Aspects of Wardrop Equilibria

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften
der RWTH Aachen University zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Mathematiker

Lars Olbrich

aus Lünen
in Westfalen

Berichter: Universitätsprofessor Dr. Berthold Vöcking
Universitätsprofessor Dr.-Ing. Ekkehard Wendler

Tag der mündlichen Prüfung: 22. Februar 2010

Aspects of Wardrop Equilibria

Lars Olbrich

Aachen • November 2009

Abstract

Global communication networks like the Internet often lack a central authority that monitors and regulates network traffic. Mostly even cooperation among network users is not possible. Network users may behave selfishly according to their private interest without regard to the overall system performance.

Such highly complex environments prompted a paradigm shift in computer science. Whereas traditional concepts are designed for stand-alone machines and manageable networks, a profound understanding of large-scale communication systems with strategic users requires to combine methods from theoretical computer science with game-theoretic techniques. This motivates the analysis of network traffic in the framework of non-cooperative game theory. The principal aspect of this theory is the notion of equilibrium that describes stable outcomes of a non-cooperative game.

In his seminal paper, Wardrop introduced a game-theoretic model in the 1950s for describing resource sharing problems in the context of road traffic systems. Wardrop's traffic model has attracted a lot of interest and inspired a great deal of research, especially after the emergence of huge non-cooperative systems like the Internet. In this thesis, we follow this line of research and study equilibrium situations in Wardrop's traffic model. In Wardrop's model a rate of traffic between each pair of vertices of a network is modeled as network flow, i. e., traffic is allowed to split into arbitrary pieces. The resources are the network edges with latency functions quantifying the time needed to traverse an edge. The latency of an edge depends on the congestion. It increases the more flow traverses that edge. A common interpretation of the Wardrop model is that flow is controlled by an infinite number of agents each of which is responsible to route an infinitesimal amount of traffic between its origin and destination vertex. Each agent plays a pure strategy in choosing one path from its origin to its destination, where the agent's disutility is the sum of edge latencies on this path. Note that this game-theoretic model permits extremely complex mutual dependencies among the agents' disutilities precluding application of standard optimization methods. A solution concept for this network game is provided by the theory of Wardrop equilibria. A *Wardrop equilibrium* denotes a strategy profile in which all used paths between a given

origin-destination pair have equal and minimal latency. Wardrop equilibria are also Nash equilibria as no agent can decrease its experienced latency by unilaterally deviating to another path.

Wardrop equilibria are known to possess a number of desirable properties. Foremost, they are optimal solutions to a related convex optimization problem which guarantees their existence and essential uniqueness. Moreover, Wardrop equilibria can be computed efficiently using general purpose algorithms for convex programming. All of these positive results do not hold for Nash equilibria in general games. In fact, in general games Nash equilibria are guaranteed to exist only in mixed strategies, there may exist multiple Nash equilibria, and finding a Nash equilibrium is PPAD-complete. However, like Nash equilibria in general, Wardrop equilibria do not optimize any global objective per se. In particular, the total latency of all agents is not minimized at Wardrop equilibrium. Addressing this issue, Roughgarden and Tardos gave tight bounds on the *price of anarchy* measuring the worst-possible inefficiency of equilibria with respect to the incurred latency. Further, the famous *Braess's paradox* states that adding edges to a network may in fact worsen the unique equilibrium.

The primary goal of this thesis is to provide a deeper understanding of Wardrop equilibria. We identify several problems whose solution captures the essence of Wardrop equilibria. All of the problems we analyze find their motivation in the inefficiency of Wardrop equilibria or the counterintuitive phenomenon of Braess's paradox. First, we study natural and innovative means to reduce the price of anarchy. Secondly, we analyze the stability of equilibria regarding modifications of the network environment. Finally, we propose a distributed algorithm for computing approximate equilibria.

The inefficiency of equilibria motivates our first line of research. We employ the elegant theory of mechanism design that provides a large arsenal of methods for coping with selfish behavior and turn to the question of how to improve the performance of equilibria. The goal of mechanism design is the design of protocols that interact with selfish actors following their individual objective function and steer them to a socially desirable outcome. In the context of selfish routing most prominent protocols regulate the behavior of agents by imposing taxes on the network edges. In Wardrop's model, imposing *marginal cost taxes* on every edge completely eliminates the inefficiency of selfish routing. However, in many networks there might be technical or legal restrictions that prevent an operator from imposing a tax on certain edges. Thus, we concentrate on optimal taxes for the crucial and more realistic case in which only a given *subset of the edges* can be taxed. We establish NP-hardness of this optimization problem in general networks. On the positive side, we

provide a polynomial time algorithm for computing optimal taxes in parallel link networks with linear latency functions.

We also propose a novel approach to improve the performance of selfish flow in networks by additionally routing flow, called *auxiliary flow*. In opposition to most well-established concepts designed to deal with negative effects of selfish behavior, optimal utilization of auxiliary flow is neither detrimental from an agents' perspective nor does it assume partly control over the network infrastructure or the agents. Contrary to classical taxing for instance, optimally assigning auxiliary flow does not increase the agents' disutility. We focus on the computational complexity for the optimal utilization of auxiliary flow and present strong inapproximability results. In particular, the minimal amount of auxiliary flow needed to induce an optimal flow as the outcome of selfish behavior cannot be approximated by any subexponential factor.

Further, we study the *sensitivity* of Wardrop equilibria. Whereas the notion of Wardrop equilibrium captures stability in closed systems, traffic is typically subject to external influences. However, an equilibrium would be a rare event if it were not sufficiently robust against environmental changes. Thus, from both the practical and the theoretical perspective it is a natural and intriguing question, how equilibria respond to slight modifications of either the network topology or the traffic flow. We show positive and negative results on the stability of flow pattern and flow characteristics at equilibrium. Remarkably is our finding, that an arbitrarily small environmental change may well cause the entire flow to redistribute. We also prove that the flow on every edge and the unique path latency at equilibrium are stable.

As it is fundamental for the above studies that selfish behavior in network routing games yields an equilibrium, it is not clear how the set of agents can attain an equilibrium in the first place. Moreover, the definition of Wardrop equilibrium requires agents to possess complete knowledge about the game. In previous work it was shown that an infinite set of selfish agents can approach Wardrop equilibria quickly by following a simple round-based load-adaptive rerouting policy relying on very mild assumptions only. We convert this policy into an efficient, distributed algorithm for computing *approximate Wardrop equilibria* for a slightly different setting in which the flow is controlled by a finite number of agents only each of which aims at balancing the entire flow of one commodity. We show that an approximate equilibrium in which only a small fraction of the agents sustains latency significantly above average is reached in expected polynomial time.

Zusammenfassung

Weltweite Kommunikationsnetzwerke wie das Internet können nicht zentral gesteuert werden. Benutzer solcher Netzwerke handeln eigennützig, ohne die Gesamtleistung des Systems zubeachten. Solch komplexe Strukturen führten zu einem Paradigmenshift in der Informatik. Während traditionelle Konzepte für überschaubare Netzwerke konzipiert wurden, stellt die nicht-kooperative Spieltheorie die benötigten Techniken zur Analyse von Verkehr in heutigen Netzwerken zur Verfügung.

Gegenstand dieser Arbeit sind Gleichgewichtszustände im von Wardrop in den 1950er Jahren eingeführten Verkehrsmodell. In Wardrops Modell wird Verkehr als Fluß zwischen Paaren von Knoten in einem Graphen modelliert. Latenzfunktionen beschreiben die flußabhängigen Latenz einer Kante. Eine weitverbreitete Interpretation des Modells ist, das unendlich viele Agenten jeweils einen infinitesimal kleinen Flußbetrag kontrollieren. Die Kosten jedes Agenten sind genau die Summe der Kantenlatenzen auf dem von ihm gewählten Pfad. Ein *Wardrop Gleichgewicht* ist einen Zustand, in dem jeder Agent einen latenzminimalen Pfad zwischen seinem Start- und Zielknoten gewählt hat. Es ist bekannt, dass die Netzwerklatenz in Wardrop Gleichgewichten nicht minimiert wird. Darüberhinaus zeigt das Braess Paradox, dass das Hinzufügen von Kapazität die Netzwerkleistung sogar verschlechtern kann.

In dieser Arbeit analysieren wir wichtige Probleme, die zum Verständnis der Wardrop Gleichgewichte beitragen. Es ist lange bekannt, dass wenn beliebige Steuern auf jeder Kante erhoben werden können, ein bezüglich der Gesamtlatenz optimaler Gleichgewichtsfluss erreicht werden kann. Wir untersuchen den Fall, dass Steuern nur auf einigen Kanten erhoben werden können. Für beliebige Netzwerke zeigen wir dass optimale Steuern NP-schwer zu berechnen sind. Auf der anderen Seite präsentieren wir für einfache Netzwerkstrukturen einen effizienten Algorithmus. Anschließend führen wir mit dem Konzept des Hilfsflusses einen alternativen Ansatz zur Verbesserung von Gleichgewichten ein. Wir konzentrieren uns auf die Komplexität der wesentlichen damit verbundenen Optimierungsprobleme. In einem weiteren Kapitel studieren wir die Sensitivität von Wardrop Gleichgewichten bezüglich

Änderungen entscheidender Netzwerkparameter und erhalten positive und negative Ergebnisse zu allen wichtigen Gleichgewichtsmerkmalen. Abschließend analysieren wir wie Agenten mit nur wenig Information ein Gleichgewicht erreichen können. Basierend auf einer existierenden rundenbasierten Imitationsdynamik entwickeln wir einen verteilten Algorithmus, der in erwarteter polynomieller Zeit zu einem approximativem Gleichgewicht konvergiert.

Acknowledgements

First and foremost I am grateful to my supervisor Berthold Vöcking for offering me the possibility to work in his group. I thank him for his constant support and guidance, but also for allowing me to work very independently on research topics I found interesting. I thank Ekkehard Wendler for his interest in this work and for acting as a co-referee. Thanks to the DFG Research Training Group "Algorithmic synthesis of reactive and discrete-continuous systems" for providing an inspiring research atmosphere and to the DFG for financial support.

This thesis would hardly exist without the support and input of my coauthors Matthias Englert, Simon Fischer, Thomas Franke, Martin Hoefer, Alexander Skopalik, and Berthold Vöcking. Thanks to all of you! I am further indebted to Martin Hoefer for proofreading an earlier draft of this thesis.

Not least, I bow my thanks to the entire algorithms and complexity group for constantly providing hilarious material for the *Liebling des Monats* and to the East Westphalian local reporter for referring to our community as *chaos calculators*.

Contents

List of Figures

Chapter 1

Introduction

The Internet differs in many respects from classical networks studied in computer science. Whereas traditional network optimization proceeds under the assumption of a central authority that controls the entire network, here the communication infrastructure is built and governed by a huge number of economic entities that interact in an uncoordinated and distributed fashion following their individual interest. The fact that globally optimal solutions are apparently not viable prompted a paradigmatic change in theoretical computer science. The field of *algorithmic game theory* resulted from the combination of classical methods from traditional network optimization and concepts provided by the framework of game theory.

Following this line of research, we study the game-theoretic traffic model due to Wardrop [**?**]. Introduced in the 1950s in the context of road traffic, this model captures key features of resource sharing among many selfish agents. It has been utilized to analyze many problems in transportation and communication networks. Suppose we are given a road network and a large number of agents traveling through the network from their origin to their destination. Each agent aims to minimize its experiences travel time, which is the duration needed to traverse every road segment on the selected route. Here, the time it takes to traverse a road segment is dependent on both the road segment's characteristics and its congestion, i. e., the number of agents using it. Large-scale communication networks like the Internet provide another scenario of individuals sharing the same network, where congestion effects on edges generate interdependencies between the routing decisions. More precisely, in Wardrop's traffic model a network equipped with non-decreasing latency functions mapping flow on edges to latencies is given. Between each of several source-destination pairs a certain amount of flow demand has to be routed via a collection of paths.

The situation can be described as a non-cooperative game, in which infinitely many selfish flow particles (agents) try to allocate a shortest path be-

tween their origin-destination vertices. In terms of the examples discussed above, each agent could represent a vehicle in a highway system or one of the umpteen packets traversing the world wide web every minute. An important solution concept for this network game is provided by the theory of Wardrop equilibria. A *Wardrop equilibrium* denotes a network flow that incurs equal and minimal latency on all used paths between a given origin-destination pair. Assuming that all agents select their strategies independently and rationally, such a state is a Nash equilibrium [**?**] as no arbitrary small fraction of the traffic assigned to some path can benefit from unilaterally deviating to another path. It seems only natural to study Wardrop equilibria as they represent stable states of the game.

Beckmann el al. [**?**] provided a rigorous mathematical formulation of Wardrop equilibria. They formulated the network equilibrium problem as a convex optimization problem with a single objective function. In this optimization problem a potential function has to be minimized subject to natural flow constraints. This formulation directly yields the existence, the essential uniqueness and the polynomial time computability of Wardrop equilibria.

Non-cooperative selfish behavior causes a potentially higher cost at equilibrium than in socially optimal solutions. Addressing this issue, Koutsoupias and Papadimitriou [**?**] initiated investigations of the *price of anarchy*, which measures the worst-possible inefficiency of equilibria with respect to a social welfare measure. In their seminal paper, Roughgarden and Tardos [**?**] studied the price of anarchy in the Wardrop model and gave tight bounds for several classes of networks.

A large fraction of the research on Wardrop's traffic model is motivated by the so-called *Braess's paradox*. Braess [**?**] made the seminal observation that adding extra capacity to a network may change a Wardrop equilibrium in such a way that every agent experiences a *higher* latency. This counterintuitive phenomenon stems from the non-cooperative nature of the agents: every agent minimizes its individual path latency and does not care about the experienced latency of the others.

In this thesis, we analyze Wardrop equilibria in several respects. Throughout our studies, Braess's original instance and natural extensions will serve as omnipresent benchmark networks. At first, we study two different ways to reduce the price of anarchy. Certainly, the most well-studied approach is known as taxing. The idea of taxing edges is to charge agents a fee for traversing an edge. The assumption is that tax and latency can be measured on the same scale. Agents strive to minimize their disutility, i. e., the experienced latency plus the sum of the taxes on their chosen path. The classical result states that imposing *marginal cost taxes* on every edge induces the social optimum [**?**]. A serious drawback of marginal cost pricing is that it requires *every* edge of the

network to be taxable, which may not possible for legal or technical reasons. Further, the process of collecting taxes may require an infrastructure that can be costly or impossible to establish. We consider the more general case in which only a given subset of edges may be taxed striving at minimizing the network wide performance. For this case, we give positive and negative results on the computational complexity of finding optimal taxes for different classes of networks.

As mentioned above, the concept of taxing relies on the existence of direct access to the edges and potentially costly infrastructure. Further, the agents' disutility is not minimized [**?**]. Alternative approaches to influence the behavior of selfish agents in networks as network design [**?**] or Stackelberg routing [**?**] require control over the network infrastructure or the agents, respectively.

We elaborate on the conceptually simple idea of influencing network performance by routing additional flow, which is more practicable in many scenarios. We distinguish between *auxiliary flow* and *adversarial flow*, that may be utilized to influence the routing decisions of the set of selfish agents in such a way that the induced equilibrium minimizes/maximizes the total latency of the selfish flow. Adversarial flow is loosely related to the concept of spam in the Internet, while auxiliary flow would represent "useful spam". As attractive as this approach might seem, we prove several impossibility results for optimal routings of these additional flows. Interestingly, several of our results on the computational complexity of taxing subnetworks and optimal auxiliary flow sharply contrast well-known results derived in the related field of Stackelberg routing.

Most existing literature in the context of selfish routing based on Wardrop's model focuses on the static analysis of equilibria. In the majority of cases, however, uncoordinated networks are subject to traffic fluctuations. Entities constantly enter and leave the system, they establish and remove connections among each other. Braess's paradox exemplifies that selfish behavior and the consequences of such fluctuations are non-trivial to predict. Going one step further, the notion of Wardrop equilibria serves only as a solution concept and it is not clear how an equilibrium state can be actually reached. For instance, Braess's paradox shows that Wardrop equilibria are not computable by a naive algorithm, that iteratively computes shortest paths for fractions of the flow and routes the flow accordingly.

We will address these issues in the second and third part of this thesis. Following the line of research of stability and sensitivity analysis that has received a lot of attention especially after the discovery of Braess's paradox and many similarly counterintuitive and counterproductive traffic behavior, we quantify the changes of the crucial flow characteristics due to modification of the network environment.

Finally, we study how to approximate Wardrop equilibria in a distributed fashion. Motivated by the fact that in large networks agents may not have complete knowledge about the network environment, we show how agents may *learn* Wardrop equilibria in a repeated routing game under rather weak assumptions on the agents' information about the game. Previous work [**?**] considers imitation dynamics in which agents are permitted to imitate each other concurrently. Following a clever round-based protocol the infinite set of agents can approach Wardrop equilibria quickly. We transform this protocol into a feasible distributed algorithm for computing approximate Wardrop equilibria and focus on the time until a stable state is reached.

The remainder of this chapter is organized as follows. At first, we briefly describe fundamental game theoretic concepts. Then we formally introduce Wardrop's game-theoretic traffic model. We give an overview of classical results surrounding Wardrop equilibria and discuss related work. Finally, we outline the results presented in this dissertation.

1.1 Non-cooperative Game Theory in a Nutshell

The game theoretic concepts introduced in this section provide the necessary game-theoretic knowledge for the remainder of this thesis.

A finite normal form game (or simply a *game*) is a tuple of three components $(\mathcal{N}, (\mathcal{S}_i), (u_i))$ where $\mathcal{N} = \{1, \dots, n\}$ is the finite set of agents, and each agent is equipped with a finite set of *pure strategies* $\mathcal{S}_i$ and a *cost function* $u_i : \prod_{i \in \mathcal{N}} \mathcal{S}_i \to \mathbb{R}$. Every agent i can either select a pure strategy or, more generally, a mixed strategy, i. e., it can choose a probability distribution over its strategy space $\mathcal{S}_i$. In a game, we assume that every agent is interested in *minimizing the (expected) cost*.

In his seminal dissertation, Nash [**?**] proposed a solution concept of non-cooperative equilibrium that later became known as *Nash equilibrium*. A game is at Nash equilibrium if no agent can decrease its cost by unilaterally switching to an alternative strategy. Using Brouwer's fixed point theorem, Nash proves that such a stable state is guaranteed to exist for all finite non-cooperative games if the agents are allowed to utilize mixed strategies. In fact, this groundbreaking result made Nash equilibria the most popular solution concept in game theory.

The *Bach or Stravinsky* game shows, that Nash equilibria do not need to be unique. Two opera lovers want to go to the classical concerto . One prefers a Bach concert, the other one favors Stravinsky. However, they both rather like to go to a concert together than on their own. Figure 1.1(a) depicts the cost matrix. (Negative costs can be considered as positive payoffs.) In Bach

	Bach	Stravinsky
Bach	-2/-1	0/0
Stravinsky	0/0	-1/-2

(a) Bach and Stravinsky

	Heads	Tails
Heads	1/-1	-1/1
Tails	-1/1	1/-1

(b) Matching Pennies

Figure 1.1: An entry (x/y) at position i,j of the matrix means that Player 1, the row player, experiences a cost of x and that Player 2, the column player, experiences a cost of y. In (a) there are two pure Nash equilibria, in (b) there is no pure Nash equilibrium, but only a unique mixed Nash equilibria.

or Stravinsky there are two (pure) Nash equilibria: the joint visits of Bach or Stravinsky.

Pure strategy Nash equilibria, however, are not guaranteed to exist even in very simple games. In the *Matching Pennies Game* two people simultaneously decide which side of a coin to show. One wins if they both show the same side, the other wins if one shows "heads" and the other one shows "tails". The looser of the game has to pay off the winner. The game matrix is shown in Figure 1.1(b). Both people know their own strategy, but are totally uncertain about the opponents' strategy. At the unique Nash equilibrium both people play both strategies with probability 1/2.

Its one peculiarity of matching pennies that costs add up to zero. For the special class of finite normal form games with this property, namely for non-cooperative two agent zero-sum games in which the costs of the agents add up to zero for every possible strategy selection von Neumann and Morgenstern [**?**], the founder of game theory, had proposed so-called *minimax solutions*, i. e., solutions were each agent minimizes its maximum possible loss.

1.2 Wardrop's Traffic Model

The problem of resource sharing has a long history in the transportation sciences and economics. As early as in the midst of the 19th century Kohl [**?**], a German geographer, studied the time and money consuming issue of moving people and goods between different places in the context of urban planning. Congestion effects have been explicitly factored in by Pigou [**?**] and Knight [**?**] in the 1920s, who qualitatively described *selfish routing* in transportation networks and observed that selfish behavior does not necessarily maximize the overall performance. Wardrop [**?**] introduced a formal model for selfish behavior in road networks. Since its publication 1956, Wardrop's versatile traffic

model became widely accepted within the transportation sciences. Over the last decades, Wardrop's traffic model has been reinvestigated by theoretical computer scientists since it is also well-suited for the analysis of digital traffic in communication networks. Now we will describe Wardrop's traffic model formally.

An instance of the Wardrop routing game is given by a tuple $(\mathcal{G}, d)$. $\mathcal{G} = (V, E)$ denotes a directed multi-graph with latency functions $\ell = (\ell_e)_{e \in E}$

$$\ell_e : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$$

attached to the edges. We assume the latency functions to be non-decreasing, differentiable and *semi-convex*, i. e., that $x \cdot \ell_e(x)$ is convex. We explicitly mention if E is equipped with latency function different from ℓ. Furthermore, we are given a set of commodities $[k] = \{1, \ldots, k\}$ specified by source-sink pairs $(s_i, t_i) \in V \times V$ and flow demands d_i, where we can assume without loss of generality pairwise disjoint sets (s_i, t_i) for $i \in [k]$. The total demand is $d = \sum_{i \in [k]} d_i$. We call an instance *single-commodity* if $k = 1$ and *multi-commodity* if $k > 1$. Considering single-commodity instances, we drop the index i and set $d = d_1$.

Let $\mathcal{P}_i$ denote the admissible acyclic paths of commodity i, i. e., all acyclic paths connecting s_i and t_i, and let $\mathcal{P} = \bigcup_{i \in [k]} \mathcal{P}_i$. For $P \in \mathcal{P}$ let f_P denote the volume of agents on path P. A path flow vector $(f_P)_{P \in \mathcal{P}}$ induces an edge flow vector $(f_{e,i})_{e \in E, i \in [k]}$ with $f_{e,i} = \sum_{P \in \mathcal{P}_i : e \in P} f_P$. The total flow on edge e is

$$f_e = \sum_{i \in [k]} f_{e,i} = \sum_{i \in [k]} \sum_{P \in \mathcal{P}_i : e \in P} f_P = \sum_{P \ni e} f_P \ .$$

The latency of an edge $e \in E$ is given by $\ell_e(f_e)$. The total latency of an edge e is given by $\ell_e(f_e) \cdot f_e$. Slightly abusing notation, we denote $(f_P)_{P \in \mathcal{P}}$, $(f_{e,i})_{e \in E, i \in [k]}$ and $(f_e)_{e \in E}$ by f. A flow f is *feasible* either if $f_P \geq 0$ for $P \in \mathcal{P}$ and it satisfies the flow demands

$$\sum_{P \in \mathcal{P}_i} f_P = d_i$$

for all $i \in [k]$, or if it is induced by such a path flow. In this thesis, we only consider the set of feasible flows denoted by $\mathcal{F}$. The latency of a path $P \in \mathcal{P}$ is given by the sum of the edge latencies

$$\ell_P(f) = \sum_{e \in P} \ell_e(f_e) \ .$$

Note that the path latency is not a function of the corresponding path flow, because it depends on the total flow on each of its edges.

Definition 1 (Total latency)**.** *The total latency of a flow f is defined as*

$$C(f) = \sum_{P \in \mathcal{P}} \ell_P(f) f_P \ . \tag{1.1}$$

We drop the argument f whenever it is clear from the context.

Since the edge latency depends solely on the edge flow, the total latency can also be expressed in terms of edge flows only:

$$C(f) = \sum_{P \in \mathcal{P}} \left(\sum_{e \in P} \ell_e(f_e) \right) f_P = \sum_{e \in E} \left(\sum_{P \in \mathcal{P}: e \in P} f_P \right) \ell_e(f_e) = \sum_{e \in E} \ell_e(f_e) f_e \ .$$

1.2.1 Wardrop Equilibria

A natural goal for a central authority is to compute a routing f that minimizes the total latency over all commodities. This min-cost flow problem can be formulated as the following non-linear program:

$$\min_{f \in \mathcal{F}} C(f) \ ,$$

where the feasible set $\mathcal{F}$ can be expressed by a polynomial number of flow conservation and non-negativity constraints. Since the latency functions are continuous, $\mathcal{F}$ is a compact set and an optimal flow exists. Since $\ell(x) \cdot x$ is convex, we can apply the concepts of convex programming and can use, e. g., the ellipsoid method [**?**] to compute an optimal flow up to a small error term in time polynomial in the size of the instance and the number of bits of precision. This error term is unavoidable since the description of an optimal solution may require irrational numbers even if the input contains only natural numbers. For solving the classical problem of efficiently computing a minimum cost multi-commodity flow, there are also several specific algorithms known. For an overview see, e. g., [**?**] and [**?**]. Note that polynomial time computability relies crucially on the semi-convexity of the latency function, as for the general multi-commodity case no fast algorithms are known.

Taking the game theoretic perspective, we envision the flow as composed of an infinite number of agents each of which carries an infinitesimal amount of flow. Each agent plays a pure strategy in selecting one path from its origin to its destination, where the agent's cost is the chosen path's latency. Adjusting the definition of Nash equilibria to games with infinitely many agents we require that *no arbitrarily small fraction of the agents* can be shifted from their path to another without increasing their latency.

Definition 2 (Nash equilibrium)**.** *A feasible flow f is at* Nash equilibrium *if for every commodity $i \in [k]$, all paths $f_{P1}, f_{P2} \in P_i$ with $f_{P1} > 0$, and every $0 \leq \varepsilon \leq f_{P_1}$ it holds that*

$$\ell_{P_1}(f) \leq \ell_{P_2}(\tilde{f}) \ ,$$

where $\tilde{f}$ is obtained from f by shifting an amount of ε from P_1 to P_2, i. e.,

$$\tilde{f}_P = \begin{cases} f_P - \varepsilon & \text{if } P = P_1 \\ f_P + \varepsilon & \text{if } P = P_2 \\ f_P & \text{otherwise.} \end{cases}$$

Since the latency functions are continuous and non-decreasing, a flow at Nash equilibrium can be nicely characterized as a flow obeying the "First Principle of Wardrop" **[?]** or being at *Wardrop equilibrium*. A flow is at Wardrop equilibrium if all used paths of the same commodity have minimal latency whereas unused paths may have larger latency.

Lemma 1. *A feasible flow f is at Nash equilibrium if and only if for every commodity $i \in [k]$ and all paths $f_{P1}, f_{P2} \in P_i$ with $f_{P1} > 0$ it holds that*

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

The total latency of flows at Wardrop equilibrium can easily be expressed, which will come in handy several times throughout this thesis.

Lemma 2. *The total latency of a flow f at Wardrop equilibrium can be expressed as*

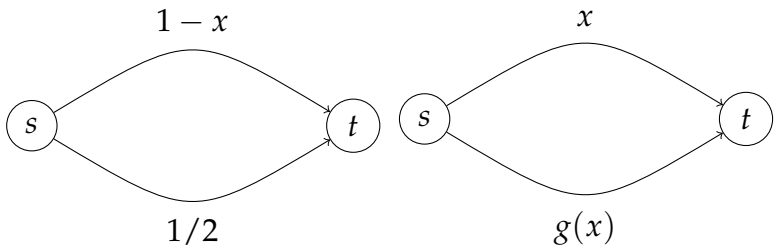
$$C(f) = \sum_{i \in [k]} L_i(f) \cdot d_i \ ,$$

where $L_i(f)$ denotes the unique path latency of an equilibrium flow in commodity i.

Note that Wardrop equilibria and Nash equilibria are two related paradigms that describe a stable network flow as a function of environmental characteristics. Yet, for arbitrary latency functions Wardrop equilibria and Nash equilibria do not coincide. Consider the network shown in Figure 1.2(a). If half of the demand is being routed over both links each, the flow is at Wardrop equilibrium. However, sending the entire flow over the lower link constitutes the unique Nash equilibrium. In Figure 1.2(b), the unique Nash equilibrium is reached if $2/3$ of the flow routes over the upper edge and $1/3$ over the lower edge with the non-continuous latency function, even though the path latencies differ, and the flow is not at Wardrop equilibrium. In fact, there is no Wardrop equilibrium for this routing instance. We conclude that our assumptions of continuity and monotonicity of the latency functions are necessary and sufficient for Lemma 1 to hold.

