LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

INSTITUT FÜR INFORMATIK
LEHRSTUHL FÜR MOBILE UND VERTEILTE SYSTEME

# BACHELOR THESIS

# Stackelberg routing with fairness considerations

Miriam Fischer

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR INFORMATIK

LEHRSTUHL FÜR MOBILE UND VERTEILTE SYSTEME

# BACHELOR THESIS

# BACHELORARBEIT

# Stackelberg routing with fairness considerations

Miriam Fischer

| | |
|---|---|
| Aufgabenstellerin:<br>(Supervisor) | Prof. Dr. Claudia Linnhoff-Popien, LMU Munich |
| Betreuer:<br>(Advisor) | Dr. Martin Gairing, University of Liverpool<br>Thomy Phan, LMU Munich<br>Robert Müller, LMU Munich<br>Steffen Illium, LMU Munich |
| Abgabetermin:<br>(Date of submission) | 10. August 2020 |

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 10. August 2020

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

*(Unterschrift des Kandidaten)*

I hereby declare that I have written this bachelor's thesis independently and without using any other means than those indicated.

Munich, 10th August 2020

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

*(Signature of the candidate)*

**Abstract**

Selfish routing leads to inefficient outcomes. Stackelberg routing, meaning that a central authority controls a share $\alpha$ of the total flow $r$, is a well-established way to improve the inefficiency of selfish routing. To reduce inefficiency and therefore total costs, the Stackelberg leader routes some players along paths that are individually more expensive for them, but lead to socially cheaper outcomes. Previous research on Stackelberg routing did not take account of the extent to which the Stackelberg leader penalises players by taking control of their routing decisions. This work introduces a new measure $\mathcal{B}(\alpha)$ that reflects the *unfairness* of Stackelberg routing. $\mathcal{B}(\alpha)$ is defined as the ratio of the maximum latency experienced by any controlled players to the minimum latency in the network, given the flow induced by the Stackelberg leader. The ratio reflects how much individually worse off controlled players are compared to if they could route themselves.

We investigate Stackelberg routing regarding its unfairness in single-commodity, parallel-edge networks. We establish upper bounds on the unfairness of the optimal flow and any opt-restricted Stackelberg strategy. We show that the unfairness $\mathcal{B}(\alpha)$ of Scale, LLF, and a wider class of strategies called *monotone opt-restricted strategies* is monotonically increasing with $\alpha$. We introduce a new class of games called *fair Stackelberg games* for which the unfairness of a given Stackelberg strategy is not allowed to exceed an external, given parameter $\beta \geq 1$. We show that there exist values for $\beta$ such that neither LLF nor Scale are fair Stackelberg strategies. Further, we give lower bounds on the Price of Anarchy for any fair Stackelberg strategy that depend not only on the share controlled by the leader, but also the maximum level of unfairness allowed. We further present some examples of fair Stackelberg strategies.

# Acknowledgements

Many, many, many thanks to Dr. Martin Gairing for exceptional academic advice. Further thanks go to my friends, family, and everyone else who supported me these past months.

# Contents


**1. Introduction** **1**

**2. Terminologies in Algorithmic Game Theory** **5**
2.1. Nash equilibria and the Price of Anarchy . . . . . 5
2.2. Selfish routing games . . . . . 7
2.2.1. Improving system performance . . . . . 11

**3. Stackelberg routing and state of the art** **13**
3.1. Stackelberg routing . . . . . 13
3.2. State of the art . . . . . 15

**4. Stackelberg routing with fairness** **17**
4.1. Preliminaries . . . . . 17
4.1.1. Remarks . . . . . 20
4.1.2. Mathematical program . . . . . 21
4.2. Bounding unfairness in standard Stackelberg games . . . . . 22
4.2.1. Unfairness in optimal flows . . . . . 23
4.2.2. Unfairness for any opt-restricted Stackelberg strategy . . . . . 25
4.3. Further analysis for specific opt-restricted strategies . . . . . 27
4.3.1. Tight unfairness bound and monotonicity properties of LLF . . . . 28
4.3.2. Unfairness monotonicity of monotone opt-restricted strategies . . . 31
4.3.3. Comparing unfairness of Scale and LLF . . . . . 33
4.4. Lower bounds on the Price of Anarchy for all fair Stackelberg games . . . 36
4.4.1. Price of Anarchy bounds for linear latencies . . . . . 37
4.4.2. Price of Anarchy bounds for polynomial latencies . . . . . 40
4.5. Fair Stackelberg strategies . . . . . 41
4.5.1. Naive fair Scale . . . . . 42
4.5.2. Flexible fair Scale . . . . . 44
4.5.3. Naive shifted LLF . . . . . 46
4.5.4. Flexible shifted LLF . . . . . 48
4.5.5. LLF on approximate equilibria . . . . . 49

**5. Conclusion and Future Work** **53**

**A. Appendix** **55**
A.1. Mathematical program for general networks . . . . . 55

**Bibliography** **57**

# 1. Introduction

Assume a road traffic network with a large number of networks users (hereafter called players). Each player controls a very small amount of flow which they want to get from source $s$ to sink $t$ as quickly as possible. The time (or latency) of a specific route in the network depends on the total traffic on the route, i.e. the flow of *all* players using this route. When players choose their route in the network, we assume them to behave selfishly and independently from each other. They take the shortest paths available and ignore the effect their routing decision has on other players and the overall traffic and travel time. This setting is called *selfish routing*. It is appealing as it guarantees stable equilibria, i.e. assignments in which no player has incentives to change their route. However, in some cases the total travel time of all players in the network, i.e. the sum of all players individual travel time, could be reduced if some players would refrain from being selfish and use paths that are individually more expensive. This is reflected in the *Price of Anarchy*, the ratio of highest total cost of selfish behaviour to the minimal total cost possible.

We illustrate this scenario with a simple example: Imagine the network seen in Figure 1.1. We assume a total traffic flow of 1, thus the sum of flow of all players on the top and bottom edge are adding up to 1. Each player wants to get their share of flow from $s$ to $t$. The travel time of using the upper edge is 1.8, independent of the traffic using this edge. The travel time of the lower edge equals the total traffic $x$ using this edge. If all players now route their share of flow selfishly, all players will use shortest paths. Easy to see, the shortest path is to use the bottom edge, as $l_2(x) = x < 1.8 = l_1(1-x)$ for all $x$ (as we only allow for $x \leq 1$). As all players behave like this, they will all route their flow along the bottom edge, thus with x=1 *all* players will have costs of 1. The total cost of this setting is therefore $1 * 1 = 1$ (the sum of the costs each share of flow has). If however, some players would sacrify their own selfishness for the greater good and use paths that are not shortest for themselves, the total cost could be reduced: To see this, imagine ten percent of the players to use the upper edge, i.e. the traffic on the upper edge is 0.1 and the traffic on the bottom edge is 0.9. This is the optimum allocation for this instance. The total cost then is $0.1 * 1.8 + 0.9 * 0.9 = 0.975$, thus smaller than if all players behave selfish.

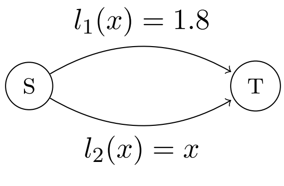



Figure 1.1.: Example of a network instance

One solution to improve the inefficiency of selfish routing is to introduce a central authority, called *Stackelberg leader*, who controls and routes a fraction $\alpha$ of the total flow.

The remaining flow is being routed selfishly by players. This is called a *Stackelberg game*. Routing then can be modelled as a two-stage game. First, the Stackelberg leader routes his share flow along edges. Then, the selfish players route their flow along shortest paths, given the already existing controlled flow on paths. The Stackelberg leader wants to route his flow in such a way that it minimises the combined total cost of controlled and selfish flow. As an example, if the Stackelberg leader controls $\alpha = 0.4$ of the flow, he could decide to route 0.1 along the top edge and 0.3 along the bottom edge. The remaining selfish users have flow 0.6 and given the underlying flow of the Stackelberg leader, they decide to route themselves along the bottom edge. Therefore, the Stackelberg leader is able to implement the optimal assignment.

We see that Stackelberg routing can reduce the inefficiency of selfish routing by routing some players along paths that are individually more expensive than paths used by selfish players. Therefore, the players controlled by the leader can experience individually higher costs than if they could route themselves. Even though Stackelberg routing has already been studied in the literature (see, for example [Rou04; BHS10; Swa07]), previous work did not put any constraints on how much more costs the players controlled by the Stackelberg leader are allowed to experience compared to the self-allocating selfish players. We call these games standard Stackelberg games.

This work introduces the consideration of *fairness* in Stackelberg games. We define the unfairness $\mathcal{B}(\alpha)$ of a given Stackelberg strategy as the ratio of maximal latency experienced by any player to minimal latency in the network for a given flow. The ratio reflects how much worse off a given Stackelberg strategy puts controlled players compared to if these players could route themselves. We analyse the unfairness of standard Stackelberg routing and give upper bounds on the unfairness $\mathcal{B}^{opt}$ of the optimum allocation (Section 4.2.1) and any opt-restricted strategy (Section 4.2.2) in parallel-edge networks. Parallel-edge networks are networks for which a source and a destination are linked with $m$ parallel edges. We analyse the LLF strategy in greater detail and show that the unfairness of LLF monotonically increases with $\alpha$ in Section 4.3.1. We extend this result to a wider class of strategies, called *monotone opt-restricted strategies* in Section 4.3.2.

We further introduce a new type of Stackelberg routing games, called *fair Stackelberg games* that have an additional external parameter $\beta \geq 1$ which restricts the level to which the Stackelberg leader is allowed to route players along paths that have higher latency than the paths used by selfish players. Fair Stackelberg strategies must obey $\mathcal{B}(\alpha) \leq \beta$. Thus, fair Stackelberg routing restricts the leader's options of controlling flow by demanding that the individual latency of any centrally controlled player is at most $\beta$-times worse than the individual latency of the selfish players. We give lower bounds on the Price of Anarchy of any fair Stackelberg strategy which both depend on $\alpha$ and $\beta$ (Section 4.4) and also present some fair Stackelberg strategies (Section 4.5).

A possible application for fair Stackelberg games is traffic routing with autonomous (AVs) and non-autonomous (non-AVs) vehicles. Drivers of non-AVs route and drive on their own, whereas passengers of AVs give control of the routing to a central authority. Drivers of non-AVs want to minimise their individual travel time as they must focus on driving throughout the ride; passengers of autonomous vehicles are willing to be routed on slower routes as they can spend time well in their cars. However, their indulgence is limited; if the extra travel time becomes too high, they prefer to take over the routing and drive manually. The central authority must not only anticipate the routing decisions of the selfish non-AV drivers, but also commit not to penalise AV passengers too badly when routing them, to prevent passengers of non-AVs from taking over and routing selfishly.

To see the effect of fairness demands on Stackelberg routing, we again use the instance presented in Figure 1.1. It is crucially to understand that no matter how the leader routes his share of flow, the selfish users will only use the bottom edge, as the latency on this edge will always be smaller than the latency on the top edge (as the flow on the bottom edge can be at most the total demand, which is 1). Thus, if the Stackelberg leader controls at least $\alpha \geq 0.1$, he can enforce the optimum $o = (0.1, 0.9)$. However, this comes with an unfairness of $\mathcal{B}^{opt} = \frac{l_1(o_1)}{l_2(o_2)} = \frac{1.8}{0.9} = 2$, i.e. the costs experienced by controlled players on the top edge is twice as much as costs of players on the bottom edge. In a fair Stackelberg game with $\beta = 1.8$, it is no longer possible for the Stackelberg leader to enforce the optimum, even if he controls sufficient flow. In fact, in a fair Stackelberg game the leader cannot route *any* amount of flow $\varepsilon$ along the top edge, as this would make the controlled players being routed via this edge at least $1.8/(1-\varepsilon) > 1.8$-times worse off than the other players. Thus, in this particular instance, the Stackelberg leader cannot implement the optimum, he is even forced to induce the Nash equilibrium $n = (0, 1)$ when faced with fairness demands.

The thesis is structured as follows: Chapter 2 gives a brief introduction into terminologies used in Algorithmic Game Theory such as Price of Anarchy, Nash equilibria, and routing games. Readers who are already familiar with these notions can skip this chapter. Chapter 3 gives an introduction into a subarea of routing games, called Stackelberg routing. We further give an overview of the literature in this field which serves as motivation for our own work. This chapter can be skipped if the reader is already familiar with Stackelberg routing. Chapter 4 is the most significant part of this thesis: We introduce our new measure of fairness in Section 4.1. We bound the unfairness of standard Stackelberg games for a variety of strategies and allocations in Section 4.2. We further show monotonicity properties for more defined strategies in Section 4.3. We then investigate fair Stackelberg strategies and bound their Price of Anarchy which now depends on the maximum unfairness allowed in Section 4.4. We present some fair Stackelberg strategies in Section 4.5. Chapter 5 concludes and gives an outlook.

# 2. Terminologies in Algorithmic Game Theory

This chapter gives a short introduction in measures used in Algorithmic Game Theory. Algorithmic Game Theory gives an algorithmic approach to problems studied in Game Theory. [Neu28] is considered as the birth of Game Theory. Since then, Game Theory is studied in Mathematics, Economics, and other sciences to analyse strategic interactions of many, mostly self-interested, players in a market. Algorithmic Game Theory both gives new tools to solve persistent problems in Game Theory as well as new applications for Game Theory. An example for new applications are online ad auctions used by popular search engines. Examples for new tools include algorithms which calculate equilibria for a variety of games as well as the analysis of their runtime. Algorithmic Game Theory covers many classes of games which we do not cover in this work. For an in-depth introduction and overview, we refer to [Nis+08; Rou16]. In the following, we mention important measures in Algorithmic Game Theory that will be of use in our work.

## 2.1. Nash equilibria and the Price of Anarchy

The notion of a *Nash equilibrium* is one of the most important notions in Game Theory. It is a situation in a game such that no player can increase his payoff or decrease his cost by *unilaterally* changing their strategy. That means given all other players stick to their current strategy, no player can improve his utility by switching strategies. Consider the following scenario as an example: two friends want to go out together. They can either go to the museum or to the theatre. The entrance fee at both locations is 3 EUR for each person. However, if they meet at the same place, they get a "2-for-1" discount, so that both of them would only have to pay 2 EUR for the theatre and even 1 EUR for the museum. Unfortunately, the two friends have not agreed where to meet in advance and both their phones' ran out of battery, so both have to choose a location without being able to communicate with the other. The social or total cost of a location combination is simply the sum of both players' individual cost.

This setting can be modelled as a game, presented in Table 2.1. The first player's actions are written in the rows and the second player's actions are written in the columns. Each entry of the table reflects the entrance fee the friends would have to pay for a given setting. For example, if the first friend goes to the museum (first row) and the second friend goes to the theatre (second column), both have to pay 3 EUR.

To see the Nash equilibria of this game, we must look for scenarios in which neither of the two friends could decrease his costs by going to a different location. Is it an equilibrium if the first friend goes to the theatre and the second friend goes to the museum? No, as both friends would only have to pay 1 EUR instead of 3 if the first friend would go to the museum instead of the theatre. Is it an equilibrium if both friends go to the museum? Yes, as if one them would decide to go to the theatre instead of the museum (given the other one goes to the museum), they would both have to pay more. None of them could decrease their cost by unilaterally changing locations.

Applying this reasoning to all four possible cases (or allocations) of the game [1], we see that this game has two pure Nash equilibria. They are "both going to the museum" or "both going to the theatre". Going to different locations is not an equilibrium, as one player (e.g. the one who went to the museum) could reduce his costs by going to the other location (e.g. the theatre) instead, thus meeting his friend and getting the "2-for-1" discount. However, it is crucial to understand that even if (museum,museum) gives lower costs for both friends than (theatre,theatre), both allocations are Nash equilibria. This is due to the fact that one player cannot force the other to go to a specific location, he can only change his own location, but not that of his friend. Given the friend goes to the theatre, it is the best response to go to the theatre as well. We will say that players play *best responses* to each other in a Nash equilibrium. We keep in mind that Nash equilibria correspond to unilateral deviation.

| 1\2 | Museum | Theatre |
|---|---|---|
| Museum | (1,1) | (3,3) |
| Theatre | (3,3) | (2,2) |

Table 2.1.: Cost table of the "friends and locations" game

Nash equilibrium is a very attractive concept as it basically says "let everybody behave on their own, with their own utility in mind", leading to stable outcome. Allocations with socially cheaper outcomes (i.e. with lower total cost), are not per se stable: there, it can be the case that some players are not pleased with what they are assigned and could reduce their individual costs if they could choose their action selfishly. Particularly, this can apply to optimal allocations, i.e. allocations which minimise overall cost. Thus, Nash equilibria sometimes come with sub-optimal, socially expensive outcomes. [KP99] introduced the idea of measuring performance degradation due to selfish, anarchistic behaviour of players. [KP99] introduced the ratio of total cost of the worst Nash equilibrium to total cost of optimal solution as a measure for the inefficiency of selfish behaviour. [Pap01] later named that ratio the *Price of Anarchy*. The smaller the Price of Anarchy, the less inefficient selfish behaviour of players is in respect of total cost.

More formally, for a particular instance (or game) $I$, the Price of Anarchy is defined as

$$PoA(I) = \max_{P \in Nash(I)} \frac{cost(P)}{opt(I)}$$

.

For a class of instances $\mathcal{I}$, the PoA equals

$$PoA(\mathcal{I}) = \max_{I \in \mathcal{I}} PoA(I)$$

Much of past and present research in Algorithmic Game Theory deals with finding upper and lower bounds of the Price of Anarchy for various classes of games. In several classes, the Price of Anarchy is unbounded, i.e. selfish behaviour is arbitrarily bad. The

[1] sophisticated readers might realise that we neglect mixed strategies in this easy example, as they do not add essential knowledge for the model we study

Price of Anarchy can never be smaller than 1, which for example occurs if the global optimum is the unique equilibrium for the instance or class of instances.

In similar manner, the *Price of Stability* (PoS) [Ans+03; SS03] is defined as the ratio between the Nash equilibrium with the lowest social cost to minimal social cost. Thus, the difference between the Price of Anarchy and the Price of Stability is that the former is the ratio of *worst* Nash equilibrium to optimum, and the latter is the ratio of *best* Nash equilibrium to optimum. The Price of Anarchy is an upper bound on the Price of Stability and the two measures coincide if there is a unique Nash equilibrium.

Recall again the game presented in Figure 2.1. The total cost of the Nash equilibrium (museum,museum) is 1+1=2, whereas the total cost of (theatre,theatre) is 2+2=4. We observe that of all four possible location combinations, (museum,museum) is the one with lowest social costs. As the optimum is a Nash equilibrium, the Price of Stability is 1. The Price of Anarchy is 2, as the total cost of both players going to the theatre (=4) is twice as high as the total cost of both going to the museum (=2).

Having introduced equilibria notions and the Price of Anarchy, we shift our focus to a specific class of games which will be the underlying model of this thesis.

## 2.2. Selfish routing games

As mentioned, Algorithmic Game Theory can be applied to a broad variety of classes of games - the game shown in Table 2.1 is one example for a simultaneous move game with two players. In this thesis, we cover a different class of games, called *selfish routing games*. This class of games is a *non-atomic* network game that consists of a network with a large number of players. There is a graph network with vertices and edges and one or multiple source and destination pairs in the network. A total traffic (or demand) $r$ must be routed from the source to the destination. Each player controls a negligible amount of the total traffic which he wants to route from a given start to endpoint in the network. Each path and each edge in the network has a cost (or latency) attached, such that routes in the network come with cost that is dependent on the traffic using this route. The players are assumed to be selfish and non-cooperative, i.e. they route their share of the traffic such that it minimises their individual costs, ignoring the total cost. Non-atomic network routing games were first introduced by [War52]. [BMW56] were first to give a mathematical model of such games.

We now present non-atomic selfish routing games as defined in [RT02; Rou05]. We are given a directed graph $G = (V, E)$ with a finite set of vertices $V$ and a finite set $E$ of directed edges. We are given $k$ source-destination pairs $(s_i, t_i)$ which we call commodities. Each commodity $i \in [1, k]$ has a demand (or traffic rate) $r_i$ of flow that has to be routed from $s_i$ to $t_i$. We call networks with only one source-destination pair single-commodity networks, otherwise they are named multi-commodity networks.

Paths that begin at vertex $u$ and terminate at vertex $v$ are $u - v$ paths. For each commodity $i \in [1, k]$, the set of $s_i - t_i$ paths is described as $\mathcal{P}_i$. For simplicity, it is assumed that $\mathcal{P}_i \neq \emptyset$ for all $i$. $\mathcal{P} = \cup_i \mathcal{P}_i$ is the set of all source-destination paths of all commodities. A flow $f$ is said to be feasible if $\sum_{P \in \mathcal{P}_i} f_P = r_i$ for all commodities $i$.

All edges $e \in E$ come with a latency function $l_e(f_e)$ that may depend on the flow $f_e$ that traverses this edge. This latency describes the cost or delay experienced by players using this edge. All players traversing the same edge experience the same latency $l_e(f_e)$

on this edge. All latency functions are assumed to be non-negative, continuous, and non-decreasing. A triple $(G, r, l)$ is called an instance.

The latency of a path $P$ with respect to a flow $f$ is defined as the sum of the latencies on all edges of the path, thus denoted by $l_P(f) = \sum_{e \in P} l_e(f_e) f_e$. The total or social cost $C(f)$ of a flow $f$ is the total latency of a flow over all paths or edges in the network. We thus can write $C(f) = \sum_{P \in \mathcal{P}} l_P(f_P) f_P = \sum_{e \in E} l_e(f_e) f_e$. Thinking of players, we can see this as the latency experienced by a player on a given path, multiplied by the number of players using this path; then summed over all paths.

Players in this setting are selfish and non-cooperative. Each player tries to use paths that minimise his latency. We say that selfish players use shortest paths, without considering the total cost this selfishness induces. Applying the idea of Nash equilibria to selfish routing games leads to the following definition:

**Definition 1** ([War52])**.** A flow $f$ feasible for $(G, r, l)$ is at Nash equilibrium if and only if for all commodities $i$ and $P_1, P_2 \in \mathcal{P}_i$ with $f_{P_1} > 0$, we have $l_{P_1}(f) \leq l_{P_2}(f)$.

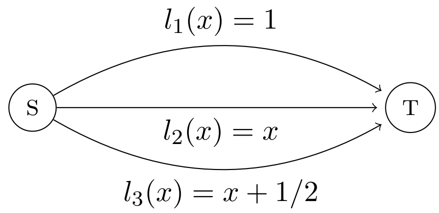
We also say $f$ is at Wardrop equilibrium, due to [War52]. Therefore, a feasible flow is at Wardrop equilibrium if no player can reduce the latency that he experiences when switching paths; given all other players stay on their paths. As we assume players to control an infinitesimal small amount of the total flow, we can neglect the effect a player switching paths has on the latency of the paths. For a given commodity $i$, all $s_i - t_i$ paths with positive flow in the Wardrop equilibrium have equal and minimal latency $L_i$.

The marginal latency function of an edge is $\widehat{l_e} : (l_e(f_e) \cdot f_e)' = l_e(f_e) + l'_e(f_e) f_e$ and describes the marginal cost of increasing flow on an edge. A flow that minimises the total cost is called an optimal flow. An optimal flow is at Nash equilibrium regarding the marginal latency functions $\widehat{l}$ of a given instance. For an optimal flow, all paths with positive flow have equal and minimal marginal latency. The latency of the paths might however have various values. Therefore, for an optimal flow it is no longer given that all players use shortest paths, i.e. it is not given that no player has an incentive to switch paths.

