesis we focus on two problems at the intersection of mathematics and theoretical computer science: the inefficiency and computation of Nash equilibria in congestion games and the uniform generation of graphs with a given degree sequence using simple Markov Chain Monte Carlo methods. In Chapter 1 we provide some background and context for both these problems.

In Chapters 2 and 3 we study congestion game models, that can be used to analyze problems such as traffic congestion and internet routing from a theoretical point of view. We are interested in so-called Nash equilibria of these games. A Nash equilibrium is in some sense a 'stable' outcome of a game, meaning that no player of the game has an incentive to act differently. The main difference between the models studied in these chapters is the influence that individual players have on the outcome of a game. In Chapter 2 we study non-atomic congestion games, that can be used to model large systems in which individual players do not significantly influence the outcome of a game. On the other hand, in Chapter 3, we study atomic congestion games, where individual players can have a significant impact on the outcome of a game. We next give a more detailed overview of the problems studied in Chapters 2 and 3.

In Chapter 2 we study the quality deterioration of Nash equilibria as a result of deviations (or perturbations) in the latency functions of non-atomic network routing games. This framework can, e.g., be used to study risk-averse behavior of players in such games, as has been done in the literature somewhat recently. For example, in the case of traffic congestion, it might be uncertain how long it will take to travel through certain parts of the road network, and, as a result, these are avoided by players that prefer to know exactly how long their journey will take (at the cost of an increased travel time). The main contributions of Chapter 2 are tight inefficiency bounds quantifying this type of quality deterioration. The quality of a Nash equilibrium here is measured as the total latency (or travel time) of all players in the network.

In Chapter 3 we study atomic congestion games. Here we provide various unifications and extensions regarding the computation and inefficiency of pure Nash equilibria. In particular, we do this by taking a polytopal point of view, following recent approaches in the literature. We identify polytopal properties, satisfied by many games with a combinatorial flavor, that are sufficient to compute a pure

Nash equilibrium in polynomial time. Moreover, we give quantitative bounds on the quality of these equilibria that outperform bounds known for arbitrary pure Nash equilibria. We also unify various extensions and variations on the classical atomic congestion game model due to Rosenthal (1973).

In Chapter 4 we present various new results related to the switch algorithm for the uniform generation of graphs with a given degree sequence. The problem of uniformly generating a graph with a given degree sequence entails the design of an algorithm that outputs every graph with the desired degree sequence with (almost) equal probability. The switch algorithm is a very simple Markov Chain Monte Carlo approach that proceeds by repeatedly selecting two edges of the current graph and switching them if possible, while preserving the degree sequence. The main question of interest here is how many switches are needed before the output is close to being a uniform random sample from the set of all graphs with the given degree sequence. We make some progress on this problem by identifying large ranges of degree sequences for which a polynomial number of switches is sufficient, thereby unifying various results in the literature. We also study two Markov chains related to the switch algorithm.

Bibliography

[1] H. Ackermann, H. Röglin, and B. Vöcking. On the impact of combinatorial structure on congestion games. *Journal of the ACM*, 55(6):25:1–25:22, 2008.

[2] H. Ackermann, H. Röglin, and B. Vöcking. Pure Nash equilibria in player-specific and weighted congestion games. *Theoretical Computer Science*, 410(17):1552–1563, 2009.

[3] E. Aigner-Horev, J. Carmesin, and J.-O. Frohlich. Infinite matroid union. *CoRR*, abs/1111.0602, 2012.

[4] S. Aland, D. Dumrauf, M. Gairing, B. Monien, and F. Schoppmann. Exact price of anarchy for polynomial congestion games. *SIAM Journal of Computing*, 40(5):1211–1233, 2011.

[5] G. Amanatidis, B. Green, and M. Mihail. Graphic realizations of joint-degree matrices. *CoRR*, abs/1701.03856, 2008.

[6] G. Amanatidis and P. Kleer. Rapid mixing of the switch Markov chain for strongly stable degree sequences and 2-class joint degree matrices. *CoRR*, abs/1803.01338, 2018.

[7] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4):1602–1623, 2008.

[8] B. Awerbuch, Y. Azar, and A. Epstein. The price of routing unsplittable flow. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 57–66, 2005.

[9] A. Barvinok and J. Hartigan. The number of graphs and a random graph with a given degree sequence. *Random Structures & Algorithms*, 42(3):301–348, 2013.

[10] S. P. Baum and L. E. Trotter. Integer rounding and polyhedral decomposition for totally unimodular systems. In *Optimization and Operations Research: Proceedings of a Workshop Held at the University of Bonn, October 2–8, 1977*, pages 15–23, 1978.

[11] M. Bayati, J. H. Kim, and A. Saberi. A sequential algorithm for generating random graphs. *Algorithmica*, 58(4):860–910, 2010.

[12] M. Beckmann, B. McGuire, and C. Winsten. Studies in the economics of transportation. *Yale University Press*, 1956.

[13] M. Bellare, O. Goldreich, and E. Petrank. Uniform generation of np-witnesses using an np-oracle. *Information and Computation*, 163(2):510–526, 2000.

[14] A. Berger and M. Müller-Hannemann. Uniform sampling of digraphs with a fixed degree sequence. In *Lecture Notes in Computer Science*, volume 6410, pages 220–231, 2010.

[15] N. Bhatnagar, D. Randall, V. V. Vazirani, and E. Vigoda. Random bichromatic matchings. *Algorithmica*, 50(4):418–445, 2008.

[16] B. Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1(4):311–316, 1980.

[17] V. Bonifaci, M. Salek, and G. Schäfer. On the efficiency of restricted tolls in network routing games. In *Lecture Notes in Computer Science*, volume 6982, pages 302–313, 2011.

[18] D. Braess. Über ein paradoxon aus der verkehrsplanung. *Unternehmensforschung*, 12:258–268, 1969.

[19] A. Brouwer and W. Haemers. *Spectra of Graphs*. Universitext. Springer New York, 2011.