Moreover, these assumptions are reasonable in applications where cost typically represents a quantity that only increases with the network congestion, delay being the prime example.

Wardrop equilibria and optimal flows exhibit a striking similarity. Both flows are Nash equilibria with respect to a different set of latency functions.



(a) (b)

Figure 1.2: A unit demand needs to be routed from s to t. The edges are labeled with their latency functions, where $g(x) = x$ for $x \leq 1/3$ and $g(x) = 1$ for $x > 1/3$. In (a) Wardrop equilibrium and Nash equilibrium differ. In (b) no Wardrop equilibrium exists.

Definition 3 (Marginal cost function)**.** *If ℓ is a differentiable function, then*

$$\ell^*(x) = \frac{d}{dx}(x \cdot \ell(x)) = \ell(x) + \ell'(x)\, x$$

denotes the corresponding marginal cost function*.*

Note that the marginal cost function of a latency function ℓ_e consists of two terms $\ell_e(x)$ and $\ell'_e(x)\, x$. The first captures the per-unit latency incurred by additional flow whereas the second accounts for the per-unit increased latency of the flow that is already using the edge.

Theorem 3. *[?, ?] Let $(\mathcal{G}, d)$ be an instance with latency functions ℓ_e for all $e \in E$. Then a flow f is optimal with respect to $(\ell_e)_{e\in E}$ if and only if f is at Wardrop equilibrium with respect to $(\ell^*_e)_{e\in E}$.*

The idea of the proof of Theorem 3 is the following. For contradiction, assume that a minimal latency flow uses paths with suboptimal marginal costs. Hence, there are paths $P_1, P_2 \in \mathcal{P}$ with $f_{P_1} > 0$ and $\ell^*_{P_1}(f) > \ell^*_{P_2}(f)$. Since the marginal costs are continuous $\ell^*_{P_1}(f - \delta) > \ell^*_{P_2}(f + \delta)$ holds for a sufficiently small $\delta > 0$. However, this flow shift changes the total latency by $(-\ell_{P_1}(f) + \ell_{P_2}(f)) \cdot \delta < 0$.

Theorem 3 establishes not only a deep connection between optimal flows and Wardrop equilibria but in fact yields an important existence and uniqueness result.

Theorem 4. *[?] The set of Wardrop equilibria coincides with the set of solutions of the following convex program:*

$$\min_{f\in\mathcal{F}} \sum_{e\in E} \int_0^{f_e} \ell_e(u)\, du \ .$$

Thus, every instance $(\mathcal{G}, d)$ admits a Wardrop equilibrium and every Wardrop equilibrium induces the same edge latencies. Further, a Wardrop equilibrium can be computed in polynomial time.

This theorem holds even for latency function that are not semi-convex. Particularly useful is the fact that the objective function $\Phi(f) = \sum_{e \in E} \int_0^{f_e} \ell_e(u)\, du$ serves as a *potential function* as it precisely absorbs progress: If an infinitesimal amount of flow du is shifted from path P_1 to P_2, thus improving its latency by $\ell_{P_1} - \ell_{P_2}$, the potential decreases by $(\ell_{P_1} - \ell_{P_2})du$. We will make use of this fact frequently. The existence of a potential function is sufficient to guarantee the existence of at least one Wardrop equilibrium. Let f be a flow minimizing the potential function Φ. If an infinitesimal amount of flow du is shifted from path P_1 to path P_2, transforming the flow f to f', it follows that $\ell_{P_2} - \ell_{P_1} = \Phi(f') - \Phi(f) \geq 0$. Hence, the fraction of deviating agents could not benefit from the migration move.

Since Wardrop equilibria are guaranteed to exist in pure strategies, they constitute the most appealing solution concept. However, there are routing scenarios that do not satisfy common game theoretic assumptions needed for the motivation of Wardrop equilibria such as accurate knowledge of the network and its latency functions. Further, agents may incur some costs when they change their strategy. Thus, it is reasonable to assume that an agent only switches its path for a significant latency gain. This assumption leads to the notion of a popular, slightly weaker notion of Wardrop equilibria. A $(1+\varepsilon)$-*approximate Wardrop equilibrium* is a state in which no arbitrary small fraction of agents can reduce their latency by more than a multiplicative factor of $(1+\varepsilon)$ by unilaterally migrating to another path. We will comment on several alternative solution concepts in Section 1.6.8.

1.3 The Price of Anarchy

It is well-known in economics and in traditional game theory that selfish behavior can yield a socially suboptimal outcome. The famous *Prisoner's dilemma* exemplifies this. Two people are arrested by the police being suspected of a crime. They are interrogated separately and simultaneously such that they have no chance to communicate or to coordinate their statements. Both suspects can either confess the crime or deny having done anything. If both confess, they are sentenced to go to prison for 5 years each. If both deny, they go to prison for only 1 year each because of lack of clear evidence. However, if they choose different strategies, the confessor is released and the denier is sent to prison for 8 years (see Figure 1.3(a)). At the unique Nash equilibrium, both suspects confess the crime. In terms of the total number of years or the maxi-

	Confess	Deny
Confess	5/5	8/0
Deny	0/8	1/1

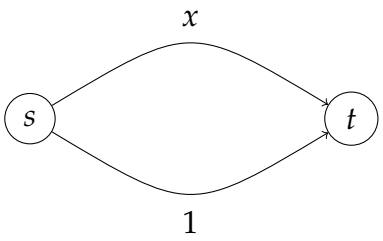


Figure 1.3: (a) Nash equilibria in the Prisoner's Dilemma can be arbitrarily bad. (b) A unit demand needs to routed from s to t. The edges are labeled with their latency functions ℓ_1 and ℓ_2. At equilibrium the entire demand utilizes the upper edge. Socially desirable, however, is splitting traffic evenly among both paths

mum number of years spent in prison denying the crime is the socially optimal (from the suspects' point of view) strategy for both suspects. The equilibrium situation degrades arbitrarily by increasing the sentences in case of confession.

In the context of selfish routing the degradation of performance was already observed by Pigou [**?**]. Braess [**?**] noticed that selfish behavior can in fact be worse for all agents. Interestingly, the natural problem of quantifying this degradation as not been addressed explicitly before the rise of the Internet. In 1999, Koutsoupias and Papadimitriou [**?**] proposed to investigate the *coordination ratio* which they defined as the worst case ratio between the social cost at Nash equilibrium and the optimal social cost. Later, Papadimitriou [**?**] dubbed this measure the *price of anarchy*.

Note that there are obvious structural similarities to other established concepts in theoretical computer science. In particular, the notion of the price of anarchy is related to the approximation ratio measuring the performance loss due to lack of computational power of approximation algorithms [**?**] and to the competitive ratio measuring the performance loss due to lack of perfect information of online algorithms [**?**]. In the same spirit the price of anarchy quantifies the loss of performance due to lack of a central authority. A small price of anarchy indicates that every equilibrium is a good approximation of a socially optimal state.

Over the last ten years, equilibrium efficiency analyses have been conducted in a large variety of games, such as job scheduling, facility location and network design (for an extensive overview see [**?**]). Arguably routing games are among the most successfully analyzed applications. In Wardrop routing games the total latency $L(f)$ is the most common performance measure.

Definition 4 (Price of anarchy)**.** *[?,?] The* price of anarchy *for an instance* $(\mathcal{G}, d)$ *is defined as*

$$\rho(\mathcal{G}, d) = \frac{C(f^*)}{C(f)} \ ,$$

where f and f^* denote an optimal flow and an equilibrium flow, respectively. The price of anarchy for a set of instances $\mathcal{I}$ is

$$\rho(\mathcal{I}) = \sup_{(\mathcal{G},d)\in\mathcal{I}} \rho(\mathcal{G},d) \ .$$

Note that by Theorem 4 every Wardrop equilibrium incurs the same total latency and the price of anarchy is well-defined.

Pigou's example [**?**] (Figure 1.3(b)) exemplifies that selfish routing does not optimize social welfare in general. Assume there is one unit of traffic routing itself from s to t. At the unique equilibrium every agent routes via the upper edge which incurs a total latency of 1. Following Theorem 3 the minimum cost flow solves $\ell_1^*(f_1) = \ell_2^*(1 - f_1)$, which holds if the flow is split evenly. While the agents on the upper edge experience a latency of $1/2$, the agents on the lower edge incur a latency of 1. This minimum cost flow incurs a total latency of $1/2 \cdot 1/2 + 1/2 \cdot 1 = 3/4$. The minimum cost flow is not at equilibrium since a small fraction of selfish agents currently using the lower edge experiences a latency of 1 and could improve their latency by switching to the upper edge. A switch would deteriorate the total latency since it would slightly increase the latency of a large fraction of the selfish agents. Thus, the fact the agents ignore the latency increase their decisions imposes on the other agents is the reason why equilibria are inefficient in general. In Pigou's example, the price of anarchy is $\frac{1}{3/4} = 4/3$. The inefficiency can be amplified by changing the non-constant latency function to $\ell_1(x) = x^p$ for some large integer $p > 0$. The equilibrium flow remains the same, but in the optimal flow almost all agents are routed over the upper edge and the total latency vanishes for large p. The price of anarchy can be computed as roughly $p/\log p$.

In their ground-breaking work Roughgarden and Tardos [**?**] analyze the price of anarchy in Wardrop's model. In fact, they show that the price of anarchy equals $4/3$ for linear latency functions and $\Theta(p/\log p)$ for polynomial latency functions with non-negative coefficients of degree at most p. Later, improved bounds on the price of anarchy for special classes of polynomial latency functions were given [**?**] . Whereas the set of latency functions were identified as the crucial parameter for the price of anarchy, the network topology is irrelevant [**?**, **?**]. In particular, Roughgarden [**?**] presents a simple procedure for computing the price of anarchy by proving that the worst-case ratio is already achieved on parallel links (see also Correa *et al.* [**?**]). Observe that in this regard Pigou's example depicted in Figure 1.3(b) exhibits the worst possible price of anarchy among all networks with polynomial latency functions. Complementing the result that the inefficiency of equilibria cannot be bounded in general, Roughgarden and Tardos [**?**] show that the total latency at Wardrop equilibrium is upper bounded by the total latency of an optimal flow rout-

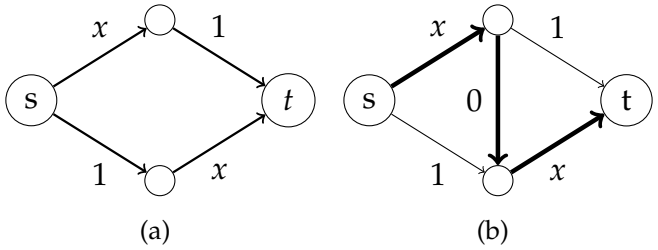


Figure 1.4: Again a unit demand needs to be routed from s to t. In the left network optimal solution and equilibrium coincide and traffic is split among both paths. After adding an extra edge, at equilibrium the flow utilizes the zig-zag-path incurring a higher latency.

ing twice the demand. In the special case with latency functions of the form $\ell_e(f_e) + \ell'_e(f_e) \cdot f_e = C \cdot \ell_e(f_e)$ for all $e \in E$ and some constant $C > 0$, the price of anarchy equals 1 [**?**]. In an inverse line of research, Roughgarden [**?**] analyzes the unfairness of the optimal solution in terms of the worst case ratio between a path latency at optimum and at Wardrop equilibrium. By definition the price of anarchy is a worst case measure. Like other traditional worst case measures in theoretical computer science it has often been criticized for being too pessimistic. For a promising approach of average case analysis of the price of anarchy see [**?**].

1.4 Braess's Paradox

A famous result on selfish routing in congested networks is the so-called *Braess paradox*. Braess [**?**] made the astonishing observation that adding extra capacity to a network may change a Wardrop equilibrium in such a way that every agent experiences a *higher* latency.

Consider the small network depicted in Figure 1.4(a). As in Pigou's example assume one unit of selfish traffic traveling from s to t. At equilibrium (and optimum) half of the agents take the upper path while the other half selects the lower path. In this case, the experienced path latency of every agent (and the total latency) is $3/2$. The addition of an edge as shown in Figure 1.4(b) yields Braess's original network. Now the entire flow uses the zig-zag-path at equilibrium, which increases the path latency of every agent (and the total latency) to 2.

Braess's Paradox fueled a huge amount of research up to today. Many researchers elaborated on Braess's Paradox in the Wardrop model [**?**,**?**,**?**,**?**] and related models [**?**,**?**,**?**,**?**,**?**,**?**]. Braess's Paradox further prompted the search for other counterintuitive observations in traffic networks [**?**,**?**,**?**,**?**].

Roughgarden [**?**] gave results on the severeness of this phenomenon in the Wardrop model. In networks with n vertices removing a set of edges may decrease the total latency by a factor of $\lfloor n/2 \rfloor$, which gives a tight bound. By removing at most k edges from a given network, the total latency can be improved by at most a factor of $k+1$. Yet, since for networks with linear latency functions the price of anarchy equals $4/3$ [**?**], Braess's original four vertex network exhibits the worst case manifestation of Braess's paradox for this class of networks. Even though Braess's paradox is common in large random graphs [**?**, **?**], it is hard to detect [**?**].

We want to emphasize that Braess's paradox is far from being merely an academic curiosity, as it has been observed many times in large road networks. For instance, the (temporal) closure of central roads in highly jammed traffic areas around the globe improved the total traffic flow notably [**?**, **?**, **?**]. In an analytical approach, Youn *et al.* [**?**] estimated the price of anarchy with respect to the travel times in road networks of several major cities to be roughly 1.3 and identified several roads, whose closure may improve traffic situation.

Remarkably, the occurrence of Braess's Paradox is not confined to selfish behavior. Similar effects have been observed in mechanical and electronic systems [**?**], indicating that also physical equilibrium principles do not always pilot the network system to optimal states.

As another line of research stimulated by the paradoxical behavior of selfish routing, stability and sensitivity analysis of equilibrium traffic characteristics have received a lot of attention. The outstanding result by Dafermos and Nagurney [**?**, **?**] states that equilibrium flow patterns depend continuously upon the demands and latency functions. In other words, small changes in the travel demands or in the latency functions induce small changes in the edge flows, path flows, and path latency at Wardrop equilibrium. In particular, for single-commodity networks the path latency at equilibrium is a monotone increasing function of the input demand. Further, they identified the structure of networks in which Braess's paradox occurs.

1.5 Reducing the Price of Anarchy

A large portion of current research is dedicated to quantifying the price of anarchy in Wardrop's traffic model. While this work is vital, it is even more valuable to design methods to reduce the inefficiency of selfish flow in scenarios with no central control. To this end, several approaches have been studied. Generally, the goal is to design a protocol that interacts with selfish agents following their individual objective and steer their incentives to a socially desirable outcome. In this section we will summarize known results about methods

to reduce the price of anarchy. We focus on introducing taxes [**?**, **?**], designing "good" networks [**?**] and controlling a subset of the agents centrally [**?**].

1.5.1 Taxes

In the context of selfish routing most prominent protocols regulate the equilibrium by the utilization of *economic means* in form of taxes. The idea of taxing is to charge agents a fee for traversing an edge. In other words, a tax $\tau_e \geq 0$ on an edge $e \in E$ raises the perceived disutility from $\ell_e(f_e)$ to $\ell_e(f_e) + \tau_e$. Subsequently, every agent selects a path minimizing its disutility, i. e., its experienced latency plus the sum of the taxes on the chosen path. The effectiveness of such taxes has been observed by Pigou [**?**] and generalized by Beckmann *et al.* [**?**]. Theorem 3 yields the fundamental result that imposing *marginal cost taxes* $\tau_e = \ell'_e(f_e) \cdot f_e$ induces the social optimum [**?**], where f denotes an optimal flow. In other words, if each agent on the edge has to pay a tax equal to the additional cost its presence causes for the other agents on the edge, one entirely eradicates the inefficiency of selfish behavior. This classic result holds since the agents are homogeneous with respect to their sensitivity to taxes. If we generalize the model to the heterogeneous case in which every agent trades off money and time in an individual manner and minimizes a weighted sum of the edge latencies and the edge taxes, marginal cost taxing does not remain optimal. Early work on taxes for *heterogeneous agents* considered unsatisfying agent-specific taxes on the edges [**?**, **?**, **?**]. Later, Cole *et al.* [**?**] were the first to consider the problem from the view of theoretical computer science. They give a non-constructive existence proof for taxes stabilizing the optimal flow in single-commodity networks and upper bound the size of the maximal tax necessary. Fleischer [**?**] reduces the bounded on the required taxes to linear functions and gives an algorithm for computing optimal taxes for series-parallel networks. In following work, the existence of taxes was proved constructively for multi-commodity networks [**?**,**?**,**?**]. Even more, Fleischer *et al.* [**?**] shows the existence of taxes that induce optimal flows for several alternative objectives, such as minimum average weighted latency and minimum maximum latency.

The underlying assumption of the above mentioned work is that taxes can be returned to the agents and therefore the network performance is determined entirely by the total latency. However, there may arise situations, in which the refunding process could be costly or infeasible. In this case we need to consider *non-refundable taxes*, that minimize the total disutility (latency plus taxes) of the agents. Under this assumption, marginal cost pricing does not improve the cost of Wardrop equilibria for linear latency functions [**?**] . But alternative tax functions can still be beneficial as the Braess network exemplifies. In networks with linear latency functions there are optimal taxes that are

either 0 or ∞ on each edge [**?**]. Still, optimal taxes are hard to approximate [**?**]. Whereas for networks equipped with linear latency functions the trivial algorithm, i. e., imposing no taxes at all, yields a $4/3$-approximation of the social optimum [**?**], it is NP-hard to approximate the social optimum within $(4/3 - \varepsilon)$ for every $\varepsilon > 0$.

1.5.2 Network Design

Braess's paradox shows that removing edges from a network may *improve* equilibrium performance. More precisely, in networks with n vertices removing a set of edges may decrease the total latency by a factor of $\lfloor n/2 \rfloor$ [**?**]. However, this approach is restricted since it does not even reduce the price of anarchy on parallel links networks. Roughgarden [**?**] considered the computational complexity of detecting a subnetwork of a given network with n vertices exhibiting the best equilibrium and presented inapproximability results and naive optimal approximation algorithms. In particular, whereas a $(n/2 - \varepsilon)$-approximation algorithm is NP-hard to compute, the trivial algorithm, i. e., choosing the entire network as the optimal subnetwork, is a $n/2$-approximation. Note that by imposing a sufficiently large tax on an edge one can simulate the removal of that edge. Thus, the network design problem can be seen as a special case of the taxing problem. As Roughgarden [**?**] points out, in selfish routing the difference between linear and nonlinear latency functions is most often only quantitative, as bounds on the price of anarchy show. Yet, there is a qualitative gap in the relative power of taxes to the power of edge removals. When moving from linear to non-linear latency functions. While for linear latency functions edge removal is as powerful as taxing [**?**], the benefit of taxes exceeds the benefit through edge removal by $\mathcal{O}(n)$ for non-linear latency functions.

1.5.3 Stackelberg Routing

Taxing and network design intend to reduce the price of anarchy by directly modifying the network topology. *Stackelberg routing* [**?**] is an alternative approach to mitigate the negative effects of selfish behavior in congested networks. The idea of Stackelberg routing is to route a fraction of flow centrally such that the latency of all flow is optimized at equilibrium. In Stackelberg routing, one assumes that an ε-fraction of the demand is controlled by a central authority, the Stackelberg leader, while the remaining $(1 - \varepsilon)$-fraction is controlled by non-atomic selfish agents. In a first phase, the Stackelberg leader fixes the routes for its fraction of the demand. In a second phase, the selfish agents enter the system and route their own flow on top of the leader demand.

The objective of the leader is to minimize the resulting total cost of the total (both leader and selfish) flow, while the selfish agents solely aim to minimize their experienced path latency. One important application of Stackelberg routing is the routing of Internet traffic within the domain of an Internet service provider [**?**]. Here, the Internet service provider centrally controls a fraction of the overall traffic traversing its domain.

The problem of the computational complexity of an optimal leader strategy is essentially solved. An optimal leader strategy is NP-hard to compute even for parallel links with linear latency functions [**?**] but the problem allows an FPTAS [**?**]. There are polynomial time algorithms to compute the minimal portion of flow needed by the leader to induce optimum cost [**?**] and the minimal value of the Stackelberg leader's demand that can improve the price of anarchy [**?**]. On the algorithmic side, Roughgarden [**?**] introduces an easy-to-implement Stackelberg strategy that reduces the price of anarchy for arbitrary latency functions on parallel links to a constant factor of $1/\varepsilon$. Thus, by controlling only a small amount of flow, the performance of equilibria can be dramatically improved. This does not remain true in arbitrary single-commodity networks [**?**]. On the positive side, Swamy [**?**] presents latency-class specific bounds on the price of anarchy in arbitrary multi-commodity networks. The obtained bounds yield a continuous trade-off between the amount of flow controlled and the price of anarchy (see also [**?**]).

1.6 Extensions and Variations

Wardrop's traffic model was originally introduced in [**?**] to model selfish behavior in road networks. Since it is also well-suited for the analyses of uncoordinated communication networks like the Internet, the model has attracted the interest of theoretical computer scientists over the last 10 years. In this section, we review various ramifications and extensions of the model that have been analyzed and outline the results that have been obtained therein.

1.6.1 Nonatomic Routing Games

In Wardrop routing games the action of a every agent has essentially no effect on the choices of the other agents. Games that possess this property are referred to as nonatomic. General *nonatomic non-cooperative games* have been introduced by Schmeidler [**?**] in the early 1970s. A nonatomic game is defined as a game in which a continuum of agents is equipped with a nonatomic measure. Strategies and cost functions can be defined similarly as for finite normal form games. However, in the case of infinitely many agents we do not need to differentiate between pure and mixed strategies. Schmeidler [**?**] gave existence

proofs for equilibria, thereby greatly generalizing the results on the existence of Wardrop equilibria [**?**] as stated in Theorem 4.

Whereas general nonatomic games are a very general concept, Wardrop's traffic model exhibits a much richer structure. Firstly, the strategy set of the agents is quite restricted as it contains only paths between the respective sources and sinks. Secondly, and more importantly, the latency of an edge does not depend on the identities but only on the measure of agents choosing this edge. The latter is indeed one of the main characteristics of congestion sensitive networks in general.

1.6.2 Congestion Games

Wardrop's model assumes an infinite number of agents. In some real-world applications, however, there are a finite number of agents competing for shared resources. To suitably model these situations Rosenthal [**?**] introduced *congestion games* in 1973. In a congestion game there are given a finite set of resources and a finite set of agents of non-negligible size. Each agents' strategy consists of a subset of the resources. The cost of a strategy is the sum of the latencies of the chosen resources, and the cost for choosing a resource depends only on the number of agents including this resource in their strategy sets. Congestion games are a discrete version of Wardrop games.

Rosenthal [**?**] provided a potential function for congestion games proving the existence of pure Nash equilibria. In fact, the class of congestion games coincides with the rich and broad class of potential games [**?**]. Rosenthal's potential function resembles the potential function given by Beckmann *et al.* [**?**] for the Wardrop model, but the potential function yields a non-convex optimization problem that allows for multiple pure Nash solutions. Correspondingly, congestion games allow for multiple equilibria. Further, in congestion games a Nash equilibrium can be achieved without incurring the same latency to all agents, contrary to the "First principle of Wardrop". On the positive side, Rosenthal's work implied that sequential best-response dynamics in congestion games converge to a pure Nash equilibrium.

While Wardrop equilibria can be computed efficiently, it is PLS-complete to compute a pure Nash equilibrium [**?**] in congestion games, i. e., there is no efficient algorithm for computing pure Nash equilibria unless $PLS \subseteq P$. This also holds for linear cost functions [**?**]. Skopalik and Vöcking [**?**] prove that even pure $(1+\varepsilon)$-approximate Nash equilibria, i. e., states in which no agent can decrease its latency by more than a factor of $(1+\varepsilon)$ by unilaterally changing its strategy, are PLS-complete to compute. On the other hand, approximate equilibria in congestion games in which the strategy spaces of the

agents coincide (*symmetric congestion games*) can be computed efficiently under mild smoothness conditions on the latency functions [**?**].

Most related to Wardrop routing games are *network congestion games*. In network congestion games the strategy sets of the agents are presented implicitly as paths in a network. Fabrikant *et al.* [**?**] show that Nash equilibria are efficiently computable for symmetric network congestion games using a reduction to min-cost flow. However $(1+\varepsilon)$-approximate Nash equilibria are still PLS-complete to compute in general network congestion games [**?**]. Feldmann *et al.* [**?**] identify properties that latency functions from natural classes have to satisfy in order to guarantee that an approximate Nash equilibrium can be computed in polynomial time.

As in the Wardrop model, in congestion games the degradation with respect to the total latency due to selfish behavior is well understood. The price of anarchy for linear latency functions is $5/2$ and $p^{\Theta(p)}$ for polynomial latency functions of degree p [**?**, **?**]. Aland *et al.* [**?**] give the exact price of anarchy for polynomial latency functions. Note that the price of anarchy in congestion games is much larger than in the Wardrop model. In both cases the set of allowed latency functions the crucial parameter and the price of anarchy is independent of the network topology.

A wide range of results for special classes of congestion games and a variety of social cost functions have been studied ([**?**,**?**,**?**,**?**,**?**,**?**]). For instance, the price of anarchy for parallel links with linear latency functions with respect to the maximum latency is $\Theta(\log m / \log\log m)$ [**?**].

As an alternative game-theoretic measure to the price of anarchy, Anshelevich *et al.* [**?**] introduced the *price of stability* as a worst-case ratio, over all instances, between the social cost of the *best* equilibrium (instead of the *worst*) and optimum social cost. The idea is that if a central authority is enabled to initially set up a solution that selfish agents are free to adopt subsequently, the best equilibrium is the prime selection. In other words, the price of stability measures the inevitable performance degradation due to the selfishness of the agents. First work shows, that for linear latency functions the price of stability is approximately $8/5$ [**?**]. For results in related models see [**?**, **?**, **?**, **?**].

Despite the considerable interest in optimal tax functions for congestion games [**?**, **?**, **?**], it is - unlike in Wardrop's model- still unknown whether there exist optimal taxes for atomic congestion games.

1.6.3 Splittable Flow

In a natural generalization of Wardrop's model, finitely many agents control a non-negligible fraction of the entire demand each. One interpretation of this setting is that agents of a commodity form coalitions to reduce the expected

latency faced by the agents in the coalition, under the assumption that all agents within a coalition are randomly assigned to the different paths used by the coalition. Motivating scenarios are route guidance systems recommending optimal routes to its users or freight companies dictating transportation routes to its truck fleet. Observe that the Wardrop model emerges as a special case in which infinitely many agents are allowed, each of them controlling a negligible amount of flow. Orda *et al.* [**?**] introduced this model and showed that Nash equilibria exist under certain conditions. Uniqueness results were obtained only for some special cases [**?**, **?**, **?**]. In fact, this model allows for multiple equilibria in general [**?**] even for only two players.

In this model the price of anarchy is not well understood yet. Some finite upper bounds on the price of anarchy for polynomial latency functions of low degrees are known [**?**, **?**], but there is still a large gap between known upper and lower bounds for polynomial latency functions of arbitrary degree [**?**, **?**]. The price of anarchy in congestion games with splittable flow can be worse than in the Wardrop game [**?**].

In light of the possibility of multiple equilibria [**?**], the situation with regard to taxing seems worse than in the Wardrop case. Nevertheless, there exists an optimal tax function for multi-commodity networks even in the presence of heterogeneous agents, in the sense that the optimal solution is realized as *some* equilibrium via taxes ([**?**], see also [**?**, **?**]). Hay *et al.* [**?**] consider *collusion games*, a variant of splittable flow games in which agents traveling between a source-sink pair may form arbitrary coalitions and measure the degradation of performance due to this behavior.

1.6.4 General Latency Functions

Wardrop's model has been extended over the years in various manners. Sticking to an infinite number of agents, one straightforward way is to allow more general latency functions. Following this line, agent-specific latency functions allow to model agents with different preferences. Gairing *et al.* . [**?**] concentrate on existence results of equilibria and give bounds on the price of anarchy. Agent-specific latency functions have also been considered in congestion games [**?**, **?**].

Most of the literature on Wardrop's traffic model deals with the case of *separable* latency functions, i. e., the latency of an edge depends only on the amount of flow on this edge. It is, however, reasonable to assume that the amount of flow on other edges influences the latency of every edge to a certain extent. *Non-separable* latency functions account for this dependency as they are functions of the entire vector of edge latencies. Dafermos and Nagurney [**?**] prove existence of equilibria for this kind of latency functions (see also [**?**, **?**]).