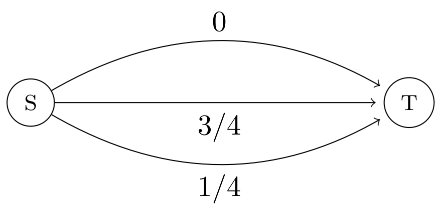
We therefore again have the trade-off between stable Nash equilibria with non-optimal total cost and optimal allocations which might not be stable. This brings us again to the Price of Anarchy - measuring the inefficiency of selfish routing in this case. For non-atomic selfish routing games, [RT02] showed tight bounds of 4/3 on the Price of Anarchy for arbitrary networks and linear latency functions. [Rou05] later showed Price of Anarchy results for a variety of networks and latency functions. For a survey on the Price of Anarchy of routing games, we also refer to [Rou05].

We illustrate selfish routing games with a few examples. Our first example can be seen in Figure 2.1. This is a single-commodity, parallel-edge instance. Parallel-edge instances consist of non overlapping, parallel edges between the source and destination, such that the notion of paths and edges coincide (as every path can be seen as a unique edge). We assume a demand of $r = 1$. Latencies of the edges are pictured in Figure 2.1a. With flow as in Figure 2.1b, we will have $l_2(3/4) = l_3(1/4) = 3/4 \leq l_1(0) = 1$. We can see that this flow is at Wardrop (or Nash) equilibrium, as the edges with positive flow have equal and minimal latency: no player can reduce his latency by switching edges. We denote this Nash flow as a vector $n = (0, 3/4, 1/4)$. Its total cost is $C(n) = \sum_{e \in E} n_e l_e(n_e) = 3/4 \cdot 3/4 + (1/4 + 1/2) \cdot (1/4) = 3/4$.

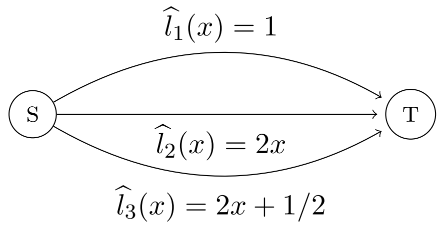
The optimal flow is simply a flow at Nash equilibrium regarding marginal latency functions. These are seen in Figure 2.1c. For example, the marginal latency of edge 3 is $(x(x + 1/2))' = 2x + 1/2$. Looking for an equilibrium regarding marginal latencies gives us the optimal flow as in Figure 2.1d, as then $\widehat{l_1}(1/4) = \widehat{l_2}(1/2) = \widehat{l_3}(1/4) = 1$. We denote this flow as $o = (1/4, 1/2, 1/4)$. Its total cost are $C(o) = 1/4 \cdot 1 + 1/2 \cdot 1/2 + (1/4 + 1/2) \cdot 1/4 = 11/16$. It is easy to see that the optimal flow does not use shortest paths only, thus is not at Wardrop equilibrium regarding its latency functions: we have latencies of $l_1(1/4) = 1$, $l_2(1/2) = 1/2$, and $l_3(1/4) = 3/4$. Players on edge 1 and 3 could therefore reduce their costs if they would use the middle edge instead. As the Nash equilibrium in this game is unique, the Price of Anarchy for this instance is $PoA = \frac{C(n)}{C(o)} = \frac{3/4}{11/16} = 12/11$.



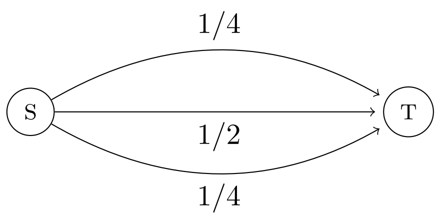

(a) Latency functions




(b) Nash flow




(c) Marginal latency functions




(d) Optimal flow

Figure 2.1.: Example of an instance

Two further examples are shown in Figure 2.2a and Figure 2.2b, both again with $r = 1$. Figure 2.2a is called the Pigou instance and goes back to [Pig20]. Many results established in selfish routing games can be derived from this instance, for example the upper bound on the Price of Anarchy for linear latency functions. The unique Wardrop equilibrium $n = (0, 1)$ for this instance is that all players use the bottom edge, each of them with individual latency $l_2(1) = 1$ and total latency $C(n) = 1 \cdot 1 = 1$. No player would like to use the top edge, as even if a small amount $\varepsilon > 0$ uses the top edge, the latency $1 - \varepsilon$ on the bottom edge is strictly less than the latency 1 on the top edge. The optimum $o$ routes half of the flow over the top edge and half of the flow over the bottom edge, thus has $C(o) = 1/2 \cdot 1 + 1/2 \cdot 1/2 = 3/4$. The Price of Anarchy for this instance is therefore 4/3, which equals the worst-case inefficiency of selfish routing instances with linear latency functions.

The instance in Figure 2.2b is an example of an instance with a Price of Anarchy of 1, i.e. an instance where optimum flow and Nash flow coincide. Routing 2/3 over the top and 1/3 over the bottom edge solves $l_1(2/3) = 2/3 = 2(1/3) = l_2(1/3)$ and

$2/3 + 1/3 = 1 = r$, therefore this allocation is a Nash flow. The flow also solves $\hat{l}_1(2/3) = 2 \cdot (2/3) = 4 \cdot (1/3) = \hat{l}_2(1/3)$, thus minimises marginal latency functions. Thus, Wardrop equilibrium and optimum in this instance coincide. As the optimum is the unique Wardrop equilibrium, the Price of Anarchy for this particular instance is 1, as both have equal total cost.


(a) Pigou instance (b) Optimal and Wardrop flow coincide

Figure 2.2.: Further instances

The instances we have seen so far had all in common that their graphs were *parallel-edge (or parallel-link) networks.* these networks only consist of parallel edges, in particular paths are disjunct and do not share edges. Further, as the network could be reduced such that every path only consist of one edge between source and destionation, the notions of paths and edges coincide: each edge is a path and each path is an edge. However, in more general networks paths consist of multiple edges. The instance seen in Figure 2.3 is an example of a network which does not only consist of parallel edges. The instance is called *Braess's paradox* and goes back to [Bra68]. Again, a total demand of $r = 1$ must be routed from $s$ to $t$. The network consists of three paths: $s - u - t$, $s - v - t$, and $s - u - v - t$. The optimum routes flow 1/2 over path $s - u - t$ and flow 1/2 over path $s - v - t$. On both of these paths, each player experiences a latency of 1/2 on the edge with variable latency (as in this case half of the flow uses this edge), and latency of 1 on the edge with constant latency. The latency a player experiences on either of these paths is therefore $1/2 + 1 = 3/2$. Regarding the total cost, both paths have total cost $1/2 \cdot 1/2 + 1/2 \cdot 1 = 3/4$ each, thus the total flow has costs of $3/4 + 3/4 = 3/2$. We again observe that the optimal flow is not a Wardrop equilibrium, as players have incentives to use paths other than $s - u - t$ or $s - v - t$: Given all other players stick to the optimal flow in the network, a player could reduce individual costs by choosing path $s - u - v - t$

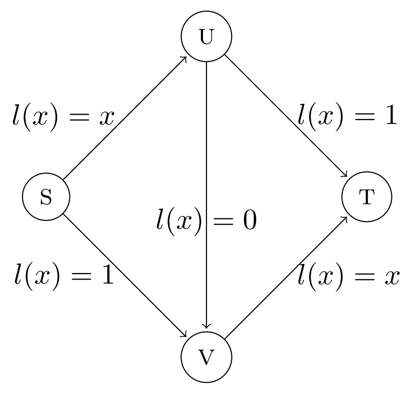



Figure 2.3.: Braess's paradox

instead, therefore having only a delay of $1/2 + 0 + 1/2$, which is less than the delay they experience on paths $s - u - t$ and $s - v - t$.

If all players route themselves selfishly, they would all use $s - u - v - t$. This results in individual latency of $1 + 0 + 1 = 2$ for every player. Interestingly, this latency is higher than the latency *every* player experiences in the optimum. However, given all players use $s - u - v - t$, no player can reduce their latency by choosing a different path, as for example taking path $s - u - t$ instead would lead to the same individual latency $1 + 1 = 2$. In this example, the Wardrop equilibrium has total cost $1 \cdot 1 + 0 \cdot 1 + 1 \cdot 1 = 2$. As the cost of the optimum are 1.5, the Price of Anarchy is $\frac{2}{1.5} = \frac{4}{3}$.

Braess's paradox is also interesting for a different reason: it shows that adding edges to a network can decrease system performance, even if the added edges come with no attached cost. So see this, imagine Braess' paradox without edge $u - v$. There are only two paths in the network. The optimum does not change, however without the optional edge $u - v$ the Wardrop equilibrium now equals the optimal allocation. Selfish behaviour is optimal. Introducing edge $u - v$ with zero costs however leads to sub-optimal selfish behaviour.

### 2.2.1. Improving system performance

Selfish routing is an attractive model as its resulting equilibria do not require supervision of the system and yet provide stable outcomes where no player has an incentive to deviate. But Nash equilibria often lack efficiency in regards to total cost. This is reflected in the fact that the Price of Anarchy is unbounded for arbitrary latency functions, see [RT02]. This is why several procedures have been developed to reduce the inefficiency of selfish routing. The most popular approaches are presented hereafter.

*Capacity augmentation* increases the total traffic that has to be routed in an optimal flow. Instead of comparing Nash and optimal flow with the same demand, it compares the total cost of an Nash flow in the original network to the total cost of an optimal flow in an augmented network with higher total demand. This approach has the advantage that it still does not demand control of routing decisions. [RT02] show that for any continuous and nondecreasing latency functions in any network, the cost of an Nash flow in the original network is at most the cost of an optimal flow in the augmented network with twice the traffic. An interpretation of this approach is to see an increase of the total traffic to the optimal flow as an increase in speed with which the traffic is routed. Then, selfish routing can be tackled by a modest increase in speed or bandwidth, see [Rou07].

*Edge pricing* discourages selfish behaviour by imposing taxes or tolls on edges that are individually profitable but socially harmful. This approach follows ideas common in *mechanism design*, a subarea of Algorithmic Game Theory that asks how underlying conditions must be set to animate socially good selfish behaviour. For a greater overview, we refer to [Nis99]. One common way of taxing edges are marginal cost taxes, suggested by [Pig20]. The tax on each edge corresponds to the harm a player induces on all players using this edge by routing along the edge. Thus, the latency of an edge is the sum of the individual latency on that edge and the tax. With this implementation, the optimal solution becomes a Wardrop equilibrium, thus is selfishly chosen by the players. For an in-depth overview of edge pricing, see [YH05].

*Edge removal* is questioning which subset of edges in a network leads to the most benevolent selfish routing. Introduced by [Rou06], it is based on the observation that removing edges from a network can reduce the inefficiency of selfish routing. Braess's paradox is an example where this applies. It is therefore asked which edges must be removed from the network to make players behave socially well. This can be seen as a variety of network design, therefore asking how to design good networks. For improved resuls on edge removal, see [Lin+11].

The approaches presented so far have in common that all players continue to route themselves selfishly and by their own will. A different approach is to establish a central authority that takes over the routing decision for a percentage $\alpha$ of all players. The players who are being routed by the central authority can no longer choose their paths themselves. The remaining players continue to route themselves selfishly. This approach is called *Stackelberg routing*. The authority is called *Stackelberg leader*. In the first step, the Stackelberg leader routes his fraction of flow through the network. In the next step, the remaining selfish players route themselves over shortest paths, with considering the already existing controlled flow on edges. The aim of the authority is to minimise total cost of controlled and selfish players in the network. Therefore, he must route the controlled players in the first step such that the selfish players route themselves over socially cheap edges in the next step. Selfish players experience minimal latency in the network, but controlled players might experience higher latency.

Before introducing the model formally, we give a short example of how Stackelberg routing can improve system performance: We again refer to the Pigou instance in Figure 2.2a. Let us assume a total demand of $r = 1$ and the Stackelberg leader controls share $\alpha = 0.5$ of the traffic. Let us further assume the Stackelberg leader routes his flow along the top edge. The remaining selfish players have flow $1 - 0.5 = 0.5$, and given there is no flow on the bottom edge, but flow 0.5 on the top edge, they will route themselves along the bottom edge. The combined flow therefore has flow 0.5 on the top and flow 0.5 on the bottom edge, which equals the optimal flow for this instance. Thus, by restricting the number of players who are allowed to route themselves selfishly, implementation of the optimal allocation is possible. We now introduce Stackelberg routing formally.

# 3. Stackelberg routing and state of the art

We have already outlined the idea of Stackelberg routing in Section 2.2.1. For the sake of readability, we will shortly repeat this here. Stackelberg games can be modelled as a two-stage game: First, the *Stackelberg leader* routes his share of players he can control through the network. We say that players that are being routed by the Stackelberg leader are *controlled*. We denote the flow controlled by the Stackelberg leader as $g$. After the Stackelberg leader has routed his share of flow in the network, the remaining selfish players route their flow along shortest paths. They consider the already underlying controlled flow on the edges. We say that selfish flow $h$ is induced by $g$. The aim of the Stackelberg leader in the first step is to route his share of flow in such a way so that the selfish players use paths that lead to low total cost for the total flow $g + h$. Therefore, the Stackelberg leader tries to minimise the cost of of the combined flow of controlled and selfish flow, thus he tries to minimise $C(g + h) = \sum_{e \in E}(g_e + h_e)l_e(g_e + h_e)$.

## 3.1. Stackelberg routing

The model we present is taken from [Rou05] and based on the model in Section 2.2. Model properties specific to single-commodity, parallel-edge networks are from [Rou04].

In addition to $(G, r, l)$, Stackelberg routing games come with an additional parameter $\alpha \in [0, 1]$. $\alpha$ measures the fraction of the total demand controlled by the Stackelberg leader. We distinguish between strong and weak Stackelberg strategies. A strong Stackelberg strategy $g$ must be a flow feasible for $(G, r', l)$ with $r' = (\alpha_1 r_1, \ldots, \alpha_k r_k)$ for some $\alpha_1, \ldots, \alpha_k \in [0, 1]$ so that $\sum_{i=1}^{k} \alpha_i r_i = \alpha \sum_{i=1}^{k} r_i$. If $\alpha_i = \alpha$ for all commodities $i$, i.e. $g$ is a flow feasible for $(G, r', l)$ with $r' = \alpha(r_1, \ldots, r_k)$, the strategy is called a weak Stackelberg strategy. The two notions coincide for single-commodity instances (thus $k = 1$).

For a given Stackelberg strategy $g$, we define $\tilde{l}_e(x) = l_e(g_e + x)$ for every edge. We then have that the Nash flow $h$ induced by Stackelberg strategy $g$ is a flow at Nash equilibrium for $(G, r - r', \tilde{l})$, or in a different notation, $(G, r - r', l(g + h))$. Thus, selfish players use shortest paths given the underlying flow set by the Stackelberg leader. The Stackelberg leader tries to minimise the combined total cost of the controlled and induced selfish flow, i.e. he tries to minimise $C(g + h) = \sum_{e \in E}(g + h)l_e(g_e + h_e)$.

We now observe the following: For all commodities $i$, paths with positive induced selfish flow, i.e. $h_{P \in \mathcal{P}_i} > 0$ have equal and minimal latency $L_i$. Paths with positive controlled flow, but no induced selfish flow cannot have smaller latency than paths with selfish flow, i.e. paths with $g_{P \in \mathcal{P}_i} > 0$ must have $l_P(g_P + h_P) \geq L_i$.

We shift our analysis to single-commodity instances. As mentioned, for these instances strong and weak Stackelberg strategies coincide and we no longer have an vector of $\alpha$-values, but just a single value. The notion simplifies as follows: in single-commodity instances, the Stackelberg leader controls $\alpha r$ of the flow. A Stackelberg strategy $g$ is a flow feasible for $(G, \alpha r, l)$. The induced selfish flow is a flow at Nash equilibrium for

$(G, (1-\alpha)r, \tilde{l})$. With $r = 1$, this further simplifies to $g$ being feasible for $(G, \alpha, l)$ and $h$ being at Nash equilibrium for $(G, 1-\alpha), \tilde{l})$.

We illustrate Stackelberg routing with the instance presented in Figure 2.1. We set $r = 1$ and assume that the leader controls $\alpha = 0.5$ of the flow. The total flow the leader can control is therefore $\alpha r = 0.5 \cdot 1 = 0.5$. The remaining selfish players have flow $1 - 0.5 = 0.5$. Let us imagine the Stackelberg leader applies the following strategy: he puts flow 1/4 on the top edge and flow 1/4 on the middle edge. We therefore have $g = (1/4, 1/4, 0)$. Given the underlying flow $g$ on the edges, the selfish players now use shortest paths: this leads to $h = (0, 3/8, 1/8)$, as $l_2(1/4 + 3/8) = 5/8 = l_3(0 + 1/8) \leq l_1(1/4 + 0) = 1$. The total flow equals $g + h = (1/4, 5/8, 1/8)$ and comes with $C(g + h) = 1 \cdot (1/4) + (5/8) \cdot (5/8) + (1/8 + 1/2) \cdot 1/8 = 0.71875$.

By reducing from the NP-hard problem "$\frac{1}{3}$-$\frac{2}{3}$-PARTITION", [Rou04] showed NP-hardness for computing the optimal Stackelberg strategy for a given instance, even for single-commodity, parallel-edge networks with linear latencies. To overcome this problem, [Rou04] introduced two Stackelberg strategies - *Scale* and *Largest Latency First (LLF)* - that are computable in polynomial time. For LLF, [Rou04] showed a performance guarantee of $\frac{4}{3+\alpha}$ for linear latencies and $\frac{1}{\alpha}$ for arbitrary latencies in parallel-edge networks. [CS07] as well as [Swa07] presented generalised versions of LLF for wider classes of networks. LLF outperforms Scale regarding the Price of Anarchy in parallel-edge networks (see [Rou04]), but Scale can perform better than LLF in general networks (see [CS07]). We first present the two strategies for general networks, then give specific examples for parallel-edge networks. We will later refer to Scale and LLF in our model too. We further note that a Stackelberg strategy that puts no more flow on every edge than the optimal flow, i.e. $g_e \leq o_e$ for all edges, is called opt-restricted.

*Scale* simply scales the optimal flow $o$ for $(G, r, l)$ according to the share $\alpha$ the leader controls. Thus, for optimal flow $o$ Scale simply sets $g_e = \alpha o_e$ for every edge. This definition holds both for general as well as parallel-edge networks.

For general networks, *Largest Latency First (LLF)* computes optimal flow $o$ for $(G, r, l)$ and puts $g_P = o_P$ on paths used by $o$ until $\alpha$ units of flow have been routed. By putting $g_P = o_P$, LLF *saturates* path $P$. LLF starts with a path with highest latency regarding $o$ and saturates paths in descending order regarding their latency regarding $o$. For networks with $m$ parallel edges, LLF simplifies as follows: LLF computes optimal flow $o$ for $(G, r, l)$ and indexes all edges from lowest to highest latency with respect to $o$, such that $l_1(o_1) \leq l_2(o_2) \leq \cdots \leq l_m(o_m)$. An edge $e$ is saturated if $g_e = o_e$. LLF starts with an edge with highest latency regarding optimal flow $o$ and puts $g_e = o_e$ on edges one-by-one, in decreasing terms regarding their latency to $o$, as long as the sum of the flow put on the saturated edges does not exceed the total flow the leader is allowed to control. If saturating edge $k$ would exceed the leader's total share of flow, he puts his remaining allowed share on that edge. No controlled flow is put on edges with index smaller than $k$. More formally, LLF indexes all edges such that $l_1(o_1) \leq \cdots \leq l_m(o_m)$ and chooses $k \leq m$ such that $\sum_{i=k+1}^{m} o_i \leq \alpha r$. LLF now puts $g_i = o_i$ on all edges with $i > k$, $g_k = \alpha r - \sum_{i=k+1}^{m} o_i$ and $g_i = 0$ for $i < k$. For linear latencies, a slightly different interpretation of LLF is that it tries to saturate edges in decreasing order of constant terms until a total flow of $\alpha$ has been allocated.

We illustrate the two strategies with two examples. Our first example is the Pigou instance (Figure 2.2a) we introduced earlier. The optimal flow in this instance is $o =$

$(0.5, 0.5)$ with $C(o) = 0.5 \cdot 1 + 0.5 \cdot 0.5 = 0.75$. Applying Scale leads to $g = \alpha o = (0.5\alpha, 0.5\alpha)$. The induced Nash flow of Scale is $h = (0, 1 - \alpha)$ for any $\alpha$, thus we have $g + h = (0.5\alpha, 1 - 0.5\alpha)$ and $C(g + h) = 0.5\alpha + (1 - 0.5\alpha)^2$.

With LLF, we have $l_1(o_1) = 1$ and $l_2(o_2) = 0.5$, thus $l_2(o_2) \leq l_1(o_1)$. We distinguish two cases: If $\alpha < 0.5$, LLF fails to saturate the most expensive edge, thus puts the entire share of controlled flow on the top edge, thus $g = (\alpha, 0)$. If $\alpha \geq 0.5$, LLF saturates the top edge and puts the remaining share of flow on the bottom edge, i.e. $g = (o_1, \alpha - o_1) = (0.5, \alpha - 0.5)$. In both cases, the induced Nash flow is $h = (0, 1 - \alpha)$ thus the total flow $g + h$ is $(\alpha, 1 - \alpha)$ if $\alpha < 0.5$ and $(0.5, 0.5)$ otherwise.

Our second example is the instance seen in Figure 2.1. We pick two cases, one with $\alpha = 0.2$ and one with $\alpha = 0.4$. The optimal flow is $o = (0.25, 0.5, 0.25)$ and the latencies regarding the optimal flow are $l_1(1/4) = 1$, $l_2(1/2) = 1/2$ and $l_3(1/4) = 1/4 + 1/2 = 3/4$. Thus, the top edge is most expensive, then comes the bottom edge, and the middle edge is cheapest. Given $\alpha = 0.2$, Scale leads to $g = 0.2 \cdot o = (0.2 \cdot 0.25, 0.2 \cdot 0.5, 0.2 \cdot 0.25) = (0.05, 0.1, 0.05)$, and the total controlled flow sums up to 0.2, as required. Given $\alpha = 0.4$, Scale leads to $g = 0.4 \cdot o = (0.1, 0.2, 0.1)$. For LLF, we get $g = (0.2, 0, 0)$ if $\alpha = 0.2$, as $\alpha < o_1$ and the most expensive edge cannot be saturated. If $\alpha = 0.4$, the top edge is saturated, thus $g_1 = o_1 = 0.25$. We fail to saturate the bottom edge as $o_3 > \alpha - o_1 = 0.15$, thus we put $g_3 = 0.15$. No flow is put on the middle edge. This leads to $g = (0.25, 0, 0.15)$.

## 3.2. State of the art

Having introduced Stackelberg routing, we now summarise previous research done in this field. This will also serve as a motivation for our own work.