[20] I. Caragiannis, A. Fanelli, N. Gravin, and A. Skopalik. Computing approximate pure Nash equilibria in congestion games. *ACM SIGecom Exchanges*, 11(1):26–29, 2012.

[21] I. Caragiannis, M. Flammini, C. Kaklamanis, P. Kanellopoulos, and L. Moscardelli. Tight bounds for selfish and greedy load balancing. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*, pages 311–322, 2006.

[22] I. Caragiannis, C. Kaklamanis, and P. Kanellopoulos. Taxes for linear atomic congestion games. *ACM Transactions on Economics and Computation*, 7(1):13:1–13:31, 2010.

[23] I. Caragiannis, C. Kaklamanis, P. Kanellopoulos, M. Kyropoulou, and E. Papaioannou. The impact of altruism on the efficiency of atomic congestion games. In *Lecture Notes in Computer Science*, volume 6084, pages 172–188, 2010.

[24] M. Carol. Network flow problems and congestion games: complexity and approximation results. *Ph.D. thesis*, 2006.

[25] C. J. Carstens, A. Berger, and G. Strona. Curveball: a new generation of sampling algorithms for graphs with fixed degree sequence. *CoRR*, abs/1609.05137, 2016.

[26] C. J. Carstens and P. Kleer. Comparing the switch and curveball Markov chains for sampling binary matrices with fixed marginals. *CoRR*, abs/1709.07290, 2017.

[27] H. Chan and A. X. Jiang. Congestion games with polytopal strategy spaces. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 165–171, 2016.

[28] P.-A. Chen, B. de Keijzer, D. Kempe, and G. Schäfer. Altruism and its impact on the price of anarchy. *ACM Transactions on Economics and Computation*, 2(4):17:1–17:45, 2014.

[29] P.-A. Chen and D. Kempe. Altruism, selfishness, and spite in traffic routing. In *Proceedings of the 9th ACM conference on Electronic Commerce*, pages 140–149, 2008.

[30] Y. Chen, P. Diaconis, S. P. Holmes, and J. S. Liu. Sequential monte carlo methods for statistical analysis of tables. *Journal of the American Statistical Association*, 100(469):109–120, 2005.

[31] S. Chien and A. Sinclair. Convergence to approximate Nash equilibria in congestion games. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 169–178, 2007.

[32] G. Christodoulou and M. Gairing. Price of stability in polynomial congestion games. *ACM Transactions on Economics and Computation*, 4(2):10:1–10:17, 2016.

[33] G. Christodoulou and E. Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congestion games. In *Proceedings of the 13th Annual European Conference on Algorithms*, pages 59–70, 2005.

[34] G. Christodoulou and E. Koutsoupias. The price of anarchy of finite congestion games. In *Proceedings of 37th Annual ACM Symposium on Theory of Computing*, pages 67–73, 2005.

[35] G. Christodoulou, E. Koutsoupias, and P. G. Spirakis. On the performance of approximate equilibria in congestion games. *Algorithmica*, 61(1):116–140, 2011.

[36] G. Christodoulou, V. S. Mirrokni, and A. Sidiropoulos. Convergence and approximation in potential games. *Theoretical Computer Science*, 438:13–27, 2012.

[37] R. Cole, Y. Dodis, and T. Roughgarden. Bottleneck links, variable demand, and the tragedy of the commons. *Networks*, 60(3):194–203, 2012.

[38] R. Cole, T. Lianeas, and E. Nikolova. When does diversity of agent preferences improve outcomes in selfish routing? In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 173–179, 2018.

[39] R. Colini-Baldeschi, R. Cominetti, P. Mertikopoulos, and M. Scarsini. The asymptotic behavior of the price of anarchy. In *Lecture Notes in Computer Science*, volume 10674, pages 133–145, 2017.

[40] R. Cominetti. Equilibrium routing under uncertainty. *Mathematical Programming*, 151(1):117–151, 2015.

[41] C. Cooper, M. E. Dyer, and C. S. Greenhill. Sampling regular graphs and a peer-to-peer network. *Combinatorics, Probability & Computing*, 16(4):557–593, 2007.

[42] C. Cooper, M. E. Dyer, and C. S. Greenhill. Corrigendum: Sampling regular graphs and a peer-to-peer network. *CoRR*, abs/1203.6111, 2012.

[43] C. Cooper, M. E. Dyer, C. S. Greenhill, and A. J. Handley. The flip Markov chain for connected regular graphs. *Discrete Applied Mathematics*, 254:56–79, 2018.

[44] J. Correa, J. de Jong, B. Keijzer, and M. Uetz. The curse of sequentiality in routing games. In *Lecture Notes in Computer Science*, volume 9470, pages 258–271, 2015.

[45] J. R. Correa, A. S. Schulz, and N. E. Stier-Moses. Selfish routing in capacitated networks. *Mathematics of Operations Research*, 29(4):961–976, 2004.

[46] J. R. Correa, A. S. Schulz, and N. E. Stier-Moses. A geometric approach to the price of anarchy in nonatomic congestion games. *Games and Economic Behavior*, 64(2):457–469, 2008.

[47] M. Cryan, M. Dyer, L. A. Goldberg, M. Jerrum, and R. Martin. Rapidly mixing markov chains for sampling contingency tables with a constant number of rows. *SIAM Journal on Computing*, 36(1):247–278, 2006.

[48] É. Czabarka. Partition adjacency matrices in network modeling. *Technical report on GRAPHS project FA9550-12-1-040*, 2014.

[49] É. Czabarka, A. Dutle, P. L. Erdös, and I. Miklós. On realizations of a joint degree matrix. *Discrete Applied Mathematics*, 181:283–288, 2015.

[50] É. Czabarka, L. A. Szekely, Z. Toroczkai, and S. Walker. An algebraic Monte-Carlo algorithm for the bipartite partition adjacency matrix realization problem. *CoRR*, abs/1708.08242, 2017.

[51] G. B. Dantzig. *Linear programming and extensions.* Princeton University Press, 1963.