For results on the price of anarchy for non-separable latency function s ee [**?**, **?**, **?**].

A more accurate description of traffic flows can be obtained by introducing *edge capacities* [**?**, **?**, **?**, **?**]. In this model multiple equilibria are possible and the price of anarchy becomes unbounded even for linear latency functions. However, the best equilibrium is still as efficient as in absence of edge capacities [**?**].

1.6.5 Non-Increasing Latency Functions

Throughout this work, we assume the latency functions on the edges to be continuous and non-decreasing. The remark following Lemma 2 highlights that these assumptions are necessary (and in fact sufficient) for flows obeying the "First Principle of Wardrop" to be at Nash equilibrium. Further, these assumptions seem reasonable in real-world applications, because in congestion dependent networks the latency mostly represents delay. However, applications such as multi-cast routing with multiple duplication of flow motivate the analysis of selfish routing in the presence of strictly *non-increasing* latency functions [**?**]. As it turns out, this model exhibits rather demotivating characteristics. Equilibria are not unique, and an optimal flow is not approximable by selfish behavior even for linear latency functions in a small network with only six vertices.

1.6.6 Maximum Latency, Bottleneck and Elastic Demands

In the vast majority of the literature on the Wardrop model, the network performance is measured in total latency. As can be observed in Pigou's example in Figure 1.3(b), a flow minimizing total latency may be unfair from the agents' perspective [**?**]. In order to attain a system optimal routing, some agents may take costly detours that reduce the congestion encountered by the others. This unfairness makes such a solution unattractive for the affected agents. Arguably, the most intuitive way to establish a higher degree of fairness is to minimize the maximum latency incurred by a user. The price of anarchy for the maximum path latency as social cost has been considered by several researchers [**?**, **?**, **?**, **?**]. For single-commodity networks the price of anarchy is $n - 1$ [**?**], contrasting results for the total latency. For multi-commodity instances the situation is worse as even the removal of a single edge may decrease the maximum latency by a factor of $2^{\mathcal{O}(n)}$ [**?**].

An underlying assumption in Wardrop's traffic model is that the agents' performance is determined by the sum of edge latencies. However, there are many practical scenarios in which the agents follow *bottleneck* objectives [**?**], i. e., performance is determined by the worst component (highest edge la-

tency). Note that in Wardrop's setting the bottleneck latency of a path corresponds to the ∞-norm of the vector of edge latencies whereas the total latency equals the 1-norm. More generally, Cole *et al.* [**?**] focus on selfish routing networks under the p-norm for $1 < p \leq \infty$ and give several performance guarantees of equilibria. In particular, for single-commodity the price of anarchy under the p-norm for $1 < p < \infty$ is bounded by the price of anarchy with respect to the total latency (i. e., under the 1-norm), but for multi-commodity networks the price of anarchy under the p-norm for $1 < p \leq \infty$ can be arbitrarily larger.

In many scenarios, the demand is not fixed a priori but is dependent on the prevailing network congestion. Models allowing these so-called *elastic* demands have been extensively studied in the transportation science literature [**?**]. Recent work on elastic demands in Wardrop's model focuses on the efficiency of equilibria [**?**, **?**] and optimal taxes [**?**].

1.6.7 Non-Selfish Agents

Recent trends in the Internet like open source software development establish that selfishness may be not as rampant as we might expect. Instead, people voluntarily contribute to public goods projects without direct personal benefit. On the contrary, large uncoordinated systems often have to deal with spiteful adversaries who single-mindedly strive to degrade the network wide performance, Internet viruses being an infamous example. These examples exhibit cooperative behavior through the evolution of social norms or altruism and forms of spite as subjects aim to destruct systems. Thus, selfishness is not the only challenge to optimize network performance.

In the Wardrop model *altruistic* and *malicious* behavior has been modeled in several ways [**?**, **?**, **?**, **?**]. Babaioff *et al.* [**?**] introduce a model in which a certain fraction of agents act rationally and wish to minimize their individual latency. The remaining fraction of flow consists of malicious agents that wish to maximize the total latency of the rational agents. The authors study the existence of equilibria for these games and demonstrate a counterintuitive phenomenon which they coin "windfall of malice": malicious agents can improve the latency experienced by the selfish agents. Chen and Kempe [**?**] assume that agents trade off the benefit of themselves against the benefit of the others and prove that Wardrop equilibria are guaranteed to exist. They further show that the price of anarchy for parallel link networks is merely a constant in the presence of a non-negligible amount of altruists, thereby generalizing the Stackelberg routing result of Roughgarden [**?**].

The existence and computational complexity of equilibria in presence of altruistic or malicious agents has also been considered for discrete congestion games [**?**,**?**].

1.6.8 Alternative Solution Concepts

Wardrop equilibria are the most prevalent solution concept in non-atomic selfish routing. But yet, some scenarios may require more general solution concepts.

For instance, agents often face the problem of uncertain latency estimates. The uncertainty may be caused by random effects, such as accidents, weather, or varying traffic conditions in road traffic as well as noise or signal degradation in the context of telecommunication networks [**?**]. Motivated by this problem, Ordonez and Stier-Moses [**?**] introduced *robust Wardrop equilibria* that account for the agents' imperfect information. Robust Wardrop equilibria are appealing as they always exist and can be computed in polynomial time.

In a related approach, Fisk [**?**] generalizes Wardrop's traffic model in that he formalizes a network optimization problem whose solution is a *probabilistic equilibrium* that contains the original Wardrop equilibrium in a special case.

In congestion games several alternative solution concepts have been studied. Closely related to robust Wardrop equilibria, the concept of *Bayesian equilibria* has been applied to congestion games, in which agents possess only imperfect information about the game [**?**]. *Correlated equilibria* rely on a trusted authority telling the agents how to play to minimize their cost. Correlated equilibria can be computed efficiently [**?**] and exhibit a small price of anarchy [**?**]. *Strong equilibria* [**?**] are strategy profiles in which no *coalition* of agents may improve the latency of each of its members by deviating from the current strategies. Their existence and their efficiency have also been studied [**?**]. Finally, *sink equilibria* constitute an attractive solution concept, since they exist even in weighted congestion games. The price of sinking has been analyzed by Goemans *et al.* [**?**].

1.7 Outline

In this thesis we study a variety of algorithmic problems in Wardrop's model that revolve around the price of anarchy and Braess's paradox. In the first part we study a general taxing problem and propose a novel approach to reduce the influence of selfish routing. Secondly, we analyze the stability of Wardrop equilibria with respect to network parameter changes. Lastly, we provide a distributed approximation algorithm for Wardrop equilibria.

As a prerequisite for our results, we need to specify how to encode an instance $(\mathcal{G}, d)$. The network G can be represented using adjacency matrices or adjacency lists, and the demand vector d consisting of k rational entries can be encoded in a canonical way using binary representation. A natural representation of polynomial latency functions is the coefficient representation which lists the coefficients of all monomials. All our positive results that require the set of latency functions as input only hold for networks with linear or polynomial latency functions. The impossibility results hold even for linear latency functions. Hence, it is sufficient to have efficient encodings for these sets of latency functions.

Reducing the price of anarchy via taxes The most popular approach to reduce the inefficiency of Wardrop equilibria utilizes edge taxes. *Marginal cost taxes* are known to reduce the price of anarchy to 1 [**?**]. Since imposing taxes on *every* network edge may be impossible or costly, we consider the more general problem of minimizing the network wide performance by setting taxes for a given subset of edges only. While we prove that the problem is NP-complete in general networks, we provide a polynomial time algorithm solving this problem for single-commodity parallel link networks with linear latency functions.

The results are presented in Chapter 2. In preliminary form these results already appeared at the following conference:

- [**?**] Martin Hoefer, Lars Olbrich, and Alexander Skopalik. Taxing Subnetworks. In *Proc. of the 4th Workshop Workshop on Internet and Network Economics (WINE)*, pages 286-294, 2008.

Reducing the price of anarchy via auxiliary flow Taxing, Stackelberg routing, and network design are the most prominent means to reduce the inefficiency of selfish flow in scenarios without central control. Nevertheless, all of these approaches either require costly infrastructure or the possibility of manipulating the network structure or the agents. We propose a novel approach to reduce the price of anarchy that circumvents all of these problematic issues. We observe that routing an additional amount of flow, which we coin *auxiliary flow*, can actually improve the equilibrium situation for the selfish flow. We prove that the *optimal auxiliary flow* is NP-hard to approximate to less than a factor of $4/3$ and the *minimal amount of an optimal auxiliary flow* is NP-hard to approximate within any subexponential factor. These results are complemented by proving that the worst *adversarial flow*, i. e., flow that aims to maximize the total latency, is also NP-hard to compute. In fact, in all cases we obtain strong NP-hardness.

The results are presented in Chapter 3. In preliminary form these results already appeared at the following conference:

- [**?**] Martin Hoefer, Lars Olbrich, and Alexander Skopalik. Doing Good with Spam is Hard. In *Proc. of the 2nd Symposium on Algorithmic Game Theory (SAGT)*, pages 263-274, 2009.

Sensitivity of Wardrop Equilibria Braess's paradox displays intriguing aspects of selfish behavior. In fact, it triggered the stability and sensitivity analysis of Wardrop equilibria. While most existing literature concentrates on qualitative questions [**?**,**?**,**?**], we upper and lower bound the change of the main flow parameters at the induced equilibrium due to an ε-change. An *ε-change* is defined as a demand increase by a factor of $(1+\varepsilon)$ or the removal of an edge carrying only an ε-fraction of flow. For single-commodity networks, we show how an ε-change may force every agent to change its path in order to recover equilibrium. Our proof employs a family of networks generalizing Braess' original graph. On the other hand, an ε-change in the demand increases the path latency and the price of anarchy at most by a factor of $(1+\varepsilon)^p$ for polynomial latency functions of degree at most p with nonnegative coefficients. In contrast, the relative increase in the latency of an edge can be unbounded. For multi-commodity networks neither the change in edge flows nor the increase in the path latency can be bounded.

The results are presented in Chapter 4. In preliminary form these results already appeared at the following conference:

- [**?**] Matthias Englert, Thomas Franke, and Lars Olbrich. Sensitivity of Wardrop Equilibria. In *Proc. of the 1st Symposium on Algorithmic Game Theory (SAGT)*, pages 158–169, 2008.

They also appeared as invited contribution to a special issue of *Theory of Computing Systems* with selected papers from *SAGT 2008*:

- [**?**] Matthias Englert, Thomas Franke, and Lars Olbrich. Sensitivity of Wardrop Equilibria. In *Theory of Computing Systems*, pages 263-274, 2009.

Distributed Approximation of Wardrop Equilibria The notion of Wardrop equilibrium requires complete knowledge about the latency dependence of the edges as well as unbounded reasoning capabilities of the agents. Further, most research on selfish routing focuses on the agents' behavior at equilibrium and exclude the question how the set of agents may attain such a stable state. We study how approximate Wardrop equilibria can be computed efficiently under rather weak assumptions on the agents' information about the game. Previous work [**?**] shows that the set of agents can approach Wardrop equilibria quickly by following a simple round-based rerouting policy. Following the so-called *replication policy*, in each round every agent concurrently samples another agent uniformly at random. If the sampled agent's path latency is lower

than its current path latency, the agent switches to the other agent's path with a probability increasing with the offered improvement. The policy avoids the problem of oscillation due to its carefully chosen switching probability. A state, in which only a small fraction of the agents sustains latency significantly above average is reached in a number of rounds that mainly depends on the approximation parameters and the *elasticity* of the latency functions.

We consider a setting, in which the flow is controlled by a finite number of agents only, each of which is responsible for the entire flow of one commodity. Each agent has a set of admissible paths among which it may distribute its flow. Each agent aims to balance its own flow such that the jointly computed allocation will be at Wardrop equilibrium

Since the replication policy is designed for an infinite set of agents and potentially exponentially many paths it does not directly yield a feasible distributed algorithm. However, applying a randomized sampling technique we turn the replication policy into a distributed algorithm executable by finitely many agents. The distributed algorithm achieves essentially the same convergence rates as in the setting with an infinite number of agents. Thus, an approximate Wardrop equilibrium is reached in a number of rounds that is independent of the size and the topology of the network and can be computed in expected polynomial time.

The results are presented in Chapter 5. In preliminary form these results already appeared at the following conference:

- [**?**] Simon Fischer, Lars Olbrich, and Berthold Vöcking. Approximating Wardrop Equilibria with Finitely Many Agents In *Proc. of the 21st International Symposium on Distributed Computing (DISC)*, pages 238–252, 2007.

They also appeared as invited contribution to a special issue of *Distributed Computing* with selected papers from *DISC 2007*:

- [**?**] Simon Fischer, Lars Olbrich, and Berthold Vöcking. Approximating Wardrop Equilibria with Finitely Many Agents In *Distributed Computing*, 21(2) pages 129–139, 2008.

Chapter 2

Taxing Subnetworks

We have already seen in the introduction of this dissertation that the set of Wardrop equilibria embodies the set of stable states in Wardrop's game theoretic traffic model. Such equilibria are solutions of related convex programs and can thus be found in polynomial time. In general a Wardrop equilibrium is not socially optimal, i. e., it does not minimize the total latency. The inefficiency of selfish flows has been extensively studied in previous work [**?**, **?**, **?**, **?**, **?**, **?**]. In fact, Roughgarden and Tardos [**?**] proved that even in parallel link networks the price of anarchy may well be unbounded.

There are several approaches that have been proposed to address the inefficiency of equilibria, most notably via taxing network edges. Agents are assumed to minimize the sum of their latencies and taxes. A fundamental result is that using *marginal cost pricing* to set a tax on every edge results in equilibrium flows that are optimal with respect to total latency [**?**]. Therein, the tax an agent has to pay on an edge equals the additional delay its presence causes for other agents on this edge.

Marginal cost pricing is widely accepted as a benchmark solution. However, the necessary underlying assumptions for marginal cost pricing will most often be an obstacle in real world applications. In this regard, a serious drawback is that marginal cost pricing requires *every* edge of the network to be taxable. In many networks there might be technical or legal restrictions that prevent an operator from imposing a tax on all edges. Even assuming direct access to the edges, the monitoring costs for many edges and the process of collecting taxes may be considered prohibitive. On many edges it further may generate only negligible benefits to social welfare. Therefore, we consider the more realistic problem of computing taxes for a *subset of taxable edges* that minimize the total latency of the resulting equilibrium. Nevertheless, we do not impose further restrictions on the taxes itself besides non-negativity and computability. This problem is certainly relevant in the context of road traffic, as an increasing number of urban areas are installing electronic road charg-

ing [**?**]. For example the optimal taxes for a toll-ring around a city center can be the problem under consideration. This exemplifies that even if there are no imposed restrictions, a central authority may have a self interest in restricting the number or structure of taxed roads. In related work, Yang and Lam [**?**] give heuristics for the same problem in a queuing network, while Verhoef [**?**] presents analytically tractable solutions for small networks.

2.1 Our Results

Taxing subnetworks can be difficult and non-trivial. Consider the parallel link network of two links and linear latency functions shown in Figure 2.1(a). If one can tax only one edge, the total latency is generally not monotone in the imposed tax. Using this insight, we carefully construct networks with one taxable edge and several distinct optimal taxes. A combination of these networks establishes NP-hardness of the problem for two commodities and linear latency functions (Section 2.3). On the other hand, for parallel link networks with linear latency functions, we derive a precise structural analysis of optimally taxed equilibrium flows in Section 2.4. This allows to construct a polynomial time algorithm to find optimal taxes. The main ingredients are insights on the set of links carrying flow, dependencies between total latency and demand, and linearity of latency functions. Unlike a large part of related work, we do not need to resort on convex programming.

2.2 Preliminaries

We have already formally introduced Wardrop's model [**?**] in this thesis. Considering the problem of taxing a subset of network edges, we slightly extend the model and reformulate some classic results, which we will rely on throughout this chapter.

We are given a directed graph $G = (V, E)$ with vertex set V and edge set E. Considering only parallel edges, we speak of parallel link networks and denote the set of links by $[n] = \{1, \dots, n\}$. We allow a set of non-negative taxes $\tau = \{\tau_e\}_{e \in T}$ to be imposed on a subset of edges $T \subset E$. We call edges in T *taxable* and edges in $N = E \setminus T$ *non-taxable*. For simplicity, we set $\tau_e = 0$ for $e \in N$. The disutility of an agent choosing a path P is defined as latency plus tax, i. e., $\ell_P(f) + \sum_{e \in P} \tau_e$. Finally, we call the quadruple (V, T, N, d) an *instance*.

Bearing the definition of an agents' disutility in mind, at Wardrop equilibrium no fraction of the flow can improve its sustained latency plus taxes by moving unilaterally to another path.

Definition 5 (Wardrop equilibrium with taxes)**.** *A flow vector f is at* Wardrop equilibrium *if for every commodity $i \in [k]$ and paths $P_1, P_2 \in \mathcal{P}_i$ with $f_{P_1} > 0$ it holds that $\ell_{P_1}(f) + \sum_{e \in P_1} \tau_e \leq \ell_{P_2}(f) + \sum_{e \in P_2} \tau_e$.*

Remember that without taxes at Wardrop equilibrium all used paths in commodity i have equal latency $L_i(f)$ and the total latency can be expressed as $\sum_{i \in [k]} L_i(f) \cdot d_i$ ([**?**, **?**], 2). In the presence of taxes, however, all agents in commodity i experience a unique *disutility*. A classical result on taxing selfish flow, called *marginal cost pricing*, is that with taxes $\tau_e = x_e \cdot \ell_e'(x_e)$ for all $e \in E$ the resulting equilibrium flow minimizes the total latency. With $\ell_e^*(x) = (x \cdot \ell_e(x))' = \ell_e(x) + x \cdot \ell_e'(x)$ denoting the marginal cost of increasing flow on edge e, Theorem 3 yields the following lemma.

Lemma 5. *For an instance $(V, E, \emptyset, d)$ a flow f minimizes the total latency with respect to $(\ell_e)_{e \in E}$ if and only if it is at Wardrop equilibrium with respect to $(\ell_e^*)_{e \in E}$.*

In the restricted case with only a subset of edges being taxable such a result is obviously out of reach as Pigou's example (cf. Figure 2.1(a)) exemplifies. If only the constant latency edge is taxable, no improvement is possible. This directly leads us to the following definition.

Definition 6 (Optimal taxes)**.** *Given an instance (V, T, N, d), a set of taxes $\{\tau_e\}_{e \in T}$ is called* optimal *if there is an equilibrium flow f_τ with respect to $\ell + \tau$ with*

$$C(f_\tau) \leq C(f_{\tau'})$$

for all equilibrium flows $f_{\tau'}$ with respect to $\ell + \tau'$ for any $\{\tau_e'\}_{e \in T}$.

2.3 NP-Hardness for Multi-Commodity Networks

In this section we study the computational complexity of Subnetwork-Tax. In the decision problem Subnetwork-Tax we are given a multi-commodity selfish routing instance (V, T, N, d) and a threshold value C. The problem is to decide if there are taxes for the edges $e \in T$ such that the induced equilibrium incurs total latency of at most C. Subnetwork-Tax turns out to be NP-hard even for the two-commodity case with linear latency functions. We start with an observation that will allow us to discretize the problem.

Lemma 6. *There is a family of instances $(V, T, N_A, d_A)_{A \in \mathbb{N}}$ with parallel link networks allowing for two distinct optimal tax values.*

Proof. Consider the network shown in Figure 2.1(b). Two nodes s and t are connected via three links, with latency functions $\ell_1(x) = x + A$ and $\ell_2(x) = \ell_3(x) = x$. Suppose we can only tax the third link.

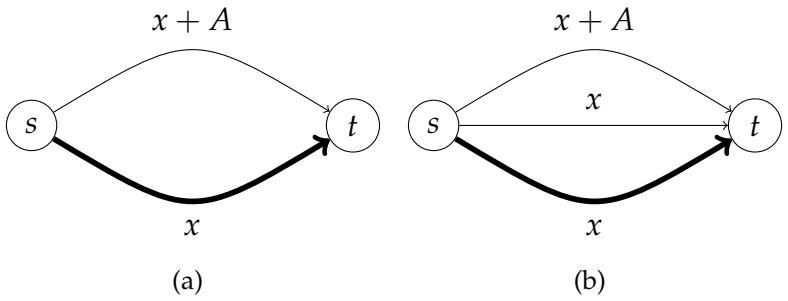


Figure 2.1: (a) A unit demand needs to be routed from s to t. Let $A > 0$. We can tax the bold edge only. The total latency at equilibrium is minimized for $\tau = A/2$. In (b) routing a demand of $A \cdot (1 + \frac{\sqrt{3}}{2})$, the optimal tax for the third link is $\tau = 0$ or $\tau = A/4$.

In the following we study the latency dependence on the imposed tax and denote the total latency as $C(\tau)$.
Routing a demand of $d_A = A \cdot (1 + \frac{\sqrt{3}}{2})$ taxing has the following effect. For tax $0 \leq \tau \leq A \cdot (1 - \frac{\sqrt{3}}{2})$, at equilibrium the total demand is split among links two and three. Since both links have no offset the total latency increases with increasing tax.

For every tax τ the flows on the bottom edges fulfill the equations

$$\begin{aligned} f_2 &= f_3 + \tau \text{ and} \\ f_2 + f_3 &= d_A. \end{aligned}$$

Thus,

$$\begin{aligned} f_2 &= \left(\frac{1}{2} + \frac{\sqrt{3}}{4}\right) \cdot A + \frac{1}{2} \cdot \tau \text{ and} \\ f_3 &= \left(\frac{1}{2} + \frac{\sqrt{3}}{4}\right) \cdot A - \frac{1}{2} \cdot \tau. \end{aligned}$$

The total latency of this equilibrium flow is

$$\begin{aligned} C_1(\tau) &= \left(\left(\frac{1}{2} + \frac{\sqrt{3}}{4}\right) \cdot A + \frac{1}{2} \cdot \tau\right)^2 + \left(\left(\frac{1}{2} + \frac{\sqrt{3}}{4}\right) \cdot A - \frac{1}{2} \cdot \tau\right)^2 \\ &= \frac{1}{2} \cdot \tau^2 + \left(\frac{7}{8} + \frac{\sqrt{3}}{2}\right) \cdot A^2 \ . \end{aligned}$$

$C_1(\tau)$ is minimized for $\tau = 0$.
For $A \cdot (1 - \frac{\sqrt{3}}{2}) < \tau < A \cdot (1 + \frac{\sqrt{3}}{4})$ all links are used and the total latency is not monotone as a function of the imposed tax. The corresponding flows and the incurred latency can be calculated is a similar fashion as above. The link flows satisfy

$$\begin{aligned} f_1 + A &= f_2, \\ f_2 &= f_3 + \tau \text{ and} \\ f_1 + f_2 + f_3 &= d_A. \end{aligned}$$

Thus,

$$\begin{aligned} f_1 &= \frac{1}{3}\left(A \cdot \left(\frac{\sqrt{3}}{2} - 1\right) + \tau\right), \\ f_2 &= \frac{1}{3}\left(A \cdot \left(\frac{\sqrt{3}}{2} + 2\right) + \tau\right) \text{ and} \\ f_3 &= \frac{1}{3}\left(A \cdot \left(\frac{\sqrt{3}}{2} + 2\right) - 2 \cdot \tau\right). \end{aligned}$$

induce a total latency of

$$C_2(\tau) = \frac{2}{3} \cdot \tau^2 - \frac{1}{3} \cdot A \cdot \tau + \left(\frac{11}{12} + \frac{\sqrt{3}}{2}\right) \cdot A^2$$

that is minimized for a tax of $A/4$.
For $\tau \geq A \cdot (1 + \frac{\sqrt{3}}{4})$ the total latency at equilibrium is $C_2(A \cdot (1 + \frac{\sqrt{3}}{4}))$.
Hence, the total latency C is

$$C(\tau) = \begin{cases} C_1(\tau) & \text{for } 0 \leq \tau \leq A \cdot (1 - \frac{\sqrt{3}}{2}) \\ C_2(\tau) & \text{for } A \cdot (1 - \frac{\sqrt{3}}{2}) < \tau \leq A \cdot (1 + \frac{\sqrt{3}}{4}) \\ C_2(A \cdot (1 + \frac{\sqrt{3}}{4})) & \text{for } \tau > A \cdot (1 + \frac{\sqrt{3}}{4}) \ . \end{cases}$$

Since $C(0) = C(A/4)$ the instance admits two optimal taxes 0 and $A/4$. □

Having discretized the problem we are now able to prove the main result of this section.

Theorem 7. SUBNETWORK-TAX *is* NP*-hard, even for instances with only two commodities and linear latency functions.*

Proof. We reduce from the PARTITION problem: given k positive integers a_i, is there a subset $S \subseteq [k]$ satisfying $\sum_{i \in S} a_i = \frac{1}{2} \sum_{i=1}^{k} a_i$? We will show that deciding the PARTITION problem reduces to deciding if a given 2-commodity

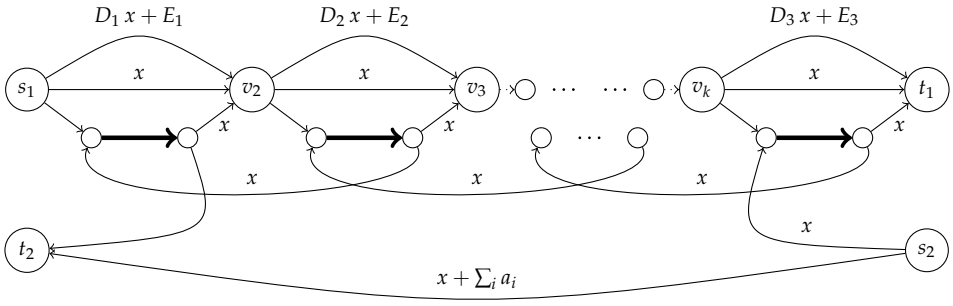


Figure 2.2: The network of an instance $(V_{\{a_i\}}, T_{\{a_i\}}, N_{\{a_i\}}, (d_{\{a_i\}}))$. The edges are labeled with the latency functions. Unlabeled edges have latency 0. Taxes can be imposed on the set of bold edges only.

instance $(V, T, N, (d_i))$ with latency functions admits taxes inducing a Wardrop equilibrium with a given total latency.

Given an instance of Partition specified by non-negative integers $a_1, a_2, \ldots, a_k$ we define an instance $(V_{\{a_i\}}, T_{\{a_i\}}, N_{\{a_i\}}, (d_{\{a_i\}}))$ as depicted in Figure 2.2. Let the set of taxable edges T consist of the bold edges.

Commodity one has a demand of $A = \prod_{i=1}^{k} a_i$ to route between $s_1 = v_1$ and $t_1 = v_{k+1}$, the second commodity has to route a demand of $\sum_i a_i$ between s_2 and t_2. For $i \in [k]$ define the following constants:

$$
\begin{aligned}
A_{-i} &= \prod_{j \neq i}^{k} a_j \ , \\
D_i &= \frac{2 - 4A_{-i} + A_{-i}^2}{4A_{-i} - 2} \quad \text{and} \\
E_i &= 2a_i(D_i + 1) = \frac{AA_{-i}}{2A_{-i} - 1} \ .
\end{aligned}
$$

Note that all values can be encoded by a number of bits that is polynomial in the size of the instance of Partition.

We show that $\{a_1, \ldots, a_k\}$ is a Yes instance if and only if there are taxes for the instance $(V_{\{a_i\}}, T_{\{a_i\}}, N_{\{a_i\}}, (d_{\{a_i\}}))$ inducing a Wardrop equilibrium with total latency of at most $C = \frac{k}{2}A^2 + \frac{7}{8}(\sum_i a_i)^2$. Since both commodities do not share any latency incurring edge, the constructed network allows for a separated consideration of both commodities. The idea is that the minimal latency is reached if and only if the tax between v_i and v_{i+1} is 0 or a_i and the sum of all taxes is exactly $\sum_i a_i/2$.

First, consider the set of vertex disjoint paths between v_i and v_{i+1} for some $i \in [k]$. A demand of A needs to be routed between those two nodes. The

situation resembles the situation described in the proof of Lemma 6. The total latency is

$$C_i(\tau) = \begin{cases} \frac{A^2}{2} + \frac{\tau^2}{2} & \text{for } 0 \leq \tau \leq 2E_i - A \\ \frac{(D_i+1)\tau^2 - E_i\tau + A(D_iA+E_i)}{2D_i+1} & \text{for } 2E_i - A < \tau < \frac{D_iA+E_i}{D_i+1} \\ C_i(\frac{D_iA+E_i}{D_i+1}) & \text{for } \tau \geq \frac{D_iA+E_i}{D_i+1} \quad . \end{cases}$$

One can easily check that for $A_{-i} > 1$ there are two non-empty intervals for tax τ, in which the total latency as a function of the imposed tax τ is quadratic. The constants D_i and E_i are chosen in such a way that the total latency is minimized for 0 and $\frac{E_i}{2 \cdot D_i + 2} = a_i$. Thus, both taxes 0 and a_i are optimal for the set of parallel paths connecting v_i with v_{i+1}. The incurred total latency for the optimal taxes is $A^2/2$.