Stackelberg games originate from [Sta34], introduced as an oligopoly model in Economics. [KLO97] first proposed the idea to use this leader/follower model for system performance improvements in selfish routing games. [KLO97] established conditions on the network instance to guarantee the existence of Stackelberg strategies that enforce the optimum. In this thesis we are more interested in how Stackelberg strategies perform not only in optimal, but also in sub-optimal settings. The basis of our work is [Rou02; Rou04], who analysed to what degree Stackelberg routing improves the performance of selfish routing in parallel-edge networks. [Rou04] proved NP-hardness of computing the best Stackelberg strategy and provided two heuristics, namely Largest Latency First (LLF) and Scale, to overcome the NP-hardness problem. Furthermore, [Rou04] provided upper bounds of $4/(3 + \alpha)$ and $1/\alpha$ on the Price of Anarchy for LLF for linear and arbitrary latency functions, respectively.

Several papers extended the analysis by [Rou04]. [KM02] investigated an approximation scheme that comes with performance guarantee that is only a constant factor worse than the optimum. [Swa07] as well as [CS07] showed upper bounds of $1 + 1/\alpha$ on the Price of Anarchy of LLF for single-commodity, serial-parallel networks with arbitrary latency functions. Furthermore, [Swa07] obtained upper bounds on the Price of Anarchy for Scale and LLF for polynomial latency functions in general networks. [KK09] presented improved upper and lower bounds on the Price of Anarchy for Scale and LLF in general multi-commodity networks with linear or polynomial latency functions. [Fot10] established bounds on LLF and Scale for atomic congestion games with affine latency functions. These bounds were later improved by [BV19]. [CS07] proved that opt-restricted strategies, i.e.

strategies such that the Stackelberg leader does not send more flow than the system optimum does over every edge, do not increase the Price of Anarchy. More recently, [BHS10] answered the open question of whether Stackelberg routing bounds the Price of Anarchy in general networks negatively. They also developed an efficiently computable Stackelberg strategy that induces a flow whose cost is at most the cost of an optimal flow with respect to demands increased by a factor of $1 + \sqrt{1 - \alpha}$. [KS09] developed an algorithm that computes efficiently the smallest fraction of controlled players needed to induce the optimum in parallel-edge networks. [SW09] investigated the smallest fraction of controlled players needed in order to improve the social costs in comparison to the Nash equilibrium with selfish players only.

However, all this work has in common that it ignores the factor to which players experience individually unequal latencies. The Stackelberg leader was allowed to route players to his liking, putting controlled players arbitrarily worse off than selfish players. Some papers in the field of transportation science consider the degree to which the Stackelberg leader puts controlled players unfair off: [Biy+18] models altruistic players who still behave selfishly, but are willing to bear higher costs to a certain extent. [Jah+05] consider the idea that paths with positive flow may only be inferior to each other to a certain degree. These models, however, are highly different to the model studied in this thesis: they either follow empirical approaches or use a different theoretical model.

We use the model of Stackelberg routing and add fairness considerations. We define the unfairness of a given flow $f$ as the ratio of highest latency experienced by any player to lowest latency in the network for the given flow $f$. Thus, unfairness can be seen as the maximum degree to which any player could reduce their experienced latency if they would take over their routing decision and route themselves. Regarding the selfish routing model used in our work, [Rou02] and [CSS07] are closest to the idea of considering unfairness. Both mention and analyse unfairness, however their unfairness measure is different to ours and not applied to Stackelberg routing, but different settings. Interestingly though, some of our results coincide with results shown by [Rou02] and [CSS07].

[Rou02] briefly touches the unfairness of optimal allocations. [Rou02] defined the unfairness of an instance as the maximum ratio between the latency of any path in an optimal flow to the latency of any path in a Nash flow. Therefore, latencies of paths across allocations are compared to each other. This is very different to our unfairness measure, as we compare latencies of paths of the same flow, instead of across flows. We believe our measure to be the more intuitive measure as we believe players to compare themselves to other players in the same setting, instead of to themselves in a different setting.

Closer to our unfairness measure is [CSS07], who compare latencies of paths in the same flow, instead of across flows. Their unfairness measure is similar to ours, however, they demand that both, the paths with highest and lowest latency come with positive flow. We on the other hand only demand the highest path to have positive flow, the cheapest path players compare themselves too might come with no flow for the given instance. Braess's paradox is an easy example for which the optimal flow gives same costs for all paths with positive flow, however puts no flow on a path with lower latency (regarding the optimum flow). Further, [CSS07] do not analyse unfairness specifically for Stackelberg routing, they focus on min-max flows.

# 4. Stackelberg routing with fairness

Having introduced the field of research this thesis belongs to, this chapter contains our contribution to the topic. As mentioned, previous research on Stackelberg routing disregarded how much worse off the Stackelberg leader puts players compared to if they would route themselves; no constraints were made on how much worse off the leader is allowed to put players. We therefore introduce unfairness considerations into the model and ask how much worse off the Stackelberg leader puts a player by controlling him compared to if that could route himself. We define the unfairness $\mathcal{B}(\alpha)$ for a given Stackelberg strategy as the ratio of highest latency experienced by any player to lowest latency in the network, for the same given flow. Thus, this ratio can be interpreted as how much a controlled player could reduce his individual cost if he could unilaterally take over the routing decision and route himself. We introduce this unfairness measure in Section 4.1. We bound the unfairness of Stackelberg routing in single-commodity, parallel-edge networks in Section 4.2 and show that the unfairness of some specific Stackelberg strategies increases with $\alpha$ in Section 4.3.

We introduce a new measure $\beta$ which limits the degree of allowed unfairness. We introduce fair Stackelberg games for which the Stackelberg leader is not allowed to route players along paths that would make them more than $\beta$-times worse off than if they would route themselves. We present lower bounds on the Price of Anarchy of all fair Stackelberg strategies in Section 4.4 and further give algorithms of fair strategies in Section 4.5.

## 4.1. Preliminaries

We adapt our model from the model presented in Section 2.2 and Section 3, respectively. However, we focus on parallel-edge, single-commodity networks. Parallel-edge networks consist of parallel edges only, therefore paths do not share edges. Further, the notions of edges and paths overlap, as every path can be modelled as one unique edge. If not stated otherwise, all results in this Chapter refer to parallel edges only. Instead of referring to the latency of paths, it is sufficient to talk about the latency of individual edges. We denote open intervals not including their limit points with parentheses and closed intervals with square brackets.

We are given a directed network $G = (V, E)$ with vertices $V$ and parallel edges $E \subseteq V \times V$. Let $m$ be the number of edges in the network. The network has a source $s$ and a sink $t$ with demand $r = 1$ of flow that should be transported from $s$ to $t$. $r$ represents a large population of non-atomic players, each of them controlling an infinitesimally small amount of the entire demand that they want to be send from $s$ to $t$. We assume w.l.o.g throughout this thesis the demand to be normalised to 1, i.e. $r = 1$ and will only specify the demand for a given instance $(G, r, l)$ if it is not 1. Otherwise, we just refer to $(G, l)$.

$x = (x_1, \ldots, x_m)$ is an $m$-vector which for each edge specifies how much flow is routed on the edge. Flow $x$ is called feasible if $\sum_{e \in E} x_e = r = 1$. Each edge $e \in E$ has an associated latency function $l_e : \mathbb{R}_+ \to \mathbb{R}_+$, indicating the latency each player experiences if they use this edge. The latency on edge $e$ may be dependent on the flow $x_e$ using the edge, otherwise it is constant. We assume $l_e$ to be differentiable, continuous, and non-decreasing. We define linear latency functions as $l_e(x) = a_e x + b_e$ with $a_e, b_e \geq 0$ and polynomial latency functions as $l_e(x) = \sum_{i=0}^{d} a_i x^i$ with $a_e \geq 0$ and $d \in \mathbb{N} \geq 2$.

Each player traversing edge $e$ will have the same individual latency $l_e(x_e)$, with $x_e$ being the total flow of all players on that edge. The total (or social) latency of flow $x_e$ traversing edge $e$ is written as $x_e l_e(x_e)$. We assume $x_e l_e(x_e)$ to be a convex function of $x_e$. The social (or total) cost of total flow $x \in \mathbb{R}_+^m$ then is $C(x) = \sum_{e \in E} x_e l_e(x_e)$. Marginal latency functions are denoted by $\widehat{l_e}(x_e) = (x_e l_e(x_e))' = l_e(x_e) + x_e l_e'(x_e)$.

$n = (n_1, n_2, \cdots, n_m) \in \mathbb{R}_+^m$ is a feasible flow at Wardrop (or Nash) equilibrium for instance $(G, l)$. $n$ is also called Nash flow. This corresponds to the idea that all players route themselves selfishly in the network, each of them aiming to minimise their own latency. In $n$, no player can decrease his latency by switching edges. More formally, it must hold that $l_i(n_i) \leq l_j(n_j)$ for all $i, j \in E$ with $n_i > 0$. Thus, all edges with $n_e > 0$ have equal and minimal latency $L$ in the network, thus $L = l_e(n_e)$ for any edge with $n_e > 0$.

In a similar manner, $o = (o_1, o_2, \cdots, o_m)$ denotes an optimal flow for $(G, l)$. $o$ is a feasible flow for $(G, l)$ with minimal total cost among all feasible flows. All edges with $o_e > 0$ have equal and minimal marginal latency with respect to $o$, therefore $o$ is a feasible flow at Nash equilibrium for $(G, \widehat{l})$. However, $o$ is not guaranteed to be at Nash equilibrium regarding $l$, thus players can experience different latencies in flow $o$. Both, $n$ and $o$ are computable in polynomial time and essentially unique in our setting.

Instances of Stackelberg routing come with an additional parameter $\alpha \in [0, 1]$. $\alpha$ equals the share of flow that is controllable by the Stackelberg leader. A given Stackelberg strategy consists of two flows: $g$ is the flow routed by the Stackelberg leader. We will call this *controlled flow*. $h$ is the flow of the remaining selfish players, who route themselves along shortest paths after the leader sets his flow. We call $h$ *induced selfish flow*. First, the Stackelberg leader sets $g$, then the remaining selfish players choose their individual shortest paths given the underlying controlled flow. The Stackelberg leader is interested to set his flow such that total cost of controlled and selfish flow are minimised, thus he wants to minimise $C(g + h) = \sum_{e \in E} (g_e + h_e) l_e(g_e + h_e)$. Total flow $g + h$ must be feasible for $r$, therefore $\sum_{e \in E} g_e + h_e = 1$ and further $g_e, h_e \geq 0$ for all edges.

Stackelberg strategies as defined in [Rou04] demand that the Stackelberg leader routes exactly his share $\alpha$ of flow among edges, thus $\sum_{e \in E} g_e = \alpha$. $h$ is then a feasible flow at Nash equilibrium for instance $(G, 1 - \alpha, l(g + h))$. Scale and LLF are strategies that satisfy this definition. We extend the set of Stackelberg strategies and allow the Stackelberg leader to route less flow than he is allowed to control. This implies $\sum_{e \in E} g_e \leq \alpha$. The induced Nash flow $h$ then must have a total flow of $1 - \sum_{e \in E} g_e$. $h$ is at Wardrop equilibrium for $(G, 1 - \sum_{e \in E} g_e, l(g + h))$. In both cases, the induced selfish flow uses shortest paths given the controlled flow set by the Stackelberg leader, the only difference is the amount of underlying controlled flow. We call strategies with $\sum_{e \in E} g_e < \alpha$ *reduced Stackelberg strategies*. Thus, in a reduced Stackelberg strategy the Stackelberg leader does not make use of his power entirely. The reason for this is that we introduce a model called *fair Stackelberg routing* which limits the degree of highest to lowest latency experienced by players. In that model, the leader is not allowed to route players to his liking, as it could put them too unequally off. Thus, to make players more equally off, the Stackelberg leader could reallocate some flow from edges with highest latency to edges with lowest latency. However, as edges with lowest latency are edges used by the induced selfish players, this can be seen as if the players routed among shortest paths by the Stackelberg leader would route themselves selfishly on these paths. We later show that any reduced Stackelberg strategy can be easily transformed into a full Stackelberg strategy for which the Stackelberg leader strictly controls $\alpha$. The difference between full and reduced strategies will only be relevant in Section 4.5.

We now introduce new fairness notations in our model. For a given Stackelberg strategy $g$ with induced selfish flow $h$ and a given share $\alpha$ of flow controllable by the leader, let $L_N(\alpha)$ denote the minimal latency of all edges given flow $h+g$. If $\alpha < 1$, i.e. selfish players exist in the network, $L_N(\alpha)$ corresponds to the latency of any edge with $h_e > 0$, as selfish players use shortest paths. Otherwise, it could be the case that $L_N(\alpha)$ is defined by an edge with no flow attached. Likewise, let $L_S^{max}(\alpha)$ be the maximal latency of all edges with positive flow given $g+h$, i.e. $L_S^{max}(\alpha) = \max_{e \in E: g_e + h_e > 0} l_e(g_e + h_e)$. It is here crucially to note that $L_S^{max}(\alpha)$ is only defined for edges with positive flow regarding $g+h$.

We can then define the unfairness $\mathcal{B}(\alpha)$ for a given Stackelberg strategy as the ratio of $L_S^{max}(\alpha)$ to $L_N(\alpha)$. $\mathcal{B}(\alpha)$ denotes how much maximally worse off the given Stackelberg strategy puts a player compared to if that player could route himself. If selfish players exist, this is similar to comparing the highest latency experienced by controlled players to the latency experienced by selfish players. It compares the maximum latency a player experiences to the minimal latency in the network, both for the same given flow. This measure therefore compares latencies *within* a flow. This can be seen as the degree to which a player could reduce his individual latency if he could unilaterally deviate from his current route and route himself over shortest edges instead. All other players are assumed to not change routes. Our unfairness measure is therefore a measure of how much unilateral deviation can improve a player's individual latency in the setting of Stackelberg routing: it considers unilateral deviation and the effect a single player switching paths has on the latency of paths is negligible. All other players do not change paths.

**Definition 2.** For a given Stackelberg instance $(G, l, \alpha)$ for which the leader applies Stackelberg strategy $g$ with induced selfish flow $h$, the unfairness $\mathcal{B}(\alpha)$ of the resulting total flow $g+h$ is denoted by

$$\mathcal{B}(\alpha) = \frac{L_S^{max}(\alpha)}{L_N(\alpha)} = \frac{\max_{e \in E: g_e + h_e > 0} l_e(g_e + h_e)}{\min_{e \in E} l_e(g_e + h_e)}$$

We normally assume the strategy used to induce $g+h$ is clear from the context. If it is not clear what strategy we refer to, we will write $g^{\text{STRAT}}$, $h^{\text{STRAT}}$, $L_N(\alpha, \text{STRAT})$, $L_S^{max}(\alpha, \text{STRAT})$, and $\mathcal{B}(\alpha, \text{STRAT})$, with STRAT the name of the strategy used.

Regarding optimal flow $o$, let $\mu$ denote one of the edges with highest latency among all edges with positive flow regarding $o$. Let $L_O^{max} = \max_{e \in E: o_e > 0} l_e(o_e) = l_{m_\mu}(o_{m_\mu})$ denote the maximal latency on any edge with positive flow regarding $o$ and let $L_O^{min} = \min_{e \in E} l_e(o_e)$ denote the minimal latency of all edges. Note that the minimal latency could be described by an edge with $o_e = 0$. We then come to the following definition.

**Definition 3.** For a given instance $(G, l)$, the unfairness of an optimal flow $o$ is denoted by

$$\mathcal{B}^{opt} = \frac{L_O^{max}}{L_O^{min}} = \frac{\max_{e \in E: o_e > 0} l_e(o_e)}{\min l_e(o_e)}$$

Stackelberg games studied so far ignored fairness considerations. We call Stackelberg games without any demands on fairness *standard Stackelberg games* and describe them by $(G, l, \alpha)$. We introduce a new type of Stackelberg games, called *fair Stackelberg games*. Instances of this type come with an additional parameter $\beta \geq 1$ which restricts the degree to which the Stackelberg leader can put controlled players worse off than selfish players.

**Definition 4.** A fair Stackelberg instance $(G, l, \alpha, \beta)$ is a Stackelberg instance $(G, l, \alpha)$ in which no player is allowed to experience individual latency higher than $\beta$-times the latency of any other path in the network, given all other players stick to their current routing decision.

*Fair Stackelberg strategies* are strategies that come with an unfairness with no more than $\beta$. Thus, fair Stackelberg strategies must come with $\mathcal{B}(\alpha) \leq \beta$ for any $\beta \geq 1$, for all $\alpha$. In contrast, standard Stackelberg strategies might have $\mathcal{B}(\alpha) > \beta$.

**Definition 5.** For a fair Stackelberg instance $(G, l, \alpha, \beta)$, a Stackelberg strategy is called fair if it comes with $\mathcal{B}(\alpha) \leq \beta$.

We further note that as we only consider parallel-edge networks, we have defined $\mathcal{B}(\alpha)$ only for single edges, instead of paths. For more general networks, the unfairness would be defined for paths instead of edges, i.e. the highest latency of a path with positive flow to the lowest latency of any path in the network for the given flow.

### 4.1.1. Remarks

We start with a couple of observations regarding our model. Any Wardrop equilibrium $n$ (i.e. any flow with selfish players only) comes with $\mathcal{B} = 1$. This is due to the fact that in a Nash equilibrium, players use the shortest paths available in the network. No player has an incentive to deviate as there is no path in the network that is cheaper for any player than the one they are currently using. Thus, 1 is a natural lower bound both on the unfairness of any Stackelberg strategy as well as the parameter $\beta$.

We can make use of the shortest paths argument when analysing Stackelberg routing as well. For flow $g + h$ induced by a specific Stackelberg strategy, it is easy to see that edges with induced selfish flow cannot be put at disadvantage relatively to other edges. This is because the induced selfish flow will use the shortest paths in the network. Thus, $L_N(\alpha)$ is simply the latency of any edge with induced selfish flow, thus any edge with $h_e > 0$. The only case where this not applies is if there is no induced selfish flow (thus if $\alpha = 1$). In that case, $L_N(\alpha)$ is simply the smallest latency in the network and can possible be determined by an edge that does not have any flow attached. Similarly, $L_O^{min}$ might be determined by a path with no attached flow[1].

For $L_S^{max}(\alpha)$ and $L_O^{max}$, it is guaranteed that their latency is determined by edges (or paths) with positive flow with respect to $g+h$ and $o$, respectively. This is because $L_S^{max}(\alpha)$ and $L_O^{max}$ are only defined for edges with positive flow. Therefore, our unfairness measure is the ratio of highest latency of any player in the network to lowest latency possible for a given flow.

We further note that our unfairness measure is a measure of *maximal* unfairness for a given flow; it does not tell us how equally (or unequally) the unfairness is distributed among players. To see this, we consider the instance in Figure 4.1. Imagine $\alpha = 0.5$ and a Stackelberg strategy which puts $g = (1/4, 0, 1/4)$. This induces $h = (0, 1/2, 0)$. The latencies on each edge given $g + h$ are $l_1(g_1 + h_1) = 1$, $l_2(g_2 + h_2) = 1/2$, and $l_3(g_3 + h_3) = 3/4$, respectively. If a controlled player on edge 1 and 3 could route himself, he would use edge 2 instead, experiencing latency of 1/2 (as we consider unilateral deviation, thus all other players do not change their routing decision, and the effect a single player has on the latency of edges is negligible). Thus, a player on the first edge is put off $\frac{1}{1/2} = 2$-times as worse as selfish players, whereas a player on the third edge is only put off $\frac{3/4}{1/2} = 1.5$-times as worse as selfish players. Even though $g$ comes with $\mathcal{B}(0.5) = 2$, it does not put *all* controlled players equally bad off.


(a) Instance (b) Flow $g + h$

Figure 4.1.: Unfairness is not necessarily equally spread

### 4.1.2. Mathematical program

Stackelberg routing can be seen as a bi-level optimisation problem: the Stackelberg leader chooses his controlled flow in the upper level such that selfish players play a Wardrop equilibrium with low overall costs in the lower level. We now express the problem faced by the Stackelberg leader in a fair Stackelberg game as a mathematical non-linear program, seen in Table 4.1. As we focus on single-commodity networks with parallel edges with $r = 1$ in this thesis, the optimisation problem is specifically adjusted to this kind of network. A more general version that allows for multi-commodity, arbitrary networks with variable demand can be found in Table A.1.

The objective function is the aim of the Stackelberg leader to minimise the social cost of the total (i.e. controlled and selfish) flow. By doing so, he faces several constraints. There are seven constraints in total. Constraints (1), (2), (3), and (7) are flow conservation constraints, requiring that the Stackelberg flow $g$, the induced selfish flow $h$, and any

[1] Braess's paradox is an example for which the individually shortest path in the network $(s - u - v - t)$ given the optimal flow is a path that comes with no optimal flow

$$
\begin{array}{llcll}
\min & \sum_{e\in E}(g_e+h_e)l_e(g_e+h_e) & & & \\
\text{s.t.} & \sum_{e\in E} g_e & \le & \alpha & (1)\\
& \sum_{e\in E} h_e & = & 1-\sum_{e\in E} g_e & (2)\\
& \sum_{e\in E} x_e & = & 1-\sum_{e\in E} g_e & (3)\\
& \sum_{e\in E} h_e l_e(g_e+h_e) & \le & \sum_{e\in E} x_e l_e(g_e+h_e) \quad \forall e\in E & (4)\\
& l_{e:g_e>0}(g_e+h_e) & \le & \beta\cdot l_{e'}(g_{e'}+h_{e'}) \quad \forall e\in E: g_e>0, \forall e'\in E & (5)\\
& l_{e:h_e>0}(g_e+h_e) & \le & 1\cdot l_{e'}(g_{e'}+h_{e'}) \quad \forall e\in E: h_e>0, \forall e'\in E & (6)\\
& g_e, h_e, x_e & \ge & 0 \quad \forall e\in E & (7)
\end{array}
$$

Table 4.1.: Mathematical program of fair Stackelberg routing

arbitrary flow $x$ are feasible flows that carry their respective share of the demand. We further note that (1), (2), and (3) are adapted to our more flexible range of strategies: we only demand that the Stackelberg leader routes at most $\alpha$, thus the induced selfish flow must carry the remaining flow, which might be higher than $1-\alpha$.

Constraint (4) demands the selfish flow $h$ induced by Stackelberg flow $g$ to be at Wardrop equilibrium, thus using shortest paths. The variational inequality is due to [Smi79, equation 9].

Constraints (5) and (6) are our new fairness constraints. (6) states that edges that are being used by selfish players have smallest latency in the network. This constraint applies to both Stackelberg games and fair Stackelberg games. (5) only applies to fair Stackelberg games. It demands that the latency experienced by any controlled player is at most $\beta$ times the latency of every other edge in the network. Thus, the mathematical program without (5) applies to standard Stackelberg strategies, which allow the Stackelberg leader to put players arbitrarily bad off. Constraint (5) demands the strategy to be fair. We conclude that the introduction of fairness constraints in Stackelberg routing is reflected as one additional constraint in the mathematical program faced by the Stackelberg leader.

We now continue with presenting our results regarding the impact of fairness considerations on Stackelberg routing. We begin with analysing how unfair the Stackelberg leader puts off players if he is given no external fairness constraint.