[52] J. de Jong, W. Kern, B. Steenhuisen, and M. Uetz. The asymptotic price of anarchy for k-uniform congestion games. In *Lecture Notes in Computer Science*, volume 10138, pages 317–328, 2018.

[53] J. de Jong, M. Klimm, and M. Uetz. Efficiency of equilibria in uniform matroid congestion games. In *Lecture Notes in Computer Science*, volume 9928, pages 105–116, 2016.

[54] A. Del Pia, M. Ferris, and C. Michini. Totally unimodular congestion games. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 577–588, 2017.

[55] P. Diaconis and L. Saloff-Coste. Comparison techniques for random walk on finite groups. *The Annals of Probability*, 21(4):2131–2156, 1993.

[56] P. Diaconis and L. Saloff-Coste. Comparison theorems for reversible Markov chains. *The Annals of Applied Probability*, 3(3):696–730, 1993.

[57] P. Diaconis and M. Shahshahani. Time to reach stationarity in the bernoulli-laplace diffusion model. *SIAM Journal on Mathematical Analysis*, 18(1):208–218, 1987.

[58] P. Donnelly, P. Lloyd, and A. Sudbury. Approach to stationarity of the bernoulli-laplace diffusion model. *Advances in Applied Probability*, 26(3):715–727, 1994.

[59] M. Dyer, A. Frieze, and M. Jerrum. Approximately counting hamilton paths and cycles in dense graphs. *SIAM Journal on Computing*, 27(5):1262–1272, 1998.

[60] M. Dyer, L. A. Goldberg, M. Jerrum, and R. Martin. Markov chain comparison. *Probability Surveys*, 3:89–111, 2006.

[61] M. Dyer, C. S. Greenhill, and M. Ullrich. Structure and eigenvalues of heat-bath Markov chains. *Linear Algebra and its Applications*, 454:57–71, 2014.

[62] M. Dyer, M. Jerrum, and H. Müller. On the switch Markov chain for perfect matchings. *Journal of the ACM*, 64(2):12:1–12:33, 2017.

[63] J. Edmonds. Paths, trees, and flowers. *Canadian journal of mathematics*, 17(3):449–467, 1965.

[64] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In *Lecture Notes in Computer Science*, volume 2570, pages 11–26, 2003.

[65] J. Edmonds and R. Giles. A min-max relation for submodular functions on graphs. *Annals of Discrete Mathematics*, 1:185–204, 1977.

[66] M. Englert, T. Franke, and L. Olbrich. Sensitivity of Wardrop equilibria. In *Lecture Notes in Computer Science*, volume 4997, pages 158–169, 2008.

[67] P. L. Erdős, C. S. Greenhill, T. R. Mezei, I. Miklós, D. Soltész, and L. Soukup. The mixing time of the swap (switch) markov chains: a unified approach. *CoRR*, abs/1903.06600, 2019.

[68] P. L. Erdős, S. G. Hartke, L. van Iersel, and I. Miklós. Graph realizations constrained by skeleton graphs. *Electronic Journal of Combinatorics*, 24(2), 2017.

[69] P. L. Erdős, Z. Király, and I. Miklós. On the swap-distances of different realizations of a graphical degree sequence. *Combinatorics, Probability & Computing*, 22(3):366–383, 2013.

[70] P. L. Erdős, S. Z. Kiss, I. Miklós, and L. Soukup. Approximate counting of graphical realizations. *PLOS ONE*, 10(7):1–20, 2015.

[71] P. L. Erdős, T. R. Mezei, and I. Miklós. Efficiently sampling the realizations of irregular, but linearly bounded bipartite and directed degree sequences. *CoRR*, abs/1712.01709, 2017.

[72] P. L. Erdős, T. R. Mezei, I. Miklós, and D. Soltész. Efficiently sampling the realizations of bounded, irregular degree sequences of bipartite and directed graphs. *PLOS ONE*, 13(8):1–20, 08 2018.

[73] P. L. Erdős, I. Miklós, and Z. Toroczkai. A decomposition based proof for fast mixing of a Markov chain over balanced realizations of a joint degree matrix. *SIAM Journal on Discrete Mathematics*, 29(1):481–499, 2015.

[74] E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence time to Nash equilibrium in load balancing. *ACM Transactions on Algorithms*, 3(3):1–32, 2007.

[75] A. Fabrikant, C. Papadimitriou, and K. Talwar. The complexity of pure Nash equilibria. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 604–612, 2004.

[76] T. Feder, A. Guetz, M. Mihail, and A. Saberi. A local switch Markov chain on given degree graphs with application in connectivity of peer-to-peer networks. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 69–76, 2006.

[77] M. Feldman, N. Immorlica, B. Lucier, T. Roughgarden, and V. Syrgkanis. The price of anarchy in large games. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing*, pages 963–976, 2016.

[78] L. Fleischer. Linear tolls suffice: New bounds and algorithms for tolls in single source networks. *Theoretical Computer Science*, 348(2):217–225, 2005.

[79] D. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 2010.

[80] D. Fotakis. Congestion games with linearly independent paths: Convergence time and price of anarchy. *Theory of Computing Systems*, 47(1):113–136, 2010.

[81] D. Fotakis, D. Kalimeris, and T. Lianeas. Improving selfish routing for risk-averse players. In *Lecture Notes in Computer Science*, volume 9470, pages 328–342, 2015.

[82] A. Frank and É. Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987.

[83] H. Gabow. A matroid approach to finding edge connectivity and packing arborescences. *Journal of Computer and System Sciences*, 50(2):259–273, 1995.

[84] M. Gairing, T. Lücking, M. Mavronicolas, B. Monien, and M. Rode. Nash equilibria in discrete routing games with convex latency functions. *Journal of Computer and System Sciences*, 74(7):1199–1225, 2008.

[85] D. Gale. A theorem on flows in networks. *Pacific Journal of Mathematics*, 7(2):1073–1082, 1957.

[86] T. Gallai and P. Erdos. Graphs with prescribed degree of vertices. *Lapok*, 11:264–274, 1960.