For the second commodity consider the sum $g = \sum_i \tau_i$ of all taxes in the network. At equilibrium a flow volume of $\sum_i a_i - \frac{g}{2}$ is routed via the path including the taxable edges, and a flow of volume $\frac{g}{2}$ is routed via the lower edge. The total latency of $\bar{C}(g) = (\sum_i a_i - \frac{g}{2})^2 + \frac{g}{2}(\frac{g}{2} + \sum_i a_i)$ is then minimized for $g = \sum_i a_i/2$.

First, assume $\{a_1, \ldots, a_k\}$ is a YES instance. We reach optimality for both commodities by choosing

$$\tau_i = \begin{cases} a_i & \text{for } i \in S \\ 0 & \text{for } i \notin S \quad . \end{cases}$$

The total latency sums up to

$$\begin{aligned} C &= \sum_{i \in S} C_i(a_i) + \sum_{i \in [k] \setminus S} C_i(0) + \bar{C}\left(\sum_i a_i/2\right) \\ &= \frac{k}{2}A^2 + \frac{7}{8}\left(\sum_i a_i\right)^2 \quad . \end{aligned}$$

Now suppose $\{a_1, \ldots, a_k\}$ is a NO instance. To obtain a total latency of $\frac{k}{2}A^2$ for commodity one, the tax on the taxable edge between v_i and v_{i+1} needs to be 0 or a_i for every i. For the second commodity a total latency of $\frac{7}{8}(\sum_i a_i)^2$ can be obtained only if the sum of the taxes adds up to $\sum_i a_i/2$. Obviously, in a NO instance both conditions can not be achieved at the same time. Thus, the total latency is above the threshold composed of the two optima and the reduction is complete. □

2.4 Parallel Links with Linear Latency Functions

The parallel link instances in the proof of Lemma 6 show that the total latency is generally not monotone as a function of the imposed taxes. That holds even

in the case of linear latency functions and one taxable link. Further, these examples show that such instances do not necessarily admit a unique optimal tax. These observations indicate that studying optimal taxes in parallel link networks might be intriguing.

Our main goal in this section is to provide an algorithm for finding optimal taxes in single-commodity parallel link networks $(V, T, N, 1)$ in which every link $i \in [n]$ is equipped with a latency function $\ell_i(x) = a_i x + b_i$. This setting has been of special interest in the related problem of computing a Stackelberg leader strategy **[?]** described in the introduction. While this problem is already NP-hard in this setting, it may be surprising that we will be able to formulate a polynomial time algorithm for computing optimal taxes.

Suppose the links are numbered by $N = \{1, \ldots, k\}$ and $T = \{k+1, \ldots, n\}$, such that $b_1 \leq \ldots \leq b_k$ and $b_{k+1} \leq \ldots \leq b_n$. We use this labelling for convenience, but note that the ordering conditions apply only *within* N and T. In particular, we do not require $b_i \leq b_j$ for any $i \in N$ and $j \in T$ or any other particular restriction or relation between the links of N and T.Without loss of generality we assume at most one constant latency link in $N \cup T$. Thus, equilibrium flow and optimal flow are unique.

2.4.1 Candidate Supports Sets

Recall that a flow f is at Wardrop equilibrium if and only if there is a constant $L > 0$, such that all used links $i \in [n]$ have the same latency $L = \ell_i(f_i)$, whereas $L \leq \ell_{i'}(0) = b_{i'}$ for unused links $i' \in [n]$. Lemma 5 shows that a flow f is socially optimal if and only if there is a constant $C > 0$ such that $C = \ell_j^*(f_j) = 2a_j f_j + b_j$ for all used links $j \in [n]$, whereas $C \leq \ell_{j'}^*(0) = b_{j'}$ for unused links $j' \in [n]$.

Observation 8. *Consider a routing instance with $d = 0$. Both equilibrium and optimum satisfy the following condition: when increasing the demand from 0, the links will be filled with flow in order of their offset b .*

We will use this property for the problem of finding optimal taxes $(\tau_j)_{j \in T}$. Regarding the agents disutility (latency plus tax) the set of taxes will induce an equilibrium assigning flow to some link set $S \subset N \cup T$. All used non-taxable links have the same latency L. Since we allow for non-negative taxes only, all used taxable links will not have higher latency. This property allows us to parametrize the problem by the set of taxable and non-taxable links filled with flow. These sets turn out to be *candidate support sets* defined as follows.

Definition 7 (Candidate support set)**.** *Every set of the form $S = \{1, \ldots, l_1\} \cup \{k+1, \ldots, l_2\}$ with $1 \leq l_1 \leq k$ and $k+1 \leq l_2 \leq n$ is called a* candidate support set.

Note that there are at most $n^2/4$ candidate support sets for any instance.

Lemma 9. *Let f denote a socially optimal flow for a parallel link network where every edge is taxable. Then*

$$\ell_1(f_1) \leq \ell_2(f_2) \leq \ldots \leq \ell_n(f_n) \ .$$

Proof. The set of used links is of the form $\{1, \ldots, l\}$ for some $l \leq n$. Since f is a minimal latency flow, all links $j \in \{1, \ldots, l\}$ have equal marginal cost, and there is a constant $C > 0$ with $2a_j f_j + b_j = C$. Thus, $\ell_j(f_j) = a_j f_j + b_j = C/2 + b_j/2$. □

Let us first argue that the consideration of candidate support sets is indeed sufficient to find optimal taxes. Imagine two separate commodities, routing fixed demands d_N and $1 - d_N$ exclusively over N and T, resp. In such a scenario, it would be optimal to set marginal cost taxes on T. According to Observation 8 the set of used links form a candidate support set.

The difference to our setting is that demand can change between N and T, and thus we also need to ensure that latency and taxes create an equilibrium flow. In particular, latency plus tax of any used link in T must be equal to the common latency L of all used links in N. Furthermore, the offset plus tax of any unused link in T must be higher than L.

If the optimal flow of all links in T yields latencies only smaller than L, then we can satisfy the latency constraint by setting appropriate non-negative taxes. Otherwise, the latency restriction reduces the flow on some used links. Naturally, this can happen to all links $j \in T$ with $b_j \leq L$. However, if the flow on a link is smaller than in the optimum due to the latency constraint, the marginal cost on this link is also smaller. Therefore, it is still optimal to fill the link with flow to the maximal possible extent, which we will prove in Lemma 10. For all links not affected by the latency restriction, however, it is optimal to equalize the marginal costs, and the allocation of flow follows the ordering of offsets. In conclusion, it can be observed that the set of links allocated with flow remains a candidate support set. Thus, it is sufficient to restrict our attention to these sets.

2.4.2 Problem Parametrization

Fixing numbers n_S and t_S yields a candidate support set $S = N_S \cup T_S$ with $N_S = \{1, \ldots, n_S\}$ and $T_S = \{k+1, \ldots, t_S\}$. For S denote by d_{N_S} and $1 - d_{N_S}$ the demand routed over N_S and T_S, respectively. $C_{N_S}(d_{N_S})$ is the total latency for an equilibrium flow $(f_i)_{i \in N_S}$ of demand d_{N_S}. Denote by $C_{T_S}(1 - d_{N_S})$ the total latency for an optimal flow $(f_j)_{j \in T_S}$ of demand $1 - d_{N_S}$ additionally fulfilling

the latency constraint $\ell_j(f_j) \leq L(d_{N_S})$ for all links $j \in T_S$, where $L(d_{N_S})$ denotes the unique latency of all used links in N_S for a demand of d_{N_S}. Finally, let

$$C(d_{N_S}) = C_{N_S}(d_{N_S}) + C_{T_S}(1 - d_{N_S})$$

denote the total latency of the entire flow.

We can further parametrize the problem of finding a set of optimal taxes for a fixed set S by the demands routed over N_S and T_S. We formulate it in a compact way:

$$\begin{aligned}
&\text{minimize } C(d_{N_S}) \\
&\qquad s.t.\ (f_i)_{i \in N_S} \text{ equilibrium for demand } d_{N_S} \\
&\qquad\quad (f_j)_{j \in T_S} \text{ optimal for demand } (1 - d_{N_S}) \\
&\qquad\qquad s.t.\ \ell_j(f_j) \leq L(d_{N_S}) \quad \forall j \in T_S \\
&\qquad\quad 0 \leq d_{N_S} \leq 1 \ .
\end{aligned}$$

Note that we require the equilibrium and the optimum to hold for N and T (and not only for N_S and T_S). We will show that if this minimization problem has a solution, the total latency $C(d_{N_S})$ is piecewise quadratic with at most n breakpoints and the optimal demand distribution $(d^*_{N_S}, 1 - d^*_{N_S})$ for N_S and T_S for a candidate support set S is efficiently computable. Iterating this for all possible sets S enables us to find optimal taxes.

Definition 8 (Full and relaxed links)**.** *We call a link $j \in T$* full *with respect to some $L > 0$ if $f_j > 0$ and its latency equals the constraint value, i.e., if $\ell_j(f_j) = L$ or if $f_j = 0$ and $\ell_j(0) = b_j \geq L$. We call a link* relaxed *if $f_j > 0$ and $\ell_j(f_j) < L$.*

When shifting demand from N to T, the common latency L of used links in N decreases, while the demand on T increases. In the corresponding optimal flow on T respecting the constraint value, however, a full link never becomes relaxed.

More formally, consider an instance $(V, T, \emptyset, d)$ and let f denote the optimal flow respecting $\ell_i(f_i) \leq L$ for all i. With Lemma 9 we can assume the full links to form a set $\{p, \dots, n\}$ for some $p \geq 1$. Furthermore, assume there are $L' \leq L$ and $d' \geq d$ such that there is a flow of demand d' to T such that all used links have latency at most L'. For all non-constant links, we define $\ell_i^{-1}(L)$ to be the flow f_i such that $a_i f_i + b_i = L$ if $b_i \leq L$, and 0 otherwise.

Lemma 10. *The optimal flow f' respecting $\ell_i(f'_i) \leq L'$ for all i assigns $\ell_i^{-1}(L')$ flow to all non-constant links $i \in \{p_1, \dots, n\}$ for some uniquely defined $p_1 \leq p$.*

Proof. Restricting the latencies to at most L' removes flow from links $p, \dots, n$ and distributes it among the remaining links. Therefore, more links might become full with respect to L'. Let the newly affected links be $p_1, \dots, p-1$.

Let f' denote the flow of volume d' that assigns $\ell_j^{-1}(L')$ of flow to every non-constant link $j \in \{p_1, \ldots, n\}$ and that is optimal on links $1, \ldots, p_1 - 1$. Thus, for all $i \in \{1, \ldots, p_1 - 1\}$ and for all $j \in \{p_1, \ldots, n\}$

$$\ell_j^*(f_j') \leq \ell_i^*(f_i') \ . \tag{2.1}$$

For contradiction, assume that respecting L' there is an optimal flow $\bar{f}$ with

$$\bar{f}_{j_0} < f'_{j_0} \text{ for some } j_0 \in \{p_1, \ldots, n\} \ . \tag{2.2}$$

The latency ℓ_{j_0} is not constant since either j_0 is full and, thus, $f'_{j_0} = 0$ or j_0 is relaxed and therefore it is not optimal to reduce the flow on j_0. Since $\bar{f}$ is assumed to be optimal, for all links $i \in \{1, \ldots, p_1 - 1\}$

$$\ell_{j_0}^*(\bar{f}_{j_0}) \geq \ell_i^*(\bar{f}_i) \ . \tag{2.3}$$

Further, since $\{p_1, \ldots, n\}$ loses flow there is a link $i_0 \in \{1, \ldots, p_1 - 1\}$ that gains some flow, i. e.,

$$\bar{f}_{i_0} > f'_{i_0}$$

and therefore (even if ℓ_{i_0} is constant)

$$\ell_{i_0}^*(\bar{f}_{i_0}) \geq \ell_{i_0}^*(f'_{i_0}) \ . \tag{2.4}$$

Altogether,

$$\ell_{j_0}^*(f'_{j_0}) >^{(2.2)} \ell_{j_0}^*(\bar{f}_{j_0}) \geq^{(2.3)} \ell_{i_0}^*(\bar{f}_{i_0}) \geq^{(2.4)} \ell_{i_0}^*(f'_{i_0}) \geq^{(2.1)} \ell_{j_0}^*(f'_{j_0}) \ ,$$

which yields a contradiction. □

2.4.3 A Polynomial-Time Algorithm for Computing Optimal Taxes

Considering an optimal flow for an increasing demand, the links become used in order of their offsets. Note that Lemma 9 and Lemma 10 show that the links become full with respect to some bound in reverse order. The fact that we know the order in which the links become both used and full and the linearity of the latency functions enable us to determine the lower and the upper bound $d_{N_S}^{\min}$ and $d_{N_S}^{\max}$ for d_{N_S} such that the following holds. The equilibrium flow of demand d_{N_S} on N exactly uses the set of links N_S and there is a flow of demand $1 - d_{N_S}$ on T respecting the bound $L(d_{N_S})$ exactly using the set of links T_S, whose total latency can not be improved by using links in $T \setminus T_S$.
Given a candidate support set S we need to compute the optimal demand distribution $(d_{N_S}, 1 - d_{N_S})$. If a distribution exists that fulfills the above requirements we call S *feasible*. We call the corresponding demand intervals $[d_{N_S}^{\min}, d_{N_S}^{\max}]$ *feasible demand intervals*.

Lemma 11. *The feasible demand intervals can be computed in polynomial time.*

Proof. We will compute the feasible demand intervals by solving systems of linear equations. As mentioned above, certain conditions on the flows on both N and T must be met. Without loss of generality assume no constant latency link.

Let us first consider N. We must ensure that at equilibrium there is some flow on all links of N_S and no flow on $N \setminus N_S$. Thus, for all $i \in N_S$

$$\ell_i(f_i) = a_i f_i + b_i = L(d_{N_S}) \leq b_{n_S+1}$$

must hold. Thus, it is not possible to obtain an equilibrium flow for N using exactly the set N_S for a demand exceeding

$$d^+_{N_S} = \sum_{i \in N_S} f_i = \sum_{i \in N_S} \ell_i^{-1}(b_{n_S+1}) \ .$$

Similarly we get a lower bound of

$$d^-_{N_S} = \sum_{i \in N_S \setminus \{n_S\}} f_i = \sum_{i \in N_S \setminus \{n_S\}} \ell_i^{-1}(b_{n_S}) \ .$$

For $N_S = N$, we set $d^+_{N_S} = \infty$. For $N_S = \emptyset$, we set $d^-_{N_S} = d^+_{N_S} = 0$.

Considering the set T, we must ensure that exactly the set $T_S = \{k+1, \ldots, t_S\}$ is filled with flow, such that the latency of no link exceeds the bound L and that shifting flow to links in $T \setminus T_S$ is not socially rewarding. Consider a flow f of demand $1 - d'_{N_S}$ such that

$$\ell_j(f_j) = L(d'_{N_S}) \tag{2.5}$$

for all $j \in T_S$. Note that both f and d'_{N_S} are unique.
If for the marginal costs

$$\ell^*_j(f_j) \leq b_{t_S+1}$$

holds for all $j \in T_S$, the upper bound for the demand routed over T_S is $d^+_{T_S} = 1 - d'_{N_S}$. Here let $b_{t_S+1} = \infty$ for $T_S = T$.
Otherwise set for all j with $\ell^*_j(f_j) > b_{t_S+1}$

$$f'_j = (\ell^*_j)^{-1}(b_{t_S+1})$$

such that these links exhibit the same marginal cost values. Now, subtract $\sum_{\ell^*_j(f_j) > b_{t_S+1}} (f_j - f'_j)$ from the current demand $1 - d'_{N_S}$ and iterate the procedure beginning at Equation 2.5. The desired upper bound is

$$d^+_{T_S} = \sum_{\ell^*_j(f_j) \leq b_{t_S+1}} f_j + \sum_{\ell^*_j(f_j) > b_{t_S+1}} \tilde{f}_j \ ,$$

where $\tilde{f}$ denotes the flow that has been obtained in the last iteration.
The lower bound for $1 - d_{N_S}$, denoted by $d^-_{T_S}$, can be computed in an analogous fashion.
Since we need to meet the conditions for the flows on both N and T at the same time, we get demand bounds

$$d^{\min}_{N_S} = \max\{1 - d^+_{T_S}, d^-_{N_S}\} \text{ and}$$
$$d^{\max}_{N_S} = \min\{1 - d^-_{T_S}, d^+_{N_S}\} \ .$$

Finally, the candidate support set S is feasible if and only if $[d^{\min}_{N_S}, d^{\max}_{N_S}]$ is non-empty. □

Lemma 12. *The common latency L of all used non-taxable links at equilibrium is linear as a function of d_{N_S} for $d_{N_S} \in [d^{\min}_{N_S}, d^{\max}_{N_S}]$.*

Proof. Suppose f is at equilibrium for demand d_{N_S}. Then $L(d_{N_S})$ fulfills

$$L(d_{N_S}) = a_i f_i + b_i \text{ for every } i \text{ and}$$
$$\sum_{N_S} f_i = d_{N_S} \ ,$$

which proves the claim. □

Corollary 13. *The total latency $C_{N_S}(d_{N_S})$ is quadratic for every feasible candidate support set S and $d_{N_S} \in [d^{\min}_{N_S}, d^{\max}_{N_S}]$.*

Proof. Suppose f is at equilibrium for demand d_{N_S}. Since $L(d_{N_S})$ is linear, the total latency $C_{N_S}(d_{N_S}) = L(d_{N_S}) \cdot d_{N_S}$ is quadratic for $d_{N_S} \in [d^{\min}_{N_S}, d^{\max}_{N_S}]$. □

Neglecting the constraints $\ell_j(f_j) \leq L(d_{N_S})$, the total latency C_{T_S} of an optimal flow on T_S of demand $1 - d_{N_S}$ is a quadratic function for similar reasons. Respecting the constraints for increasing $1 - d_{N_S}$, we need to handle the full links of decreasing latency. Due to the linearity of L, C_{T_S} turns out to be quadratic with at most n breakpoints.

Lemma 14. *The breakpoints, i. e., the demand values for which the number of full links increases, can be computed in polynomial time.*

Proof. As in the proof of Lemma 11 solving systems of linear equations is the key. Respecting the constraint for increasing $1 - d_{N_S}$, an increasing number of links in T_S becomes affected by the bound $L(d_{N_S})$. This effect turns out to yield a piecewise quadratic total latency with at most n breakpoints. For readability let $T_S = \{1, \ldots, t_S\}$ for the remainder of the proof.

Computing an optimal flow respecting the bound L
We first efficiently compute an optimal flow x on T_S of minimal demand, i. e., of demand $1 - d^{\max}_{N_S}$, respecting $\ell_j(x_j) \leq L(d^{\max}_{N_S})$. We start with a socially

optimal flow $(f_j)_{j\in T_S}$, which can be computed in polynomial time. If $\ell_j(f_j) \leq L(d_{N_S}^{\max})$, we are done and $x = f$. Otherwise, due to Lemma 9 there is a $j_0 \in T_S$ such that

$$\ell_j(f_j) \leq L(d_{N_S}^{\max}) < \ell_{j'}(f_{j'})$$

for $1 \leq j < j_0 \leq j' \leq t_S$ and $\sum_{T_S} f_j = 1 - d_{N_S}^{\max}$. Lemma 10 shows that it is optimal to set $x_j = \ell_j^{-1}(L(d_{N_S}^{\max}))$ for the affected links $j_1, \ldots, t_S$ for some $j_1 \leq j_0$. In order to compute x, we first determine j_1. We set $f_{j'} = \ell_{j'}^{-1}(L(d_{N_S}^{\max}))$ for $j' = j_0, \ldots, t_S$. Computing the optimal flow for the links $1, \ldots, j_0 - 1$ for a demand of $(1 - d_{N_S}^{\max} - \sum_{j_0 \leq j' \leq t_S} \ell_{j'}^{-1}(L(d_{N_S}^{\max}))$ and proceeding as described above until $\ell_j(f_j) \leq L(d_{N_S}^{\max})$ for all links completes the computation of j_1. Thus we can compute the desired optimal flow x, in which the links $j_1, \ldots, t_S$ are full. If $j_1 = 1$, we are done. Otherwise, the imposed latency bound $L(d_{N_S})$ allows for a higher demand for the set T_S. Nevertheless, the total latency becomes non-differentiable and we need to compute the corresponding breakpoints.

Determine the breakpoints

Now increasing the input demand $1 - d_{N_S}$, the latency bound of $L(d_{N_S})$ becomes more restrictive. Determining the first breakpoint, i. e., the demand value d_{N_S} for which link $j_1 - 1$ becomes full with respect to $L(d_{N_S})$, amounts to solving

$$\ell_{j_1-1}(f_{j_1-1}) = L(d_{N_S}) \ , \tag{2.6}$$

where (f_j) denotes an optimal flow with respect to the additional constraint $f_j \leq L(d_{N_S})$ for all j. The equations

$$\begin{aligned} 2a_1 f_1 + b_1 = 2a_j f_j + b_j \qquad & \text{for } j = 2, \ldots, j_1 - 1 \quad \text{and} \\ a_j f_j + b_j = L(d_{N_S}) \qquad & \text{for } j = j_1 - 1, \ldots, t_S \end{aligned}$$

uniquely define the link flows on T_S. They can be written as

$$f_j = \alpha_j(1 - d_{N_S}) + \beta_j$$

with α_j and β_j being rational functions in the coefficients of the latency functions. Solving

$$\sum_{j\in T_S} f_j = \sum_{j\in T_S} (\alpha_j(1 - d_{N_S}) + \beta_j) = 1 - d_{N_S}$$

for d_{N_S} yields the first breakpoint.

Lemma 15. *The total latency functions $C_{T_S}(1 - d_{N_S})$ and $C(d_{N_S})$ are piecewise quadratic for $d_{N_S} \in [d_{N_S}^{\min}, d_{N_S}^{\max}]$ with at most n breakpoints for every feasible candidate support set S. The breakpoints can be computed in polynomial time.*

Algorithm 1 OPTTAX $(V, T, N, 1)$

for every candidate support set S **do**
  if S feasible **then**
    compute the breakpoints $d_{N_S}^{\min} = d_{N_{S_{k+1}}}, \ldots, d_{N_{S1}}, d_{N_{S0}} = d_{N_S}^{\max}$
    $d_{N_S}^* \leftarrow \operatorname{argmin}_{0 \le j \le k} \min_{d_{N_S} \in [d_{N_{S_j}}, d_{N_{S_{j+1}}}]} C(d_{N_S})$
  end if
end for
$\gamma(S) := C(d_{N_S}^*)$
$S^* \leftarrow \operatorname{argmin}_S \gamma(S)$
compute optimal flow on T_{S^*} respecting $L(d_{N_{S^*}}^*)$ with $\sum_{T_{S^*}} f_j^* = 1 - d_{N_{S^*}}^*$ and set $f_j^* := 0$ for $j \in T \setminus T_{S^*}$.
set taxes $\tau_j \leftarrow L(d_{N_{S^*}}^*) - \ell_j(f_j^*)$ for $j \in T$

For $d_{N_S} \in [d_{N_{S1}}, d_{N_S}^{\max}]$ the bound $L(d_{N_S})$ restricts exactly links $j_1, \ldots, t_S$ and the total latency

$$C_{T_S}(1 - d_{N_S}) = \sum_{j \in T_S} \ell_j(f_j) f_j = \sum_{j \in T_S} \ell_j(\alpha_j(1 - d_{N_S}) + \beta_j)(\alpha_j(1 - d_{N_S}) + \beta_j)$$

is quadratic. Further increasing $1 - d_{N_S}$, we get a piecewise quadratic total latency function C_{T_S} in $[d_{N_S}^{\min}, d_{N_S}^{\max}]$ with at most n breakpoints. □

Theorem 16. *Given an instance $(V, T, N, 1)$ with parallel links and linear latency functions Algorithm* OptTax(V,T,N,1) *computes a set of optimal taxes $(\tau_j)_{j \in T}$ in polynomial time.*

Proof. Correctness We have already argued that restricting to candidate support sets is sufficient for finding optimal taxes.

Runtime For each of at most $n^2/4$ candidate support sets the total latency and all breakpoints can be computed in polynomial time. Obviously, the minimization steps can be carried out in polynomial time as well. □

Chapter 3

Improving Equilibria with Auxiliary Flow

Marginal cost pricing assumes a central authority that has direct access to every network edge and that agrees to build and maintain the possibly very costly infrastructure necessary to collect taxes. We have seen in the previous chapter that the problem of computing optimal taxes for the more restricted case where only a given subset of edges is taxable becomes intractable for two-commodity networks. As another major drawback, marginal cost pricing charges the agents higher taxes than necessary [**?**, **?**]. Especially if the latency functions on the edges have large derivatives, the marginal cost taxes can be extremely large. While minimal optimal taxes for single-commodity networks can be computed in polynomial time [**?**], they are NP-hard to determine in multi-commodity networks [**?**]. Addressing a quite related question, several researchers give worst-case bounds on the largest tax needed to induce an optimal flow [**?**, **?**, **?**]. Moreover, a look at classical taxing procedures from an agents' perspective reveals that, albeit taxes improve the latency of the networks, they do not improve the disutility of agents for a large set of networks, e. g., for linear latency functions [**?**].
Alternative approaches to reduce the price of anarchy as Stackelberg routing or network design directly manipulate the network or the agents both of which are quite strong assumptions.

In this chapter, we study a means of reducing the inefficiency of selfish flow applicable in scenarios with no central control that circumvents all of the above mentioned problems. Our approach is motivated by the observation, that routing some flow in addition to the given amount of selfish flow, may in fact improve the performance of selfish flow. We introduce two sorts of additional flow, which we we call auxiliary and adversarial flow.

The goal is to route the additional flow in such a way that the induced equilibrium minimizes/maximizes the total latency of the selfish flow. The

routed packets solely alter the latency of the used edges and have no intrinsic value. Therefore we assume that the latency of the additional flow does not contribute to the total latency. Note that we equip an instance either with auxiliary or with adversarial flow depending on our goal. The demand value of the additional flow is given independently in addition to the given selfish flow demand. We want to remark that our approach has similarities to the concept of spam in the Internet. However, while in large uncoordinated networks like the Internet spam does not accompany regular digital traffic, we route the given extra amount of flow within the same commodity in order to influence its performance.

3.1 Our Results

We first present networks where auxiliary flow eradicates the inefficiency of the Wardrop equilibrium (Section 3.2). However, it turns out that both the *optimal auxiliary flow* of given value and the *minimal amount of an optimal auxiliary flow* are NP-hard to compute (Subsection 3.3.1 and 3.3.2). Further, we prove that for auxiliary flow there is no polynomial time approximation with a factor of less than $\frac{4}{3}$. The minimal amount of the optimal auxiliary flow needed to induce the best possible equilibrium cannot be approximated even by any subexponential factor. These results are complemented by proving NP-hardness for adversarial flow (Subsection 3.3.3).

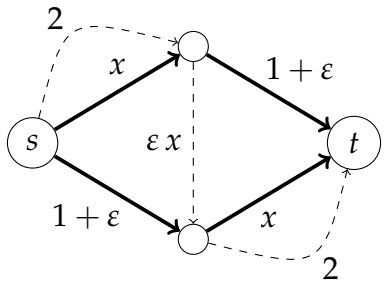
3.2 Preliminaries and Initial Results

Again we rely on Wardrop's model as described in the introduction, but we will slightly extend the model and reformulate some of the classic results we will rely on throughout this chapter.

We are given a directed graph $G = (V, E)$, one commodity specified by a source-sink pair $(s, t) \in V \times V$, and a unit flow demand. Additionally to the given selfish flow, we introduce two kinds of flows - auxiliary flow and adversarial flow of demand $\delta > 0$. The objective of the auxiliary/adversarial flow is to minimize/maximize the total latency of the induced equilibrium of the selfish flow. Given the routes of the additional flow and the selfish flow, the total latency equals

$$C(f, \delta) = \sum_{e \in E} \ell_e(f_e + \delta_e) f_e \ .$$

If not specified further, we refer by flow to the selfish flow. Finally, we call the tuple $\Gamma = (G, (s, t), \delta)$ an *instance*. Since in this chapter we will confine ourselves to single-commodity networks the definition of a Wardrop equilibrium now reads as follows.