## 4.2. Bounding unfairness in standard Stackelberg games

In this section we focus on standard Stackelberg games. We recall that standard Stackelberg games do not have any externally given limits of allowed unfairness. We therefore ask how much worse off the Stackelberg leader puts players naturally if he is given no external constraints. We first bound how much worse off an optimal allocation puts players in Section 4.2.1. We give upper bounds on the unfairness of optimal flows of $\mathcal{B}^{opt} = 2$ for linear latencies and $\mathcal{B}^{opt} = d + 1$ for polynomial latencies up to degree $d \geq 2$. Section 4.2.2 bounds $\mathcal{B}(\alpha)$ for opt-restricted strategies, that are strategies in which the Stackelberg leader puts no more flow than an optimal flow on every edge. In detail, we show that the unfairness of any opt-restricted strategy is no worse than the unfairness of the optimum for a given instance.

### 4.2.1. Unfairness in optimal flows

We start with showing upper bounds on $\mathcal{B}^{opt}$, i.e. we show to what maximum extend the optimum puts players unfair off. Applied to Stackelberg routing, this can be seen as a special case of routing in case the leader enforces the optimum, as in case $g + h = o$, we will have $\mathcal{B}(\alpha) = \mathcal{B}^{opt}$. The results we establish in this section match bounds established by [Rou02, Theorem A.3.1], however we recall that the unfairness measure by [Rou02] is defined as the maximal ratio of latency of a path in an optimal flow to latency of a path in a Nash flow. [Rou02] therefore defines unfairness across flows, whereas we define unfairness within flows. Therefore, the results correspond to different measures.

**Theorem 1.** *For parallel-edge networks with linear latency functions, the unfairness of an optimal flow $o$ is upper-bounded by 2, i.e. $\mathcal{B}^{opt} \leq 2$.*

*Proof.* Imagine an instance with linear latency functions $l_e(x_e) = a_e x_e + b_e$ and $a_e, b_e \geq 0$. The marginal latency functions are $\widehat{l_e}(x_e) = 2a_e x_e + b_e$. As $2 \cdot l_e(x_e) = 2(a_e x_e + b_e)$ it holds that $\widehat{l_e}(x_e) \leq 2 \cdot l_e(x_e)$ for any edge $e$ with feasible flow $x_e$.

Let $o$ describe an optimal flow for $(G, l)$. Consider an arbitrarily chosen edge $i$ with positive flow in the optimum, i.e. $o_i > 0$. Clearly, $l_i(o_i) \leq L_O^{max}$. Let $j$ denote an edge with minimal latency given $o$, i.e. $l_j(o_j) = L_O^{min}$. We distinguish two cases: in case $o_j > 0$, i.e. the edge with minimal latency has positive flow regarding $o$, it must be that $\widehat{l_i}(o_i) = \widehat{l_j}(o_j)$, as the optimal flow allocates the flow such that all edges with positive flow have equal and minimal marginal latency. Thus, the following inequality holds:

$$l_i(o_i) \leq l_i(o_i) + o_i \cdot l_i'(o_i) = \widehat{l_i}(o_i) = \widehat{l_j}(o_j) = l_j(o_j) + o_j \cdot l_j'(o_j) \leq 2 \cdot l_j(o_j) \tag{4.1}$$

If $j$ had no flow with respect to $o$, we will have $l_j(o_j) = l_j(0)$. It must then follow that $j$ does not have minimal marginal latency, i.e. $\widehat{l_i}(o_i) \leq \widehat{l_j}(0)$. We then get:

$$l_i(o_i) \leq l_i(o_i) + o_i \cdot l_i'(o_i) = \widehat{l_i}(o_i) \leq \widehat{l_j}(o_j) = l_j(o_j) + o_j \cdot l_j'(o_j) \leq 2 \cdot l_j(0) \tag{4.2}$$

In both cases, the equation holds particular for edge $i = \mu$, i.e. an edge with highest latency among all edges with positive flow. We therefore get that the latency on edge $\mu$ is at most twice as large as the latency on any other edge, or differently, $L_O^{max} \leq 2L_O^{min}$, thus $\mathcal{B}^{opt} = \frac{L_O^{max}}{L_O^{min}} \leq 2$. ■

**Corollary 1.1.** *For parallel-edge networks with polynomial latency functions of degree $d \geq 2$, the unfairness of an optimal flow $o$ is upper-bounded by $d+1$, i.e. $\mathcal{B}^{opt} \leq d+1$.*

*Proof.* For polynomial functions it generally holds that $l(x) = \sum_{i=0}^{d} a_i \cdot x^i$ and $l'(x) = \sum_{i=1}^{d} i \cdot a_i \cdot x^{i-1}$. Furthermore,

$$x \cdot l'(x) = x \cdot \sum_{i=1}^{d} i \cdot a_i \cdot x^{i-1} = \sum_{i=1}^{d} i \cdot a_i \cdot x^i = \sum_{i=0}^{d} i \cdot a_i \cdot x^i \leq \sum_{i=0}^{d} d \cdot a_i \cdot x^i = d \cdot \sum_{i=0}^{d} a_i \cdot x^i = d \cdot l(x)$$

Applied to our setting, it therefore holds for any edge $e$ that $\widehat{l_e}(x_e) = l_e(x_e) + x_e \cdot l'_e(x_e) \leq l_e(x_e) + d \cdot l_e(x_e) = (d+1) \cdot l_e(x_e)$.

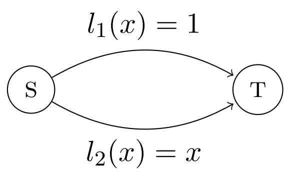
With similar notation as in Theorem 1, let $i$ be an edge with $o_i > 0$. Let $j$ be an edge with minimal latency given $o$, i.e. $L_O^{min} = l_j(o_j)$. We again distinguish two cases: in case $j$ has $o_j > 0$, it must be that $\widehat{l_i}(o_i) = \widehat{l_j}(o_j)$. If $j$ does not have any optimal flow, it must be $\widehat{l_i}(o_i) \leq \widehat{l_j}(o_j)$. In either case, we get

$$l_i(o_i) \leq l_i(o_i) + o_i \cdot l'_i(o_i) = \widehat{l_i}(o_i) \leq \widehat{l_j}(o_j) \leq (d+1) \cdot l_j(o_j) \tag{4.3}$$

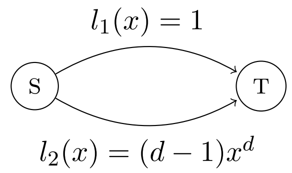
The equation particularly holds for edge $i = \mu$, i.e. an edge with highest latency of all edges with positive optimal flow. Thus, with $L_S^{max} = l_\mu(o_\mu)$, we get $L_O^{max} \leq (d+1) L_O^{min}$. ∎

Figure 4.2 illustrates the results just gained. The popular Pigou instance in Figure 4.2a is an example of an instance with linear latency functions where the optimal flow comes with $\mathcal{B}^{opt} = 2$. The optimum $o = (1/2, 1/2)$ routes equal traffic over both edges, thus the latency $l_1(o_1) = 1$ on the top edge is twice as high as the latency $l_2(o_2) = 1/2$ on the bottom edge.

Similarly, Figure 4.2b illustrates the case of $\mathcal{B}^{opt} = d+1$ of optimal flow $o$ for instances with polynomial latency functions up to degree $d \geq 2$. For any natural number $d \geq 2$, the marginal latencies of the edges equal $\widehat{l_1}(x) = 1$ and $\widehat{l_2}(x) = ((d-1)x^d \cdot x)' = (d-1)(d+1)x^d = (d^2-1^2)x^d$, thus with $\widehat{l_1}(x_1) = \widehat{l_2}(x_2)$, the optimum routes $o_2 = \frac{1}{(d^2-1^2)^{1/d}}$ along the bottom edge and $1 - o_2$ along the top edge. The experienced latency of the flow on the top edge is $l_1(o_1) = 1$, whereas the latency on the bottom edge is $l_2(o_2) = (d-1)\frac{1}{d^2-1^2} = \frac{d-1}{d^2-1^2} = \frac{1}{d+1}$, thus $\frac{l_1(o_1)}{l_2(o_2)} = (d+1)$, i.e. the latency of the flow on the top edge is (d+1) times as high as the latency of the flow on the bottom edge.




(a) Linear latency




(b) Polynomial latency (d $\geq$ 2)

Figure 4.2.: Instances matching the upper bound on the unfairness of optimal flow $o$

### 4.2.2. Unfairness for any opt-restricted Stackelberg strategy

Having bounded the unfairness of optimal flows in our model, we now try to bound the unfairness of standard Stackelberg strategies, i.e. strategies which do the leader is not required to consider fairness. We first note that for arbitrarily chosen Stackelberg strategies, unfairness can be arbitrarily bad. To see this, imagine the instance in Figure 4.3, with $K \geq 1$. Imagine a Stackelberg strategy where the Stackelberg leader puts his entire share of flow $\alpha$ on the top edge. The induced Nash flow $1-\alpha$ uses the bottom edge only, for any $\alpha$. Thus, the strategy has an unfairness of $\mathcal{B}(\alpha) = \frac{K}{1-\alpha}$, which goes to $+\infty$ with $K$. We note that this strategy is somehow counter-intuitive as it yields higher total cost than the Nash equilibrium (which would correspond to the Stackelberg leader doing nothing) and is therefore far from the leader's objective to minimise total cost. However, the strategy still complies with the definition of a feasible Stackelberg strategy, is therefore valid.

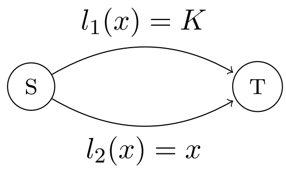



Figure 4.3.: Stackelberg routing can lead to unfairness that is arbitrarily high

Unfairness of arbitrarily chosen Stackelberg strategies is therefore unbounded. To bound the unfairness of Stackelberg routing, we restrict the class of strategies we consider to *opt-restricted* strategies. Opt-restricted strategies are strategies for which the Stackelberg leader does not put more controlled flow on every edge than an optimal flow does, i.e. $g$ is opt-restricted if $g_e \leq o_e$ for all edges. For this subclass of strategies we can show that the unfairness $\mathcal{B}(\alpha)$ of any opt-restricted strategy and any $\alpha$ is bounded by $\mathcal{B}^{opt}$, i.e. the unfairness of an optimal flow. This further implies for linear latencies $\mathcal{B}(\alpha) \leq 2$ and for polynomial latencies $\mathcal{B}(\alpha) \leq d+1$.

**Theorem 2.** *For all $\alpha \in [0,1]$, the unfairness $\mathcal{B}(\alpha)$ of any opt-restricted Stackelberg strategy $g$ is upper-bounded by the unfairness of an optimal allocation for the given instance, i.e. $\mathcal{B}(\alpha) \leq \mathcal{B}^{opt}$.*

*Proof.* Let $g$ be an opt-restricted Stackelberg strategy. Let $h$ be the selfish flow induced by $g$. We have to show that $\mathcal{B}(\alpha) = \frac{L_S^{max}(\alpha)}{L_N(\alpha)} \leq \frac{L_O^{max}}{L_O^{min}} = \mathcal{B}^{opt}$.

We introduce two disjunct subsets of edges regarding $g+h$: $E^{N+} = \{e \in E : h_e > 0\}$ and $E^{N-} = \{e \in E : h_e = 0\}$. We have $E = E^{N-} \cup E^{N+}$. If we have $\alpha = 1$, $E^{N+}$ is empty, as there are no selfish players. As the strategy is opt-restricted, the Stackelberg leader is not allowed to put $g_e > o_e$ on any edge, but as the total flow must add up to 1 (= the total demand), the leader cannot put less than the optimal flow on any edge either. It follows that $\mathcal{B}(1) = \mathcal{B}^{opt}$, as the Stackelberg leader has no other choice than to put $g_e = o_e$ on every edge, therefore inducing the optimum.

If $\alpha < 1$, $E^{N+}$ must be non-empty. We can assume $E^{N-}$ to contain at least one edge with $g_e > 0$, i.e. positive Stackelberg flow. Otherwise $E^{N-}$ is empty or only contains edges that have no flow given $g + h$. In both case, the induced Nash flow uses all edges the Stackelberg leader uses, and we must have $L_S^{max}(\alpha) = L_N(\alpha)$ and thus $\mathcal{B}(\alpha) = 1$. The theorem follows trivially, as $1 \leq \mathcal{B}^{opt}$ is always fulfilled. If $E^{N-}$ is non-empty and contains at least one edge with $g_e > 0$, an edge with highest latency regarding $g + h$ must be in $E^{N-}$ (as the induced Nash flow uses edges with lowest latency). Furthermore, as $g$ is opt-restricted, all edges in $E^{N-}$ must have $g_e + h_e = g_e \leq o_e$. Let $i \in E^{N-}$ refer to an edge with highest latency regarding $g+h$, i.e. an edge with $l_i(g_i + h_i) = l_i(g_i) = L_S^{max}(\alpha)$. As we must have $g_i \leq o_i$ for this edge, $l_i(g_i) \leq l_i(o_i)$ follows, and as $l_i(o_i) \leq L_O^{max}$ this proves $L_S^{max}(\alpha) \leq L_O^{max}$.

To prove $L_N(\alpha) \geq L_O^{min}$, we show that there is at least one edge $j \in E^{N+}$ with $g_j + h_j \geq o_j$. To see this, assume the opposite, i.e. $g_e + h_e < o_e$ for all $e \in E^{N+}$. It follows that $\sum_{e \in E^{N+}}(g_e + h_e) < \sum_{e \in E^{N+}} o_e$. As $g$ is opt-restricted, we further have $\sum_{e \in E^{N-}} g_e \leq \sum_{e \in E^{N-}} o_e$. But as $\sum_{e \in E}(g_e + h_e) = \sum_{e \in E^{N-}} g_e + \sum_{e \in E^{N+}}(g_e + h_e) < \sum_{e \in E^{N-}} o_e + \sum_{e \in E^{N+}} o_e = 1 = r$, this violates the fact that $g+h$ must be a feasible flow for $r = 1$. Thus, for at least edge $j$, $g_j + h_j \geq o_j$ and therefore $l_j(g_j + h_j) \geq l_j(o_j)$. As $j$ is used by the induced Nash flow (which uses edges with lowest latency), and all edges in $E^{N+}$ must have equal and minimal latency given $g+h$, we get $L_N(\alpha) = l_j(g_j + h_j)$. As $L_O^{min}$ is defined by the edge with smallest latency given $o$, there cannot be an edge with smaller latency than $L_O^{min}$, given $o$. Particularly, $l_j(o_j)$ cannot be smaller than $L_O^{min}$. Thus, $l_j(o_j) \geq L_O^{min}$. As we have also shown $l_j(g_j + h_j) \geq l_j(o_j)$, $L_N(\alpha) = l_j(g_j + h_j) \geq l_j(o_j) \geq L_O^{min}$ follows, proving the second part. Thus, as $L_S^{max}(\alpha) \leq L_O^{max}$ and $L_N(\alpha) \geq L_O^{min}$, we have $\mathcal{B}(\alpha) = \frac{L_S^{max}(\alpha)}{L_N(\alpha)} \leq \frac{L_O^{max}}{L_O^{min}} = \mathcal{B}^{opt}$. ∎

We shall quickly mention that it is possible that edge $j$ does not have positive flow regarding optimal flow $o$, thus $o_j = 0$. This does not change anything to our argumentation. If that was the case, we would simply have $L_O^{min} \leq l_j(o_j) = l_j(0)$, and as $g$ is opt-restricted, $g_j = 0$. As we have shown that $g_j + h_j \geq o_j$, i.e. $h_j \geq 0$, $l_j(h_j) \geq l_j(0)$ follows as latency functions are non-decreasing.

It is easy to see that Theorem 2 crucially relies on the fact that $g$ is opt-restricted. Imagine the instance in Figure 4.4. The optimum routes 5/8 over the top edge and 3/8 over the bottom edge. This leads to $\mathcal{B}^{opt} = 7/5$. Imagine now the Stackelberg controls $\alpha = 0.5$ of the flow and puts $g = (0, 1/2)$. This is not an opt-restricted strategy, as $g_2 = 1/2 > 3/8 = o_2$. $g$ enforces $h = (1/2, 0)$, we therefore have $\mathcal{B}(\alpha) = 2 > \mathcal{B}^{opt}$ for this strategy.

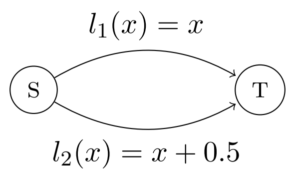



Figure 4.4.: Strategies that are not opt-restricted can have $\mathcal{B}(\alpha) > \mathcal{B}^{opt}$

We conclude this section with a weaker statement that follows from the results just gained.

**Corollary 2.1.** *In parallel-edge networks, any opt-restricted Stackelberg strategy comes with $\mathcal{B}(\alpha) \leq 2$ (for linear latencies) and $\mathcal{B}(\alpha) \leq d+1$ for polynomial latencies up to degree $d \geq 2$, for all values of $\alpha$.*

*Proof.* This immediately follows from previous results. We showed upper bound $\mathcal{B}^{opt} \leq 2$ (for linear latencies) and $\mathcal{B}^{opt} \leq d+1$ (for polynomial latencies) for the optimum in Theorem 1 and Corollary 1.1, respectively. Further, as we showed $\mathcal{B}(\alpha) \leq \mathcal{B}^{opt}$ in Theorem 2 for any opt-restricted strategy, the statement follows. ■

## 4.3. Further analysis for specific opt-restricted strategies

Having shown unfairness bounds for all opt-restricted strategies, we now investigate more specific opt-restricted strategies. We show that 2 and $d+1$ are tight bounds on the unfairness of the LLF strategy in Section 4.3.1. We further show that the unfairness $\mathcal{B}(\alpha)$ of LLF is monotonic and increasing with $\alpha$. We show that the latter result also applies to a greater class of opt-restricted strategies, which we call *monotone opt-restricted strategies*, in Section 4.3.2. Lastly, we compare Scale and LLF regarding their unfairness for a given $\alpha$ in Section 4.3.3.

We begin with a few simple observations regarding the strategies *Scale* and *Largest Latency First (LLF)*. It is easy to see that Scale as well as LLF are opt-restricted strategies. Scale puts $\alpha o_e \leq o_e$ on all edges, and LLF puts at most $g_e = o_e$ on edges. Thus, Theorem 2 and Corollary 2.1 apply. We therefore keep the following in mind:

**Remark.** *In parallel-edge networks, both the Scale and LLF strategy come with $\mathcal{B}(\alpha) \leq \mathcal{B}^{opt} \leq 2$ for linear latencies and $\mathcal{B}(\alpha) \leq \mathcal{B}^{opt} \leq d+1$ for polynomial latencies, for any $\alpha$.*

Further, we continue by showing that LLF and Scale both fail to be fair Stackelberg strategies for any $\alpha > 0$. For any $\alpha > 0$, there exists a fair Stackelberg game $(G, l, \alpha, \beta)$ such that LLF and Scale result in $\mathcal{B}(\alpha) > \beta$, i.e. they fail to satisfy the fairness constraint given in the fair Stackelberg game.

**Remark.** *Neither Scale nor LLF are fair Stackelberg strategies for any $\alpha > 0$.*

*Proof.* To show that LLF and Scale are not fair Stackelberg strategies it is sufficient to show that for every $\alpha > 0$ there exists a $\beta$ such that the respective unfairness $\mathcal{B}(\alpha)$ of LLF and Scale is higher than $\beta$.

Consider once again the Pigou instance depicted in Figure 2.2a. The optimal flow is $o = (0.5, 0.5)$, thus applying Scale leads to $g = \alpha o = (0.5\alpha, 0.5\alpha)$. The induced Nash flow is $h = (0, 1-\alpha)$ for any $\alpha$, thus we have $g+h = (0.5\alpha, 1-0.5\alpha)$ and $\mathcal{B}(\alpha, \text{Scale}) = 1/(1-0.5\alpha)$. If our fair Stackelberg game requires $\beta < (1-0.5\alpha)^{-1}$, Scale fails to be a satisfying strategy.

For LLF, we have $l_1(o_1) = 1$ and $l_2(o_2) = 0.5$, thus $l_2(o_2) \leq l_1(o_1)$. We distinguish two cases: If $\alpha \leq 0.5$, LLF puts $g = (\alpha, 0)$, otherwise we have $g = (o_1, \alpha - o_1) = (0.5, \alpha - 0.5)$. In both cases, the induced Nash flow is $h = (0, 1-\alpha)$ thus the total flow $g+h$ is $(\alpha, 1-\alpha)$ if $\alpha \leq 0.5$ and $(0.5, 0.5)$ otherwise. We have $\mathcal{B}(\alpha, \text{LLF}) = 1/(1-\alpha)$ if $\alpha < 0.5$ and $\mathcal{B}(\alpha, LLF) = 1/0.5 = 2$ if $\alpha \geq 0.5$. If our fair Stackelberg instance demands $\beta < \mathcal{B}(\alpha, LLF)$, LLF fails to be fair. ■

For LLF, we can further refine these results and show that there exists an instance such that LLF results in $\mathcal{B}(\alpha) = 2$ for any $\alpha > 0$ and linear latency functions. Likewise, we can show the same for polynomial latency functions, thus $\mathcal{B}(\alpha) = d + 1$ when applying LLF. Therefore, we now show that LLF fails to be a fair Stackelberg strategy for any $\beta < 2$ (linear latencies) and $\beta < d + 1$ (polynomial latencies), for any $\alpha > 0$.

### 4.3.1. Tight unfairness bound and monotonicity properties of LLF

We already established that $\mathcal{B}(\alpha) \leq \mathcal{B}^{opt} \leq 2$ holds for linear latencies when applying the LLF strategy, as LLF is an opt-restricted strategy. We continue with showing that for any $\alpha > 0$, the unfairness of LLF is not only upper-bounded, but also lower-bounded by 2 (for linear latencies). This makes the bound tight. We do so by presenting an instance for which LLF implements the optimum, for any $\alpha > 0$. Further, the optimum for this instance comes with $\mathcal{B}^{opt} = 2$. Therefore, it follows that for any $\alpha > 0$, LLF comes with an unfairness of $\mathcal{B}(\alpha) = 2$. We further generalise this result to polynomial latencies, showing that $\mathcal{B}(\alpha) = d + 1$ is a tight bound on the unfairness of LLF in that case.

**Theorem 3.** *Assuming the Stackelberg leader applies the LLF strategy to a given Stackelberg instance $(G, l, \alpha)$ with parallel edges, $\alpha > 0$, and linear latency functions, $\mathcal{B}(\alpha) = 2$ is a tight bound on the unfairness of the LLF strategy.*

*Proof.* We have already shown that 2 is an upper bound on $\mathcal{B}(\alpha)$ when applying LLF. Providing instances for which LLF results in $\mathcal{B}(\alpha) = 2$ for any $\alpha > 0$ makes the bound tight.

For $\alpha = 1$, we simply use the Pigou instance in Figure 4.2a. As mentioned earlier, LLF implements $g = (0.5, 0.5)$, thus we have $\mathcal{B}(1) = 2$. For $\alpha \in (0, 1)$, imagine the instance depicted in Figure 4.5a. For any value of $\alpha \in (0, 1)$, the optimum is routing $1 - \alpha$ via the top edge and $\alpha$ via the bottom edge, thus $o = (1 - \alpha, \alpha)$. The latencies of the edges with respect to $o$ are $l_1(o_1) = 1 - \alpha$ and $l_2(o_2) = 2(1 - \alpha)$. This leads to $\mathcal{B}^{opt} = \frac{2(1-\alpha)}{1-\alpha} = 2$. Assuming the Stackelberg leader controls share $\alpha$ of the total flow and uses the LLF strategy, he will put $g = (0, \alpha)$. The induced Nash flow is $h = (1 - \alpha, 0)$, thus we have total flow $g + h = (1 - \alpha, \alpha)$. This equals the optimal flow, thus $\mathcal{B}(\alpha) = 2$ for total flow $(1 - \alpha, \alpha)$. ■

We now generalise this result to polynomial latencies. The instances we provide are generalised versions of the instances used in Theorem 3.