[87] P. Gao and C. S. Greenhill. Uniform generation of spanning regular subgraphs of a dense graph. *CoRR*, abs/1807.00964, 2018.

[88] P. Gao and N. Wormald. Enumeration of graphs with a heavy-tailed degree sequence. *Advances in Mathematics*, 287:412–450, 2016.

[89] P. Gao and N. C. Wormald. Uniform generation of random regular graphs. *SIAM Journal of Computing*, 46(4):1395–1427, 2017.

[90] P. Gao and N. C. Wormald. Uniform generation of random graphs with power-law degree sequences. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1741–1758, 2018.

[91] A. L. Gibbs and F. E. Su. On choosing and bounding probability metrics. *International Statistical Review*, 70(3):419–435, 2002.

[92] M. Gjoka, B. Tillman, and A. Markopoulou. Construction of simple graphs with a target joint degree matrix and beyond. In *Proceedings of IEEE Conference on Computer Communications*, pages 1553–1561, 2015.

[93] C. S. Greenhill. A polynomial bound on the mixing time of a Markov chain for sampling regular directed graphs. *Electronic Journal of Combinatorics*, 18(1), 2011.

[94] C. S. Greenhill. Making Markov chains less lazy. *CoRR*, abs/1203.6668, 2013.

[95] C. S. Greenhill. The switch Markov chain for sampling irregular graphs. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1564–1572, 2015.

[96] C. S. Greenhill and M. Sfragara. The switch Markov chain for sampling irregular graphs and digraphs. *Theoretical Computer Science*, 719:1–20, 2018.

[97] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988.

[98] S. Hakimi. On realizability of a set of integers as degrees of the vertices of a linear graph. I. *Journal of the Society for Industrial and Applied Mathematics*, 10(3):496–506, 1962.

[99] T. Harks, M. Hoefer, M. Klimm, and A. Skopalik. Computing pure Nash and strong equilibria in bottleneck congestion games. *Mathematical Programming*, 141(1):193–215, 2013.

[100] T. Harks, M. Klimm, and B. Peis. Sensitivity analysis for convex separable optimization over integral polymatroids. *SIAM Journal on Optimization*, 28(3):2222–2245, 2018.

[101] A. Haurie and P. Marcotte. On the relationship between Nash—Cournot and Wardrop equilibria. *Networks*, 15(3):295–308, 1985.

[102] V. Havel. A remark on the existence of finite graphs. *Časopis pro pěstování matematiky*, 080(4):477–480, 1955.

[103] M. Hoefer, L. Olbrich, and A. Skopalik. Taxing subnetworks. In *Lecture Notes in Computer Science*, volume 5385, pages 286–294, 2008.

[104] D. A. Holton and J. Sheehan. *The Petersen graph*. Cambridge University Press Cambridge, 1993.

[105] T. Homma. A theorem on continuous functions. *Kodai Mathematical Seminar Reports*, 4(1):13–16, 1952.

[106] S. Ieong, R. McGrew, E. Nudelman, Y. Shoham, and Q. Sun. Fast and compact: A simple class of congestion games. In *Proceedings of the 20th National Conference on Artificial Intelligence*, pages 489–494, 2005.

[107] M. Jerrum, B. D. McKay, and A. Sinclair. When is a graphical sequence stable? *Department of Computer Science: Internal report*, 1989.

[108] M. Jerrum and A. Sinclair. Approximating the permanent. *SIAM Journal on Computing*, 18(6):1149–1178, 1989.

[109] M. Jerrum and A. Sinclair. Fast uniform generation of regular graphs. *Theoretical Computer Science*, 73(1):91–100, 1990.

[110] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM*, 51(4):671–697, 2004.

[111] M. R. Jerrum, L. G. Valiant, and V. V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

[112] S. Kakutani. A generalization of brouwer's fixed point theorem. *Duke Mathematical Journal*, 8(3):457–459, 1941.

[113] R. Kannan, P. Tetali, and S. Vempala. Simple Markov-chain algorithms for generating bipartite graphs and tournaments. *Random Structures and Algorithms*, 14(4):293–308, 1999.

[114] J. H. Kim and V. H. Vu. Generating random regular graphs. *Combinatorica*, 26(6):683–708, 2006.

[115] P. Kleer and G. Schäfer. Tight inefficiency bounds for perception-parameterized affine congestion games. *Theoretical Computer Science*, 754:65–87, 2019.

[116] M. Klimm and P. Warode. Computing all Wardrop equilibria parametrized by the flow demand. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 917–934, 2019.

[117] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Conference on Theoretical Aspects of Computer Science*, pages 404–413, 1999.

[118] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov chains and mixing times*. American Mathematical Society, 2009.

[119] T. Lianeas, E. Nikolova, and N. E. Stier-Moses. Asymptotically tight bounds for inefficiency in risk-averse selfish routing. *CoRR*, abs/1510.02067, 2015.

[120] T. Lianeas, E. Nikolova, and N. E. Stier-Moses. Asymptotically tight bounds for inefficiency in risk-averse selfish routing. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 338–344, 2016.

[121] A. Liebenau and N. Wormald. Asymptotic enumeration of graphs by degree sequence, and the degree sequence of a random graph. *CoRR*, abs/1702.08373, 2017.

[122] H. Lin, T. Roughgarden, É. Tardos, and A. Walkover. Stronger bounds on braess's paradox and the maximum latency of selfish routing. *SIAM Journal on Discrete Mathematics*, 25(4):1667–1686, 2011.

[123] T. Lücking, M. Mavronicolas, B. Monien, and M. Rode. A new model for selfish routing. *Theoretical Computer Science*, 406(3):187–206, 2008.

[124] P. Mahadevan, D. V. Krioukov, K. R. Fall, and A. Vahdat. Systematic topology analysis and generation using degree correlations. In *Proceedings of the ACM SIGCOMM 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 135–146, 2006.

[125] C. McDiarmid. Integral decomposition in polyhedra. *Mathematical Programming*, 25(2):183–198, 1983.