(a)

Figure 3.1: In absence of additional flow, a unit demand of the selfish flow uses only the zig-zag-path at equilibrium. Routing auxiliary flow of demand $1/(2\varepsilon)$ over the dashed path increases the latency on the top down edge. The selfish flow then splits half-half among the bold paths and reaches the social optimum.

Definition 9 (Wardrop equilibrium with additional flow)**.** *Given an instance* Γ *and fixed routes for the additional flow* δ*, a flow vector* f *is at* Wardrop equilibrium *if and only if for paths* $P_1, P_2 \in \mathcal{P}$ *with* $f_{P_1} > 0$ *it holds that* $\ell_{P_1}(f+\delta) \leq \ell_{P_2}(f+\delta)$*.*

Note that the extra commodity δ is not composed of stabilizing selfish agents. Instead, the aim is to allocate this flow in a coordinated way to influence the total latency of the Wardrop equilibrium. Our optimization problem is similar to Stackelberg routing [**?**]. In particular, it can be formulated as a bilevel problem, where in a first phase the extra flow flow is allocated to the routes. The additional flow naturally changes the latency on the used edges. In a second phase the selfish flow stabilizes at Wardrop equilibrium depending on the allocation in the first phase. The resulting latency of the selfish flow is to be optimized by the allocation of auxiliary/adversarial flow in the first place.

Let us note two initial observations about auxiliary flow. Figure 3.1 yields our first observation.

Observation 17. *There are networks in which auxiliary flow eradicates the inefficiency of selfish routing.*

One can easily modify the network in Figure 3.1, such that even an arbitrary small amount of auxiliary flow does the job.

Observation 18. *Adding auxiliary flow to selfish flow increases the path latency in series-parallel graphs. Since the total latency at equilibrium equals the path latency L, auxiliary flow of arbitrary value does not improve the total latency at equilibrium.*

3.3 Computational Complexity of Optimal Additional Flows

In this section, we discuss the computational complexity of problems related to auxiliary and adversarial flow.

In the decision problem Optimal-Flow we are given a single-commodity selfish routing instance, some auxiliary flow, and a threshold value C. The problem is to decide if there is a routing of the auxiliary flow such that the total latency at equilibrium is at most C.

In the decision problem Threshold-Flow we are given a single-commodity selfish routing instance and auxiliary flow of amount δ. The problem is to decide if there is a routing of the auxiliary flow such that the total latency of the equilibrium is less or equal than the total latency at equilibrium induced by any auxiliary flow $\delta' > \delta$.

In the decision problem Worst-Flow a single-commodity selfish routing instance is given, some adversarial flow, and a threshold value C. The problem is to decide if there is a routing of the adversarial flow such that the total latency at equilibrium is at least C.

We will show NP-hardness of these decision problems and give strong inapproximability results. Our results are based on extensions of Roughgarden's proof of NP-hardness for the Network-Design problem [**?**]. Motivated by Braess's paradox, the author formulates the problem as: Given an instance (G, ℓ) of the routing problem, which subnetwork $H \subset G$ allows a Wardrop equilibrium with minimal total latency? Roughgarden shows that for linear latency functions Network-Design is NP-hard to approximate with a factor less than $4/3$. This negative result carries over to the case, in which taxes are to minimize the total user disutility (latency plus tax) at equilibrium [**?**]. The main result of this section is the strong inapproximability of Threshold-Flow. This results sharply contrasts the work of Kaporis and Spirakis [**?**] on Stackelberg routing, which stated that the minimal amount of flow that a Stackelberg leader needs to induce an optimal flow can be computed in polynomial time by virtue of a surprisingly simple algorithm. They dubbed the minimal amount of flow needed "Price of Optimum".

3.3.1 Complexity of Optimal-Flow

Observation 17 shows that auxiliary flow can improve the total latency at Wardrop equilibrium. Here, we show that computing the optimal routing for a given auxiliary flow is NP-hard.

Theorem 19. Optimal-Flow *is* NP*-hard.*

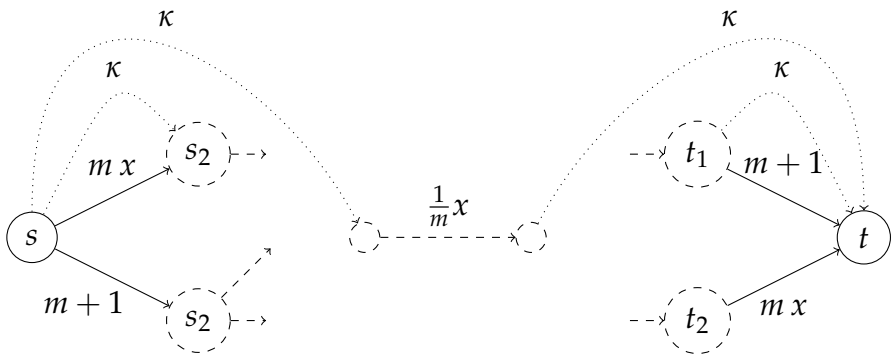


Figure 3.2: This figure outlines the construction of G'. The dashed edges are the edges of G and the dotted edges are the edges in P. The edges are labeled with their latency functions.

Proof. We reduce from the problem 2 Directed Disjoint Path (2DDP) which is known to be NP-hard [**?**]. An instance $I = (G, (s_1, t_1), (s_2, t_2))$ is a directed graph G with two distinguished pairs of vertices (s_1, t_1) and (s_1, t_2). An instance I belongs to 2DDP, that is $I \in$ 2DDP if and only if there exist two vertex disjoint paths in G from s_1 to t_1 and from s_2 to t_2, respectively. Without loss of generality, we assume that there exist arbitrary paths from s_1 to t_1 and from s_2 to t_2, respectively.

Given an instance $I = (G, (s_1, t_1), (s_2, t_2))$ with $G = (V, E)$ and $|E| = m$, we construct a single-commodity selfish routing game $\Gamma = (G', (s, t), 3m^2)$ that has the following properties: If and only if $I \in$ 2DDP, optimal auxiliary flow induces a Wardrop equilibrium with total latency of less than $C = \frac{3}{2}m + \frac{5}{2}$.

We construct $G' = (V', E')$ as follows: $V' = V \cup \{s, t\}$ and

$$E' = E \cup \{(s, s_1), (s, s_2), (t_1, t)(t_2, t)\} \cup P$$

with $P = \{(s, u), (v, t) \mid \text{for all } (u, v) \in E\}$. The latency function of each edge $e \in E$ is $\ell_e(x) = \frac{1}{m}x$, for the edges $e \in \{(s, s_1), (t_2, t)\}$ it is $\ell_e(x) = mx$, for the edges $e \in \{(s, s_2), (t_1, t)\}$ it is $\ell_e(x) = m + 1$, and for all edges $e \in P$ it is $\ell_e(x) = \kappa$, where κ is a large constant only depending on m, e. g., $\kappa = m^3$. Note that in equilibrium no selfish flow is assigned to an edge $e \in P$, because the latency of κ is much larger than the latency of any s-t-path not including an edge $e \in P$.

If $I \in$ 2DDP, then in G' there exist two disjoint paths from s_1 to t_1 and from s_2 to t_2, respectively. Let $D \subseteq E$ be the set of edges of these two paths. An auxiliary flow that assigns for all $(u, v) \in E \setminus D$ flow of at least $3m$ to each of the edges $(s, u), (v, t) \in P$, and (u, v) essentially forces the selfish flow to use

the two disjoint paths only. The latency for flow demand d' on such a path is at least $md' + m + 1$ and at most $md' + m \cdot \frac{1}{m} + m + 1$. Solving

$$md' + m + 1 + \alpha/m = m(1 - d') + m + 1 + \beta/m$$

with $0 \leq \alpha \leq \beta \leq m$ for d' and $1 - d'$ shows that at equilibrium the maximal flow on each of the two paths is upper bounded by $\frac{m+1}{2m}$. Therefore, the latency of a path at a resulting Wardrop equilibrium is at most $\frac{3}{2}m + \frac{5}{2}$ and the total latency is at most C. In particular, there is an optimal routing of the auxiliary flow such that the total latency at equilibrium is at most C.

If $I \notin$ 2DDP, we show that there no auxiliary flow that induces an equilibrium flow with total latency of less than $2m$. We distinguish several cases by the usage of the four edges incident to s and t. Since we have unit demand and all used paths have the same length at equilibrium, it suffices to show that there is a used path with latency of at least $2m$.

1. If a flow uses a path starting with (s, s_2) and ending with (t_1, t), this path has total latency of at least $2m + 2$.

2. If a flow uses only paths starting with (s, s_1) and ending with (t_2, t), it has total latency of at least $2m$.

3. If a flow uses only paths starting with (s, s_1) and ending with (t_2, t) or (t_1, t), the latency from s_1 to t must be the same on all paths. Therefore every path has latency of at least $2m + 1$.

4. If a flow uses only paths starting with (s, s_1) or (s, s_2) and ending with (t_2, t), the same argument holds.

5. If a flow uses at least one path starting with (s, s_1) and ending with (t_1, t) and at least one path starting with (s, s_2) and ending with (t_2, t), there exists a vertex v^* that is contained in both paths. Due to the equilibrium constraint, all path segments from s to v^* and from v^* to t must have the same latency. Thus, every path has latency of at least $2m + 2$.

Thus, the optimal auxiliary flow induces an equilibrium with total latency less or equal C in Γ if and only if $I \in$ 2DDP.

□

Note that the decision in the previous instances is whether the total latency of the selfish flow can be reduced to at most $C = \frac{3}{2}m + \frac{5}{2}$. If this is impossible, for every flow the total latency is at least $2m$. Now suppose there is a polynomial time approximation algorithm, which computes a $\left(\frac{4}{3} - \varepsilon\right)$-approximation for optimizing the total latency of selfish flow. Then, such an algorithm could be used to decide sufficiently large instances of 2DDP using the previously outlined construction. We therefore get the following corollary.

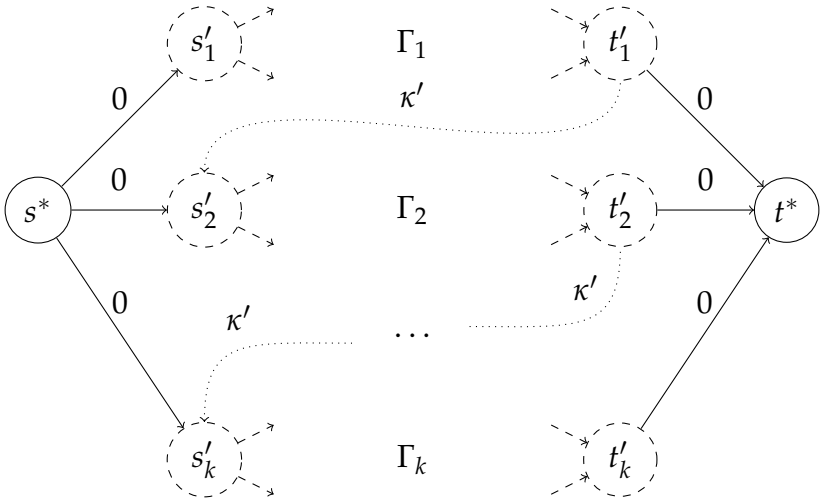


Figure 3.3: The network contains $k = 3m^2 \cdot \lceil \varepsilon^{-1} \rceil$ copies of the network G' of the proof of Theorem 19. Between s^* and t^* there is a demand of k.

Corollary 20. *For every $\varepsilon > 0$ it is* NP*-hard to approximate* Optimal-Flow *on instances with linear latency functions to a factor of $\frac{4}{3} - \varepsilon$.*

In addition, note that in the NP-hardness reduction the auxiliary flow is much larger than the demand of selfish flow. However, we can show that the result even holds if the auxiliary flow is only a (polynomially) small fraction of the selfish demand.

Theorem 21. Optimal-Flow *is* NP*-hard to approximate to a factor of $\frac{4}{3} - \varepsilon$ for every constant $\varepsilon > 0$ on instances with linear latency functions and auxiliary flow $\delta \in \mathcal{O}\left(\frac{d}{poly(m)}\right)$.*

Proof. Again, we reduce from 2DDP. Given an instance I and an $\varepsilon > 0$, we construct a selfish routing game Γ as described in the proof of Theorem 19. We use $k = 3m^2 \cdot \lceil \varepsilon^{-1} \rceil$ copies $\Gamma_1, \ldots, \Gamma_k$ of this game to create a new game Γ' as follows. We add a source vertex s^* and a target vertex t^*. The vertex s^* is connected to each source vertex s'_i of Γ_i (for all $1 \leq i \leq k$) by an edge (s^*, s_i) with latency function $\ell_{(s^*,s_i)}(x) = 0$. Likewise, there is an edge with $\ell_{(t'_i,t^*)}(x) = 0$ from each vertex t'_i to t^*. Additionally, for every $i \in \{1, \ldots, k-1\}$, there is an edge from t'_i to s'_{i+1} with $\ell_{(t'_i,s_{i+1})}(x) = \kappa'$, where $\kappa' = k^4$. The demand of the selfish flow is $d = k$, and the auxiliary flow is limited to $3m^2$ and $C = d \cdot (\frac{3}{2}m + \frac{5}{2})$.

If $I \in$ 2DDP, there is an auxiliary flow that yields an equilibrium flow with total latency of at most $d \cdot (\frac{3}{2}m + \frac{5}{2})$: We assign auxiliary flow of at most $3m^2$

between the vertices s'_i and t'_i in each copy Γ_i as described in the proof of Theorem 19. We assign the same amount of flow to the edges $\{(s^*, s'_1), (t'_1, s'_2), \ldots, (t'_{k-1}, s'_k), (t'_k, t^*)\}$ to obtain a flow of at most $3m^2$ from s^* to t^*. At the resulting Wardrop equilibrium, there is a flow of 1 that is assigned to each copy Γ_i and the edges that connect it to s^* and t^*. Each of these flows has latency of at most $\frac{3}{2}m + \frac{5}{2}$. Thus, the total latency sums up to at most $d \cdot (\frac{3}{2}m + \frac{5}{2})$.

If $I \notin$ 2DDP, the total latency of the selfish flow is more than $d \cdot 2m$. Note that at equilibrium the selfish flow never chooses an edge that connects two of the copies because it has latency of κ', and there is always a s^*-t^*-path with lower latency. Therefore, there is at least one copy Γ_i in which flow of at least 1 is routed from s'_i to t'_i. As shown in the proof of Theorem 19, the latency of the s'_i-t'_i-paths is at least $2m$. Since the flow is at Wardrop equilibrium, every path between s'_j and t'_j for every $1 \leq j \leq k$ has latency of at least $2m$. Thus, the total latency sums up to more than $d \cdot 2m$. □

Due to the known price of anarchy result [**?**], this hardness result can be restated in view of algorithm design as: For linear latency functions the trivial algorithm, i. e., routing no auxiliary flow, is the optimal algorithm.

3.3.2 Complexity of Threshold-Flow

The previous result showed that it is computationally infeasible to compute the best possible auxiliary flow. In this section we show that even the minimal amount of auxiliary flow that is needed to achieve the best possible Wardrop equilibrium is hard to approximate.

Note that this result strongly contrasts the corresponding result of Kaporis and Spirakis [**?**] for Stackelberg routing. In Stackelberg routing the minimal fraction of flow needed by the Stackelberg leader to induce optimal latency can be computed in polynomial time for arbitrary multi-commodity networks using a surprisingly simple algorithm.

Theorem 22. Threshold-Flow *is* NP*-hard.*

Proof. Again, we reduce from 2 Directed Disjoint Path (2DDP). Given an instance $I = (G, (s_1, t_1), (s_2, t_2))$ with $G = (V, E)$ and $|E| = m$, we construct a single-commodity selfish routing game that has an optimal auxiliary flow of at most polynomial in m if and only if $I \in$ 2DDP. Construct $\Gamma = (G', (s, t), \text{poly}(m))$ as described in the proof of Theorem 19 and modify it as follows. Remove the edge (t_2, t) and replace it with the following gadget. Add the vertices u and v and the edges $(t_2, u), (u, v), (u, t), (t_2, v), (v, t)$. Latency functions are $\ell_e(x) = (\frac{m}{2} - \frac{1}{2^{m+1}})x$ for the edges $e \in \{(t_2, u), (v, t)\}$ and $\ell_e(x) = \frac{m}{2} + \frac{1}{2^{m+1}}$ for the edges $e \in \{(u, t), (t_2, v)\}$ and $\ell_{(u,v)}(x) = \frac{1}{2^m}x$. In

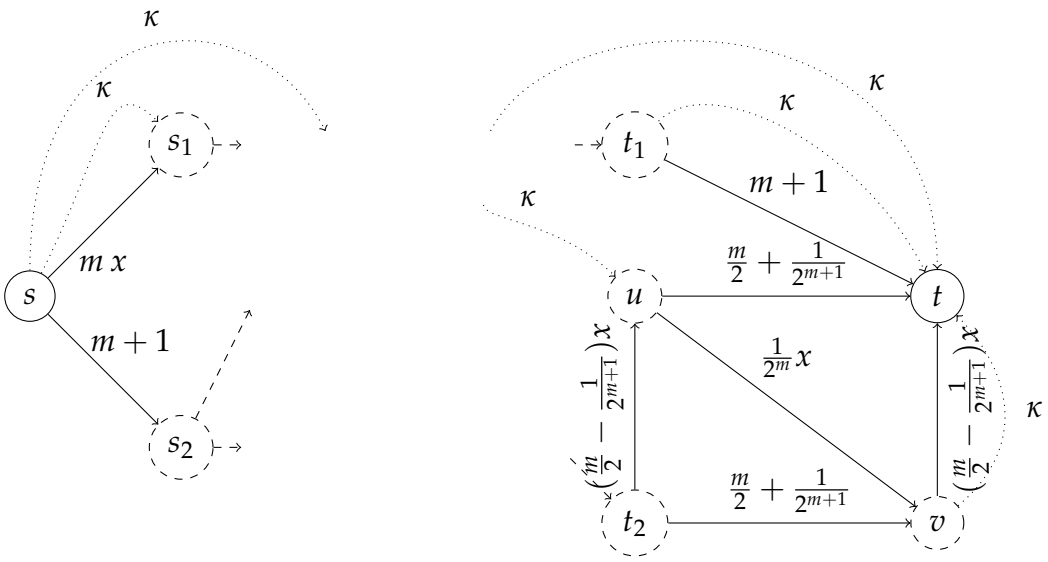


Figure 3.4: This figure outlines the modified construction of G' for the proof of Theorem 22.

addition, we add edges $(s,u),(s,v),(u,t)$ and (v,t) with latency κ to the set P (cf. proof of Theorem 19).

Observe that for routing flow demand $d' \leq \frac{m2^{m+1}+2}{3m2^m+1}$ from t_2 to t, it is optimal to leave all selfish flow on the zig-zag path, which generates path latency md' and also yields an equilibrium. Note that the optimum assignment of selfish flow that is achievable by (marginal cost) taxing might split the flow along all three possible paths from t_2 to t. However, the resulting latency of such a flow is larger here, as the auxiliary flow, which can used to simulate taxes in our gadget, is accounted in the latency of selfish flow. For flow larger than $\frac{m2^{m+1}+2}{3m2^m+1}$, splitting the flow and assigning $\frac{d'}{2}$ to the edges (t_2,u), (t_2,v), (u,t), and (v,t) yields a better latency. This flow and its improved latency can be induced using a sufficiently large auxiliary flow along edge (u,v). The auxiliary flow needs only to be large enough to prevent the selfish flow from using the edge (u,v). Observe, that for large demand values in our gadget splitting the flow among (t_2,u,t) and (t_2,v,t) by blocking (u,v) yields also a better latency than assigning an amount of flow to (u,v) that still allows some selfish flow to use this edge as well.

If $I \in$ 2DDP, then in G' there exist two disjoint paths from s_1 to t_1 and from s_2 to t_2, respectively. Again, let $D \subset E$ be the set of edges of these two paths. Then an auxiliary flow that assigns, for all $(e,e') \in E \setminus D$, flow of volume $3m$ to each of the edges (s,e), (e,e') and (e',t) forces the selfish flow to use the two disjoint paths only. Thus, the flow becomes almost balanced between the two disjoint paths. The best possible Wardrop equilibrium can be

reached by sending additional auxiliary flow slightly unevenly over the edges of both disjoint paths. Nevertheless, a polynomial amount of auxiliary flow is sufficient to totally balance the selfish flow. To see this, assume some edge in D receives a super-polynomial amount of auxiliary flow. The resulting latency of this edge would then surpass the path latency of the other disjoint path. Hence, an auxiliary flow of demand $\text{poly}(m)$ is sufficient to obtain the best possible Wardrop equilibrium.

Note that selfish flow of demand close to $1/2$, i. e., less than $\frac{m2^{m+1}+2}{3\,m2^m+1}$ is routed through the gadget from t_2 to t. Therefore, it is not necessary to route auxiliary flow over the edge (u, v).

If $I \notin$ 2DDP, then optimal auxiliary flow yields a Wardrop equilibrium in which the whole selfish demand is routed from s via s_1 and t_2 to t. Especially, a unit demand is being routed through the gadget between t_2 and t. The optimal auxiliary flow thus must block edge (u, v). Hence, it needs to route auxiliary flow of demand δ over (u, v), such that

$$\frac{1}{2^m}\delta + \left(\frac{m}{2} - \frac{1}{2^{m+1}}\right) \cdot \frac{1}{2} \geq \frac{m}{2} + \frac{1}{2^{m+1}} \ ,$$

i. e., $\delta \in \Omega(2^m)$. □

Note that the latency functions in our gadget use exponentially large coefficients. Nevertheless, the latency functions can be represented by a polynomial number of bits in the input size, assuming that the numbers in our instance are represented in binary coding.

The proof shows that deciding 2DDPcan be reduced to the decision whether a polynomial auxiliary flow can be optimal in the previous instances or not. Thus, we have the following corollary.

Corollary 23. *For any constant $\varepsilon > 0$, it is* NP*-hard to approximate* Threshold-Flow *by a factor of $2^{m(1-\varepsilon)}$.*

3.3.3 Complexity of Worst-Flow

We have seen that the optimal auxiliary flow is NP-hard to compute. We now turn to the computational complexity of computing the optimal adversarial flow.

Theorem 24. Worst-Flow *is* NP*-hard.*

Proof. We reduce from the NP-hard problem Hamilton. A graph $G \in$ Hamilton if and only if G contains a Hamiltonian path. Given a directed graph $G = (V, E)$ with $|V| = n$ and $|E| = m$ and two vertices $x, y \in V$, we construct a selfish routing game $\Gamma = (G', (s, t), \delta)$ with the property that the latency

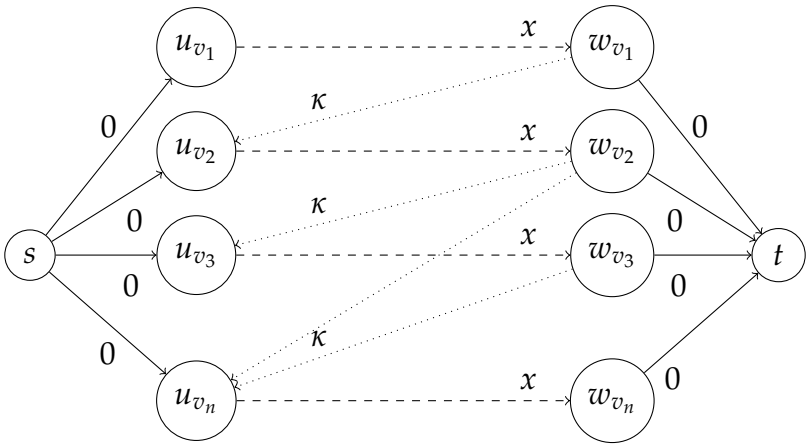


Figure 3.5: This figure depicts the corresponding graph G' for an instance G of the problem Hamilton. The dashed edges correspond to vertices in G and the dotted edges correspond to edges in G'.

maximizing adversarial flow induces total latency of at least $C = \frac{1}{n} + \delta$ if and only if $G \in$ Hamilton.

We construct $G' = (V', E')$ as follows: For every vertex v in G there is a pair of vertices u_v, w_v in G' and, additionally we have a source and a sink vertex s and t. That is, $V = \{s, t\} \cup \{u_v, w_v \mid \forall v \in V\}$. There are edges from s to all vertices u_v, from each vertex u_v to w_v and from all vertices w_v to t. The selfish flow will use only these edges. Additionally, we have edges (with high latency) that connect a vertex w_v with a vertex $u_{v'}$ if there is an edge from v to v' in the graph G for $v' \in V - \{x\}$. To summarize, $E' = S' \cup U' \cup W'$ with

$$\begin{aligned} S' &= \{(u_v, w_v) \mid \forall v \in V\}, \\ U' &= \{(s, u_v), (w_v, t) \mid \forall v \in V\}, \text{ and} \\ W' &= \{(w_v, u_{v'}) \mid \forall (v, v') \in E \text{ and } v' \in V - \{x\}\}. \end{aligned}$$

For all edges $e \in S'$ we set $\ell_e(x) = x$, for all edges $e \in U'$ we set $\ell_e(x) = 0$, and for all edges $e \in W'$ we set $\ell_e(x) = \kappa$ for a constant $\kappa > 0$. Note that the selfish flow never uses edges $e \in W'$ and therefore assigns flow to the n paths s, u_v, w_v, t for all $v \in V$. Without adversarial flow, the equilibrium flow is equally distributed among these paths, and the total latency is $n\frac{1}{n^2} = \frac{1}{n}$.

Assume $G \in$ Hamilton and $x = v_{i_1}, \ldots, v_{i_n} = y$ is a Hamiltonian path in the network G. Then it is possible to assign adversarial flow of amount δ to all edges $e \in S'$ by choosing the path $s, u_{v_{i_1}}, w_{v_{i_1}}, u_{v_{i_2}}, w_{v_{i_2}}, \ldots, u_{v_{i_n}}, w_{v_{i_n}}, t$. Note, that the edges between the w and u vertices exist by construction. All edges with non-constant latencies carry the maximal amount of adversarial flow. This maximizes the total latency and yields $n(\frac{1}{n} + \delta) \cdot \frac{1}{n} = \frac{1}{n} + \delta$.

Consider a graph $G \notin$ Hamilton. Then there is no path in G' from s to t that visits all vertices $e \in U'$. Therefore, the adversarial flow δ can not be sent via all edges $e \in S'$, and there is at least one edge $e \in S'$ with adversarial flow less than δ. Thus, the equilibrium flow will balance accordingly among all paths containing an edge $e \in S'$. The resulting path latency and thus the total latency at equilibrium is strictly less than $\frac{1}{n} + \delta$. □

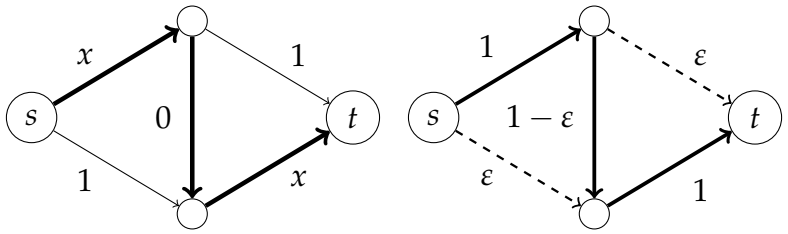
Chapter 4

Sensitivity of Wardrop Equilibria

In the preceding chapters we have tackled the problem of reducing the price of anarchy for routing instances, where fixed amounts of flow need to be routed among source-sink pairs through the network. In uncoordinated networks, however, neither the demands remain constant nor does the network topology remain unchanged. In this regard, we study how equilibrium flows react to slight modification of the network environment.

To analyze this issue, we suppose we are given an equilibrium flow for unit demand and increase the demand by ε or remove an edge carrying only an ε-fraction of flow. How does the equilibrium responds to such an ε-change in terms of change in flow and latency?

The Braess network depicted in Figure 4.1 exhibits that, in general, neither path flows nor edge flows at equilibrium are monotone functions of the demand. This observation has already been made by Braess [**?**] and suggests that studying the effects of environmental changes might be intriguing. As one immediate implication Wardrop equilibria are not computable or approximable



(a) Latency functions ... (b) ... and flow.

Figure 4.1: (a) The links are labeled with the latency functions. At equilibrium the entire demand routes over the zig-zag-path. (b) The links are now labeled with the unique equilibrium flow. Increasing the demand by $0 \leq \varepsilon \leq 1$, the zig-zag-path loses an ε-fraction.

using the following naive algorithm (for simplicity we will restrict to single-commodity instances). Given some $N > 0$, divide the total demand in chunks of size d/N. Compute a shortest path and place a flow of d/N on this path. Proceed by allocating the second flow fraction to a shortest path integrating the already located flow. Iterating this N times. As Braess's network shows, the so-established flow unfortunately can be far from the unique equilibrium flow.