**Corollary 3.1.** *Assuming the Stackelberg leader applies the LLF strategy to a given Stackelberg instance $(G, l, \alpha)$ with parallel edges, $\alpha > 0$, and polynomial latency functions up to degree $d \geq 2$, $\mathcal{B}(\alpha) = d + 1$ is a tight bound on the unfairness of the LLF strategy.*

*Proof.* We recall that we have already shown that $d + 1$ is an upper bound on $\mathcal{B}(\alpha)$ when applying LLF. To show that the bound is tight, we use a similar technique as in Theorem 3.

For $\alpha = 1$, we imagine the instance in Figure 4.2b. LLF implements the optimum, which comes with $\mathcal{B}^{opt} = d + 1$, proving our statement for this case. For $\alpha \in (0, 1)$, we refer to the instance depicted in Figure 4.5b. For any value of $\alpha \in (0, 1)$, the optimum is routing $1 - \alpha$ via the top edge and $\alpha$ via the bottom edge, thus $o = (1 - \alpha, \alpha)$. The costs of the edges regarding $o$ are $l_1(o_1) = (1 - \alpha)^d$ and $l_2(o_2) = (d + 1)(1 - \alpha)^d$, thus $\mathcal{B}^{opt} = d + 1$. Assuming the Stackelberg leader controls share $\alpha \in (0, 1)$ of the total flow and uses the LLF heuristic, he routes $g = (0, \alpha)$. The induced Nash flow is $h = (1 - \alpha, 0)$, thus we have $l_1(g_1 + h_1) = (1 - \alpha)^d$ and $l_2(g_2 + h_2) = (d + 1)(1 - \alpha)^d$, thus $\mathcal{B}(\alpha) = d + 1$ for total flow $(1 - \alpha, \alpha)$. ■


(a) $\mathcal{B}(\alpha) = 2$ (linear latencies) (b) $\mathcal{B}(\alpha) = d + 1$ (polynomial latencies)

Figure 4.5.: Lower bounds on unfairness for LLF

As we have presented instances for which LLF always comes with $\mathcal{B} = 2$ and $\mathcal{B} = d + 1$, for any $\alpha > 0$, the following corollary follows from Theorem 3 and Corollary 3.1, with Figure 4.5 as the given instance.

**Corollary 3.2.** *There is a fair Stackelberg game $(G, l, \alpha, \beta)$ with $\alpha > 0$ and linear latency functions such that LLF fails to be fair Stackelberg strategy for any $\beta < 2$. There is a fair Stackelberg game $(G, l, \alpha, \beta)$ with $\alpha > 0$ and polynomial latency functions up to degree $d$ such that LLF fails to be fair Stackelberg strategy for any $\beta < d + 1$.*

We now continue analysing how the unfairness $\mathcal{B}(\alpha)$ of LLF depends on the share $\alpha$ the Stackelberg leader is allowed to control. We will prove that given the leader applies LLF, $\mathcal{B}(\alpha)$ is monotonically increasing with $\alpha$. We first show a Lemma which comes in handy for the proof.

**Lemma 1.** *Let $(G, l, \alpha)$ with $\alpha > 0$ be a Stackelberg instance in a parallel edge network. Let $\mu \leq m$ be an edge with highest latency among all edges with positive flow regarding the optimum $o$, i.e. $L_O^{max} = l_\mu(o_\mu)$. If the Stackelberg leader applies the LLF strategy, $\mu$ will also have highest latency among all edges with positive flow regarding $g + h$, i.e. $L_S^{max}(\alpha) = l_\mu(g_\mu + h_\mu)$.*

*Proof.* We argue that we can restrict the set of edges we consider to edges with $o_e > 0$, i.e. edges with positive flow regarding the optimum. We recall that $L_O^{max}$ is only defined for edges with $o_e > 0$. Likewise, we recall that $L_S^{max}(\alpha)$ is only defined for edges with $g_e + h_e > 0$. We now argue that no edge $i$ with $o_i = 0$ but $g_i + h_i > 0$ can have higher latency regarding $g + h$ than any edge with $o_i > 0$. As LLF is an opt-restricted strategy, LLF must put $g_e = 0$ whenever $o_e = 0$. The only case where an edge $i$ with $o_i = 0$ comes with positive flow regarding $g + h$ is thus if that edge has induced selfish flow, thus $h_i > 0$ (as $g_i = 0$). But as the selfish flow uses shortest paths in the network, $i$ must have minimal latency regarding $g + h$. As the optimal flow must contain flow, there then must be at least a different edge $j$ with $o_j > 0$ which must have latency as least as high as edge $i$ regarding $g + h$ (otherwise the induced selfish flow does not use shortest paths), and as the Stackelberg leader routes some flow along edges, this edge comes with positive flow given $g + h$.

We therefore restrict our set of edges to edges with $o_e > 0$ and order edges with positive flow in the optimum in ascending order regarding their latency with respect to $o$, i.e. $l_1(o_1) \leq l_2(o_2) \leq \cdots \leq l_\mu(o_\mu)$. We first observe that as the Stackelberg leader saturates edges in descending order regarding their latency to the optimal flow, the same order applies to the flow $g$ put by the Stackelberg leader, i.e. $l_1(g_1) \leq l_2(g_2) \leq \cdots \leq l_\mu(g_\mu)$.

We distinguish two cases: if $h_\mu > 0$, i.e. edge $\mu$ has positive induced selfish flow, $\mu$ must have minimal latency in the network given flow $g + h$. However, as $\mu$ is an edge first tried to be saturated by LLF, all other edges with $g_e > 0$ must have $l_e(g_e) \leq l_\mu(g_\mu)$. Further, all other edges with $g_e + h_e > 0$ cannot have latency smaller or greater than latency of edge $\mu$, as the induced Nash flow uses shortest paths. Therefore all edges with positive flow have equal and minimal latency, proving the statement. If the induced selfish flow does not use $\mu$, latency on edges used by $h$ cannot have higher latency than edge $\mu$, as otherwise the Nash flow could have reduced its costs by using edge $\mu$, a contradiction. Further, other edges with $g_e > 0$ but $h_e = 0$ cannot have latency bigger than $\mu$, due to the ordering of edges regarding $g$. It follows that $\mu$ is an edge with highest latency among all edges with positive flow $g_e + h_e$. ∎

We now show that the both values $L_S^{max}(\alpha)$ and $L_N(\alpha)$ which determine the unfairness ratio $\mathcal{B}(\alpha)$ are monotonic with $\alpha$ when applying LLF.

**Theorem 4.** *Let $(G, l, \alpha)$ be a Stackelberg instance in a parallel-edge network. Applying LLF to $(G, l, \alpha)$ with varying $\alpha$ we have:*

*a) $L_S^{max}(\alpha)$ is monotonically increasing in $\alpha$, and*
*b) $L_N(\alpha)$ is monotonically decreasing in $\alpha$.*

*Proof.* Recall that LLF will assign flow to edges depending on their latencies in an optimum assignment $o$. It will saturate edges one-by-one regarding their latency in the optimum assignment, starting from an edge with highest latency regarding $o$.

We recall that $\mu$ is an edge that determines $L_O^{max}$, i.e. $l_\mu(o_\mu) = L_O^{max}$. $\mu$ has highest latency among all edges with positive flow regarding the optimal flow. By Lemma 1, $\mu$ will also be an edge with highest latency regarding flow $g + h$ when applying LLF. Thus, $\mu$ also determines $L_S^{max}(\alpha)$. If $\alpha$ is big enough to saturate edge $\mu$, we will have $g_\mu = o_\mu$, thus increasing $\alpha$ will not change $L_S^{max}$. Otherwise, increasing $\alpha$ will lead to putting more flow on $\mu$, thus $L_S^{max}$ will only increase or stay the same. This proves a).

For b), assume that we increase the Stackelberg flow from $\alpha$ to $\alpha + \varepsilon$ with $\varepsilon > 0$. For a sufficiently small $\varepsilon$, we can assume w.l.o.g that the additional flow $\varepsilon$ will be assigned by LLF to the same edge $j$. We recall that $h$ denotes the induced selfish flow by LLF before the flow controlled by the Stackelberg leader increases. We distinguish two cases.

If $h_j = 0$, i.e. the additional controlled flow is put on an edge that previously was not used by selfish players, selfish player will continue to not use this edge, as the underlying flow on this edge only increased. But as the total amount of selfish flow reduced from $1-\alpha$ to $1-\varepsilon-\alpha$, we get $L_N(\alpha+\varepsilon) \leq L_N(\alpha)$. If $h_j > 0$, we distinguish two subcases: If $\varepsilon \leq h_j$, i.e. the additional controlled flow does not exceed the selfish flow previously on the edge, nothing changes, as we will only rename previously selfish flow as controlled flow. We will have $L_N(\alpha) = L_N(\alpha+\varepsilon)$. If $\varepsilon > h_j$, the selfish flow now only uses edges 1 to $j-1$, as edge $j$ became too expensive. But the total demand on these edges reduced by $\varepsilon - h_j$, so $L_N(\alpha+\varepsilon) \leq L_N(\alpha)$, proving a). ■

The statement that $\mathcal{B}(\alpha, \mathrm{LLF})$ is monotonically increasing with $\alpha$ then follows from the definition of $\mathcal{B}(\alpha)$.

**Corollary 4.1.** *Let $(G, l, \alpha)$ and $(G, l, \alpha')$ with $\alpha < \alpha'$ and $\alpha' \leq 1$ be two Stackelberg instances of the same underlying parallel-edge network. Let the Stackelberg leader apply LLF in both cases. It then follows that $\mathcal{B}(\alpha) \leq \mathcal{B}(\alpha')$.*

*Proof.* Using Theorem 4, we will have $L_S^{max}(\alpha') \geq L_S^{max}(\alpha)$ and $L_N(\alpha') \leq L_N(\alpha)$. As the degree to which controlled players are put off worse compared to if they would route themselves can be expressed by the ratio of $L_S^{max}(\alpha)$ to $L_N(\alpha)$ for a given instance, we will have $\mathcal{B}(\alpha) = \dfrac{L_S^{max}(\alpha)}{L_N(\alpha)} \leq \dfrac{L_S^{max}(\alpha')}{L_N(\alpha)} \leq \dfrac{L_S^{max}(\alpha')}{L_N(\alpha')} = \mathcal{B}(\alpha')$, proving the statement. ■

The corollary implies the degree to which controlled players are put worse off than selfish players is monotonically increasing with the share of flow the Stackelberg leader controls when using LLF. We now try to generalise this result to more general Stackelberg strategies.

### 4.3.2. Unfairness monotonicity of monotone opt-restricted strategies

Having shown that the unfairness of LLF monotonically increases with $\alpha$, this raises the question whether this property can be generalised. A simple example shows that this does not hold for arbitrary opt-restricted strategies. We recall that a strategy $g$ is opt-restricted if $g_e \leq o_e$ for all edges.

To see this, consider again the Pigou instance (Figure 2.2a) and the following opt-restricted Stackelberg strategy: For $\alpha \leq 0.2$, the Stackelberg leader puts $\alpha$ on the top edge (the edge with constant latency). For $\alpha > 0.2$ and $\alpha \leq 0.5$, the leader puts $\alpha$ on the bottom edge, and for $\alpha > 0.5$, the leader puts flow 0.5 on the top edge and $\alpha - 0.5$ on the bottom edge. In all cases, the induced Nash flow will only use the bottom edge. It follows that the unfairness for $\alpha \leq 0.2$ is bigger than for $0.2 < \alpha \leq 0.5$. For $\alpha = 0.2$, we get $\mathcal{B}(0.2) = 1/0.8 = 1.25$, and for $\alpha = 0.5$ we have $\mathcal{B}(0.5) = 1$.

Arbitrary opt-restricted Stackelberg strategies fail to show monotonicity characteristics regarding their unfairness. However, with a newly defined type of Stackelberg strategies which we call *monotone* opt-restricted strategies, we are able to show that the unfairness is monotonically increasing with $\alpha$ for any monotone opt-restricted strategy.

**Definition 6.** A monotone opt-restricted Stackelberg strategy is an opt-restricted Stackelberg strategy that for any two $\alpha, \alpha'$ with $\alpha' > \alpha$ puts no less flow on every edge when controlling share $\alpha'$ of the flow than when controlling share $\alpha$. Therefore, if $g^{\alpha}$ and $g^{\alpha'}$ denote the leader's flow when being allowed to control $\alpha$ and $\alpha'$, every monotone opt-restricted strategies has $g_e^{\alpha} \leq g_e^{\alpha'}$ on all edges.

Easy to see, the strategy we presented above is not monotone, as it puts positive flow on the top edge for $\alpha \leq 0.2$, but no flow on this edge for $\alpha > 0.2$ but $\alpha \leq 0.5$. We further note that both LLF and Scale are monotone opt-restricted strategies. We now generalise the monotonicity property of LLF to all monotone opt-restricted strategies.

**Theorem 5.** *Let $(G, l, \alpha)$ be a Stackelberg instance in a parallel-edge network. Applying the same monotone opt-restricted strategy to $(G, l, \alpha)$ with varying $\alpha$, we have:*

*a) $L_S^{max}(\alpha)$ is monotonically increasing in $\alpha$, and*
*b) $L_N(\alpha)$ is monotonically decreasing in $\alpha$.*

*Proof.* We model an increase of controlled flow as if we would increase the controlled flow edge-by-edge, one at the time. Our analysis reduces to analysing the effect of a change in the controlled flow on one edge only, as this can then be repeated for all edges, one by one.

For a given $\alpha$ and a given monotone opt-restricted strategy, let $i$ be an edge with $g_i + h_i > 0$. Assume the controlled flow on $i$ changes by a sufficiently small amount $\varepsilon$. The controlled flow on all other edges does not change. Let $g_i'$ and $h_i'$ denote the resulting controlled flow and induced selfish flow on edge $i$ after the change. We have $g_i' = g_i + \varepsilon$. By Definition 6, $\varepsilon$ must be nonnegative, as the strategy is monotone and the controlled flow is therefore not allowed to decrease on any edge.

We distinguish two cases: If $g_i' < g_i + h_i$ we observe that $\varepsilon < h_i$, as $g_i' = g_i + \varepsilon$. As our strategy is monotone and demands $g_i' \geq g_i$, $\varepsilon \geq 0$ follows and therefore $h_i > 0$. Further, it must be the case that $h_i' > 0$, as some selfish flow continues to use edge $i$. In this case, both $L_S^{max}$ and $L_N$ do not change, as the flow on edge $i$ was simply relabelled from selfish to controlled flow, without changing the overall flow on $i$ and any other edge. Therefore $L_S^{max}(\alpha) = L_S^{max}(\alpha + \varepsilon)$ and $L_N(\alpha) = L_N(\alpha + \varepsilon)$, proving both parts.

If $g_i' \geq g_i + h_i$, two cases could have occurred: $i$ already did not have induced selfish flow before the increase in controlled flow, i.e. $h_i = 0$, or $i$ had induced selfish flow before the increase but no longer has induced selfish flow after the increase. In the latter, we must have $\varepsilon \geq h_i$, i.e. the additional controlled flow is at least as high as the induced selfish flow that was on the edge, as otherwise at least some selfish flow should continue to use the edge (see the first case). Trivially $\varepsilon \geq h_i$ also holds if $h_i = 0$, as our strategy is monotone and $\varepsilon$ therefore non-negative. In both cases, we get $l_i(g_i + h_i) \leq l_i(g_i + \varepsilon) = l_i(g_i' + h_i')$. As $L_S^{max}$ is determined by an edge with positive controlled flow, the controlled flow on no edge is allowed to decrease and $L_S^{max}(\alpha) \geq l_i(g_i + h_i)$ and $L_S^{max}(\alpha + \varepsilon) \geq l_i(g_i' + h_i')$, we get $L_S^{max}(\alpha) \leq L_S^{max}(\alpha + \varepsilon)$, proving a). For b), we see that the total amount of selfish flow reduces by $\varepsilon$, as this is put on edge $i$ which has no induced selfish flow after the increase in controlled flow. $L_N(\alpha) \geq L_N(\alpha + \varepsilon)$ then simply follows because the total selfish flow decreases and latency functions are non-decreasing. ■

Similar to LLF, we use the theorem to get the following corollary.

**Corollary 5.1.** *Let $(G, l, \alpha)$ and $(G, l, \alpha')$ with $\alpha < \alpha'$ and $\alpha' \leq 1$ be two Stackelberg instances of the same underlying parallel edge network. Let the Stackelberg leader apply the same monotone opt-restricted strategy in both cases. It holds that $\mathcal{B}(\alpha) \leq \mathcal{B}(\alpha')$.*

We can therefore conclude that the degree to which controlled players are put worse off than self-routing players is monotonically increasing with the share of flow the Stackelberg leader controls for any monotone opt-restricted strategy. We further observe that Scale is a monotone opt-restricted strategy. This implies the following corollary.

**Corollary 5.2.** *Let $(G, l, \alpha)$ and $(G, l, \alpha')$ with $\alpha < \alpha'$ and $\alpha' \leq 1$ be two Stackelberg instances of the same underlying parallel-edge network. Let the Stackelberg leader apply Scale in both cases. It holds that $\mathcal{B}(\alpha) \leq \mathcal{B}(\alpha')$.*

Having shown that the unfairness of both LLF and Scale increases with $\alpha$, we now investigate how the unfairness of Scale and LLF compare to each other for a given $\alpha$.

### 4.3.3. Comparing unfairness of Scale and LLF

Aim of this section is to get an understanding of how the unfairness of Scale and LLF for a given $\alpha$ compare to each other. We first give a simple Lemma for Scale that is similar to Lemma 1 for LLF.

**Lemma 2.** *Let $(G, l, \alpha)$ be a Stackelberg instance in a parallel-edge network. Let $\mu \leq m$ denote the index of an edge with highest latency and positive flow regarding optimum flow $o$, i.e. $L_O^{max} = l_\mu(o_\mu)$. If the Stackelberg leader applies the Scale strategy, $\mu$ will also have highest latency among all edges with positive flow regarding $g + h$, i.e. $L_S^{max}(\alpha) = l_\mu(g_\mu + h_\mu)$.*

*Proof.* Due to a similar reason as in Lemma 1, we can restrict our set of edges to edges with $o_e > 0$. We order edges with $o_e > 0$ in ascending order regarding their latency with respect to $o$, i.e. $l_1(o_1) \leq l_2(o_2) \leq \cdots \leq l_\mu(o_\mu)$. We first observe that as Scale puts $g_e = \alpha o_e$ on all edges, the same order applies to the flow $g$ put by the Stackelberg leader, i.e. $l_1(g_1) \leq l_2(g_2) \leq \cdots \leq l_\mu(g_\mu)$.

We distinguish two cases: if $h_\mu > 0$, i.e. edge $\mu$ has positive induced selfish flow, $\mu$ must have minimal latency in the network given flow $g + h$. But as $\mu$ comes with the highest underlying latency of controlled flow, it follows that all edges with positive flow regarding $g + h$ must have equal and minimal latency. The only edges which might have higher latency are edges without flow, but these edges are not considered in $L_S^{max}(\alpha)$. The statement follows.

If $h_\mu = 0$, the induced Nash flow does not use edge $\mu$. Latency on edge $\mu$ then is $l_\mu(g_\mu) = l_\mu(\alpha o_\mu)$. Edges used by $h$ cannot have higher latency than $l_\mu(\alpha o_\mu)$, as otherwise the Nash flow could have reduced its costs by using edge $\mu$, a contradiction. Further, edges with positive controlled flow but no induced Nash flow cannot have latency higher than $l_\mu(\alpha o_\mu)$ as for these edges $g_e + h_e = g_e$ and $l_e(g_e) \leq l_\mu(g_\mu)$ due to the ordering of edges regarding $g$ for Scale. It again follows that $\mu$ is an edge with highest latency among all edges with positive flow regarding $g + h$. ■

We now state the conjecture that in parallel-edge networks, the unfairness resulting from applying Scale cannot be worse than the unfairness resulting from applying LLF to a given Stackelberg instance $(G, l, \alpha)$.

**Conjecture 1.** *Let $(G, l, \alpha)$ be a Stackelberg instance with parallel edges where the Stackelberg leader controls fraction $\alpha$ of the flow. We suppose $\mathcal{B}(\alpha, Scale) \leq \mathcal{B}(\alpha, LLF)$.*

Let $L_N(\alpha, \text{LLF})$ and $L_N(\alpha, \text{Scale})$ denote the minimal latency for flow $g^{LLF} + h^{LLF}$ and $g^{Scale} + h^{Scale}$, respectively. Likewise, let $L_S^{max}(\alpha, \text{LLF})$ and $L_S^{max}(\alpha, \text{Scale})$ denote the highest latency experienced on any edge with positive flow with regard to $g^{LLF} + h^{LLF}$ and $g^{Scale} + h^{Scale}$, respectively. In order to make this conjecture a theorem, we would have to show for a given $\alpha$ that

$$\mathcal{B}(\alpha, \text{Scale}) = \frac{L_S^{max}(\alpha, \text{Scale})}{L_N(\alpha, \text{Scale})} \leq \frac{L_S^{max}(\alpha, \text{LLF})}{L_N(\alpha, \text{LLF})} = \mathcal{B}(\alpha, \text{LLF})$$

We first note that in case $L_S^{max}(\alpha, \text{Scale}) = L_N(\alpha, \text{Scale})$, it follows that all edges with positive flow with regard to $g^{Scale} + h^{Scale}$ have equal and minimal latency when applying Scale. It then follows that no controlled player can decrease his own latency when routing himself, and as $L_S^{max}(\alpha, \text{Scale}) = L_N(\alpha, \text{Scale})$, $\frac{L_S^{max}(\alpha,\text{Scale})}{L_N(\alpha,\text{Scale})} = 1 \leq \frac{L_S^{max}(\alpha,\text{LLF})}{L_N(\alpha,\text{LLF})}$ holds trivially.