[126] B. D. McKay and N. C. Wormald. Uniform generation of random regular graphs of moderate degree. *Journal of Algorithms*, 11(1):52–67, 1990.

[127] B. D. McKay and N. C. Wormald. Asymptotic enumeration by degree sequence of graphs with degress $o\left(n^{1/2}\right)$. *Combinatorica*, 11(4):369–382, 1991.

[128] R. Meir and D. C. Parkes. Congestion games with distance-based strict uncertainty. *CoRR*, abs/1411.4943, 2014.

[129] R. Meir and D. C. Parkes. Playing the wrong game: Smoothness bounds for congestion games with behavioral biases. *SIGMETRICS Performance Evaluation Review*, 43(3):67–70, 2015.

[130] R. Meyer. A class of nonlinear integer programs solvable by a single linear program. *SIAM Journal on Control and Optimization*, 15(6):935–946, 1977.

[131] I. Miklós, P. L. Erdős, and L. Soukup. Towards random uniform sampling of bipartite graphs with given degree sequence. *Electronic Journal of Combinatorics*, 20(1), 2013.

[132] I. Milchtaich. Congestion games with player-specific payoff functions. *Games and economic behavior*, 13(1):111–124, 1996.

[133] I. Milchtaich. Network topology and the efficiency of equilibrium. *Games and Economic Behavior*, 57(2):321–346, 2006.

[134] D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996.

[135] B. Morris and A. Sinclair. Random walks on truncated cubes and sampling 0-1 knapsack solutions. *SIAM journal on computing*, 34(1):195–226, 2004.

[136] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

[137] J. F. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

[138] E. Nikolova and N. E. Stier-Moses. A mean-risk model for the traffic assignment problem with stochastic travel times. *Operations Research*, 62(2):366–382, 2014.

[139] E. Nikolova and N. E. Stier-Moses. The burden of risk aversion in mean-risk selfish routing. In *Proceedings of the 16th ACM Conference on Economics and Computation*, pages 489–506, 2015.

[140] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[141] B. P. Olding and P. J. Wolfe. Inference for graphs and networks: Adapting classical tools to modern data. In *Data Analysis for Network Cyber-Security*, pages 1–31, 2014.

[142] S. Onn, U. G. Rothblum, and Y. Tangir. Edge-directions of standard polyhedra with applications to network flows. *Journal of Global Optimization*, 33(1):109–122, 2005.

[143] A. Orda, R. Rom, and N. Shimkin. Competitive routing in multiuser communication networks. *IEEE/ACM Transactions on Networking*, 1(5):510–521, 1993.

[144] C. H. Papadimitriou and M. Yannakakis. The complexity of restricted spanning tree problems. *Journal of the ACM*, 29(2):285–309, 1982.

[145] A. Patrinos and S. Hakimi. Relations between graphs and integer-pair sequences. *Discrete Mathematics*, 15(4):347–358, 1976.

[146] G. Piliouras, E. Nikolova, and J. S. Shamma. Risk sensitivity of price of anarchy under uncertainty. In *Proceedings of the 14th ACM Conference on Electronic Commerce*, pages 715–732, 2013.

[147] J. Quastel. Diffusion of color in the simple exclusion process. *Communications on Pure and Applied Mathematics*, 45(6):623–679, 1992.

[148] A. R. Rao, R. Jana, and S. Bandyopadhyay. A Markov chain Monte Carlo method for generating random (0, 1)-matrices with given marginals. *Sankhyā: The Indian Journal of Statistics, Series A*, 58(2):225–242, 1996.

[149] Rijkswaterstaat, Ministry of Infrastructure and Water Management. Eerste auto's rijden over nieuwe A4. *https://www.rijkswaterstaat.nl/nieuws/2015/12/eerste_autos_rijden_over_nieuwe_a4.aspx*, 2015.

[150] R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

[151] R. W. Rosenthal. The network equilibrium problem in integers. *Networks*, 3(1):53–59, 1973.

[152] T. Roughgarden. The price of anarchy is independent of the network topology. *Journal of Computer and System Sciences*, 67(2):341–364, 2003.

[153] T. Roughgarden. *Selfish Routing and the Price of Anarchy.* The MIT Press, 2005.

[154] T. Roughgarden. On the severity of braess's paradox: Designing networks for selfish users is hard. *Journal of Computer and System Sciences*, 72(5):922–953, 2006.

[155] T. Roughgarden. Intrinsic robustness of the price of anarchy. *Journal of the ACM*, 62(5):32, 2015.

[156] T. Roughgarden and É. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, 2002.

[157] H. J. Ryser. Combinatorial properties of matrices of zeros and ones. *Canadian Journal of Mathematics*, 9:371–377, 1957.

[158] J. G. Sanderson. Testing ecological patterns. *American Scientist*, 88(4):332, 2000.

[159] D. Schmeidler. Equilibrium points of nonatomic games. *Journal of Statistical Physics*, 7(4):295–300, 1973.

[160] A. Schrijver. *Theory of Linear and Integer Programming.* John Wiley & Sons, Inc., 1986.

[161] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency. Vol. B, Matroids, trees, stable sets. Chapters 39–69.* Springer-Verlag, 2003.

[162] A. Sinclair. Improved bounds for mixing rates of Markov chains and multicommodity flow. *Combinatorics, Probability & Computing*, 1:351–370, 1992.

[163] I. Stanton and A. Pinar. Constructing and sampling graphs with a prescribed joint degree distribution. *ACM Journal of Experimental Algorithmics*, 17(1), 2011.

[164] D. Štefankovič, E. Vigoda, and J. Wilmes. On counting perfect matchings in general graphs. In *Proceedings of the 13th Latin American Symposium on Theoretical Informatics*, pages 873–885, 2018.

[165] A. Steger and N. C. Wormald. Generating random regular graphs quickly. *Combinatroics, Probability & Computing*, 8(4):377–396, 1999.

[166] G. Strona, D. Nappo, F. Boccacci, S. Fattorini, and J. San-Miguel-Ayanz. A fast and unbiased procedure to randomize ecological binary matrices with fixed row and column totals. *Nature Communications*, 5(4114), 2014.