4.1 Our Results

Our findings for single-commodity networks are as follows. Allowing non-decreasing, continuous latency functions, we show in Section 4.2 that for every $\varepsilon > 0$,

- there are networks, in which after an ε-change every agent is forced to change its path in order to recover equilibrium and
- the flow increase or decrease on every edge, however, is at most ε for every network.

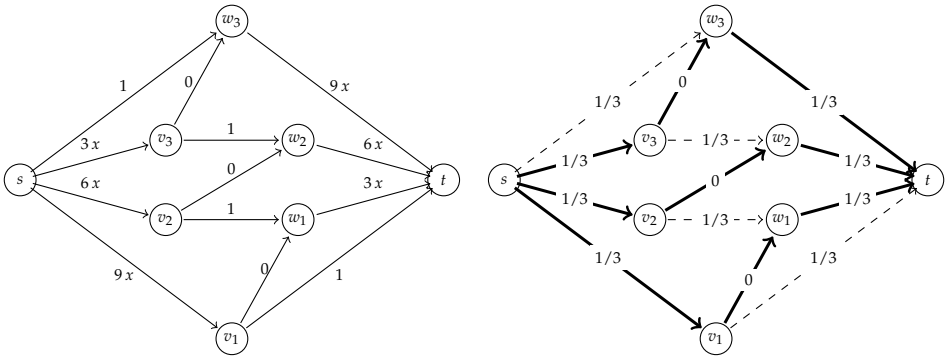
Thus, in contrast to our remarkable finding of global instability of equilibrium flow, we can prove that edge flows are locally stable. Examining the latency at equilibrium, we concentrate on polynomial latency functions of degree at most p with nonnegative coefficients. We show in Section 4.3 that, due to an ε-change in the demand,

- the path latency at equilibrium increases at most by a factor of $(1+\varepsilon)^p$ (even though the relative increase in the latency of an edge can be unbounded).

This result yields the same bound on the increase in the *Price of Anarchy*, as well. All presented bounds are best possible.

For the multi-commodity case, we present examples for every $\varepsilon > 0$ showing that neither the change in edge flows nor the increase in the path latency can be bounded. This holds already for networks equipped with linear latency functions (Section 4.4).

Most related to our work is a series of papers conducting qualitative analyses of the equilibrium under demand changes. While in multi-commodity networks the increase of one flow demand might decrease other path latencies at equilibrium [**?**], the vector of path flows and the vector of the path latencies are continuous functions of the input demand [**?**]. For single-commodity networks the path latency at equilibrium is a monotone function of the input demand. These positive results hold even for non-separable latency functions [**?**].



(a) Latency functions ... (b) ... and flow.

Figure 4.2: (a) Having unit demand, the solid paths in $B_{k=3}$ carry $1/3$ of flow each, and the dashed edges carry zero flow. (b) After increasing the demand by $(1+\varepsilon) = (1+1/3)$, the solid paths lose all their flow, and the paths containing the dashed edges gain flow of $(1+\varepsilon)/(k+1) = 1/3$ each.

4.2 Sensitivity of Equilibrium Flows

For most of the section we concentrate on the single-commodity case. First, we present a network with linear latency functions for any given $\varepsilon > 0$, in which every agent needs to change its current path to recover equilibrium. Then we prove that due to ε-changes the flow on every edge does not change by more than ε.

4.2.1 Instability of Equilibria: Every Agent Needs to Move

In [**?**] Roughgarden uses the *generalized Braess graphs* to show, that the path latency at equilibrium can arbitrarily decrease by removing several edges from a network. Our definition of B_k differs from the definition in [**?**] in the non-constant latency functions.

Definition 10 (Generalized Braess graph)**.** *For every $k \in \mathbb{N}$, let $B_k = (V_k, E_k)$ be the graph with $V_k = \{s, v_1, \ldots, v_k, w_1, \ldots, w_k, t\}$ and $E_k = \{(s, v_i), (v_i, w_i), (w_i, t) : 1 \leq i \leq k\} \cup \{(v_i, w_{i-1}) : 2 \leq i \leq k\} \cup \{(s, w_k)\} \cup \{(v_1, t)\}$. Let B_k be equipped with the following latency functions.*

- $\ell^k_{v_i,w_i}(x) = 0$ *and* $\ell^k_{s,v_{k-i+1}}(x) = \ell^k_{w_i,t}(x) = i \cdot k \cdot x$ *for* $1 \leq i \leq k$,
- $\ell^k_{v_i,w_{i-1}}(x) = 1$ *for* $2 \leq i \leq k$ *and*
- $\ell^k_{s,w_k}(x) = \ell^k_{v_1,t}(x) = 1$.

*Let B_k be called the k*th Braess graph.

Let $\varepsilon > 0$ and consider the instance $(B_{\lceil 1/\varepsilon \rceil}, 1)$.
Let $(P_1, \dots, P_{2k+1})^T = (P_{s,w_k,t}, P_{s,v_k,w_k,t}, P_{s,v_k,w_{k-1},t}, P_{s,v_{k-1},w_{k-1},t}, \dots, P_{s,v_1,t})^T$ denote the corresponding path vector. The equilibrium flow is described by the vector (f_{P_j}) of path flows

$$f_{P_j} = \begin{cases} 0 & \text{for } j = 1, 3, \dots, 2k+1 \\ 1/k & \text{for } j = 2, 4, \dots, 2k \end{cases}$$

summing up to $\sum_P f_P = \sum_{j=1}^{2k+1} f_{P_j} = 1$.

All paths have path length $\ell_P(f) = k + 1$, and since any unilateral deviation strictly increases the sustained latency, the edge flows at equilibrium are unique (Figure 4.2).

By increasing the demand by $(1 + \varepsilon)$ the equilibrium flow vector becomes (f'_{P_j}) with

$$f'_{P_j} = \begin{cases} (1+\varepsilon)/(k+1) & \text{for } j = 1, 3, \dots, 2k+1 \\ 0 & \text{for } j = 2, 4, \dots, 2k \end{cases}$$

which sums up to $\sum_P f'_P = \sum_{j=1}^{2k+1} f'_{P_j} = 1 + \varepsilon$. The path latency can easily be computed to be $1 + \frac{k^2(1+\varepsilon)}{k+1}$.

Note that the path flow decomposition at equilibrium does not need to be unique. Nevertheless, we have uniqueness in B_k.

Definition 11 (ε-edge). *An edge $e \in E$ carrying flow of at most ε is called* ε-edge.

Theorem 25. *Let $\varepsilon > 0$ and consider $(B_{\lceil \frac{1}{\varepsilon} \rceil}, 1)$. Then increasing the flow by ε causes the entire demand to be redistributed to recover a Wardrop equilibrium, i. e., every agent is forced to change its path. Adding another edge to the network, one can achieve the same result for the removal of an ε-edge.*

Proof. For the path flow vector (f_{P_j}) and (f'_{P_j}) it holds that $f_{P_j} = 0 \Leftrightarrow f'_{P_j} > 0$. For the second assertion, simply simulate a demand increase in $B_{\lceil \frac{1}{\varepsilon} \rceil}$ by directly connecting source s with sink t and choose the latency function such that (s, t) carries an ε-fraction of flow. Then remove this edge. □

Let us remark that Theorem 25 can easily be transferred to optimal flows, i. e., flows minimizing the total cost. This is since for semi-convex latency functions optimal flows are Wardrop equilibria with respect to the marginal cost functions. Thus, it is sufficient to change the linear latency functions in $B_{\lceil \frac{1}{\varepsilon} \rceil}$.

4.2.2 Edge Flows are Locally Stable

Let $f, f' \in \mathcal{F}$ be feasible flows for demands $d \leq d'$ and let $\Delta(f, f')$ denote the difference of f' and f,

$$\Delta_e(f, f') = f'_e - f_e \,, \forall e \in E \ .$$

An edge e is *positive* (with respect to f' and f) if $f'_e - f_e > 0$, and it is *negative* if $f'_e - f_e < 0$. A path is positive (or negative) if all its edges are positive (or negative). Let us remark that considering $\Delta(f, f')$ negative edges carry a positive amount of flow equal to $f_e - f'_e$ and have their directions reversed. Observe that the flow conservation property holds for the difference of two network flows.

Definition 12 (Alternating flow cycle)**.** *A cycle consisting of flow carrying edges is called an* alternating flow cycle*.*

Lemma 26. *Let f denote an equilibrium flow for an instance $(G, 1)$ with non-decreasing, continuous latency functions. Then there is an equilibrium flow f' for $(G, 1+\varepsilon)$, such that $\Delta(f, f')$ does not contain an alternating flow cycle.*

Proof. Let f' denote an equilibrium flow for $(G, 1+\varepsilon)$. Assume there is an alternating flow cycle C in $\Delta(f, f')$. Since we can assume both equilibrium flows to be cycle free, we can assume that the alternating flow cycle C contains positive and negative edges. C can thus be divided into positive and negative path segments, $C = p_1 n_1 p_2 \dots n_k$, where p_i denotes a sequence of positive edges and n_i denotes a sequence of negative edges. Let u_i be the first node of p_i and denote the last node of n_i by v_i. Thus, there are two paths from u_1 to v_k in C. For $u, v \in V$, let $\ell(u, v)$ denote the minimum path latency from u to v under f. For $u = s$ simply write $\ell(v)$. For f' write $\ell'(u, v)$ and $\ell'(v)$.

There are two facts we will make consistently use of. Since at equilibrium the flow routes only on shortest paths, we have

$$\ell(v) \leq \ell(u) + \ell(u, v) \text{ for any } u, v \in V \ , \text{ and} \tag{4.1}$$

$$\ell(v) = \ell(u) + \ell(u, v) \tag{4.2}$$

if there is a flow carrying path between s and v containing u. We show that assuming f and f' being at equilibrium yields $\ell'(u_1, v_k) = \ell(u_1, v_k)$. On the one hand, since n_k connects u_1 with v_k and there is more flow on every edge of n_k under f than under f', we have

$$\ell'(u_1, v_k) \leq \sum_{e \in n_k} \ell_e(f'_e) \leq \sum_{e \in n_k} \ell_e(f_e) = \ell(u_1, v_k) \ .$$

For the reverse direction we show $\ell'(v_k) \geq \ell'(u_1) + \ell(u_1, v_k)$, since then $\ell(u_1, v_k) \leq \ell'(v_k) - \ell'(u_1) \leq \ell'(u_1, v_k)$. In the following, we repeatedly make use of Equations (4.1) and (4.2).

$$\begin{aligned}
\ell'(v_k) &= \ell'(u_k) + \ell'(u_k, v_k) \geq \ell'(v_{k-1}) - \ell'(u_k, v_{k-1}) + \ell'(u_k, v_k) \\
&= \ell'(u_{k-1}) + \ell'(u_{k-1}, v_{k-1}) - \ell'(u_k, v_{k-1}) + \ell'(u_k, v_k) \\
&\geq \ell'(u_1) + \sum_{i=1}^{k} \ell'(u_i, v_i) - \sum_{i=2}^{k} \ell'(u_i, v_{i-1}) \\
&\geq \ell'(u_1) + \sum_{i=1}^{k} \ell(u_i, v_i) - \sum_{i=2}^{k} \ell(u_i, v_{i-1}) \\
&\geq \ell'(u_1) + \sum_{i=1}^{k} (\ell(v_i) - \ell(u_i)) - \sum_{i=2}^{k} (\ell(v_{i-1}) - \ell(u_i)) \\
&= \ell'(u_1) - \ell(u_1) + \ell(v_k) = \ell'(u_1) + \ell(u_1, v_k) \ .
\end{aligned}$$

The third inequality is valid since f and f' route only on shortest paths. Explicitly, $\ell'(u_i, v_i) = \sum_{e \in p_i} \ell_e(f'_e) \geq \sum_{e \in p_i} \ell_e(f_e) \geq \ell(u_i, v_i)$ for each $i \in [k]$ and $\ell'(u_i, v_{i-1}) \leq \sum_{e \in n_i} \ell_e(f'_e) \leq \sum_{e \in n_i} \ell_e(f_e) = \ell(u_i, v_{i-1})$ for each $i \in \{2, \ldots, k\}$. Thus, $\ell'(u_1, v_k) = \ell(u_1, v_k)$.

We deduce that the latency on every edge $e \in n_k$ does not change due to the flow change. Since the same analysis can be conducted for any path segment p_i and n_i, the latency of both paths on C connecting two arbitrary nodes remains unchanged. Therefore, by removing the bottleneck edge flow in C no edge latency is affected and the alternating flow cycle is eliminated. We may remove the set of alternating flow cycles in any order. Adding f to the altered difference, one gets the desired equilibrium flow for demand $1 + \varepsilon$. □

Thus, $\Delta(f, f')$ can be assumed a network flow of volume ε when edges are allowed to be traversed in both directions. We can now state the following theorem.

Theorem 27. *Let f denote an equilibrium flow for an instance $(G, 1)$ with non-decreasing, continuous latency functions ℓ.*

- *Then there is an equilibrium flow f' for $(G, 1 + \varepsilon)$, such that for all $e \in E$ it holds that $|\Delta_e(f, f')| \leq \varepsilon$.*
- *Consider an ε-edge (u, v) in G. There is an equilibrium flow f' for $(G' = (V, E - \{(u, v)\}), 1)$ such that $|\Delta_e(f, f')| \leq \varepsilon$ for all $e \in E$.*

Proof. Since the difference of f and f' can be assumed alternating flow cycle free, it constitutes a network flow of volume ε. To show the second assertion, let a single ε-edge (u, v) be removed. With the same argumentation as in

Lemma 26, we can exclude alternating flow cycles in $\Delta(f, f')$ that do not include (u, v). Due to the flow conservation property for every node $u \neq w \neq v$, $\Delta(f, f')$ is a network flow from u to v of volume ε. □

Note that since every edge gains or loses at most an ε amount of flow (Theorem 27), with respect to the number of paths $B_{\lceil \frac{1}{\varepsilon} \rceil}$ is a minimal example exhibiting global instability.

4.3 Stability of the Path Latency

The latency increase at equilibrium due to a demand increase clearly depends on the latency functions. Considering polynomials with nonnegative coefficients, the maximal degree is the critical parameter. Note that the results in this section do not trivially result from Theorem 27, since the relative flow increase on an edge might be unbounded.

Theorem 28. *Let f and f' be equilibrium flows for instances $(G, 1)$ and $(G, 1+\varepsilon)$ with polynomial latency functions ℓ of degree at most p with nonnegative coefficients. Let L and L' denote the corresponding path latencies. Then $L' \leq (1+\varepsilon)^p \cdot L$.*

Proof. Due to a scaling argument it is sufficient to consider monic monomials as latency functions. For equilibrium flows f and f' we have

$$L = \sum_{P \in \mathcal{P}} f_P \ell_P(f) = \sum_e f_e \ell_e(f_e) \quad \text{and} \quad (1+\varepsilon) \cdot L' = \sum_e f'_e \ell_e(f'_e) \ ,$$

and we want to show that $\sum_e {f'_e}^{p_e+1} \leq (1+\varepsilon)^{p+1} \sum_e f_e^{p_e+1}$, where $\ell_e(x) = x^{p_e}$. Since equilibrium flows f and f' minimize the potential function

$$\Phi(x) = \sum_e \int_0^{x_e} \ell_e(u)\, du$$

over feasible flows x of volume 1 and $(1+\varepsilon)$, respectively, it holds that

$$\Phi(f) \leq \Phi\left(\frac{f'}{1+\varepsilon}\right) \quad \text{and} \quad \Phi(f') \leq \Phi((1+\varepsilon) \cdot f)$$

More explicitly,

$$(1+\varepsilon)^{p+1} \cdot \Phi(f) = (1+\varepsilon)^{p+1} \cdot \sum_e \frac{1}{p_e+1} f_e^{p_e+1} \leq \sum_e \frac{(1+\varepsilon)^{p-p_e}}{p_e+1} {f'_e}^{p_e+1} \ , \quad \text{(A)}$$

and similarly,

$$\Phi(f') = \sum_e \frac{1}{p_e+1} {f'_e}^{p_e+1} \leq \sum_e \frac{(1+\varepsilon)^{p_e+1}}{p_e+1} f_e^{p_e+1} \ . \quad \text{(B)}$$

For contradiction, assume

$$(1+\varepsilon)^{p+1}\sum_e f_e^{p_e+1} < \sum_e f'^{p_e+1}_e \ . \tag{C}$$

Calculating $p\cdot(A)+((p+1)(1+\varepsilon)^p-1)\cdot(B)+((1+\varepsilon)^p-1)\cdot(C)$ yields

$$\sum_{k=0}^{p} c_k \sum_{p_e=k} f_e^{p_e+1} < \sum_{k=0}^{p} c'_k \sum_{p_e=k} f'^{p_e+1}_e \ , \tag{4.3}$$

with

$$\begin{aligned} c_k &= p\cdot\frac{(1+\varepsilon)^{p+1}}{k+1} \\ &- ((p+1)(1+\varepsilon)^p-1)\cdot\frac{(1+\varepsilon)^{k+1}}{k+1} \\ &+ ((1+\varepsilon)^p-1)\cdot(1+\varepsilon)^{p+1} \ . \end{aligned}$$

and

$$c'_k = p\cdot\frac{(1+\varepsilon)^{p-k}}{k+1} - ((p+1)(1+\varepsilon)^p-1)\cdot\frac{1}{k+1} + ((1+\varepsilon)^p-1) \ .$$

In the following we show that $c'_k \le 0$ for $0 \le k \le p$. Similar arguments can be used to show $c_k \ge 0$. Hence, we have a contradiction to Equation (4.3).

For any $0 \le k \le p$ and $\varepsilon = 0$, we have $c'_k = 0$. We show that c'_k is monotonically decreasing in ε (for $\varepsilon \ge 0$). The derivative of c'_k with respect to $(1+\varepsilon)$ is

$$\frac{\partial c'_k}{\partial(1+\varepsilon)} = p\cdot(p-k)\cdot\frac{(1+\varepsilon)^{p-k-1}}{k+1} - p\cdot(p+1)\frac{(1+\varepsilon)^{p-1}}{k+1} + p\cdot(1+\varepsilon)^{p-1} \ .$$

Thus, it is sufficient to show that

$$\frac{1}{(1+\varepsilon)^{p-k-1}}\cdot\frac{\partial c'_k}{\partial(1+\varepsilon)} \le 0 \ .$$

The inequality is equivalent to $(p-k) \le (p-k)\cdot(1+\varepsilon)^k$ which concludes the proof of the theorem. □

The bound is tight as shown by the network consisting of two nodes connected by an edge equipped with the latency function $\ell(x) = x^p$. Allowing negative coefficients the relative increase obviously can be unbounded.

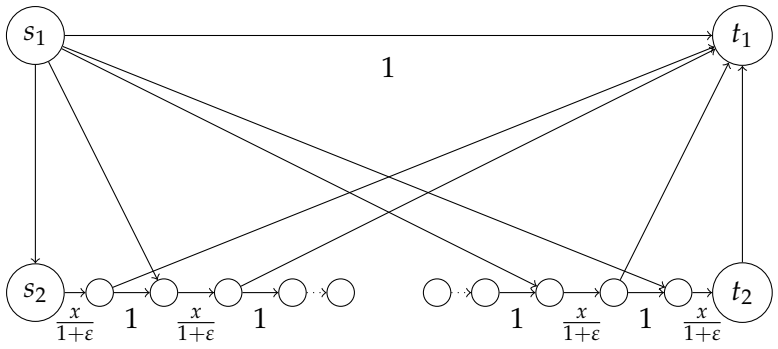


Figure 4.3: Unlabeled edges cause no latency. Assume there are $2 \cdot \lceil \frac{1}{\varepsilon} \rceil - 1$ many edges on the unique path connecting s_2 with t_2. For $d_1 = d_2 = 1$, the flow demand of commodity 1 is uniformly spread over all $\lceil 1/\varepsilon \rceil$ paths using one edge on the path connecting s_2 and t_2. After increasing d_2 by ε, we have $f_{(s_1,t_1)} = 1$.

4.3.1 Increase of the Price of Anarchy

The price of anarchy quantifies the degradation of performance due to selfish behavior. Recall that the price of anarchy for an instance $(\mathcal{G}, d)$ is defined as $\rho(\mathcal{G}, d) = \frac{C(f^*)}{C(f)}$, where f and f^* denote an optimal flow and an equilibrium flow, respectively. In **[?]** the price of anarchy is shown to be asymptotically $\Theta(\frac{p}{\ln p})$ for polynomial latency functions of degree at most p with nonnegative coefficients.

Corollary 29. *Let ρ and ρ' denote the price of anarchy for instances $(\mathcal{G}, 1)$ and $(\mathcal{G}, 1+\varepsilon)$ with polynomial latency functions of degree at most p with nonnegative coefficients. Then $\rho' \leq (1+\varepsilon)^p \cdot \rho$.*

Proof. Let $\bar{L}_d$ denote the average path latency for an optimal flow in $(\mathcal{G}, d)$. Let C, C^*, C' and C'^* denote the costs of an optimal flow and an equilibrium flow for demands 1 and $1+\varepsilon$, respectively. Then $\rho = C^*/C$ and $\rho' = C'^*/C'$. Since $C = 1 \cdot \bar{L}_1$ and $C' = (1+\varepsilon) \cdot \bar{L}_{1+\varepsilon}$, we have

$$(1+\varepsilon) \cdot C = (1+\varepsilon) \cdot \bar{L}_1 \leq (1+\varepsilon) \cdot \bar{L}_{1+\varepsilon} = C' \ ,$$

since the average latency is clearly monotone in the demand. Thus, the increase of the price of anarchy can be bounded by

$$\frac{\rho'}{\rho} = \frac{C'^*/C'}{C^*/C} = \frac{L' \cdot (1+\varepsilon) \cdot C}{L \cdot C'} \leq \frac{L \cdot (1+\varepsilon)^p \cdot (1+\varepsilon) \cdot C}{L \cdot C \cdot (1+\varepsilon)} = (1+\varepsilon)^p \ ,$$

where the inequality is due to Theorem 28. □

This upper bound is tight in the following sense: There is a network family (G, d) with latency function $\ell(p)$, such that $\lim_p \frac{\rho'/\rho}{(1+\varepsilon)^p} = 1$ for every $\varepsilon > 0$.

This holds for mildly modified instances of Pigou's example [**?**]. Replace the latency functions 1 and x in Pigou's example with $(1+\varepsilon)^p$ and x^p. We calculate $C^* = 1, C'^* = (1+\varepsilon)^{p+1}$,

$$C = \frac{(1+\varepsilon)^{p+1}}{(p+1)^{(p+1)/p}} + (1 - \frac{1+\varepsilon}{(p+1)^{1/p}})(1+\varepsilon)^p \ ,$$

and

$$C' = \frac{(1+\varepsilon)^{p+1}}{(p+1)^{(p+1)/p}} + (1+\varepsilon - \frac{1+\varepsilon}{(p+1)^{1/p}})(1+\varepsilon)^p \ .$$

Thus, we have

$$\frac{\rho'}{\rho} = (1+\varepsilon)^p \cdot \left(1 - \frac{(p+1)^{1/p}\varepsilon\, p}{(p+1)^{(p+2)/p} - p(p+1)^{1/p}}\right) \ ,$$

and it holds that $\lim_p \frac{\rho'/\rho}{(1+\varepsilon)^p} = 1$ for every fixed $\varepsilon > 0$.

4.4 Instability in Multi-Commodity Networks

There are no analogous results to Theorem 27 and 28 for the multi-commodity case. Figure 4.3 shows a network with two commodities, with both demands being 1, in which after increasing the demand of the second commodity or both demands by ε the entire demand of the first commodity needs to be shifted to a single edge to recover an equilibrium state. If a single ε-edge is being removed, other edges might also lose an arbitrary fraction of the commodity's demand.

Also, the path latency of one commodity can increase arbitrarily in multi-commodity networks. Consider a network of three nodes $s = s_1 = s_2$, t_1 and t_2, three edges (s, t_1), (s, t_2), and (t_1, t_2), latency functions $\ell_{s,t_1}(x) = x$, $\ell_{s,t_2}(x) = k\,x$, and $\ell_{t_1,t_2}(x) = k^2 - 1$, and demands $d_1 = 1$ and $d_2 = k$. If both demands are increased by a factor of $(1+\varepsilon)$, the path latency of the first commodity multiplicatively increases by $1 + k \cdot \varepsilon$. (Insisting on unit demands, one can split commodity 2 into k small commodities.) Simple examples exhibit an even higher increase.

Chapter 5

Distributed Approximation

Given complete information about the game, Wardrop equilibria can be formulated as convex programs (under some mild assumptions on the latency functions) and can thus be solved by centralized algorithms in polynomial time. In particular, the convex programming formulation requires the exact latency functions and the demand of every commodity. In this chapter, we refrain from the complete information premise and study *distributed* algorithms to compute Wardrop equilibria.

The common game-theoretic interpretation of the Wardrop model, which we heavily made use of in the previous chapters, assumes an infinite number of agents, each of which carries an infinitesimal amount of flow. In [**?**] it was shown that such a set of agents approaches Wardrop equilibria quickly by following a simple round-based load-adaptive rerouting policy (for a thorough treatment cf. the dissertation of Fischer [**?**]). This policy, called the *replication policy*, is executed by all agents in parallel and proceeds in the following way. Each agent samples another agent at random and, if this improves the latency, migrates to this agent's path with a probability proportional to the latency gain.

For scenarios in which detailed information about the environment are rare and the consequences of a strategy change may be hard to assess, the replication policy describes natural behavior: Imitation of successful agents. In this setting, a natural goal is to reach approximate equilibria in the following bicriterial sense. We say that a flow is at δ-ϵ-equilibrium if at most an ϵ-fraction of the flow utilizes paths whose latency exceeds the average latency of their commodity by more than a δ-fraction of the overall average latency. Remarkably, the number of rounds to reach an approximate equilibrium in this sense is independent of the size and the topology of the underlying network and chiefly depends on the approximation parameters and the *elasticity* of the latency functions.

We consider a different setting, in which the flow is controlled by a *finite number of agents* only, each of which is responsible for the entire flow of one commodity. Each agent has a set of admissible paths among which it may distribute its flow. To be able to represent exponentially large collections of paths we assume that these are represented by a DAG connecting the source and the sink of the agent. Each agent aims at balancing its own flow such that the jointly computed allocation will be at Wardrop equilibrium.

In each round each agent can observe the edge flows of its own commodity and the latency values of the paths it uses, but it does not know the latency functions themselves or the other agents' flow demands. Let us remark that agents do not aim at minimizing the overall latency of their flow as in the splittable demand model [**?**], but seek to minimize the maximum latency of their commodity.

5.1 Our results

Unfortunately, the replication policy does not yield a feasible distributed algorithm in this setting directly. Simulating an infinite number of agents each of which chooses one out of the given collection of paths would require maintaining one variable for each path and computing a quadratic number of migration rates between pairs of paths. As the number of paths may be exponential in the size of the network this approach is rendered computationally infeasible.

We present two approaches to circumvent this problem. Our first approach exploits the fact that, for a simplified variant of the replication policy, the updates of the edge flows can be expressed in a way that merely uses the edge flow variables themselves (rather than the path flow variables). Thus, the updates can be computed in polynomial time. The convergence time of this variant is only pseudopolynomial in the latency functions since it depends on the maximum slope of the latency functions.

Since the original replication policy cannot be expressed in this compact way, we consider a second approach to achieve convergence in a polynomial number of communication rounds. Consider a collection of paths for one of the commodities. In a first step, our algorithm samples a polynomial number of paths with probability proportional to their flow. We thus obtain a *randomized path decomposition*. We consider paths in this decomposition with above-average latency. From such paths, a fraction of the flow is removed and reallocated proportionally among all admissible paths. If this is done carefully, oscillations can be avoided, and a potential function argument ensures convergence towards Wardrop equilibria. Thus, we achieve essentially the same convergence rates as in the setting with an infinite number of agents and keep the

computation time of one communication round polynomial. Altogether, we can compute approximate Wardrop equilibria in expected polynomial time.

Let us comment on the inherent weakness of the underlying replication policy. Because agents only imitate each other, they are not able to explore currently unused, possibly very cheap paths. Hence, convergence to the set of Wardrop equilibria can not be guaranteed. To this end, we assume a positive amount of flow on each network edge, which ensures not only that we reach an δ-ϵ-equilibrium, but in fact convergence to a Wardrop equilibrium.

5.2 Related Work

The computation of Wardrop equilibria can be formulated as a min-cost flow problem. For an overview of classical methods for finding a minimum cost multi-commodity flow see, e. g., [**?**] and [**?**]. Nevertheless, no fast algorithms for the multi-commodity case are known in general networks. Thus, most work analyzed confined and related problems.