In case $L_S^{max}(\alpha, \text{Scale}) > L_N(\alpha, \text{Scale})$, it would be sufficient to show that $L_S^{max}(\alpha, \text{LLF}) \geq L_S^{max}(\alpha, \text{Scale})$ and that $L_N(\alpha, \text{LLF}) \leq L_N(\alpha, \text{Scale})$. We are able to prove the former, but fail to prove the latter: Let $\mu$ denote an edge with highest latency and positive flow regarding optimal flow $o$. Applying Lemma 2 we can conclude that edge $\mu$ also has highest latency regarding $g^{Scale} + h^{Scale}$, thus $L_S^{max}(\alpha) = l_\mu(g_\mu^{Scale} + h_\mu^{Scale})$. As $L_S^{max}(\alpha, \text{Scale}) > L_N(\alpha, \text{Scale})$, it follows that $h_\mu^{Scale} = 0$. Using Lemma 1, $\mu$ further has highest latency when applying LLF. Thus, $\mu$ determines $L_S^{max}(\alpha)$ both for Scale and LLF. Showing that $L_S^{max}(\alpha, \text{LLF}) \geq L_S^{max}(\alpha, \text{Scale})$ thus reduces to showing that $l_\mu(g_\mu^{LLF} + h_\mu^{LLF}) \geq l_\mu(g_\mu^{Scale}) = l_\mu(g_\mu^{Scale} + h_\mu^{Scale})$ as $h_\mu^{Scale} = 0$. If $\alpha \geq o_\mu$, the Stackelberg leader saturates edge $\mu$ with LLF, thus we have $g_\mu^{LLF} = o_\mu \geq \alpha o_\mu = g_\mu^{Scale}$. As $h_\mu^{LLF} \geq 0$ and latency functions are non-decreasing, it then follows that $l_\mu(g_\mu^{LLF} + h_\mu^{LLF}) \geq l_\mu(g_\mu^{LLF}) = l_\mu(o_\mu) \geq l_\mu(\alpha o_\mu) = l_\mu(g_\mu^{Scale})$. If LLF cannot saturate $\mu$, we have $g_\mu^{LLF} = \alpha$, thus $l_\mu(g_\mu^{LLF} + h_\mu^{LLF}) \geq l_\mu(g_\mu^{LLF}) = l_\mu(\alpha) \geq l_\mu(\alpha o_\mu)$ as $o_\mu \leq r = 1$. In both cases, we will have $l_\mu(g_\mu^{LLF} + h_\mu^{LLF}) \geq l_\mu(g_\mu^{LLF}) \geq l_\mu(g_\mu^{Scale}) = l_\mu(g_\mu^{Scale} + h_\mu^{Scale})$, and due to Lemma 1 and Lemma 2 this proves $L_S^{max}(\alpha, \text{LLF}) \geq L_S^{max}(\alpha, \text{Scale})$.

It is yet missing to show that $L_N(\alpha, \text{Scale}) \geq L_N(\alpha, \text{LLF})$. Intuitively, it makes sense to think that the latency of the selfish flow induced by LLF cannot be bigger than the latency of the selfish flow induced by Scale, as LLF tends to allocate the same amount $\alpha$ of flow more unequally on edges than Scale. This is as Scale puts flow proportionally on all edges, whereas LLF tries to completely saturates edges with high latency first. However, we fail to find a proof for $L_N(\alpha, \text{Scale}) \geq L_N(\alpha, \text{LLF})$. The statement is therefore only a conjecture.

[Rou04] showed that for single-commodity, parallel edge networks with linear latencies, LLF is a better performing strategy than Scale. Therefore, for a given $\alpha$, Scale cannot yield a lower Price of Anarchy than LLF in this kind of network. Thus, $PoA(\alpha, LLF) \leq PoA(\alpha, \text{Scale})$. If Conjecture 1 was true, i.e. $\mathcal{B}(\alpha, \text{Scale}) \leq \mathcal{B}(\alpha, \text{LLF})$ holds, we might therefore think that there is a trade-off between efficiency and fairness of Stackelberg strategies when we compare them to each other, at least for Scale and LLF. We are inclined to think that a lower Price of Anarchy comes with a higher level of unfairness and a lower level of unfairness comes with a higher Price of Anarchy. However, a simple example shows that this relation between LLF and Scale certainly does not hold in general networks, even for linear latencies.

The instance we use is Braess's paradox in Figure 2.3. We recall that due to edge $u - v$, this instance no longer has only parallel edges. We are therefore temporarily out of the framework of parallel-edge networks. Thus, the unfairness measure $\mathcal{B}(\alpha)$ here corresponds to $s - t$ paths (which consist of one or multiple edges), instead of individual edges. We now show that for any $\alpha$, LLF is at least as inefficient *and* unfair as Scale.

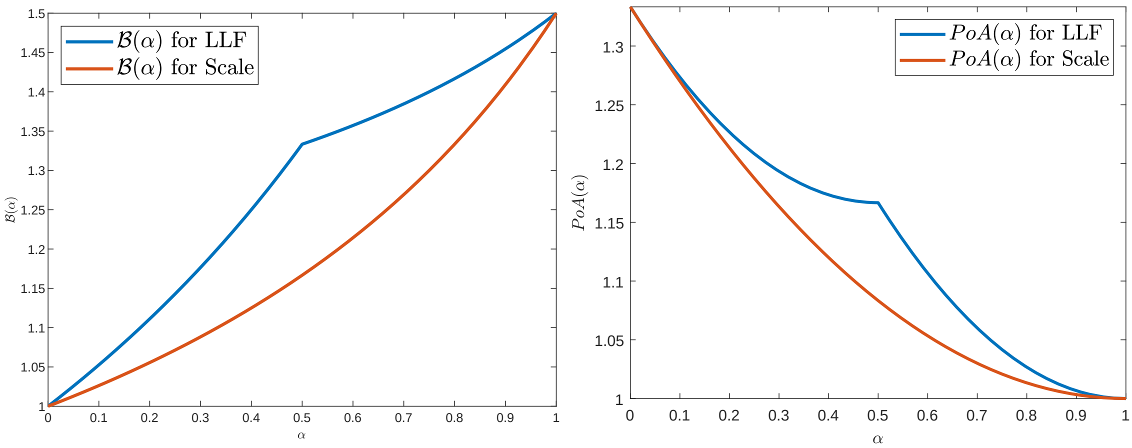
Assuming $r = 1$, the optimum routes half of the traffic over $s - u - t$ and half of the traffic over $s - v - t$. If players could route themselves, they would use the path $s - u - v - t$, experiencing individual latency of 1. Thus, the optimum comes with $\mathcal{B}^{opt} = \frac{0.5+1}{0.5+0+0.5} = 1.5$. We now assume that the Stackelberg leader can control share $\alpha$ of the flow and investigate the two common strategies LLF and Scale. As LLF is originally only defined for parallel-edge networks, we make use of a generalised version of LLF presented by [CS07] which works on this graph.

For any $\alpha$, $g^{Scale}$ will route flow $\frac{\alpha}{2}$ along $s - u - t$ and $\frac{\alpha}{2}$ along $s - v - t$, inducing $h^{Scale}$ with flow $1 - \alpha$ along path $s - u - v - t$. Thus, controlled players experience personal costs of $(1 - \alpha + \frac{\alpha}{2}) + 1 = 2 - \frac{\alpha}{2}$, whereas the selfish players only experience costs of $(1 - \alpha + \frac{\alpha}{2}) + 0 + (1 - \alpha + \frac{\alpha}{2}) = 2 - \alpha$. We thus have $\mathcal{B}(\alpha, \text{Scale}) = \frac{2-\alpha/2}{2-\alpha}$.

For LLF, we distinguish two cases: if $\alpha \leq 0.5$, the Stackelberg leader either puts $\alpha$ on path $s - u - t$ or he puts $\alpha$ on path $s - v - t$. In both cases, the induced Nash flow $1 - \alpha$ uses path $s - u - v - t$. Controlled players thus have costs of $(1 - \alpha + \alpha) + 1 = 2$ whereas selfish players only experience costs $(1 - \alpha + \alpha) + 1 - \alpha = 2 - \alpha$. We thus have $\mathcal{B}(\alpha, \text{LLF}) = \frac{2}{2-\alpha}$ if $\alpha \leq 0.5$. If $\alpha > 0.5$, LLF either puts flow 0.5 over $s - u - t$ and $\alpha - 0.5$ over $s - v - t$ or 0.5 over $s - v - t$ and $\alpha - 0.5$ over $s - u - t$. In both cases the induced Nash flow is again $1 - \alpha$ on $s - u - v - t$. Controlled players on the saturated path (with Stackelberg flow 0.5) experience highest latency, which is $1.5 - \alpha + 1 = 2.5 - \alpha$, controlled players on the path with Stackelberg flow $\alpha - 0.5$ have latency 1.5 and selfish players have latency $2 - \alpha$. We thus have $\mathcal{B}(\alpha, \text{LLF}) = \frac{2.5-\alpha}{2-\alpha}$.

Furthermore, we have $C(o) = 1.5$, $C(g^{Scale} + h^{Scale}) = 2(1 - \frac{\alpha}{2})^2 + \alpha$ and $C(g^{LLF} + h^{LLF}) = 1 + \alpha + (1 - \alpha)^2$ for $\alpha \leq 0.5$ and $(1.5 - \alpha)^2 + \alpha + 0.25$ otherwise. Thus, Scale has $PoA(\alpha, \text{Scale}) = \frac{4}{3} - \frac{1}{3}(2\alpha - \alpha^2)$, whereas $PoA(\alpha, \text{LLF}) = \frac{4}{3} - \frac{1}{3}(2\alpha - 2\alpha^2)$ for $\alpha \leq 0.5$ and $PoA(\alpha, \text{LLF}) = \frac{4}{3} - (\frac{4\alpha}{3} - \frac{2\alpha^2}{3} - \frac{1}{3})$ otherwise.

Figure 4.6 gives a graphical illustration of the unfairness and efficiency of Scale and LLF is. Easy to see, LLF both has a weakly higher level of unfairness and a weakly higher Price of Anarchy, for all $\alpha$. We therefore have for all $\alpha$ that $\mathcal{B}(\alpha, \text{Scale}) \leq \mathcal{B}(\alpha, \text{LLF})$ *and* $PoA(\alpha, \text{Scale}) \leq PoA(\alpha, \text{LLF})$. Scale not only results in a smaller unfairness than LLF, but does so without sacrificing efficiency gains. However, we note that not *all* players who are put off worse by the Stackelberg leader experience lower individual costs for Scale than they do for LLF. Assuming $\alpha > 0.5$, controlled players on the not saturated path in LLF have an unfairness degree of $\frac{1.5}{2-\alpha}$ whereas all players in Scale have an unfairness degree of $\frac{2-\alpha/2}{2-\alpha}$. Therefore, the players on the path not completely saturated by the Stackelberg leader for LLF are relatively less worse off than all players in Scale. However, as the unfairness measure takes the maximum ratio of highest to lowest latency and players on the saturated path in LLF are put off worse than all controlled players for Scale, $\mathcal{B}(\alpha)$ is higher for LLF than for Scale. It seems that LLF distributes unfairness less equal than Scale.




(a) Unfairness for LLF and Scale

(b) Price of Anarchy for LLF and Scale

Figure 4.6.: Braess's paradox showing that Scale can dominate LLF both regarding performance and fairness

We therefore conclude this section with two open questions: the first is whether Scale always results in an unfairness which is at most the unfairness of LLF. We were not able to prove this conjecture, but also fail to give a counterexample. Interestingly, even our example in general networks which showed that Scale might be less costly than LLF is still inline with the conjecture that Scale is less unfair than LLF. It is therefore conceivable that Conjecture 1 not only applies to parallel-edge networks, but also to general networks. Further, Braess's paradox showed us that there is no clear trade-off between unfairness and efficiency of strategies in general networks. The question remains whether this applies to parallel-edge networks as well or if there is indeed a trade-off, at least for specific strategies. If we were able to prove Conjecture 1, we would be able to confirm a trade-off between unfairness and efficiency of LLF and Scale for parallel-edge networks. These questions therefore remain open and we come back to them in Chapter 5.

## 4.4. Lower bounds on the Price of Anarchy for all fair Stackelberg games

We shift our focus from standard Stackelberg games to fair Stackelberg games. We recall that fair Stackelberg games come with an additional parameter $\beta \geq 1$ which restricts the level of unfairness allowed for a given Stackelberg strategy. Any fair Stackelberg strategy must comply with $\beta$, i.e. any fair Stackelberg strategy must have $\mathcal{B}(\alpha) \leq \beta$.

We establish lower bounds on the Price of Anarchy for *any* fair Stackelberg strategy in parallel-edge networks. The lower bound both depends on the share $\alpha$ the Stackelberg leader is allowed to control and the degree $\beta$ to which he is allowed to put players worse off. We show such bounds both for linear (Section 4.4.1) and polynomial latency functions (Section 4.4.2). We do so by presenting two simple instances, each of them depends on one of the two parameters $\alpha$ and $\beta$ of a fair Stackelberg game $(G, l, \alpha, \beta)$. Both instances have in common that any fair Stackelberg strategy cannot give better performance than the respective Nash equilibrium. Therefore, the maximum of the two resulting Prices of Anarchy for the two instances yields a lower bound on the Price of Anarchy of any fair Stackelberg strategy for a given fair Stackelberg game $(G, l, \alpha, \beta)$.

### 4.4.1. Price of Anarchy bounds for linear latencies

In this section we show that given a fair Stackelberg game $(G, l, \alpha, \beta)$ with linear latency functions and $\beta \leq 2$, the Price of Anarchy of any fair Stackelberg strategy will be at least $PoA(\alpha, \beta) = \max\left\{\frac{4}{3+\alpha}, \left(\beta - \frac{\beta^2}{4}\right)^{-1}\right\}$. This bound holds for all Stackelberg strategies that comply with $\alpha$ and $\beta$. As mentioned, we show this bound by presenting two instances which have a Price of Anarchy of at least $\frac{4}{3+\alpha}$ and $(\beta - \frac{\beta^2}{4})^{-1}$, irrespective of the chosen fair Stackelberg strategy.

The first instance depends on $\alpha$ and is pictured in Figure 4.7a. We now show that any fair Stackelberg strategy for this instance leads to a Price of Anarchy of at least $PoA(\alpha) = \frac{4}{3+\alpha}$.

For any $\alpha < 1$, the Wardrop equilibrium $n$ is $n = (\alpha, 1-\alpha)$, thus routing flow $\alpha$ over the top edge and flow $1-\alpha$ over the bottom edge. The optimum is $o = (\frac{1+\alpha}{2}, \frac{1-\alpha}{2})$, therefore results in $\mathcal{B}^{opt} = 2$. For any $\alpha < 1$, the Stackelberg leader does not control sufficiently flow to route along the top edge to make a positive difference to the Wardrop equilibrium. In fact, no matter how he allocates his share of flow, he always induces a flow that is at least as bad as the Wardrop equilibrium. We note that for arbitrary fair Stackelberg strategies, depending on the value of $\beta$ it is possible that the strategy of the leader leads to higher total cost than the Wardrop equilibrium (if $\beta$ allows outcomes other than the Wardrop equilibrium). However, the outcomes the Stackelberg leader can achieve with his fair strategy will always have total cost at least as high as $n$, but never better. As the bound we establish is a lower bound, we can w.l.o.g assume that the Stackelberg leader implements the Wardrop equilibrium instead of implementing allocations with even higher cost (as the lower bound would also hold in that case). Thus, the cost of any Stackelberg strategy $g$ and induced Nash flow $h$ will at least as high as the cost of Nash equilibrium $n$, thus $C(g+h) \geq C(n) = \alpha(1-\alpha) + (1-\alpha)^2 = 1-\alpha$. The optimum comes with cost $C(o) = (1-\alpha)(\frac{1+\alpha}{2}) + (\frac{1-\alpha}{2})^2 = \frac{1}{4}(1-\alpha)(3+\alpha)$, therefore $\frac{C(g+h)}{C(o)} \geq \frac{C(n)}{C(o)} \geq \frac{4}{3+\alpha}$. For $\alpha = 1$, we will have $C(g+h) \geq C(n) = C(o) = 0$, and we shall define $\frac{0}{0} = 1$. In both cases, we get $PoA(\alpha) = \frac{C(g+h)}{C(o)} \geq \frac{4}{3+\alpha}$.

What is interesting about this instance is that the Price of Anarchy matches the $\frac{4}{3+\alpha}$ performance guarantee of the Largest Latency First strategy for linear latencies shown by [Rou04]. As this lower bound on the inefficiency of any fair Stackelberg strategy matches the upper bound on the LLF heuristic, this might give us an idea of the efficiency of LLF in cases where LLF is fair. However, we shall keep in mind that LLF does not always guarantee to be a fair Stackelberg strategy (see Theorem 3).


(a) Instance varying with $\alpha$

(b) Instance varying with $\beta$

Figure 4.7.: Instances varying with one parameter that holds for all values of the other parameter

The second instance is seen in Figure 4.7b. The instance is taken from [CKS11] with slightly different notation. [CKS11] used this instance to analyse the Price of Anarchy and Price of Stability of $\varepsilon$-approximate Nash equilibria. We will explain $\varepsilon$-approximate Nash equilibria and their connection to fair Stackelberg routing in greater detail in Section 4.4.2. For this instance, we demand $\beta \leq 2 = \mathcal{B}^{opt}$, thus the upper bound on the unfairness of the optimal flow. For any $\beta \geq 1$ (which is trivially given), the Nash equilibrium is $n = (0, 1)$. As we are looking at fair Stackelberg games, the Stackelberg leader must consider $\beta$ in his decision. For the given instance in Figure 4.7b (with $\beta$ as constant cost), the Stackelberg leader is then forced to implement the Nash equilibrium, regardless the strategy he uses and how much flow he controls. As $\beta \geq 1$ and $x_2 \leq r = 1$, the Nash flow is $n = (n_1, n_2) = (0, 1)$. The optimal flow will be $o = (1 - \frac{\beta}{2}, \frac{\beta}{2})$. Assume the Stackelberg leader routes flow $\varepsilon$ along the top edge (may $\varepsilon$ be arbitrarily small). This flow has individual latency of $\beta$. The remaining flow (including the induced Nash flow) will take the bottom edge, having costs $1 - \varepsilon$. Thus, every player on the top edge experiences an unfairness of $\frac{\beta}{1-\varepsilon} > \beta$, higher than the allowed threshold. Thus, we conclude that for fair Stackelberg games, the Stackelberg leader does not have a choice than to route his entire share along the bottom edge, regardless the share he controls and the strategy he uses. This implements the Nash equilibrium, as all selfish users will also use the bottom edge. Thus, $PoA(\beta) = \left(\beta(1 - \frac{\beta}{4})\right)^{-1}$, as we have $C(g + h) = C(n) = 1$ and $C(o) = \beta(1 - \frac{\beta}{2}) + (\frac{\beta}{2})^2 = \beta(1 - \frac{\beta}{4})$.

If we demand all players to experience equal latency (i.e. $\beta = 1$), this lower bound on the Price of Anarchy of any Stackelberg strategy becomes the well-known upper bound 4/3 on the Price of Anarchy of selfish routing for linear latency functions established in [RT02]. It decreases to 1 as $\beta$ increases to 2.

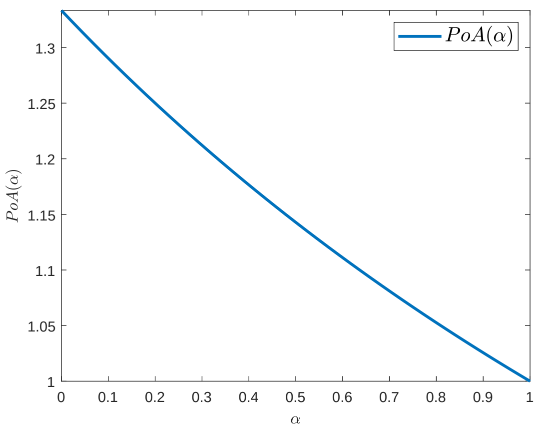
We continue with our main theorem in this section.

**Theorem 6.** *For any fair Stackelberg game $(G, l, \alpha, \beta)$ with linear latency functions, $\alpha \in [0, 1]$ and $\beta \in [1, 2]$ the Price of Anarchy of any fair Stackelberg strategy is at least*

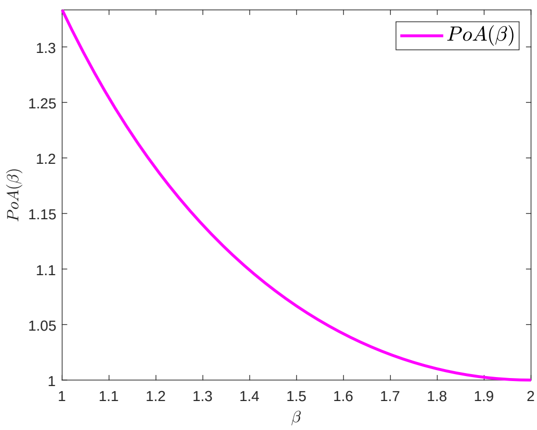
$$PoA(\alpha, \beta) = \max\left\{\frac{4}{3+\alpha}, \left(\beta - \frac{\beta^2}{4}\right)^{-1}\right\}$$

*Proof.* For any fair given Stackelberg strategy, thus any Stackelberg strategy which complies with the given values $\alpha$ and $\beta$ of a fair Stackelberg instance $(G, l, \alpha, \beta)$, the instance in Figure 4.7a gives us a Price of Anarchy of at least $\frac{4}{3+\alpha}$. Similarly, any fair Stackelberg strategy leads to a Price of Anarchy of $(\beta(1-\frac{\beta}{4}))^{-1}$ for the instance in Figure 4.7b. For a given $(\alpha, \beta)$ pair, the Price of Anarchy of any fair Stackelberg strategy will therefore be at least the maximum of the two instances' Price of Anarchy. ∎

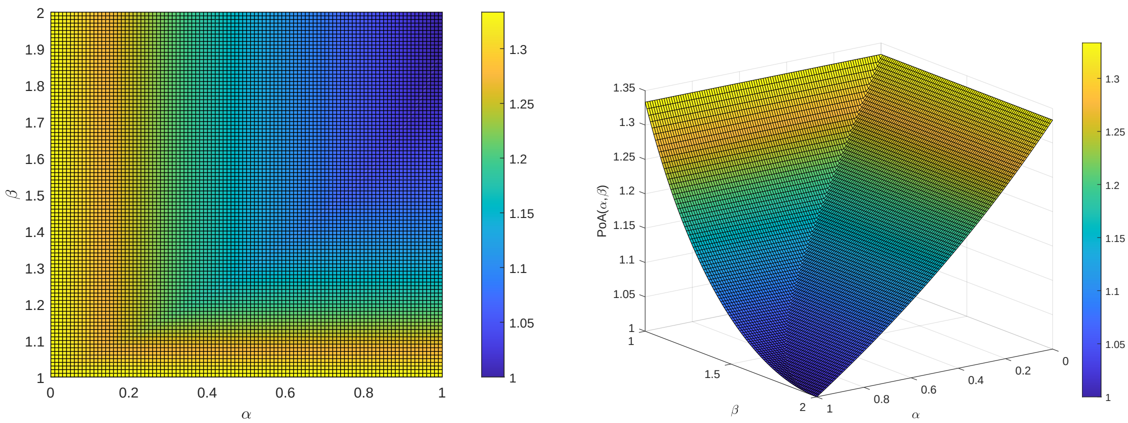
A graphical illustration of this bound is given in Figure 4.8. Figure 4.8a illustrates the function $\frac{4}{3+\alpha}$. Figure 4.8b illustrates the function $(\beta - \frac{\beta^2}{4})^{-1}$. Figures 4.8c illustrate the maximum of the two functions. For this plot, yellow indicates a high Price of Anarchy and blue indicates a low Price of Anarchy. We can see that the Price of Anarchy is highest for low values of $\alpha$ and $\beta$, respectively. This is intuitively as it means the Stackelberg leader does either not have a lot of power to improve the performance of selfish routing or is not allowed to improve it much, as it would assign latencies too unequally to players.




(a) Price of Anarchy for Figure 4.7a




(b) Price of Anarchy for Figure 4.7b




(c) Combined lower bound on Price of Anarchy for any fair Stackelberg strategy

Figure 4.8.: Graphical illustration of Price of Anarchy used in Theorem 6

### 4.4.2. Price of Anarchy bounds for polynomial latencies

We extend our results established in Section 4.4.1 to fair Stackelberg games with polynomial latency functions. For these instances, we show improved, thus higher, lower bounds.