[167] R. Taylor. Constrained switchings in graphs. In *Proceedings of the 8th Australian Conference on Combinatorial Mathematics*, pages 314–336, 1981.

[168] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.

[169] L. Tran-Thanh, M. Polukarov, A. Chapman, A. Rogers, and N. R. Jennings. On the existence of pure strategy Nash equilibria in integer–splittable weighted congestion games. In *Lecture Notes in Computer Science*, volume 6982, pages 236–253, 2011.

[170] W. T. Tutte. The factors of graphs. *Canadian Journal of Mathematics*, 4(3):314–328, 1952.

[171] W. T. Tutte. A short proof of the factor theorem for finite graphs. *Canadian Journal of Mathematics*, 6:347–352, 1954.

[172] L. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

[173] A. F. Veinott and G. B. Dantzig. *Integral Extreme Points.* Defense Technical Information Center, 1968.

[174] N. D. Verhelst. An efficient MCMC algorithm to sample binary matrices with fixed marginals. *Psychometrika*, 73(4):705, 2008.

[175] J. G. Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, 1:325–378, 1952.

[176] J. V. Whittaker. A mountain-climbing problem. *Canadian Journal of Mathematics*, 18(873-882):4, 1966.

[177] F. Zhang. *Matrix Theory: Basic Results and Techniques.* Springer, 1999.

Appendix A

Combinatorial SDD of matroid congestion games

In this section, we describe a combinatorial approach for computing the symmetric difference decomposition of non-symmetric matroid congestion games. Our analysis also provides a local search algorithm which can be seen as a natural generalization of best response dynamics.

Throughout this section, we let $\Gamma = (N, E, (\mathcal{S}_i), (c_e))$ be a non-symmetric matroid congestion game, where the strategy set $\mathcal{S}_i$ of each player $i \in N$ is given by the bases $\mathcal{B}_i$ of a matroid $\mathcal{M}_i = (E, \mathcal{I}_i)$.

A.1 Symmetric difference decomposition

We start by deriving the symmetric difference decomposition.

Let $s = (s_1, \ldots, s_n)$ and $t = (t_1, \ldots, t_n)$ be two feasible strategy profiles with (feasible) load profiles f and g, respectively. We need the following result for matroids.

Proposition A.1 ([161])**.** *Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid and let $\mathcal{B}$ denote the set of bases of $\mathcal{M}$. Then for all $B, B' \in \mathcal{B}$, there exists a bijection $\tau : B \setminus B' \to B' \setminus B$ such that $B - x + \tau(x) \in \mathcal{B}$ for all $x \in B \setminus B'$.*

For $i \in N$, let $\tau^i : s_i \setminus t_i \to t_i \setminus s_i$ be a bijection satisfying Proposition A.1. Let $G = (V, A)$ be a directed multigraph defined by $V = E$, and the multiset

$$A = \bigcup_{i \in N} \{(e, \tau^i(e)) : e \in s_i \setminus t_i\}.$$

Note that, implicitly, every arc corresponds to a unique player. Since, for a fixed player i the bases s_i and t_i have the same size, it follows that

$$K := \sum_{e\colon f_e > g_e} (f_e - g_e) = \sum_{e\colon g_e > f_e} (g_e - f_e).$$

In particular, the graph G can be decomposed into K edge-disjoint paths (or chains) $(P_j)_{j=1,\dots,K}$ with the property that they start at an *overloaded* resource e with $f_e > g_e$ and end at an *underloaded* resource e with $g_e > f_e$ (and possibly some cycles). If follows that we can write

$$g - f = \sum_{j=1}^{K} x^j,$$

where for each path $P_j = (a_j, \dots, b_j)$ we have

$$x_e^j = \begin{cases} 1 & \text{if } e = b_j, \\ -1 & \text{if } e = a_j, \\ 0 & \text{otherwise.} \end{cases}$$

We claim that the load profiles $f + x^j$ are again feasible.

Let us first introduce some more terminology. Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid and let $I \in \mathcal{I}$. Let $\mathcal{D}_{\mathcal{M}}(I) = (E, A_{\mathcal{M}}(I))$ be the directed *exchange* graph defined by

$$A_{\mathcal{M}}(I) = \{(y,z) : y \in I, z \in E \setminus I, I - y + x \in \mathcal{I}\}.$$

The following proposition will be used below.

Proposition A.2 ([161]). *Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid and let $I \in \mathcal{I}$. Let $D_{\mathcal{M}}(I)$ be as defined above, and let $J \subseteq E$ be such that $|I| = |J|$ and such that $A_{\mathcal{M}}(I)$ contains a unique perfect matching on $I \Delta J$. Then $J \in \mathcal{I}$.*

Moreover, for matroids $\mathcal{M}_i = (E, \mathcal{I}_i)$, $i \in N$, let $\mathcal{D}(I_1, \dots, I_n) = (E, A)$ be the multigraph defined by the

$$A = \bigcup_{i \in N} A_{\mathcal{M}_i}(I_i).$$

We implicitly label every $a \in A$ with a player i, namely the player for which $a \in A_{\mathcal{M}_i}(I_i)$. For a path $Q = (e_1, \dots, e_p)$ in $\mathcal{D}$, we denote by A_Q^i the set of arcs corresponding to player i, i.e.,

$$A_Q^i = \{a \in Q : a = (e_j, e_{j+1}) \text{ has label } i\}.$$

Further, we let T_Q^i and H_Q^i contain the tails and heads of the arcs in A_Q^i, respectively, i.e.,

$$T_Q^i = \{v \in E : a = (v,w) \in A_Q^i\} \quad \text{and} \quad H_Q^i = \{w \in E : a = (v,w) \in A_Q^i\}.$$

We let $\sigma_Q^i : T_Q^i \to H_Q^i$ denote the bijection that maps every tail to its head. We say that a shift over the path Q is feasible, if for $i = 1, \dots, n$, it holds that $I_i - T_Q^i + H_Q^i \in \mathcal{I}_i$.[1]

[1] Subsequently, we omit the subscript Q if it is clear from the context which path is meant.