In [**?**,**?**] an efficient distributed steepest-descent algorithm for solving multi-commodity flow problems with linear latency functions has been presented recently. Several authors (e. g.. [**?**,**?**]) consider dynamic routing from an online-learning perspective. Awerbuch and Kleinberg [**?**] present an algorithm for the online shortest path problem in an end-to-end feedback model. Blum *et al.* [**?**] show that approximate Wardrop equilibria defined in a similar way can be attained if the agents follow no-regret algorithms. Their bounds on the convergence time depend polynomially on the regret bounds and network size and depend pseudopolynomially on the maximum slope of the latency functions.

The problem of load-balancing has also been studied in various discrete settings for networks of parallel links. For the case of identical links, both sequential [**?**] and concurrent distributed algorithms were considered [**?**]. Even-Dar *et al.* [**?**] consider distributed algorithms for load balancing on links with speeds using sampling rules which depend pseudopolynomially on the speed of the links. A variant of the replication policy [**?**] has also been applied in congestion games [**?**].

5.3 Preliminaries and Initial Results

In this chapter we consider arbitrary non-negative, non-decreasing and differentiable latency function. In particular, we do not require semi-convex latency functions. We assume that the set of allowed paths $\mathcal{P}_i$ for commodity i is represented by a directed acyclic graph (DAG) and may assume that the sets $\mathcal{P}_i$ are disjoint and define i_P to be the unique commodity to which path P

belongs. We normalize the demand by $\sum_{i\in[k]}\sum_{P\in\mathcal{P}_i} f_P = \sum_{i\in[k]} d_i = 1$. Furthermore, for $v \in V$ and $i \in [k]$, the total flow of commodity i through vertex v is $f_{v,i} = \sum_{(u,v)\in E} f_{(u,v),i} = \sum_{(v,w)\in E} f_{(v,w),i}$ for $v \notin \{s_i, t_i\}$ and $f_{s_i,i} = f_{t_i,i} = d_i$. Finally, while in previous chapters $L_i(f)$ denoted the unique path latency at equilibrium in commodity i, now $L_i(f) = \sum_{e\in E} \ell_e(f) \cdot (f_{e,i}/d_i)$ denotes the weighted average latency of commodity $i \in [k]$. Note that for unit demand the total latency $C(f) = \sum_{e\in E} \ell_e(f) \cdot f_e$ is also the overall average latency.

Recall that Wardrop equilibria are exactly those allocations that minimize the following potential function introduced in [**?**]:

$$\Phi(f) = \sum_{e\in E} \int_0^{f_e} \ell_e(u)\, du \ .$$

The minimum potential is denoted by $\Phi^* = \min_{f\in\mathcal{F}} \Phi(f)$. Every flow f with $\Phi(f) = \Phi^*$ is then at Wardrop equilibrium. We assume that Φ^* is positive. The case that $\Phi^* = 0$ can be treated by adding virtual offsets to the latency functions. For a detailed treatment see [**?**].

The algorithms presented in this paper will compute approximate equilibria in the following bicriterial sense.

Definition 13 (δ-ϵ-equilibrium)**.** *Consider a flow f of unit demand and let*

$$\mathcal{P}_i^\delta = \{P \in \mathcal{P}_i \mid \ell_P(f) > L_i(f) + \delta\, C(f)\}$$

denote the set of δ-expensive *paths of commodity $i \in [k]$. A flow is at a δ-ϵ-equilibrium if*

$$\sum_{i\in[k]} \sum_{P\in\mathcal{P}_i^\delta} f_P \le \epsilon.$$

This definition of approximate Wardrop equilibria requires that almost all flow utilizes paths with a latency that is close to the average of their own commodity. A similar definition of approximate Nash equilibria is used, e. g., in [**?**].

5.4 Elasticity of Latency Functions

Our algorithms take the steepness of the latency functions into account when deciding how much flow to shift from one path to another. In [**?**] it was shown that the critical parameter in this setting is not the slope but the elasticity.

Definition 14 (Elasticity)**.** *For any positive differentiable function $\ell : \mathbb{R}_{\ge 0} \to \mathbb{R}_{\ge 0}$, the* elasticity *of ℓ at x is $r(x) = \frac{x\cdot\ell'(x)}{\ell(x)}$.*

In other words, the elasticity of a function is bounded from above by r if the (absolute) slope at any point is at most by a factor of r larger than the slope of the line connecting the origin and the point $(x, \ell(x))$. Note that a polynomial with positive coefficients and degree r has elasticity at most r, hence, elasticity can be considered as a generalization of the degree of such a polynomial. The function $a \cdot \exp(\lambda x)$, $x \in [0,1]$ has maximum elasticity λ.

5.5 Implicit Path Decomposition

Wardrop equilibria are defined with respect to path flows. Our algorithms, however, will make use only of the edge flow vectors, which do not determine a vector of path flows uniquely. However, in a DAG, an edge flow vector $(f_e)_{e \in E}$ induces a natural vector of path flows by starting with the flow injected at the source, and splitting the flow at each vertex v such that the set of paths containing the outgoing edge e receives a flow proportional to f_e. Since the decomposition for one commodity $i \in [k]$ is independent of the flow of other commodities, we can omit the index i for simplicity.

Definition 15. *Consider any edge flow vector $(f_e)_{e \in E}$ (for some commodity i). For any path $P = (v_1, \ldots, v_l)$ let*

$$\tilde{f}_P = f_{v_1} \cdot \prod_{j=1}^{l-1} \frac{f_{(v_j, v_{j+1})}}{f_{v_j}} \ .$$

Whereas the path flow vector $(\tilde{f}_P)_{P \in \mathcal{P}}$ may contain an exponential number of components, the definition of $\tilde{f}$ allows us to compute sums of path flows that contain a common subpath easily and without summing up an exponential number of terms. This is made precise by the following results that also show that the vector $\tilde{f} = (\tilde{f}_P)_{P \in \mathcal{P}}$ is a valid path decomposition of $f = (f_e)_{e \in E}$.

Lemma 30. *Let P denote a path (not necessarily contained in $\mathcal{P}$). Then,*

$$\sum_{Q \supseteq P, Q \in \mathcal{P}} \tilde{f}_Q = \tilde{f}_P \ .$$

Proof. Let $P = (v_1, \ldots, v_k)$. If $v_1 = s$ and $v_k = t$, for every $Q \in \mathcal{P}$ with $Q \supseteq P$ we have $Q = P$ and the statement is trivial. The proof is by reverse induction on the length of P. For $|P| = n$, we know that for any $Q \supseteq P$, $Q = P$ and the statement is trivial again. Now assume that the statement holds for all paths of length k and consider some path $P = (v_1, \ldots, v_k = w)$ with either $v_1 \neq s$ or $v_k \neq t$. There are two (overlapping) cases:

1. $v_k \neq t$. Then,

$$\begin{aligned}\sum_{Q \supseteq P, Q \in \mathcal{P}} \tilde{f}_Q &= \sum_{u \in \mathrm{Succ}(v_k)} \sum_{Q \supseteq P \cdot (v_k,u), Q \in \mathcal{P}} \tilde{f}_Q \\ &= \sum_{u \in \mathrm{Succ}(v_k)} \tilde{f}_{P \cdot (v_k,u)} \\ &= \sum_{u \in \mathrm{Succ}(v_k)} \tilde{f}_P \cdot \frac{f_{(v_k,u)}}{f_{v_k}} \\ &= \tilde{f}_P \ .\end{aligned}$$

   The second equality is the induction hypothesis and the third is the definition of $\tilde{f}$.

2. $v_1 \neq s$. Analogous.

□

As a corollary, we see that the flow decomposition $(\tilde{f}_P)_{P \in \mathcal{P}}$ is actually compatible with the original path flows $(f_e)_{e \in E}$. In addition, for every $P \in \mathcal{P}$, $\tilde{f}_P$ is obviously non-negative.

Corollary 31. *For any $e \in E$,*

$$f_e = \sum_{P \ni e} \tilde{f}_P \ .$$

Proof. Consider an edge $e = (v, w)$. Then,

$$f_e = f_v \cdot \frac{f_e}{f_v} = \tilde{f}_e = \sum_{P \ni e, P \in \mathcal{P}} \tilde{f}_P \ ,$$

where the second inequality is the definition of $\tilde{f}_e$ and last equality is due to Lemma 30. □

5.6 Distributed Computation Model

Our algorithms operate in the following setting. Agents operate in a synchronous, round-based fashion. We assume that there is a billboard via which the agents are able to share information. On this billboard each agent can observe the edge flows of its own commodity and the latency values of the paths it uses. Agents know an upper bound r on the elasticity of the latency functions, but they do not know the latency functions themselves. However, it is easily possible to extend our algorithm such that it does not rely on the knowledge of a bound on the elasticity.

In every round an agent can update the edge flows of its own commodity on the billboard. These updates become visible to all agents only in the next

round. All agents execute the same algorithm in parallel. Therefore, in the descriptions of our algorithms, we may omit the index for the commodity, i. e., f_e refers to the flow $f_{e,i}$ of commodity i on edge e.

Let us remark that the billboard is a purely theoretical construction. It may model systems where this information is collected centrally and polled by or broadcast to the agents at intervals, but may also model scenarios in which agents concurrently perform measurements over finite intervals of time in order to obtain the necessary latency information.

5.7 A Pseudopolynomial Algorithm

Our first approach works by simulating the replication policy presented in [**?**]. We will see that this can be done in polynomial time although this policy operates on an exponential number of paths.

5.7.1 The Replication Policy

Let us start by introducing the replication policy formally. We consider an infinite set of agents each of which controls an infinitesimal amount of flow which it assigns to a path. In each round agents may migrate their flow from the current path to another one. Consider an agent in commodity $i \in [k]$ currently using path $P \in \mathcal{P}_i$. Whenever activated it performs two steps.

1. *Sampling.* Sample another path Q where the probability to sample any path Q' equals $f_{Q'}/d_i$.

2. *Migration.* There are two cases:

   (a) $\ell_Q \geq \ell_P$. In this case the agent stays with its old path.

   (b) $\ell_Q < \ell_P$. The agent migrates to the sampled path Q with probability $\lambda \cdot (\ell_P - \ell_Q)$ for some constant $\lambda > 0$ to be determined later.

Altogether, we can characterize our policy by specifying the rate of agents migrating from one path $P \in \mathcal{P}_i$ to another path $Q \in \mathcal{P}_i$ with $\ell_Q(f) < \ell_P(f)$ within one round. This rate can be obtained by multiplying the probabilities specified in steps (1) and (2) with the volume of agents using path P. For this rate we obtain

$$\rho_{PQ} = \lambda \cdot f_P \cdot \frac{f_Q}{d_i} \cdot (\ell_P - \ell_Q)$$

if $\ell_Q < \ell_P$ and $\rho_{PQ} = 0$ otherwise. Thus, we can compute a sequence of flow vectors $(f_P(t))_{P\in\mathcal{P}}$ generated by this policy by summing over all paths Q:

$$\begin{aligned} f_P(t+1) &= f_P(t) + \sum_{Q\in\mathcal{P}_i} \rho_{QP} - \sum_{Q\in\mathcal{P}_i} \rho_{PQ} \\ &= f_P(t) + \lambda\, f_P \sum_{Q\in\mathcal{P}_i} \frac{f_Q}{d_i}(\ell_Q - \ell_P) \\ &= f_P(t) + \lambda\, f_P\,(L_i - \ell_P)\ . \end{aligned} \tag{5.1}$$

5.7.2 Convergence Towards Equilibria

For the time being assume that agents are migrating in a continuous fashion as described by the above rules. Then an infinitesimal amount of flow dx migrating from a path P to another path Q improving its latency from ℓ_P to ℓ_Q causes the potential Φ to reduce by $(\ell_P - \ell_Q)\,dx$. Since we only accept migrations that improve the latency, this implies that the potential always decreases which in turn implies convergence towards a Wardrop equilibrium if all paths are used in the initial flow. However, in our concurrent round-based model, flow is not shifted continuously, but in finite chunks. Thus, if these chunks are chosen too large, overshooting and oscillation effects may occur. This issue can be resolved by choosing the migration rate in step 2(b) of the replication policy carefully. In **[?]** it was shown that if we choose $\lambda = \Theta(1/\ell'_{\max})$ small enough with

$$\ell'_{\max} = \max_{P\in\mathcal{P}} \max_{f\in\mathcal{F}} \sum_{e\in P} \ell'_e(f)\ ,$$

convergence towards Wardrop equilibria can be guaranteed (provided that initially all paths have non-zero flow and hence positive sampling probability). We may assume that $\ell'_{\max} > 0$ since otherwise all latency functions are constant, and our problem can be solved trivially by assigning the entire flow to the path with lowest latency.

Theorem 32 (**[?, ?, ?]**). *If $\lambda = \Theta(1/\ell'_{\max})$ sufficiently small, the replication policy given by Equation (5.1) with initial flow $f(0) = f^0$ converges towards a Wardrop equilibrium if $f^0_P > 0$ for all $P \in \mathcal{P}$. Furthermore, the number of rounds in which the flow is not at a δ-ϵ-equilibrium is*

$$\mathcal{O}\left(\frac{1}{\epsilon^2\,\delta^2} \cdot \frac{\ell'_{\max}}{\ell_{\min}} \cdot \log\left(\frac{\Phi(f^0)}{\Phi^*}\right)\right)$$

where $\ell_{\min}$ denotes a lower bound on the latency on any edge.

One may observe that the ratio between maximum slope and minimum latency used in this theorem depends on the scale by which we measure flow. This scale, however, is fixed since we have normalized the total flow demand to be $d = 1$.

5.7.3 Simulating the Replication Policy

By a naive application of Theorem 32 we can compute a sequence of flow vectors $(f(t))_{t\geq 0}$ according to Equation (5.1) to obtain approximate Wardrop equilibria. However, this approach is rendered computationally intractable by the fact that there may be an exponential number of variables f_P.

In the following, we describe an algorithm that computes the iterative change rates of the edge flows according to the implicit flow decomposition $\tilde{f}$ described in the preceding section. To that end, we show that the change rates of the edge flows f_e can be expressed solely in terms of edge flows and edge latencies (i. e., without explicit reference to the f_P variables). It suffices to know the weighted average latencies of all paths containing e defined as

$$L_e = \sum_{P \ni e} \frac{f_P}{f_e} \cdot \ell_P \ . \tag{5.2}$$

Recall that we have fixed a commodity here, so we may drop the index i.

Lemma 33. *Consider an edge flow vector $(f_e(t))_{e\in E}$ and its path decomposition $\tilde{f}(t)$, and let $\tilde{f}(t+1)$ denote the flow generated by the replication policy in Equation (5.1) from $\tilde{f}(t)$. Finally, let $f_e(t+1) = \sum_{P\ni e} \tilde{f}_P(t+1)$. Then,*

$$f_e(t+1) = f_e(t) + \lambda \cdot f_e(t) \cdot (C - L_e) \ .$$

Proof. Let $f = f(t)$ and $f' = f(t+1)$. By definition of f_e,

$$\begin{aligned} f'_e - f_e &= \sum_{P\ni e} (f'_P - f_P) \\ &= \lambda \cdot \sum_{P\ni e} f_P \cdot (C - \ell_P) \\ &= \lambda \cdot f_e \cdot \left(C - \frac{\sum_{P\ni e} f_P \ell_P}{f_e} \right) \ , \end{aligned}$$

where the last term equals L_e. □

In order to obtain the value of L_e, we implicitly compute the path decomposition $\tilde{f}$, i. e., for every edge e' we compute the flow caused by paths containing e on edge e'. This is done by Algorithm SimulatedReplication $e \in E$. Since there are m edges, each iteration can be performed in time $\mathcal{O}\left(m^2\right)$.

Corollary 34. *The sequence of flow vectors computed by Algorithm* SimulatedReplication *converges towards the set of Wardrop equilibria. Furthermore, the number of rounds in which the flow is not at a δ-ϵ-equilibrium with respect to $\tilde{f}$ is bounded by*

$$\mathcal{O}\left(\frac{1}{\epsilon^2 \delta^2} \cdot \frac{\ell'_{\max}}{\ell_{\min}} \cdot \log\left(\frac{\Phi(f^0)}{\Phi^*} \right) \right) \ ,$$

where f^0 is the initial flow vector. Each iteration takes time $\mathcal{O}\left(m^2\right)$.

Algorithm 2 SIMULATEDREPLICATION() (executed by all commodities in parallel; $(f_e)_{e\in E}$ denotes the edge flows vector of commodity i)

1: **for** all edges $e \in E$ **do**
2: sort all edges (v, w) in the subgraph reachable from e topologically
3: compute total flow of all paths containing e and (v, w): $\tilde{f}_e^{(v,w)} = \sum_{(u,v)\in E} \tilde{f}_e^{(u,v)} \cdot \frac{f_{(v,w)}}{f_v}$
4: reverse all edges and repeat steps 2 and 3 for edges between e and s
5: compute $L_e = \sum_{e'} \frac{\tilde{f}_e^{e'}}{f_e} \ell_{e'}$
6: $f'_e \leftarrow f_e + \lambda \cdot f_e \cdot (C - L_e)$ with $\lambda = 1/\ell'_{\max}$
7: **end for**
8: replace $(f_e)_{e\in E}$ on the billboard with $(f'_e)_{e\in E}$

Proof. First note that for any edge e' the value of $\tilde{f}_e^{e'}$ computed in line 3 of the algorithm equals the volume of all paths containing e and e' with respect to our implicit decomposition $\tilde{f}$, i. e., $\tilde{f}_e^{e'} = \sum_{P\supseteq\{e,e'\}} \tilde{f}_P$. Thus, the value L_e computed in line 5 equals the definition of L_e in Equation (5.2). Hence, Lemma 33 implies that the edge flow vector computed by our algorithm after one round equals the edge flow vector obtained by applying the replication policy given by Equation (5.1) to the path decomposition $(\tilde{f})_{P\in\mathcal{P}}$. Combining this with the upper bounds on the convergence time given in [**?**, **?**], the claim follows. □

5.8 The Polynomial Time Algorithm

The migration probability specified for step 2(b) of the replication policy can get very small since the latency difference $\ell_P - \ell_Q$ may become small in relation to $\ell'_{\max}$ if $\lambda = 1/\ell'_{\max}$ is chosen constant. This causes the algorithm to achieve only a pseudopolynomial convergence time depending on the maximum slope of the latency functions. In this section we present an approach that avoids this dependence.

To this end, we choose the amount of flow removed from a path proportional to its *relative* deviation $(\ell_P - L_{i_P})/\ell_P$ from the average and the reciprocal of the elasticity r to obtain a polynomial number of communication rounds. Whereas in the preceding section the amount of flow removed or added to a path within one round could be expressed in a nice closed form as $\lambda \cdot f_P \cdot (C - \ell_P)$ (Equation (5.1)), this is now no longer possible.

To compute flow updates in polynomial time we use a randomized flow decomposition. First, we sample a path at random according to the implicit path decomposition $\tilde{f}$, i. e., the probability to sample path P is $\tilde{f}_P/d_{i_P}$. Since the length of a path is bounded by n, this is possible in time $n \log n$ by representing adjacent vertices and their flows in a binary tree. Now, the path is assigned a

certain flow volume f_P. For the time being, assume that we assign the entire bottleneck flow to P. Then, if P has latency above L_{i_P}, we remove a portion of

$$x = \Theta\left(f_P \cdot \frac{\ell_P - L_{i_P}}{r\,\ell_P}\right)$$

of its flow and distribute it proportionally among all admissible paths, i. e., after removing a flow of x from path P, the flow on every edge $e \in E$ is increased by $(f_{e,i}/d_i) \cdot x$. Thus, the computed flow remains feasible.

Why does this process decrease the potential quickly? As long as we are not at a δ-ϵ-equilibrium, the probability of sampling a δ-expensive path is at least ϵ. In this case, the latency gain and thus the potential gain *per flow unit* will be large and proportional to $\tilde{f}_P$. If we sample only a single path, we may in fact assign the entire bottleneck flow to it. We can lower bound the probability that this bottleneck flow is not too small (Lemma 40). To increase the potential gain further, we repeat this process T times. Doing this, we can no longer assign the entire bottleneck flow to a path since it may happen that an edge is sampled several times. Consider an edge e. If this edge is sampled k times, we may consume at most f_e/k of its flow in every round. If $f_e = \Theta(1)$, we will have $k = \Theta(T)$, so in this case we limit the amount of flow consumed in one round to $f_e/k = \Theta(1/T)$. For edges with less flow, however, we may consume more than f_e/k per round, since these edges are sampled less often. It turns out that we can increase the potential gain by a factor of $\Omega(m)$ if we choose $T = \Theta(m \log m)$ and set the flow assigned to a sampled path to an $\Theta(1/\log m)$ fraction of the bottleneck flow. More precisely, let

$$\Delta_e = \min\left\{\frac{1}{7\,m\,\log m}, \frac{f_e}{7\,\log m}\right\} .$$

We start with an empty decomposition. In a round in which path P is sampled we increase f_P by Δ_{e^*}, where e^* is a bottleneck edge in P. We say that an edge is *alive* if the overall flow assigned to paths containing e is at most $f_e - \Delta_e$ (i. e.there is still a flow of Δ_e remaining, so it can safely be sampled one more time without having our decomposition exceeding the flow of e). Our algorithm terminates as soon as there are any edges that are not alive. The final algorithm RandomizedBalancing is described in Algorithm 3.

Under the assumption that the latency functions are constant, we can thus show that the potential decreases in every round by a factor that only depends on ϵ and δ, and the elasticity r (Lemma 44). We furthermore show that due to our careful migration rate the potential gain with respect to the true latency functions is still at least half of the potential gain with respect to constant latencies (Lemma 42). Finally, we show that the expected potential gain implies a bound on the time to reach a minimum potential (Lemma 46). Altogether, this yields the following upper bound for our algorithm.

Theorem 35. *Assume that for the initial flow vector f_0 it holds that $f_e > 0$ for all $e \in E$. Then, the sequence of flow vectors computed by Algorithm* RandomizedBalancing *converges towards the set of Wardrop equilibria. Furthermore, the expected number of rounds, in which the flow is not at a δ-ϵ-equilibrium with respect to $\tilde{f}$, is bounded by*

$$\mathcal{O}\left(\frac{r}{\epsilon^3\,\delta^2}\log\left(\frac{\Phi(f_0)}{\Phi^*}\right)\right) \ ,$$

if r is an upper bound on the elasticity of the latency functions. The computation time of each round is bounded by $\mathcal{O}\left(n\log n \cdot m\log m\right)$.

We present the proof after establishing the necessary lemmas.

Algorithm 3 RandomizedBalancing(r) (executed by all commodities in parallel; $(f_e)_{e\in E}$ denotes the edge flows vector of commodity i)

1: $F \leftarrow 0$
2: **for** $T = m\log m$ times **do**
3: sample a path P where $\mathbb{P}[P] = \frac{\tilde{f}_P}{d_i}$
4: let e^* denote the bottleneck edge of P; let $f_P = \Delta_{e^*}$
5: **if** $\ell_P > L_i$ **then**
6: reduce the flow on all edges $e \in P$ by $\Delta f_P = f_P \cdot \frac{\ell_P - L_i}{4\,r\,\ell_P}$
7: $F \leftarrow F + \Delta f_P$
8: **if** for any $e \in P$, e is not alive **then**
9: abort loop and continue in line 13
10: **end if**
11: **end if**
12: **end for**
13: increase the flow on all edges $e \in E$ proportionally by $\frac{f_e}{d_i} \cdot F$

Note that our algorithm can be easily modified for the case that the elasticity of the latency functions is not known to the algorithm in advance. In that case, we can find an upper bound r on the maximum elasticity by using an exponential search technique. We continue doubling the value of r until for the first time it holds that for all edges relevant to the commodity, the elasticity of the latency functions is bounded by r within the interval defined by the old and new flow values.

5.8.1 Useful Inequalities

To establish the necessary lemmas we make use of the following well known facts.

Lemma 36 (Chernoff Bound **[?]**). *Let X be a real valued random variable that is the sum of* 0-1 *random variables. Then*

$$\mathbb{P}\left[X \geq q \cdot \mathbb{E}\left[X\right]\right] \leq 2^{-q \cdot \mathbb{E}[X]}$$

for $q \geq 6$.

Lemma 37 (Markov's Inequality). *Let X be a real valued random variable and $h : \mathbb{R} \to \mathbb{R}$ monotone non-decreasing. If $\mathbb{E}\left[h(X)\right]$ is defined, then*

$$\mathbb{P}\left[X \geq a\right] \leq \frac{\mathbb{E}\left[h(X)\right]}{h(a)} \ .$$

Lemma 38 (Jensen's Inequality). *Let X be real valued random variable and $f : \mathbb{R} \to \mathbb{R}$ a convex function. If $\mathbb{E}\left[X\right]$ and $\mathbb{E}\left[f(X)\right]$ are defined, then*

$$\mathbb{E}\left[f(X)\right] \leq f(\mathbb{E}\left[X\right]) \ .$$

Lemma 39 (Cauchy Schwarz Inequality). *For two vectors $(a_i), (b_i) \in \mathbb{R}^n$*

$$(\sum_{i=1}^{n} a_i \, b_i)^2 \leq (\sum_{i=1}^{n} a_i^2)(\sum_{i=1}^{n} b_i^2) \ .$$

5.8.2 Randomized Decomposition

Our algorithm generates a randomized flow decomposition using a sampling process based on $\tilde{f}$. In this section, we lower bound the probability that the bottleneck flows of the sampled paths are not too small. Furthermore, we show that the total flow volume removed from every edge is at most f_e with high probability.

For a unit flow f, let $\mathbb{P}_{P \sim f}\left[X(P)\right]$ denote the probability that event $X(P)$ occurs if the probability to draw a path P equals f_P.

Lemma 40. *Consider a flow f of unit demand and a set of paths $\mathcal{P}_\epsilon$ with $\sum_{P \in \mathcal{P}_\epsilon} \tilde{f}_P = \epsilon$. Then,*

$$\mathbb{P}_{P \sim \tilde{f}}\left[P \in \mathcal{P}_\epsilon \wedge \min_{e \in P} f_e \geq \frac{\epsilon}{2\,m}\right] \geq \frac{\epsilon}{2} \ .$$

Proof. We consider a scaled flow vector which supports only paths in $\mathcal{P}_\epsilon$.

$$f'_P = \begin{cases} \frac{\tilde{f}_P}{\epsilon} & P \in \mathcal{P}_\epsilon \\ 0 & P \notin \mathcal{P}_\epsilon \ . \end{cases}$$

Observe that the total demand of f' is 1 again, hence

$$\mathbb{P}_{P \sim f'}\left[P = Q\right] = \mathbb{P}_{P \sim \tilde{f}}\left[P = Q \mid P \in \mathcal{P}_\epsilon\right] \ .$$

Now,

$$\begin{aligned} & \mathbb{P}_{P\sim\tilde{f}}\left[P \in \mathcal{P}_\epsilon \wedge \min_{e\in P} f_e \geq \frac{\epsilon}{2\,m}\right] \\ = \;& \mathbb{P}_{P\sim\tilde{f}}\left[P \in \mathcal{P}_\epsilon\right] \cdot \mathbb{P}_{P\sim\tilde{f}}\left[\min_{e\in P} f'_e \geq \frac{1}{2\,m} \mid P \in \mathcal{P}_\epsilon\right] \\ = \;& \epsilon \cdot \mathbb{P}_{P\sim f'}\left[\min_{e\in P} f'_e \geq \frac{1}{2\,m}\right] \;, \end{aligned}$$

where the first equality uses the definition of f' and the second one uses the above observation. It remains to show that

$$\mathbb{P}_{P\sim f'}\left[\min_{e\in P} f'_e \geq \frac{1}{2\,m}\right] \geq 1/2 \;. \tag{5.3}$$

To see this, let $d(x,y)$ denote the number of edges of a shortest path connecting x and y. We can show that $\mathbb{P}\left[e = (v,w) \in P\right] = f_e$ by induction on $d(s,v)$. This holds for $d(s,v) = 0$ by definition of $\tilde{f}$. Now, assume that the statement holds for all edges (u,v) with $d(s,u) = k$ and consider an edge $e = (v,w)$ with $d(s,v) = k+1$.

$$\begin{aligned} \mathbb{P}\left[e \in P\right] &= \mathbb{P}\left[v \in P\right] \cdot \mathbb{P}\left[e \in P \mid v \in P\right] \\ &= \sum_{(u,v)} \mathbb{P}\left[(u,v) \in P\right] \cdot \frac{f_e}{f_v} \\ &= \sum_{(u,v)} f_{(u,v)} \cdot \frac{f_e}{f_v} = f_e \;. \end{aligned}$$

With $E' = \{e \in E \mid f_e \leq 1/(2\,m)\}$,

$$\mathbb{P}\left[P \ni e : e \in E'\right] \leq \sum_{e\in E'} \mathbb{P}\left[e \in P\right] \leq \sum_{e\in E'} f_e \leq \frac{|E'|}{2\,m} \leq \frac{1}{2} \;.$$

Thus, Equation (5.3) holds which completes the proof. □

We now consider a sequence of $T = m \log m$ rounds. Observe that Δ_e is an upper bound on the flow removed from a path containing e by our algorithm, since for the bottleneck edge e^*, $\Delta_{e^*} = \min_{e\in P}\{\Delta_e\}$. The flow on e may decrease to below zero only if it is contained in the sampled path at least f_e/Δ_e times. In the following we show that this is unlikely.