**Theorem 7.** *For any fair Stackelberg game $(G, l, \alpha, \beta)$ with polynomial latency functions up to degree $d \geq 2$ and $\alpha \in [0, 1]$ and $\beta \in [1, d+1]$, the Price of Anarchy of any fair Stackelberg strategy is at least*

$$PoA(\alpha, \beta) = \max \left\{ \left[(\alpha - 1)d(d+1)^{-(d+1)/d} + 1\right]^{-1}, \left[\beta\left(1 - \frac{d}{d+1}\left(\frac{\beta}{d+1}\right)^{1/d}\right)\right]^{-1} \right\}$$

*Proof.* We take the instances presented in Figure 4.7 and adapt them to polynomial latencies. The respective instances are seen in Figure 4.9. For the instance dependent on $\alpha$, seen in Figure 4.9a, we have $n = (a, 1-\alpha)$ and $o = \left(1 - \frac{1-\alpha}{(d+1)^{1/d}}, \frac{1-\alpha}{(d+1)^{1/d}}\right)$. If the Stackelberg leader controls $\alpha$ of the flow, the cost of the resulting flow $g + h$ will be at least as high as the cost of the Nash equilibrium $n$, no matter the strategy chosen. This is because he does not control a sufficient share to induce better performance. We thus have $C(g+h) \geq C(n) = (1-\alpha)^d\alpha + (1-\alpha)^{d+1} = (1-\alpha)^d$ for any strategy. Furthermore, $C(o) = (1-\alpha)^d\left(1 - \frac{1-\alpha}{(d+1)^{1/d}}\right) + \left(\frac{1-\alpha}{(d+1)^{1/d}}\right)^{d+1}$, which leads to

$$PoA(\alpha) = \frac{C(g+h)}{C(o)} \geq \frac{1}{(\alpha-1)d(d+1)^{-(d+1)/d} + 1} \tag{4.4}$$

For the instance dependent on parameter $\beta$, seen in Figure 4.9b, we restrict $\beta$ to $\beta \in [1, d+1]$, with degree $d \geq 2$. The Nash flow of this instance is $n = (n_1, n_2) = (0, 1)$ for any $\beta \in [1, d+1]$. The optimal flow is $o = (o_1, o_2) = (1 - (\frac{\beta}{d+1})^{1/d}, (\frac{\beta}{d+1})^{1/d})$. Similar to Figure 4.7b, the Stackelberg leader cannot route any flow along the top edge if he has to comply with $\beta$. Thus the social cost of *any* fair Stackelberg strategy $g$ and induced Nash flow $h$ will be $C(g+h) = C(n) = 1$, for any $\alpha$. The optimal flow has cost $C(o) = \left(\left(\frac{\beta}{d+1}\right)^{1/d}\right)^d \cdot \left(\frac{\beta}{d+1}\right)^{1/d} + \beta\left(1 - \left(\frac{\beta}{d+1}\right)^{1/d}\right) = \beta\left(1 - \frac{d}{d+1}\left(\frac{\beta}{d+1}\right)^{1/d}\right)$, hence the Price of Anarchy for any fair Stackelberg strategy for this instance equals

$$PoA(\beta) = \frac{C(g+h)}{C(o)} = \left(\beta\left(1 - \frac{d}{d+1}\left(\frac{\beta}{d+1}\right)^{1/d}\right)\right)^{-1} \tag{4.5}$$

The maximum of Equation 4.4 and 4.5 gives us the lower bound desired. ■


(a) Polynomial instance varying with $\alpha$ (b) Polynomial instance varying with $\beta$

Figure 4.9.: Instances varying with $\alpha$ and $\beta$ for polynomial latency functions

Theorem 6 and Theorem 7 reveal an interesting link to $\varepsilon$-approximate Nash equilibria. $\varepsilon$-approximate equilibria loosen the condition of Nash equilibria that players use shortest path in the network. $\varepsilon$-approximate equilibria allow for a certain degree of "not selfish" behaviour of players. It is only demanded that all players route themselves along paths that have latency of no more than $1+\varepsilon$ times the latency of any other path. This leads to the following formal definition:

**Definition 7** ([RT02]). A flow $f$ feasible for $(G, r, l)$ is at $\varepsilon$-approximate Nash equilibrium if and only if for all commodities $i$ and $P_1, P_2 \in \mathcal{P}_i$ with $f_{P_1} > 0$, we have $l_{P_1}(f) \leq (1+\varepsilon) l_{P_2}(f)$.

With $\varepsilon = 0$, this gives Definition 1, the definition of Nash equilibrium of selfish routing games. Applied to parallel-edge networks with a single commodity, this simplifies to the following equation:

A feasible flow $f$ is at $\varepsilon$-approximate Nash equilibrium if for edges $i, j \in E$ with $f_i > 0$

$$l_i(f_i) \leq (1+\varepsilon) l_j(f_j) \tag{4.6}$$

We now observe that for $\beta = 1+\varepsilon$, Equation 4.6 corresponds to constraint (5) of the mathematical program of fair Stackelberg routing, presented in Table 4.1. We recall that constraint (5) demanded that the latency experienced by any player is not more than $\beta$-times the latency of any other edge in the network. Thus, for fair Stackelberg games $(G, l, \alpha, \beta)$ with $\alpha = 1$, i.e. the Stackelberg leader controls the entire flow, the problem faced by the Stackelberg leader reduces to calculating an approximate Nash equilibrium. Among other results, [CKS11] establish tight bounds on the Price of Stability for $\varepsilon$-approximate Nash equilibria for non-atomic congestion games with polynomial latency functions. Due to the above mentioned connection between fair Stackelberg routing and approximate equilibria, it is therefore not surprising that [CKS11, Theorem 5] exactly matches our lower bound on the Price of Anarchy for fair Stackelberg games (Equation 4.5) if we replace $1+\epsilon$ with $\beta$.

## 4.5. Fair Stackelberg strategies

We now establish fair Stackelberg strategies, i.e. heuristics that guarantee $\mathcal{B}(\alpha) \leq \beta$ for any given fair Stackelberg instance $(G, l, \alpha, \beta)$ in parallel-edge networks. The algorithms we present are adjusted versions of LLF and Scale that make these strategies fair.

Some of the algorithms we present in this section have the characteristic that the Stackelberg leader does not use all of his share $\alpha$ he is allowed to control, i.e. the strategies come with $\sum_{e \in E} g_e < \alpha$. Therefore, the induced Nash flow $h$ must have a total flow of $1 - \sum_{e \in E} g_e > 1 - \alpha$. We call these strategies *reduced* Stackelberg strategies. Reason for why the leader would not use his power completely is as follows: as we demand the strategies to be fair, the Stackelberg leader is restricted in how much flow he can put on edges that do not have minimal latency in the network. In order to reduce the unfairness of his strategy (i.e. to comply with $\beta$), he must reduce the ratio of latency on individually expensive edges to latency on edges with minimal latency. One way of reducing the unfairness ratio is to reallocate controlled players from individually expensive edges to edges used by selfish players. However, edges with minimal latency do not distinguish between selfish and controlled players using this edge, as all players on these edges experience the same latency. Instead of stating the Stackelberg leader routes some of his flow among shortest paths, we can see this as if the Stackelberg leader would allow some of his flow to route itself selfishly, therefore not control it at all.

However, it is easy to transform a reduced Stackelberg strategy to a full Stackelberg strategy for which the controlled flow $g$ adds up exactly to $\alpha$. We present one way below: for a given reduced Stackelberg strategy $g$ with $\sum_{e \in E} g_e < \alpha$, we define $\alpha' = \sum_{e \in E} g_e$. In order to transform $g$ into a full Stackelberg strategy $\hat{g}$ with $\sum_{e \in E} \hat{g}_e = \alpha$, we calculate flow $g'$ which is at Nash equilibrium for $(G, \alpha - \alpha', l(g + g'))$. The full Stackelberg strategy then is the combined flow of $g$ and $g'$, therefore $\hat{g} = g + g'$. Easily to see, $\hat{g}$ has total flow $\alpha$. The induced Nash flow $\hat{h}$ of $\hat{g}$ then is at Nash equilibrium for $(G, 1 - \alpha, l(\hat{g} + \hat{h}))$. We can see that the induced selfish flow of the reduced Stackelberg strategy equals the combination of the controlled flow $g'$ on edges with minimal latency and the induced selfish flow of the full Stackelberg strategy $\hat{h}$, thus $h = g' + \hat{h}$. Further, we note that $\hat{g}$ is just one example of a full Stackelberg strategy that corresponds to $g$, therefore the reduced strategy does not correspond to a unique full Stackelberg strategy. In fact the Stackelberg leader can put his share of flow which uses paths with minimal latency in a variety of ways on edges - as long as he does not put more flow on an edge than there is induced selfish flow.We use r educed Stackelberg strategies to keep our algorithms simple, as they regard flow that could be controlled but ends up using the same edges as the selfish flow as selfish flow.

We illustrate the transformation from reduced to full strategy with the instance we introduced in Figure 2.1. Imagine the leader is allowed to control $\alpha = 1/2$. Le the reduced Stackelberg strategy be $g = (1/4, 0, 0)$. We then have $\alpha' = 1/4$. The induced selfish flow $h$ has demand 3/4 and is $h = (0, 9/16, 3/16)$. Applying the steps presented above leads to the following full Stackelberg strategy $\hat{g}$. We first calculate $g'$ at Nash equilibrium for $(G, \alpha - \alpha', l(g + g'))$, i.e. $(G, 1/4, l(g + g'))$. This yields $g' = (0, 1/4, 0)$. The full Stackelberg strategy is thus $\hat{g} = g + g' = (1/4, 1/4, 0)$. Clearly, $\sum_{e \in E} \hat{g} = \alpha = 1/2$. The induced Nash flow must be at Nash equilibrium for $(G, 1/2, l(\hat{g} + \hat{h}))$, thus $\hat{h} = (0, 5/16, 3/16)$. If we now add $g'$ and $\hat{h}$, we get the induced Nash flow $h$ of the reduced strategy. We also observe that $\hat{g}$ is not the only full Stackelberg strategy corresponding to $g$. Other possibilities would have been $(1/4, 3/16, 1/16)$ or $(1/4, 1/16, 3/16)$. Important is only that the leader does not put more flow on edges with induced selfish flow than there was induced selfish flow on these edges with regard to the reduced strategy, and that he keeps the controlled flow of the reduced strategy on edges without selfish flow.

Having introduced reduced Stackelberg strategies, we now present some fair Stackelberg strategies. We illustrate each algorithm with instances that show their manner of functioning. Aim is not to present the best fair Stackelberg strategy, but to get an understanding of how fair Stackelberg strategies look like.

### 4.5.1. Naive fair Scale

Our first try to make Scale a fair strategy is presented in Algorithm 1. In short, the algorithm applies Scale iteratively, but reduces the number of edges Scale is applied to in every iteration. The algorithm tries to apply Scale but if an edge is individually too expensive, no flow is put on this edge and instead the selfish flow is increased by the amount Scale would have put on the edge that is too expensive.

In detail, Algorithm 1 applies the normal Scale strategy to the optimal flow of the network and checks if the unfairness of Scale satisfies $\beta$. Scale can be used if its unfairness is not too high. Otherwise, there is at least one edge that puts controlled players too much worse off. An edge $i$ with highest latency among all edges with positive flow is picked and the controlled flow on this edge is set to zero. The controlled flow on the other edges remains untouched. But as the Stackelberg leader now only puts a total flow of $\alpha' = \alpha - \alpha o_i$ instead of $\alpha$ on edges, the selfish flow must be increased by $\alpha o_i$, thus the controlled flow Scale would have put on edge $i$ if it was not too costly, so that $g + h$ remain feasible. The algorithm calculates the new induced selfish flow (with higher demand $1 - \alpha + \alpha o_i$) and checks again if the unfairness of the new Stackelberg strategy is too high (this time with the updated flows). In case the unfairness is still too high, the controlled flow on an edge $j$ with highest latency and positive flow is set to zero. The previous edge $i$ is no longer considered, as it either has no attached flow, or, if $h_i > 0$, must have smallest latency. The selfish flow is again increased by $\alpha o_j$. This continues until the strategy is fair.

---

**Algorithm 1:** Naive fair Scale

---

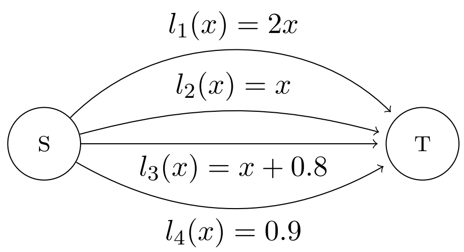
$\alpha' \leftarrow \alpha$
$o \leftarrow$ optimal flow for $(G, l)$
$g_e \leftarrow \alpha o_e$ for $e \in E$
$h \leftarrow$ induced Nash flow on $(G, 1 - \alpha, l(g + h))$
**while** $\mathcal{B} > \beta$ **do**
    $i \leftarrow$ edge with highest latency among all edges with $g_e + h_e > 0$
    $\alpha' \leftarrow \alpha' - g_i$
    $g_i \leftarrow 0$
    $h \leftarrow$ induced Nash flow on $(G, 1 - \alpha', l(g + h'))$
**end**
return $g$

---

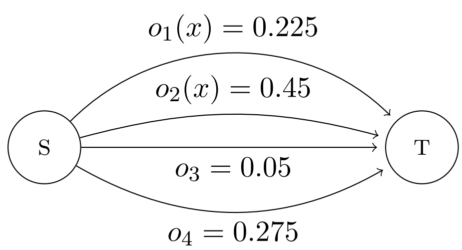
We illustrate Algorithm 1 with an example. Imagine the network instance depicted in Figure 4.10a. We again assume that $r = 1$. Let further $\alpha = 0.8$ and $\beta = 1.6$. Applying Algorithm 1 gives the following: the algorithm first applies the normal Scale strategy, thus puts $g = (g_1, g_2, g_3, g_4) = 0.8o = \alpha(0.225, 0.45, 0.05, 0.275) = (0.18, 0.36, 0.04, 0.22)$. This induced $h = (\frac{1}{15}, \frac{2}{15}, 0, 0)$ with total selfish flow 0.2. We get $\mathcal{B} = \frac{0.9}{37/75} > 1.6$ and edge 4 is the edge with highest latency among all edges with positive flow with respect to $g + h$. As latency on edge 4 relatively to latency on other edges is too high given $\beta$, the Stackelberg leader does not put any flow on this edge. The controlled flow on the remaining edges remains. The Stackelberg leader thus changes his strategy to $g = (0.18, 0.36, 0.04, 0)$. The total induced selfish flow increases to $0.2 + 0.22 = 0.42$, therefore $h = (0.14, 0.28, 0, 0)$. $\mathcal{B}$ with the updated flows $g + h$ is now $\mathcal{B} = 0.84/0.62 = 1.3125 \leq \beta$. As no player is now off too unfair, the algorithm finishes with a fair strategy. The Stackelberg leader now in fact only controls $\alpha' = 0.58$ of the flow. It is worth to point out this is different from applying Scale right away with just $\alpha = 0.58$, as this would put share $\alpha'$ equally on all edges with positive flow in the optimum, i.e. lead to $g' = 0.58o = (0.1305, 0.261, 0.029, 0.1595) \neq (0.18, 0.36, 0.04, 0)$. We further see that $g'$ is not a fair Stackelberg strategy, as it leads to $\mathcal{B}(0.58) = \frac{0.9}{0.541} \approx 1.66 > \beta$.

Imagine now the same instance but with $\beta = 1.3$. It is no longer sufficient to just remove the controlled flow on the forth edge, as $g = (0.18, 0.36, 0.04, 0)$ leads to $\mathcal{B} = 1.3125 > \beta$. The leader must also remove his flow from the third edge and let it route itself. We get $g = (0.18, 0.36, 0, 0)$, $h = (\frac{23}{150}, \frac{23}{75}, 0, 0)$ with total selfish flow $1 - \alpha + \alpha o_4 + \alpha o_3 = 0.46$. We now have $g + h = n$, thus in this case the strategy leads to $\mathcal{B} = 1$.

Even though this algorithm is very easy to implement, it fails to give any performance improvement relatively to the Wardrop equilibrium $n$, even if $\beta$ allows for a performance improvement. To see this, regard the Pigou instance with $\alpha = 0.99$ and $\beta = 1.9$. Scale first puts $g = (0.495, 0.495)$, inducing $h = (0, 0.01)$. But as $\frac{1}{0.505} > 1.9$, players on the top edge are put off worse than allowed. According to Algorithm 1, the Stackelberg leader puts no flow on the top edge, therefore $g = (0, 0.495)$. The induced selfish flow now has higher demand of 0.505, thus $h = (0, 0.505)$. Easy to see, we have $g + h = n$, therefore the strategy leads to the Nash equilibrium $n = (0, 1)$, with a Price of Anarchy of 4/3. Other strategies would have let to performance improvements without violating $\beta$, for example the strategy $g' = (\frac{9}{19}, 0)$ would induce $h' = (0, 10/19)$, therefore $\frac{1}{10/19} = 1.9 \leq \beta$ is satisfied. This would have come with $PoA(g' + h') \approx 1.0009$, a significant improvement relatively to $n$.




(a) Instance




(b) Optimal flow

Figure 4.10.: Instance used to illustrate fair Stackelberg strategies

### 4.5.2. Flexible fair Scale

Algorithm 1 followed an "all-or-nothing" approach, i.e. the Stackelberg leader either puts $\alpha o_e$ on an edge, or no flow. As we have seen, this does not yield any performance improvements relatively to the Wardrop equilibrium, even for high values of $\alpha$ and $\beta$ and linear latencies. Algorithm 2 tries to improve this shortfall: it tries to reduce the flow on edges which are currently too expensive first before it removes the flow on these edges completely. In detail, the algorithm calculates how much flow $x$ must be removed from an edge $i$ which currently has too high latency and added to the induced selfish flow such that the remaining flow on edge $i$ is not too unfair off. The controlled flow on edge $i$ is only changed to zero if that edge would still be too individually expensive if all of its flow would become selfish instead. This idea is noted in line 6 of Algorithm 2. We recall that $L_N$ denotes the minimal latency, i.e. the latency of the induced selfish flow. Line 6 then simply asks what the minimal amount of flow is that must be removed from edge $i$, such that the remaining flow on edge $i$ experiences latency at most $\beta$-times the latency of the induced selfish flow. As the flow that must be removed from $i$ now routes itself selfishly along shortest paths, the latency of the induced selfish flow might increase (as the total amount of selfish flow increases). This is reflected in the notion $L_N(\alpha' - x)$: instead of controlling $\alpha'$, the Stackelberg leader only controls $\alpha' - x$, thus $x$ is added to the induced selfish flow.

**Algorithm 2:** Flexible fair Scale

$\alpha' \leftarrow \alpha$
$g \leftarrow \alpha o$
$h \leftarrow$ induced Nash flow on $(G, 1-\alpha, l(g+h))$
**while** $\mathcal{B} > \beta$ **do**
    $i \leftarrow$ edge with highest latency among all edges with $g_e + h_e > 0$
    $x \leftarrow \text{argmin}_{0 \leq x \leq g_i} l_i(g_i - x) \leq \beta L_N(\alpha' - x)$, $x \leftarrow g_i$ if no such $x < g_i$ exists
    $g_i \leftarrow g_i - x$
    $\alpha' \leftarrow \alpha' - x$
    $h \leftarrow$ induced Nash flow on $(G, 1-\alpha', l(g+h))$
**end**
return $g$

To see the immediate improvement of this, we again consider the Pigou example with $\alpha = 0.99$ and $\beta = 1.9$. Algorithm 2 first applies Scale, giving $g = (0.495, 0.495)$ with induced Nash flow $h = (0, 0.01)$. Again, players on the upper edge experience latency that is too high compared to the latency on the bottom edge. Calculating how much flow must be removed from the upper edge and put on edges with minimum costs gives us $1 \leq 1.9 \cdot (0.505 + x) \leftrightarrow \frac{81}{3800} \leq x$. Thus, if the Stackelberg leader puts $g = (0.495 - \frac{81}{3800}, 0.495)$, inducing $h = (0, 0.01 + \frac{81}{3800})$, the strategy satisfies the unfairness constraints.

As it is the case for all fair Stackelberg strategies, Algorithm 2 cannot let to performance improvements relatively to the Wardrop equilibrium if $\beta$ does not allow for it (recall the instance presented in Figure 4.7b as an example). However, if performance improvements are possible, Algorithm 2 tries to accomplish at least some improvement better than Algorithm 1. However, Algorithm 2 still has the downfall that in case it is not possible to put any flow on an edge, all flow the Stackelberg leader would have liked to put on this edge will become selfish flow, therefore routes itself on shortest paths. It might however lead to lower overall costs if the Stackelberg leader does not automatically give away some of his power, but instead tries to put some of this flow on other edges that do not have minimal latency, but are not too expensive. Therefore, one further enhancement of the algorithm could be the following: if it is not possible for the Stackelberg leader to put any flow on an edge $i$, he removes this edge conceptionally from the network he considers and then calculates the optimal flow $o'$ for the network without edge $i$. Scale is then applied to $o'$. If the resulting strategy is still not fair, the Stackelberg leader again tries to reallocate flow from the edge with highest latency (and positive flow) $j$ to other edges. If again it is not possible to put any flow on $j$, the leader again calculates the optimal flow $o''$ for network without edges $i$ and $j$ and so forth. However, this idea comes with the downside that the controlled flow on some edges increases over time, as with the same total demand, the optimal flow on remaining edges increases if an edge is removed from the network. How this algorithm works in reality is therefore not clearly predictable in advance, which is why we will not investigate this idea in greater detail.

### 4.5.3. Naive shifted LLF

We now adjust LLF such that it is guaranteed to be fair. To do so, we make use of an adjusted version of LLF we call `Shifted LLF`. We recall that LLF first indexes edges from lowest to highest latency regarding the optimal flow, i.e. $l_1(o_1) \leq \cdots \leq l_m(o_m)$ and then tries to saturate edges $m, m-1, \cdots, 2, 1$ one-by-one, starting with edge $m$. `Shifted LLF` builds on this idea but varies the starting edge. In detail, `Shifted LLF` no longer requires to start with the most expensive edge. Instead, the algorithm starts at edge with index $m' \leq m$, and tries to saturate all edges with index $i \leq m'$. Edges with higher index, i.e. more expensive edges, are not tried to be saturated.

Our first approach to make LLF fair, presented in Algorithm 3, is an iterative algorithm which makes use of the idea of `Shifted LLF`. It works as follows. Edges are ordered in the same order as they are with LLF. In the first step of the algorithm, LLF is applied. This is the same as if `Shifted LLF` was applied with $m$. If LLF does comply with $\beta$, i.e. is not too unfair, the algorithm implements LLF and terminates. Otherwise, the Stackelberg flow on the most expensive edge regarding the optimum, i.e. edge $m$, will be put to zero. The algorithm then tries to saturate edges in descending order from edge $m' = m - 1$ onwards, ignoring edge $m$. This is `Shifted LLF` we introduced earlier. Therefore, LLF is applied on edges 1 to $m - 1$, putting a total flow of at most $\alpha$ on these edges. If the resulting strategy is still too unfair, the controlled flow on edge $m - 1$ will be put to zero, and the Stackelberg leader tries to saturate edges from edge $m - 2$ onwards. This procedure continues until the resulting strategy is fair. The approach of `Shifted LLF` can be seen in lines 11 and 12 of the algorithm.