Let us now come back to the paths P_j. By definition of $\mathcal{D}(s_1, \ldots, s_n)$, every path P_j is contained in the graph $\mathcal{D}(s_1, \ldots, s_n)$. In particular this means that there is at least one path from a_j to b_j. Then there is also a *shortest path* (in terms of number of arcs) from a_j to b_j. By Lemma A.3 (given below), the load profile $f + x^j$ is feasible since we can shift players over some shortest (a_j, b_j) path Q_j such that the resulting bases are again feasible for all players. That is, we apply Lemma A.3 with the bases $I_i = s_i$.

It remains to prove Lemma A.3.

Lemma A.3. *Let $\mathcal{D}(I_1, \ldots, I_n) = (V, A)$ be as defined above, and let $a, b \in V$. If $Q = (a, \ldots, b)$ is a shortest (a, b)-path, then $I_i - T^i_Q + H^i_Q \in \mathcal{I}_i$ for all $i = 1, \ldots, n$.*[2]

Proof. Fix some i. We let $I = I_i$ and $J = I_i - T^i + H^i$ $(= s_i - T^i + \sigma^i(H^i))$. In particular, the function σ^i as defined above gives a perfect matching on $I \Delta J$. We claim that σ^i is the unique perfect matching between T^i and H^i. It follows from Proposition A.2 that $J \in \mathcal{I}$. Let ρ be an arbitrary perfect matching. Let (v, w) be the first arc on Q corresponding to player i. If $\rho(v) \neq w$, then this means that Q was not a shortest (a, b)-path, so we must have $\rho(v) = w$. A similar argument can be given for the second arc corresponding to i, then the third arc, etc. We find that $\rho = \sigma^i$, and this concludes the proof. $\square$

A.2 Local search algorithm

We can derive a local search algorithm based on the analysis above. We first introduce some terminology. We say that the difference between two strategy profiles s (with load profile f) and s' (with load profile f') is *minimal*, if there exist resources a, b such that

$$f_e - f'_e = \begin{cases} 1 & \text{if } e = a \\ -1 & \text{if } e = b \\ 0 & \text{otherwise.} \end{cases}$$

We define the neighborhood of a strategy profile s by

$$\mathcal{N}(s) = \{s\} \cup \{s' \in \times_i \mathcal{S}_i : \text{ the difference between } s \text{ and } s' \text{ is minimal}\}. \quad \text{(A.1)}$$

Note that by definition the load profiles f and f' of two neighboring strategy profiles s, s' with $s' \in \mathcal{N}(s)$, respectively, must differ by one on exactly two resources. However, this load difference might not be achievable by a single-player deviation. In fact, it is not hard to construct examples, where a sequence of unilateral deviations is needed to reach s' from s.

We prove the following lemma.

[2] A similar statement is shown in [3, Lemma 4.5]. However, our proof is different and seems (much) shorter because of the fact that we use the result in [161].

Lemma A.4. *A strategy profile s minimizes Rosenthal's potential if and only if s is a local minimum of Rosenthal's potential with respect to the neighborhood $\mathcal{N}(s)$.*

Proof. First, let s be a a strategy profile minimizing Rosenthal's potential. It follows directly that s is a local minimum with respect to $\mathcal{N}(s)$, since s is a global optimum of the potential function.

Conversely, let s be a local minimum with respect to $\mathcal{N}(s)$ and suppose that s is not a minimizer of Rosenthal's potential. We claim that there exists a strategy profile s' such that

$$\Delta(f,g) := \sum_{e\colon f_e > g_e} (f_e - g_e) c_e(f_e) - \sum_{e\colon f_e < g_e} (g_e - f_e) c_e(f_e + 1) > 0, \tag{A.2}$$

where f and g are the load profiles of s and s', respectively. Assume for contradiction that $\Delta(f,g) \leq 0$ for all feasible load profiles g. Then

$$\begin{aligned}\Phi(f) - \Phi(g) &= \sum_{e\colon f_e > g_e} \sum_{k=g_e+1}^{f_e} c_e(k) - \sum_{e\colon g_e > f_e} \sum_{k=f_e+1}^{g_e} c_e(k) \\ &\leq \sum_{e: f_e > g_e} (f_e - g_e) c_e(f_e) - \sum_{e: f_e < g_e} (g_e - f_e) c_e(f_e + 1) \leq 0,\end{aligned}$$

where the first inequality holds because the cost functions are non-decreasing and non-negative. But then f minimizes the potential function Φ, which is a contradiction.

Let the paths $W_1, \dots, W_K$ form the path decomposition of the multigraph G (as described above) for the strategies s and s'. Because of (A.2) there must be some path $W_j = (a_j, \dots, b_j)$ such that $c_a(f_a) - c_b(f_b + 1) > 0$. This contradicts the fact that s is a local minimum with respect to $\mathcal{N}(s)$. $\square$

We next show that, given an arbitrary strategy profile s, we can determine in polynomial time whether there is an improving move with respect to the altered neighborhood defined in (A.1).

Lemma A.5. *Assume that for every player $i \in N$ we have a polynomial independence oracle for matroid $\mathcal{M}_i = (E, \mathcal{I}_i)$. Then for every strategy profile s, we can check in time polynomial in m and the independence oracles of the matroids $\mathcal{M}_i$, whether or not there exists a strategy profile $s' \in \mathcal{N}(s)$ with $\Phi(s') < \Phi(s)$.*

Proof. Note that there are at most $m(m-1)$ possibilities for the resources a and b in the description of strategy profiles with minimal difference. For a and b fixed, with $c_a(f_a) > c_b(f_b + 1)$, we can in polynomial time check whether or not there exists a chain starting at a and ending in b. For example, we can run an all-pairs shortest path algorithm on the graph $\mathcal{D}(s_1, \dots, s_n)$, which can be constructed in strongly polynomial time using the polynomial independence oracles. By construction, we know that every shortest path yields a new strategy profile s' with $f'_a = f_a - 1$ and $f'_b = f_b + 1$ (and all the other loads remain the same). Note that s' can be constructed in polynomial time. □

Exploiting the insights above, we conclude that we can find a global optimum of Φ in strongly polynomial time as follows: Starting from an arbitrary strategy profile s_0, iteratively perform local improvement steps with respect to the neighborhood $\mathcal{N}(\cdot)$ as defined in (A.1) until a local optimum is reached. By Lemma A.4, the final strategy profile is a global optimum of Rosenthal's potential. Further, by Lemma A.5, each improving move can be done in polynomial time. The fact that this local search takes only a strongly polynomial number of steps follows from arguments similar to the ones in [1] (showing that any better-response sequence in matroid congestion games has polynomial length).