Lemma 41. *With probability $1 - o(1)$, after a sequence of $T = m \log m$ iterations, all edges are still alive.*

Proof. In the proof of Lemma 40 we have seen that the probability to hit edge e in one round equals f_e. Let the random variable X denote the number of hits in T rounds. We have $\mathbb{E}\left[X\right] = T\,f_e$. An edge is alive if $X \leq f_e/\Delta_e - 1$. There are two cases:

1. $f_e < \frac{1}{m}$ implying $\Delta_e = f_e/(7 \log m)$. Then,

$$\begin{aligned}\mathbb{P}\left[X > \frac{f_e}{\Delta_e} - 1\right] &= \mathbb{P}\left[X > \mathbb{E}[X] \cdot \left(\frac{7}{f_e m} - \frac{1}{T f_e}\right)\right] \\ &\leq \mathbb{P}\left[X > \mathbb{E}[X] \cdot \left(\frac{6}{f_e m}\right)\right] \\ &\leq 2^{-\mathbb{E}[X] \cdot \frac{6}{f_e m}} = m^{-6} \ .\end{aligned}$$

   The first inequality is the definition of T and Δ_e and uses our assumption that $f_e \cdot m < 1$, and the second inequality is Chernoff's inequality (Lemma 36).

2. $f_e \geq \frac{1}{m}$ implying $\Delta_e = 1/(7\,T)$. Then, with the same arguments,

$$\begin{aligned}\mathbb{P}\left[X > \frac{f_e}{\Delta_e} - 1\right] &= \mathbb{P}\left[X > \mathbb{E}[X] \cdot \left(\frac{1}{T\,\Delta_e} - \frac{1}{T f_e}\right)\right] \\ &\leq \mathbb{P}\left[X > \mathbb{E}[X] \cdot \left(7 - \frac{1}{\log(m)}\right)\right] \\ &\leq 2^{-6 \cdot \mathbb{E}[X]} \leq 2^{-6 \cdot \log m} = m^{-6} \ .\end{aligned}$$

   This time the first inequality uses our assumption $f_e \geq 1/m$.

In both cases, the probability that edge e is not alive at the end of a sequence of T iterations is bounded by m^{-6}. Using a union bound, the probability that at least one edge does not survive is at most m^{-5}, and consequently the probability that all edges survive the sequence is at least $1 - m^{-5}$. □

5.8.3 Lower Bounding the Potential Gain

We use a potential function argument to prove convergence. In order to show that our algorithm avoids oscillations, we consider the potential gain achieved within one round. We show that this potential gain is at least half of the potential gain that would occur if latency values were fixed at the beginning of a round. A second lemma shows that, in expectation, the potential decreases by a factor in every round, as long as we are not yet at an approximate equilibrium.

Lemma 42. *Let r denote an upper bound on the elasticity of the latency functions. For a flow vector f consider a flow vector f' generated by Algorithm* RandomizedBalancing(r) *(Algorithm 3) with positive probability. For any $P \in \mathcal{P}$ let Δf_P denote the amount of flow removed from path P. Then,*

$$\Phi(f) - \Phi(f') \geq \frac{1}{2} \cdot \sum_{P \in \mathcal{P}} (\ell_P(f) - L_{i_P}) \cdot \Delta f_P \ .$$

To prove the lemma, we need the following fact about functions with bounded elasticity:

Fact 43 ([?]). *If the elasticity of a function ℓ is bounded from above by r, then for $0 \leq \delta \leq 1/(2r)$, it holds that $\ell((1+\delta)\cdot x) \leq (1+2r\delta)\cdot \ell(x)$.*

Proof (of Lemma 42). Throughout this proof, whenever we write ℓ, L_i and C without an argument, we refer to $\ell(f)$, $L_i(f)$ and $C(f)$. Let $V_P = (\ell_P - L_{i_P})\,\Delta f_P$ denote the *virtual potential gain* of any path P with $\ell_P \geq L_i$. It can easily be checked (see [?]) that

$$\Phi(f) - \Phi(f') = \sum_{P\in\mathcal{P}} V_P - \sum_{e\in E} \int_{f_e}^{f'_e} (\ell_e(u) - \ell_e)\,du \ .$$

The true potential gain $\Phi(f) - \Phi(f')$ would be achieved if the latency values did not change due to the changing flow. The terms in the second sum can be understood as *error terms* that reduce the potential gain to account for this idealistic assumption. We show that the error terms are at most half of the virtual potential gain. To that end, we attribute

$$V_P^e = V_P \cdot \frac{f_{e,i_P}\cdot \ell_e}{2\cdot d_{i_P}\cdot C} = (\ell_P - L_{i_P})\cdot \Delta f_P \cdot \frac{f_{e,i_P}\cdot \ell_e}{2\cdot d_{i_P}\cdot C}$$

of the virtual potential gain made by path P to edge e. Note that summing over all edges e, this consumes precisely half of the virtual potential gain of path P, i. e., $\sum_{e\in E} V_P^e = V_P/2$. Thus, by a reordering of the terms,

$$\Phi(f) - \Phi(f') \leq \sum_{P\in\mathcal{P}} \frac{V_P}{2} + \sum_{e\in E} \left(\sum_{P\in\mathcal{P}} V_P^e - \int_{f_e}^{f'_e} (\ell_e(u) - \ell_e)\,du \right) \ .$$

Hence, to prove the theorem it suffices to show that for any edge $e \in E$,

$$\int_{f_e}^{f'_e} (\ell_e(u) - \ell_e)\,du \leq \sum_{P\in\mathcal{P}} V_P^e \ . \tag{5.4}$$

Fix an edge $e \in E$ and assume $f'_e > f_e$. Edges with $f'_e < f_e$ can be treated symmetrically. We partition the integral over the interval $[f_e, f'_e]$ into segments of width Δf_P^e, where $\Delta f_P^e = \Delta f_P \cdot f_{e,i_P}/d_{i_P}$ is the amount of flow moved from P to paths containing e. We consider the sequence of paths P_j sampled by our algorithm in ascending order of $(\ell_P - L_{i_P})/\ell_P$. In this sequence a path may occur more than once. Let $f_e^l = f_e + \sum_{j=1}^{l} \Delta f_{P_j}^e$. To conclude the proof, we prove Equation (5.4) by showing that for $l \geq 0$

$$\int_{f_e}^{f_e^l} (\ell_e(u) - \ell_e)\,du \leq \sum_{i=1}^{l} V_{P_i}^e$$

by induction on l. Then, the increase of flow of edge e caused by the first l paths is

$$\begin{aligned} f_e^l - f_e &\leq \sum_{j=1}^{l} \Delta f_{P_j} \cdot \frac{f_{e,i_{P_j}}}{d_{i_{P_j}}} \\ &\leq \sum_{j=1}^{l} f_{P_j} \cdot \frac{\ell_{P_j} - L_{i_{P_j}}}{4\,r \cdot \ell_{P_j}} \cdot \frac{f_{e,i_{P_j}}}{d_{i_{P_j}}} \\ &\leq \frac{1}{4\,r} \cdot \frac{\ell_{P_l} - L_{i_{P_l}}}{\ell_{P_l}} \cdot \sum_{i=1}^{k} f_{e,i} \\ &\leq \frac{1}{4\,r} \cdot \frac{\ell_{P_l} - L_{i_{P_l}}}{\ell_{P_l}} \cdot f_e \ . \end{aligned}$$

The third inequality holds since we can separate the sum into sums over paths from only one commodity. Then the f_P in sum cancel out the d_i's. Now, due to the bounded elasticity of ℓ_e, we can apply Fact 43 to bound the total increase of latency caused by this increase of flow by

$$\Delta \ell_e^l = \ell_e(f_e^l) - \ell_e \leq 2\,r \cdot \ell_e \cdot \frac{f_e^l - f_e}{f_e} \leq \ell_e \cdot \frac{\ell_{P_l} - L_{i_{P_l}}}{2\,\ell_{P_l}} \ .$$

Using the definition of $\Delta f_{P_l}^e$,

$$\Delta f_{P_l}^e \cdot \Delta \ell_e^l \leq \frac{\ell_e\, f_{e,i_{P_l}}}{2\,\ell_{P_l}\, d_{i_{P_l}}} \cdot (\ell_{P_l} - L_{i_{P_l}}) \cdot \Delta f_{P_l} \leq V_{P_l}^e \ .$$

Using the induction hypothesis and the preceding inequality we have

$$\int_{f_e}^{f_e^l} (\ell_e(u) - \ell_e)\, du \leq \int_{f_e}^{f_e^{l-1}} (\ell_e(u) - \ell_e)\, du + \Delta f_{P_l}^e \cdot \Delta \ell_e^l \leq \sum_{i=1}^{l} V_{P_i}^e$$

and the proof is complete. □

Lemma 44. *Assume that f is a flow that is not at δ-ϵ-equilibrium, and let the random variable f' denote a flow generated by our algorithm. Then*

$$\mathbb{E}\left[\Phi(f')\right] \leq \Phi(f) \cdot \left(1 - \Omega\left(\frac{\epsilon^3 \delta^2}{r}\right)\right) \ .$$

Proof. For the time being, assume that the latency functions are constant. Applying Markov's inequality (Lemma 37) with $X = L_i$, $a = 2\,L/\varepsilon$ and $h = id$, the total volume of flow in commodities with $L_i > 2 \cdot C/\epsilon$ is at most $\epsilon/2$. We consider only commodities with $L_i \leq 2 \cdot C/\epsilon$. In total, at least a flow volume of ϵ utilizes δ-expensive paths, and there is still at least a volume of $\epsilon/2$ left in

the commodities we consider. Consider such a commodity $i \in [k]$, and denote the flow volume using δ-expensive paths in this commodity by ϵ_i.

Consider any iteration satisfying the precondition that all edges are alive. Let P denote the path sampled by the algorithm. Consider the event that $\ell_P \geq L_i + \delta C$ and the minimum edge flow along P is at least $\epsilon_i/(2m)$. By Lemma 40 the probability of this event is at least $\epsilon_i/(2d_i)$ (we have to scale the flow of this commodity by a factor $1/d_i$ to make it a unit flow). The amount of flow removed from this path by our algorithm is

$$\frac{\epsilon_i}{2m} \cdot \frac{1}{7 \log m} \cdot \frac{\ell_P - L_i}{4 r \ell_P} \geq \frac{\epsilon_i \epsilon \delta}{113 r m \log m}$$

where we have used that $\ell_P \geq L_i + \delta C$ and $L_i \leq 2C/\epsilon$. The latency gain of this path is then at least δC. Since this event happens with probability $\epsilon_i/(2d_i)$ the expected virtual potential gain of such a path is then at least

$$\frac{\epsilon_i^2 \epsilon \delta^2}{226 r d_i m \log m} C \ .$$

By Lemma 41 the probability that in this iteration all edges are alive is $1 - o(1)$, and the expected potential gain computed above is independent of this event. Summing up over all $T = m \log m$ iterations and all commodities, the total expected virtual potential gain of one round is at least

$$(1 - o(1)) \cdot \sum_{i \in [k]} \frac{\epsilon_i^2 \epsilon \delta^2}{226 r d_i} C \geq (1 - o(1)) \cdot \frac{\epsilon^3 \delta^2}{226 r} C \ .$$

For the last inequality we have applied the Cauchy Schwarz Inequality (see Lemma 39) with $a_i = \epsilon_i/\sqrt{d_i}$ and $b_i = \sqrt{d_i}$. This implies the claim since C is an upper bound on Φ and Lemma 42 ensures that the true potential gain with respect to the real latency functions is at least half of the potential gain with respect to the constant latency functions. □

5.8.4 From Expected Potential Gain to Expected Stopping Time

The preceding section has shown that in every round the potential decreases by a factor in expectation. Intuitively, this implies an expected running time that is logarithmic in this factor and the initial values. This intuition is made precise by the following two lemmas. Although it seems likely that similar lemmas have been proven elsewhere before, we are not aware of any formulation that can be used here.

Lemma 45. *Let $X_0, X_1, \ldots$ denote a sequence of non-negative random variables. Assume that for all $i \geq 0$*

$$\mathbb{E}\left[X_i \mid X_{i-1} = x_{i-1}\right] \leq x_{i-1} - 1$$

and let τ denote the first time t such that $X_t = 0$. Then,

$$\mathbb{E}\left[\tau \mid X_0 = x_0\right] \leq x_0 \ .$$

Proof. The proof is by induction on x_0. Let

$$T(s) = \mathbb{E}\left[\tau \mid X_0 = s\right] \ .$$

Clearly, $T(0) = 0$. For $i \in [x_0]$ let $p(i)$ denote the probability that $x_0 - X_1 = i$. By our assumption and definition of $p(i)$,

$$1 \leq x_0 - \mathbb{E}\left[X_1 \mid X_0 = x_0\right] = \sum_{i=0}^{x_0} p(i) \cdot i \ .$$

By definition of $T(j)$,

$$\begin{aligned} T(j) &= 1 + \sum_{i=0}^{j} p(i) \cdot T(j-i) \\ &\leq 1 + p(0) \cdot T(j) + \sum_{i=1}^{j} p(i) \cdot (j-i) \\ &= 1 + p(0) \cdot T(j) + j \cdot (1 - p(0)) - \sum_{i=1}^{j} p(i) \cdot i \\ &\leq p(0) \cdot T(j) + j \cdot (1 - p(0)) \end{aligned}$$

where the first inequality uses the induction hypothesis for $1 \leq i < j$. Hence, $T(j) \leq j$, implying our claim. □

Lemma 46. *Let $X_0, X_1, \ldots$ denote a sequence of non-negative random variables. Assume that for all $i \geq 0$*

$$\mathbb{E}\left[X_i \mid X_{i-1} = x_{i-1}\right] \leq x_{i-1} \cdot \alpha$$

for some constant $\alpha \in (0,1)$. Furthermore, fix some constant $x^* \in (0, x_0]$, and let τ be the random variable that describes the smallest t such that $X_t \leq x^*$. Then,

$$\mathbb{E}\left[\tau \mid X_0 = x_0\right] \leq \frac{2}{\log(1/\alpha)} \cdot \log\left(\frac{x_0}{x^*}\right) \ .$$

Proof. In order to apply Lemma 45 we transform our random variable into a new sequence of random variables

$$Y_i = 2 \cdot \frac{\log(X_i) - \log(x^*)}{\log(1/\alpha)} \quad .$$

Let $x_i = 2^{y_i \cdot \log(1/\alpha)/2 + \log x^*}$. Then,

$$\begin{aligned}
\mathbb{E}\left[Y_i \mid Y_{i-1} = y_{i-1}\right] &= \mathbb{E}\left[2 \cdot \frac{\log(X_i) - \log x^*}{\log(1/\alpha)} \mid X_{i-1} = x_{i-1}\right] \\
&= 2 \cdot \frac{\mathbb{E}\left[\log(X_i) \mid X_{i-1} = x_{i-1}\right] - \log(x^*)}{\log(1/\alpha)} \\
&\leq 2 \cdot \frac{\log\left(\mathbb{E}\left[X_i \mid X_{i-1} = x_{i-1}\right]\right) - \log(x^*)}{\log(1/\alpha)} \\
&\leq 2 \cdot \frac{\log(x_{i-1} \cdot \alpha) - \log(x^*)}{\log(1/\alpha)} \\
&= y_{i-1} - 2
\end{aligned}$$

where the first inequality is Jensen's inequality (Lemma 38) and the second is our assumption on the sequence X_i. Observe that $X_i = x^*$ if and only if $Y_i = 0$ and $\lfloor Y_i + 1 \rfloor = 0$ implies that $X_i \leq x^*$. Now,

$$\begin{aligned}
\mathbb{E}\left[\lfloor Y_i \rfloor \mid \lfloor Y_{i-1} \rfloor = y_{i-1}\right] &\leq \mathbb{E}\left[Y_i \mid \lfloor Y_{i-1} \rfloor = y_{i-1}\right] \\
&\leq \max_{z:\lfloor y_{i-1} \rfloor = z} \mathbb{E}\left[Y_i \mid Y_{i-1} = z\right] \\
&\leq \max_{z:\lfloor y_{i-1} \rfloor = z} z - 2 \\
&\leq y_{i-1} - 1
\end{aligned}$$

implying that the sequence $\lfloor Y_i \rfloor$ satisfies the conditions of Lemma 45. Let $\tilde{\tau}$ denote the smallest i such that $\lfloor Y_i + 1 \rfloor = 0$, and observe that $X_0 = x_0$ implies $\lfloor Y_0 + 1 \rfloor = \lfloor 2 \log(x_0/x^*)/\log(1/\alpha) + 1 \rfloor$. Hence,

$$\begin{aligned}
\mathbb{E}\left[\tau \mid X_0 = x_0\right] &\leq \mathbb{E}\left[\tilde{\tau} \mid X_0 = x_0\right] \\
&\leq \mathbb{E}\left[\tilde{\tau} \mid \lfloor Y_0 + 1 \rfloor = \left\lfloor 2 \cdot \frac{\log(x_0/x^*)}{\log(1/\alpha)} + 1 \right\rfloor\right] \\
&\leq \left\lfloor 2 \cdot \frac{\log(x_0/x^*)}{\log(1/\alpha)} + 1 \right\rfloor \quad ,
\end{aligned}$$

our desired bound. □

5.8.5 Convergence Time

Finally, we can prove our main result.

Proof of Theorem 35. Again, convergence follows from Lemma 42 as in [**?**]. To obtain a bound on the convergence time, let $f_0, f_1, \ldots$ denote a sequence of flow

$$\mathbb{E}\left[\Phi(f_{t+1}) \mid \Phi(f_t) = \phi\right] \leq \phi \cdot \left(1 - \Omega\left(\frac{\epsilon^3\,\delta^2}{r}\right)\right) \quad .$$

Thus, the sequence $(\Phi(f_t))_{t\geq 0}$ satisfies the conditions of Lemma 46 and the expected time until $\Phi(f_t)$ reaches its minimum Φ^* implying that f_t is a δ-ϵ-equilibrium is

$$\frac{2}{\log\left(\left(1 - \Omega\left(\frac{\epsilon^3\,\delta^2}{r}\right)\right)^{-1}\right)} \log\left(\frac{\Phi(f_0)}{\Phi^*}\right) = \mathcal{O}\left(\frac{r}{\epsilon^3\,\delta^2} \log\left(\frac{\Phi(f_0)}{\Phi^*}\right)\right) \quad ,$$

our desired bound.

One path can be sampled in time $\mathcal{O}(n \log n)$, the bottleneck edge can be found in time $\mathcal{O}(n)$, and the flow update can be computed in time $\mathcal{O}(n)$. Altogether, at most $T = m \log m$ iterations have to be computed. Finally, the removed flow can be reinserted in time $\mathcal{O}(m)$. □

Chapter 6

Concluding Thoughts

In this thesis we have studied a variety of problems in Wardrop's traffic model that revolve around the inefficiency of equilibria and their paradoxical behavior. We analyzed two different means to reduce the price of anarchy, studied the stability and sensitivity of equilibria, and designed a distributed algorithm to compute approximate equilibria. Throughout our research, structural simple networks like parallel link networks or Braess's original network served as benchmark networks providing first insight, guiding our research, and finally extending our understanding of Wardrop equilibria.

6.1 Reducing the Price of Anarchy

In the first part of this thesis, we draw connections between several established concepts of reducing the price of anarchy. Generalizing marginal cost pricing, we first investigated optimal taxes if only a subnetwork can be taxed. While we provided an efficient algorithm to compute taxes minimizing the network wide performance in parallel link networks with linear latency functions, this problem turned out to be NP-hard in arbitrary networks with multiple commodities. Our positive results may seem quite restricted, however, observe that in contrast the optimal leader strategy in Stackelberg routing is NP-hard to compute for the same class of simple networks [**?**].

Our results lead to a set of intriguing questions. The prime goal is an approximation algorithm for multi-commodity networks. Unfortunately, we are not aware of any non-trivial approximation algorithms incurring an approximation ratio less than the price of anarchy. For single-commodity networks an interesting question is whether one can close the "complexity gap" or not. Can one extend the NP-hardness results to more general classes of networks? Is there a polynomial time algorithm for parallel link networks with polynomial latency functions? Technically, the reasons that our proof fails for polynomial

latency functions are that the total latency does not remain convex and the latency threshold $L(d)$ does not remain linear.

Also, the question of how to optimally set taxes for a finite set of edges or even for a single edge remains open. Algorithms for this problem could constitute useful modules for approximation algorithms for the general case. Following this line of research, first (albeit negative) results have been obtained recently [**?**].

Towards another direction, one might want to abandon the strong homogeneity assumption. Even for heterogeneous network users that minimize their own tax versus total latency, optimal taxes can be computed efficiently for arbitrary multi-commodity networks [**?**, **?**, **?**]. What can be said about the complexity of finding optimal taxes for a given subset of edges in this case?

As outlined we tackled the problem of computing optimal taxes for a given subset of edges. But in the first place, one needs to decide, which set of links to tax, given a choice. In light of the large number of possible sets and complex interactions between taxes on different edges, this problem seems intriguing. Considering the related problem of computing optimal taxes, such that an additional tax-dependent objective function, e.g., the number of taxed edges is optimized, positive results have been obtained recently [**?**].

To dispense with direct taxing, we proposed a novel approach to reduce the price of anarchy by routing additional flow. Routing so-dubbed auxiliary flow δ raises the edge latencies for the selfish flow and thus can be considered as charging the selfish agents a non-refundable traffic dependent tax of $\ell_e(f_e + \delta_e) - \ell_e(f_e)$. There is also a strong connection to Stackelberg routing when the auxiliary flow is seen as the flow of a leader, who centrally routes its fraction of flow to improve the global performance. However, the critical difference is that the leader's latency does account for the total latency, while the auxiliary flow's latency does not. This is the reason why our results on auxiliary flow contrast those obtained for Stackelberg routing. In particular, the minimal amount of leader flow inducing the optimal flow can be computed in polynomial time in arbitrary multi-commodity networks [**?**]. For auxiliary flow this value is NP-hard to approximate even within subexponential factors. Alternatively, auxiliary flow can be interpreted as a separate altruistic commodity that tries to pilot the routing of the selfish players to a globally desirable state. In related work it was shown that when all agents are assumed to be partly altruistic, the price of anarchy can be bounded by a constant in parallel link networks [**?**]. Yet, using (altruistic) auxiliary flow does not improve the network performance in this class of networks.

Our hardness result for optimal auxiliary flow is tight for networks with linear latency functions, because the trivial algorithm, i. e., routing no spam, yields a $4/3$ approximation [**?**]. For more general sets of latency functions, e. g.,

for polynomials of bounded degree, the presented instances do not directly yield tight inapproximability results. The question if the trivial algorithm is an optimal algorithm for nonlinear latency function is still open.

Reducing the price of anarchy by simply routing an additional amount of flow seems an appealing approach. But yet, our results show that important problems related to this approach are rendered computationally infeasible. We hope that our hardness results inherently rely on the fact that we have confined ourself to extremal auxiliary flow. This would still allow efficient computation of flow that improves total latency by an arbitrary amount. Such algorithms as well as the design of non-trivial approximation algorithms for the problems considered here would certainly be of great interest.

6.2 Sensitivity Analysis

In his ground-breaking work, Braess [**?**] observed that adding capacity to a network might improve or deteriorate the total latency depending on the amount of input traffic. In other words, the occurrence of Braess's paradox is demand sensitive. We employ a family of generalized Braess graphs to bring another remarkable facet of Wardrop equilibria into the open: Even the slightest increase in the demand may cause every agent to change its path.

However, our sensitivity analysis leaves open some obvious questions. Given a unit demand flow at Wardrop equilibrium, suppose an edge carrying only an ε-fraction of flow is removed. How does the path latency change after recovering equilibrium? Considering a network with two parallel edges, one gets a lower bound of $\frac{1}{(1-\varepsilon)^p}$. Is this bound tight? Furthermore, we believe that our bound on the increase of the path latency induced by a demand increase of $(1+\varepsilon)^p$ holds not only for polynomials of bounded degree but for latency functions with bounded *elasticity*.

Less specifically, we believe that studying sensitivity of traffic equilibria is a natural and important task that is worthwhile also in related models, e. g., in the presence of heterogeneous agents or in scenarios where finitely many agents can split their non-negligible amount of flow. Further, while most existing literature on sensitivity analysis concentrates on qualitative questions, we are convinced that quantitative studies are equally important and that both kinds of results nicely complement one another. In this spirit, we have shown that the vector of edge flows is not only continuously dependent on the traffic demand [**?**], but in fact Lipschitz-continuous with constant one (Theorem 27).

6.3 Distributed Equilibrium Computation

Braess's original four vertex network demonstrates that naive algorithms for approximating equilibria fail, even if a central authority has access to complete information about the game. But yet, in a dynamic round-based setting Wardrop equilibria can be well approximated in a distributed way.

Our distributed algorithm works by redistributing flow of overloaded paths. To identify such paths we face the subproblem of finding a flow decomposition that assigns much flow to paths with high latency (induced by the current flow). In our algorithm we have used a randomized path decomposition to achieve this goal. It is a natural question whether this randomization can be avoided. In a greedy approach we could use a path decomposition that chooses a path with the largest latency, assigns to it a flow equivalent to the bottleneck flow of this path, and removes it from the network. In fact, a simple example shows that this approach does not necessarily maximize the unbalancedness of the decomposition. However, it has been shown in [**?**] that the problem of finding an unbalanced decomposition can be reduced to a Min-Cost-Flow problem. At the cost of an increased running time this could be used as a module in our algorithm derandomizing it.

In the long run, our algorithm converges towards the set of Wardrop equilibria. A weakness of our notion of approximate equilibria, however, is the fact that the average latency may be arbitrarily far away from the minimum latency. Furthermore, a δ-ϵ-equilibrium, allows some of the commodities to be very out of balance.

There are two alternative, stronger definitions of approximate equilibria. First, one could require all but an ϵ-fraction to deviate from the average of their commodity by at most δL_i rather than δC. Second, one could also consider deviations from the minimum latency rather than from the average latency. It is unclear whether convergence towards approximate equilibria in this sense can be guaranteed in polynomial time. This seems questionable in light of corresponding results in the setting of discrete network congestion games. Therein it has been proven that computing $(1+\varepsilon)$-approximate Nash equilibria is PLS-complete. Hence most likely no polynomial time algorithms for this problem exists.

Finally, it would be desirable to design specialized (not necessarily distributed) algorithms to compute (exact) Wardrop equilibria that improve upon the standard solution via convex programming.

6.4 Dynamic Extensions

Not least we want to comment on an important extension of Wardrop's model. Researchers almost exclusively concentrated on the classical static flow model. This seems a plausible assumption in networks, that are continuously used by the same number of agents. In these situations, there is no need for the introduction of a temporal component. However, in many natural network applications flow travels through a network over time and flow values on edges change over time. Already Beckmann *et al.* [**?**] stated:

> The notion of a static equilibrium of flow in a network may be thought of as somewhat limited... An understanding of dynamic aspects of the traffic really depends on an understanding of demand substitution over time.

Only recently, Koch and Skutella [**?**] were the first to explore the avenue of selfish flows over time. The authors show how flows over time can be thought of as traditional network flows plus a scheduling component. They characterized equilibria and gave first results on the price of anarchy for flows over time. Their work was followed by alternative approaches to incorporate the notion of time in selfish networks [**?**, **?**, **?**]. However, positive results are rare in existing work. We believe that extending Wardrop's traffic model (appropriately) by incorporating a temporal component embodies an important direction for future research.

Lebenslauf

Persönliche Daten

Lars Olbrich
Südstraße 47
52064 Aachen

Geb. am 05. August 1979 in Lünen
deutsch

Qualifikationen

2010	Promotion an der RWTH Aachen
2005	Diplom in Mathematik
2001–2002	Austauschstudium an der Arizona State University Tempe, Arizona, USA
1999–2005	Studium der Mathematik mit Nebenfach Informatik an der Ruprecht-Karls-Universität Heidelberg
1998	Abitur am Städtischen Gymnasium Selm

March 4, 2010