A.3 Example

We given an example illustrating the analysis above.

Let us consider the (non-symmetric) game in Figure A.1. The matroid of player a is the graphic matroid on the complete graph K_4 (with spanning trees as bases). The matroid of player b is a 1-uniform matroid on the set $E_2 = \{\{1,4\},\{1,2\}\}$. The strategies of the players in s are given by the bold edges, and the strategies in t by the dotted edges.

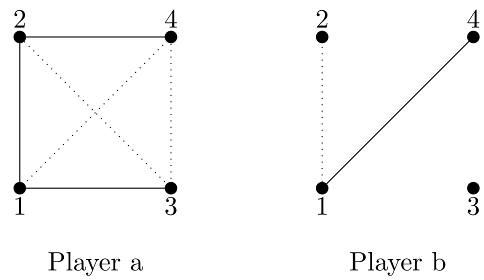

Figure A.1: Matroids and strategies of players a and b.

Let τ^a be given by $\tau^a(\{2,4\}) = \{1,4\}$, $\tau^a(\{1,3\}) = \{2,3\}$ and $\tau^a(\{1,2\}) = \{3,4\}$ (note that τ^a satisfies the condition in Proposition A.1). For τ^b there is only a unique choice defined by $\tau^b(\{1,4\}) = \{1,2\}$. The (unique) path decomposition for s and t is given in Figure A.2.

The path P_1 from $\{2,4\}$ to $\{3,4\}$ can be replaced by the shorter chain P'_1 as in Figure A.3.

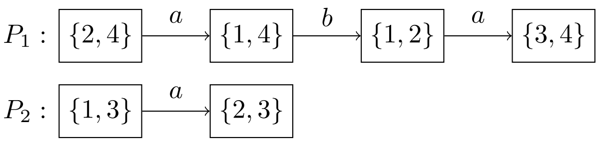

Figure A.2: Path decomposition for the profiles s and t,based on the bijections τ^1 and τ^2.

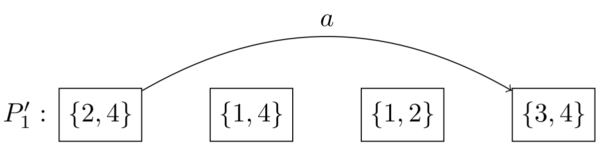

Figure A.3: The chain P_1' that is a feasible path from $\{2, 4\}$ to $\{3, 4\}$ as in Figure A.2.

Appendix B

Omitted material from Section 3.3.4

We briefly summarize the *dual greedy algorithm* of Harks et al. [99] for computing a strong equilibrium in bottleneck congestion games (see [99] for more details). Their algorithm is based on a *strategy packing oracle*.

Strategy packing oracle $\mathfrak{O}(E, (\mathcal{S}_i)_{i\in N}, (u_e)_{e\in E})$ **[99]:**	
Input:	A finite set of resources E with upper bounds $(u_e)_{e\in E}$ and strategy sets $(\mathcal{S}_i)_{i\in N}$ (given implicitly by some combinatorial property).
Output:	Strategy profile $s \in \times_{i\in N}\mathcal{S}_i$ such that $x_e(s) \leq u_e$ for all $e \in E$, or $\emptyset$ if no such strategy profile exists.

Basically, the *dual greedy algorithm* (Algorithm 2 below) works as follows: In every step of the algorithm, we have capacity constraints $(u_e)_{e\in E}$ for which it is known that there exists a strategy profile respecting these capacities. The algorithm then selects a resource e' with maximum cost (line 3) and checks if there exists a feasible strategy profile for the capacities in which $u_{e'}$ is decreased by 1. If so, then $u_{e'}$ is updated and we continue with the new vector of capacity constraints. Otherwise, the strategies of the $u_{e'}$ players using resource e' are fixed (and all variables are updated accordingly). We say that resource e' becomes *frozen*. Note that the players of which the strategies are fixed do not change anymore; these players are frozen too.

ALGORITHM 2: Dual greedy algorithm of Harks et al. [99].

Input : Bottleneck congestion game $\Gamma = (N, E, (\mathcal{S}_i), (c_e))$, strategy packing oracle $\mathfrak{O}$

Output : Strong equilibrium of Γ

set $N' = N$, $u_e = n$, $x_e = 0\ \forall e \in E$, **and** $s' = \mathfrak{O}(E, (\mathcal{S}_i)_{i \in N'}, (u_e))$
while $\{e \in E : u_e > 0\} \neq \emptyset$ **do**
  choose $e' \in \text{argmax}\{c_e(u_e + x_e) : e \in E,\ u_e > 0\}$
  $u_{e'} = u_{e'} - 1$
  if $\mathfrak{O}(E, (\mathcal{S}_i)_{i \in N'}, (u_e)) = \emptyset$ **then**
    $u_{e'} = u_{e'} + 1$
    foreach $j \in N'$ *with* $e' \in s'_j$ **do**
      $s_j = s'_j$
      set $x_e = x_e + 1, u_e = u_e - 1$ for all $e \in s'_j$
      $N' = N' \setminus \{j\}$
    end
  end
  $s' = \mathfrak{O}(E, (\mathcal{S}_i)_{i \in N'}, (u_e)\}$
end
return $s = (s_1, \ldots, s_n)$