**Algorithm 3:** Naive Shifted LLF

1 $\alpha' \leftarrow \alpha$
2 $o \leftarrow$ optimal flow on $(G, l)$
3 Order edges of $E$ so that $l_1(o_1) \leq \cdots \leq l_m(o_m)$
4 $m' \leftarrow m$
5 Let $k \leq m'$ be minimal with $\sum_{i=k+1}^{m'} o_i \leq \alpha$
6 Put $g_i = o_i$ for $k < i \leq m'$, $g_k = \min\{o_k, \alpha - \sum_{i=k+1}^{m'} o_i\}$ and $g_i = 0$ for $i < k$
7 $h \leftarrow$ induced Nash flow on $(G, 1 - \alpha, l(g + h))$
8 **while** $\mathcal{B} > \beta$ **do**
9 $\quad g_{m'} \leftarrow 0$
10 $\quad m' \leftarrow m' - 1$
11 $\quad$ Let $k \leq m'$ be minimal with $\sum_{i=k+1}^{m'} o_i \leq \alpha$
12 $\quad$ Put $g_i = o_i$ for $k < i \leq m'$, $g_k = \min\{o_k, \alpha - \sum_{i=k+1}^{m'} o_i\}$ and $g_i = 0$ for $i < k$
13 $\quad \alpha' \leftarrow \sum_{e \in E} g_e$
14 $\quad h \leftarrow$ induced Nash flow on $(G, 1 - \alpha', l(g + h))$
15 **end**
16 **return** $g$

We take a closer look to lines 6 and 12 of the algorithm. The sophisticated reader might have noticed that LLF puts $g_k = \alpha - \sum_{i=k+1}^{m} o_i$ on edge $k$, whereas we put $g_k = \min\{o_k, \alpha - \sum_{i=k+1}^{m'}\}$ on this edge. The reason for why we limit $g_k$ by $o_k$, instead of directly putting $\alpha - \sum_{i=k+1}^{m'}$ on this edge, is as follows: as `Shifted LLF` does not try to saturate all edges, the optimal flow on edges with index $i \leq m'$ might sum up to less than $\alpha$. In that case, $\alpha - \sum_{i=k+1}^{m'}$ might be bigger than $o_k$, thus the Stackelberg leader would put more than $o_k$ on edge $k$. As we wish our algorithm to be opt-restricted, we do not allow for this. However, this only is of relevance if the optimal flow on edges $i \leq m'$ is less than what the Stackelberg leader can control. In case the Stackelberg leader does not control sufficient flow to saturate all edges which are left, $\alpha - \sum_{i=k+1}^{m'} o_i$ is smaller than $o_k$, thus the leader puts the remaining of his share on edge $k$, as LLF does too.

Because of the reason that the Stackelberg leader might have a higher share of flow to his disposal than edges he is allowed to saturate, it might happen that he puts less then $\alpha$ on edges. In that case, the Stackelberg leader gives his remaining share of flow away and lets it route itself selfishly. The strategy is therefore a reduced Stackelberg strategy. This is also the reason why the algorithm keeps track of the actual flow $\alpha'$ the Stackelberg leader routes over edges, as the induced selfish flow might increase in size.

We again illustrate the algorithm with the instance seen in Figure 4.10a. The optimal flow is $o = (0.225, 0.45, 0.075, 0.275)$ and we have $l_1(o_1) = l_2(o_2) \leq l_3(o_3) \leq l_4(o_4)$. Let $\alpha = 0.4$. Applying Naive Shifted LLF in the first iteration is simply applying LLF, therefore putting $g = (0, 0.075, 0.05, 0.275)$[2]. This results in $h = (0.225, 0.375, 0, 0)$ and $\mathcal{B} = 2$. If $\beta \geq 2$, Naive Shifted LLF implements LLF. Otherwise, the Stackelberg flow on edge 4 is put to zero, i.e. $g_4 = 0$, and LLF is applied on the edges 1 to 3, trying to saturate edge 3 first. This leads to $g = (0, 0.35, 0.05, 0)$, with an induced selfish flow $h = (\frac{19}{60}, \frac{17}{60}, 0, 0)$ and $\mathcal{B} = \frac{51}{38}$. If $\beta$ was still smaller than this value, the algorithm would set $g_3 = 0$ and try to saturate edges 1 and 2 only. It leads to $g = (0, 0.4, 0, 0)$, inducing $h = (\frac{1}{3}, \frac{4}{15}, 0, 0)$, for with we have $g + h = n$, i.e. in this case the Wardrop equilibrium is implemented, trivially being fair.

Even though this algorithm is easy to implement, it fails to guarantee significant performance increments compared to the Wardrop equilibrium with selfish flow only, especially if the flow controlled by the leader is high. Similar to Algorithm 1, this is because the Stackelberg leader automatically puts zero flow on edges if the current controlled flow is too high to make the strategy fair. We again present the Pigou instance with $\alpha = 0.99$ and $\beta = 1.9$. The Stackelberg leader puts $g = (0.5, 49)$, inducing $h = (0, 0.01)$. This equals the optimal assignment, controlled players are therefore put worse off twice as much as selfish players. As we have $\mathcal{B} = 2 > 1.9 = \beta$, this assignment violates $\beta$, the algorithm therefore puts zero flow on the edge with constant latency, and now tries to apply LLF only on edge 2. He puts $g = (0, 0.5)$ (as he is not allowed to put $g_e > o_e$ on edges). This induces $h = (0, 0.5)$, therefore $g + h = n$ and thus $\mathcal{B} = 1$. The strategy is trivially fair, having costs of $C(g + h) = 0.75$, therefore leading to $PoA(g + h) = \frac{4}{3}$. However, a better Stackelberg strategy would have been possible. If the Stackelberg leader would have only reduced the flow on the edge with constant latency, instead of putting no flow on the edge at all, thus for example putting $g' = (\frac{9}{19}, 0.5)$, inducing $h' = (0, \frac{1}{38})$ would satisfy $\beta$ and lead to a Price of Anarchy of only $PoA(g' + h') \approx 1.0009$. We aim to present a more enhanced version of algorithm 3 which improves this shortfall.

[2] Due to the tie-break, it is also possible to have $g = (0.075, 0, 0.05, 0.275)$

### 4.5.4. Flexible shifted LLF

Algorithm 4 tries to improve the aforementioned shortfalls of Algorithm 3. The logic behind the algorithms are similar, however Algorithm 4 does not automatically put zero flow on the currently most expensive edge with positive flow if the latency on the edge is relatively too high. Instead, it calculates the smallest amount of Stackelberg flow that must be removed from the edge $m'$ (the edge that was tried to be saturated first in the current iteration) and put on edges with minimal latency such that players on edge $m'$ are not put off too much worse off. The Stackelberg leader then removes this amount of flow from edge $m'$ and applies `Shifted LLF` on the edges 1 to $m'-1$. He only puts no flow on edge $m'$ if the flow needed to be removed from that edge in order to make the edge fair is higher than the controlled flow currently on that edge. We have to keep in mind however that as the leader potentially puts flow on edge $m'$, the leader no longer can try to put $\alpha$ on edges 1 to $m'-1$. This is why we introduce a new measure $\alpha_r$ which subtracts the flow the Stackelberg leader puts on edges which are not completely saturated from the total flow the Stackelberg leader can control. He then only tries to saturate edges with smaller latency up to a total amount of $\alpha_r$, as otherwise he would put more flow on edges than he can in sum control.

---

**Algorithm 4:** Flexible Shifted LLF

$\alpha', \alpha_r \leftarrow \alpha$
$o \leftarrow$ optimal flow on $(G, r, l)$
Index edges of $E$ so that $l_1(o_1) \leq \cdots \leq l_m(o_m)$
$m' \leftarrow m$
Let $k \leq m'$ be minimal with $\sum_{i=k+1}^{m'} o_i \leq \alpha_r$
Put $g_i = o_i$ for $k < i \leq m'$, $g_k = \min\{o_k, \alpha - \sum_{i=k+1}^{m'} o_i\}$ and $g_i = 0$ for $i < k$
$h \leftarrow$ induced Nash flow on $(G, 1-\alpha', l(g+h))$
**while** $\mathcal{B} > \beta$ **do**
  $x \leftarrow \operatorname{argmin}_{0 \leq x \leq g_{m'}} l_{m'}(g_{m'} - x) \leq \beta L_N(\alpha' - x), x \leftarrow g_{m'}$ if no such $x \leq g_{m'}$ exists
  $g_{m'} \leftarrow g_{m'} - x$
  $\alpha_r \leftarrow \alpha_r - g_{m'}$
  $m' \leftarrow m' - 1$
  Let $k \leq m'$ be minimal with $\sum_{i=k+1}^{m'} o_i \leq \alpha_r$
  Put $g_i = o_i$ for $k < i \leq m'$, $g_k = \min\{o_k, \alpha_r - \sum_{i=k+1}^{m'} o_i\}$ and $g_i = 0$ for $i < k$
  $\alpha' \leftarrow \sum_{e \in E} g_e$
  $h \leftarrow$ induced Nash flow on $(G, 1-\alpha', l(g+h))$
**end**
**return** $g$

---

We again give an example of the algorithm. We assume the instance in Figure 4.10a with $\alpha = 0.4$. The first iteration is similar to LLF, giving $g = (0, 0.075, 0.05, 0.275)$. Unless $\beta \geq 2$, the Stackelberg leader must remove some flow from edge 4. We distinguish two cases: If $\beta \leq \frac{27}{19}$, the Stackelberg leader cannot put any flow on edge 4. We see this as follows: assuming all controlled flow on edge 4 would route itself selfishly, the selfish flow has demand of $0.6 + g_4 = 0.6 + 0.275 = 0.875$. Given the already existing controlled flow $g = (0, 0.075, 0.05, 0)$, the induced selfish flow is $h = (\frac{19}{60}, \frac{67}{120}, 0, 0)$. This leads to $\mathcal{B} = \frac{27}{19}$. In that case, the Stackelberg leader cannot put any flow on edge 4 (we would have $x = g_4$) and instead tries to saturate edges 1 to 3 instead. $\alpha_r$ does not change as no flow remains on edge 4. Therefore, the strategy changes to $g = (0, 0.35, 0.05, 0)$, thus the same as Algorithm 3, just with the difference that this case only occurs if $\beta \leq \frac{27}{19}$ instead of if $\beta \leq 2$. Then, the same is repeated with edge 3 instead of edge 4 if the unfairness was still too high.

If $\frac{27}{19} < \beta < 2$, the algorithm would now calculate how much flow it must reroute as selfish flow to comply with $\beta$. Let us assume that $\beta = 1.6$. The smallest flow that must be removed from edge 4 such that it complies with $\beta$ is $x = 0.16875$. Then, the new controlled flow on edge 4 is $0.275 - 0.16875 = 0.10625$, which leads to $\alpha_r = 0.29375$ and thus the new strategy $g = (0, 0.24375, 0.05, 0.10625)$, inducing $h = (0.28125, 0.31875, 0, 0)$ with $\mathcal{B} = 1.6$.

For the Pigou instance with $\alpha = 0.99$ and $\beta = 1.9$, Algorithm 4 implements the strategy $g = (\frac{9}{19}, 0.5)$ we already introduced in the previous section. Further, this is an example where the Stackelberg leader puts less flow on edges than he could.

### 4.5.5. LLF on approximate equilibria

We now present an alternative version of LLF which does not run iteratively. Instead, LLF is only applied once, but on a different flow. Namely, LLF is not applied on the optimal flow, but on a flow corresponding to an approximate equilibrium. It therefore makes use of the link between fair Stackelberg routing and $\varepsilon$-approximate Nash equilibria we mentioned in Section 4.4.2. We shortly recall the definition of $\varepsilon$-approximate Nash equilibria for parallel-edge networks.

**Definition 8** ([RT02], parallel edges)**.** A flow $f$ feasible for $(G, l)$ is at $\varepsilon$-approximate Nash equilibrium if for edges $i, j \in E$ with $f_i > 0$, it holds that $l_i(f_i) \leq (1 + \varepsilon) l_j(f_j)$.

We use this definition to adapt it to our $\beta$ notation used in fair Stackelberg games.

**Definition 9.** A flow $f$ feasible for $(G, l)$ is at $\beta$-approximate Nash equilibrium if for edges $i, j \in E$ with $f_i > 0$, it holds that $l_i(f_i) \leq \beta l_j(f_j)$.

Easy to see, for any fair Stackelberg game $(G, l, \alpha, \beta)$ and any given fair Stackelberg strategy $g$, any edge $i$ with $g_i + h_i > 0$ must satisfy $l_i(g_i + h_i) \leq \beta l_j(g_j + h_j)$ for any other edge $j$. Further, we note that the notions of $\beta$-approximate Nash equilibria and $\varepsilon$-approximate Nash equilibria for a given flow $f$ coincide with $\varepsilon = \beta - 1$. When we talk about approximate equilibria in the following, we always refer to $\beta$-approximate equilibria, thus Definition 4.5.5, unless otherwise stated.

The following algorithm, presented in Algorithm 5, makes use of $\beta$-approximate Nash equilibria. For a given fair Stackelberg game $(G, l, \alpha, \beta)$, the best $\beta$-approximate Nash equilibrium is calculated, that is the $\beta$-approximate Nash equilibrium with lowest total cost among all $\beta$-approximate Nash equilibria. Let $\nu$ be the flow corresponding to the best $\beta$-approximate Nash equilibrium. Algorithm 5 then simply applies LLF to $\nu$. Therefore, the algorithm works similar to LLF, but operates on the best $\beta$-approximate equilibrium instead on the optimal flow. Furthermore, as Theorem 1 showed $\mathcal{B}^{opt} \leq 2$ for linear latency functions, LLF can be seen as applying Algorithm 5 with $\beta = 2$, as the optimum clearly is the best 2-approximate Nash equilibrium. Likewise, for polynomial latencies, LLF applies Algorithm 5 with $\beta = d + 1$.

**Algorithm 5:** $\beta$-approximate LLF

1 Compute best $\beta$-approximate Nash equilibrium $\nu$ for $(G, l)$
2 Index the edges of $E$ so that $l_1(\nu_1) \leq \cdots \leq l_m(\nu_m)$
3 Let $k \leq m$ be minimal with $\sum_{i=k+1}^{m} \nu_i \leq \alpha$
4 Put $g_i = \nu_i$ for $i > k$, $g_k = \alpha - \sum_{i=k+1}^{m} \nu_i$ and $g_i = 0$ for $i < k$

We quickly illustrate the algorithm with the Pigou instance in Figure 2.2a with $\alpha = 0.5$ and $\beta = 1.8$. The best 1.8-approximate equilibrium is $\nu = (\frac{4}{9}, \frac{5}{9})$, its unfairness is $\frac{1}{5/9} = 1.8$. Being allowed to control $\alpha = 0.5$, the Stackelberg leader puts $g = (\frac{4}{9}, \frac{1}{18})$, inducing $h = (0, \frac{1}{2})$.

We now bound the performance of Algorithm 5. Specifically, we want to limit the worst Price of Anarchy the algorithm can give for a given fair Stackelberg instance $(G, l, \alpha, \beta)$.

**Theorem 8.** *Let $(G, l, \alpha, \beta)$ be a fair Stackelberg instance in a parallel-edge network with arbitrary latency functions. Let $NE(\beta)$ be the cost of the best $\beta$-approximate equilibrium for instance $(G, l)$. Applying Algorithm 5, the Stackelberg leader induces a flow of cost at most $\frac{1}{\alpha} NE(\beta)$.*

*Proof.* [Swa07, Theorem 3.4] showed performance guarantees for LLF when applied on an optimal flow. We make use of [Swa07] proof and adapt it to our $\beta$-approximate LLF algorithm applied on an approximate flow. The proof therefore reads very similar, just applied on a different strategy.

Let $NE(\beta)$ denote the total cost of the best $\beta$-approximate Nash equilibrium and $\nu$ be the flow corresponding to this equilibrium. We re-index the edges in decreasing order of $l_e(\nu_e)$, i.e. their latency regarding the best $\beta$-approximate Nash equilibrium. Therefore, $l_1(\nu_1) \geq \cdots \geq l_m(\nu_m)$. Let $g$ be the Stackelberg leader's flow when applying Algorithm 5 and $k$ be the index of the edge with highest index used by $g$, therefore the edge with smallest latency regarding $\nu$ of all edges with positive controlled flow. We define $L_g$ as the lowest latency of any edge regarding $\nu$ that comes with $g_e > 0$ for Algorithm 5. Therefore, $L_g$ is the lowest latency $l_e(\nu_e)$ of all edges with $g_e > 0$. Clearly, $L_g = l_k(\nu_k)$, as $k$ is the edge with highest index (thus lowest latency regarding $\nu$) the Stackelberg leader puts flow on. Let $L_N$ be the latency of the induced selfish flow $h$ by $g$, therefore $L_N$ denotes the minimal latency in the network given $g + h$. We claim that $L_N \leq L_g$. If not, then for all $i \geq k$, we have $l_i(g_i + h_i) \geq L_N > L_g \geq l_i(\nu_i)$ since all edges have latency at least $L_N$ under the flow $g + h$ (otherwise, selfish players would not use shortest paths). This implies that $g_i + h_i > \nu_i$ for every $i \geq k$, as latency functions are non-decreasing. For $i < k$, $g_i = \nu_i$, as the Stackelberg leader saturates these edges. This implies that the total flow $(g + h)$ has volume greater than 1, giving a contradiction.

Thus, $L_N \leq L_g$. This implies the following simple facts: (a) for edges $i \leq k$, $l_i(g_i + h_i) \leq l_i(\nu_i)$, since $l_i(\nu_i) \geq L_g \geq L_N$ for all these edges, (b) thus $\sum_{i=1}^{k} g_i l_i(g_i + h_i) \leq \sum_{i=1}^{k} \nu_i l_i(\nu_i) \leq \sum_{i \in E} \nu_i l_i(\nu_i) = NE(\beta)$, as $g_e \leq \nu_e$ by definition for all edges and $l_i(g_i + h_i) \leq l_i(\nu_i)$ for all edges 1 to $k$ as above and (c) $NE(\beta) \geq \alpha L_g \geq \alpha L_N$, thus $L_N \leq \frac{1}{\alpha} NE(\beta)$. Since the cost of $h$ is $(1-\alpha)L_N$, this implies that $C(g+h) \leq NE(\beta) + (1-\alpha)L_N \leq NE(\beta) + (1-\alpha)\frac{1}{\alpha}NE(\beta) = NE(\beta)(1 + \frac{1-\alpha}{\alpha}) = \frac{1}{\alpha}NE(\beta)$. ■

Theorem 8 gives an upper bound on the performance of Algorithm 5 for arbitrarily latency functions which depends on the total cost of the best $\beta$-approximate Nash equilibrium. [CKS11, Theorem 5] established tight bounds on the Price of Stability of approximate equilibria for polynomial latency functions. We can use this result and express the cost of the best $\beta$-approximate Nash equilibrium relatively to the cost of an optimal assignment for linear and polynomial latency functions. This allows us to not only bound the performance of the algorithm relatively to the best $\beta$-approximate Nash equilibrium, but relatively to an optimal flow. This allows us to establish upper bounds on the Price of Anarchy for Algorithm 5 for linear or polynomial latencies.

**Corollary 8.1.** *Let $(G, l, \alpha, \beta)$ be a fair Stackelberg game in a parallel-edge network with polynomial latencies up to degree $d \geq 2$. Let $OPT$ be the cost of optimal flow $o$ for $(G, l)$. Applying Algorithm 5, the Stackelberg leader induces a flow $g + h$ of total cost at most*

$$\frac{1}{\alpha}\left(\beta\left(1 - \frac{d}{d+1}\left(\frac{\beta}{d+1}\right)^{1/d}\right)\right)^{-1} \cdot OPT$$

*for $\beta < d+1$ and*

$$\frac{1}{\alpha} \cdot OPT$$

*for $\beta \geq d+1$.*

*Proof.* [CKS11] have shown that the Price of Stability of $\varepsilon$-approximate Nash equilibria with polynomial latency functions of degree $d$ is exactly

$$\left((1+\varepsilon)\left(1 - \frac{d}{d+1}\left(\frac{1+\varepsilon}{d+1}\right)^{1/d}\right)\right)^{-1}$$

for $\varepsilon < d$ and 1 for $\varepsilon \geq d$. We recall that the Price of Stability established in [CKS11] is defined as the ratio of total cost of best $\varepsilon$-approximate equilibrium (i.e. $\varepsilon$-approximate equilibrium with lowest costs) to optimum. Further, with $\beta = 1 + \varepsilon$, we see that the best $\beta$-approximate Nash equilibrium used by the algorithm corresponds to the best $\varepsilon$-approximate equilibrium for a given instance.

We therefore get that

$$\frac{NE(\beta)}{OPT} = \left(\beta\left(1 - \frac{d}{d+1}\left(\frac{\beta}{d+1}\right)^{1/d}\right)\right)^{-1}$$

if $\beta < d+1$ and $\frac{NE(\beta)}{OPT} = 1$ if $\beta \geq d+1$. Therefore, the cost of the best $\beta$-approximate Nash equilibrium is

$$NE(\beta) = \left(\beta\left(1 - \frac{d}{d+1}\left(\frac{\beta}{d+1}\right)^{1/d}\right)\right)^{-1} \cdot OPT \tag{4.7}$$

if $\beta < d+1$ and $1 \cdot OPT$ otherwise.

Replacing $NE(\beta)$ in Theorem 8 by Equation 4.7 gives the statement. ■

With $d = 1$, i.e. for linear latencies, this gives us:

**Corollary 8.2.** *Let $(G, l, \alpha, \beta)$ be a fair Stackelberg instance in a parallel-edge network with linear latencies. Let $OPT$ be the total cost of optimal flow $o$ for $(G, l)$. Applying Algorithm 5, the Stackelberg leader induces a flow $g + h$ of cost at most*

$$\frac{1}{\alpha}\left(\left(\beta - \frac{\beta^2}{4}\right)^{-1}\right) \cdot OPT$$

*for $\beta < 2$ and*

$$\frac{1}{\alpha} \cdot OPT$$

*for $\beta \geq 2$.*

# 5. Conclusion and Future Work

This thesis gave an insight into the implications of fairness considerations on Stackelberg routing. We defined the *unfairness* of a given flow as the ratio of highest latency experienced by any player to lowest latency for any $s-t$ path in the network for the given flow. For parallel-edge networks, this work showed several results:

We showed that the unfairness of optimal flow $o$ is upper-bounded by 2 for linear latency functions and upper-bounded by $d+1$ for polynomial latency functions up to degree $d$. We further showed that the unfairness of any opt-restricted Stackelberg strategy cannot exceed the unfairness of an optimal flow for a given instance. For the opt-restricted strategy LLF, we further showed that the bounds of 2 and $d+1$ are tight. We introduced a class of strategies called monotone opt-restricted strategies which have the property that the flow the leader puts on every edge monotonically increases with the share $\alpha$ of flow he can control. We showed that the unfairness of any monotone opt-restricted strategy is monotonically increasing with $\alpha$. Further, as the two common strategies Scale and LLF are monotone opt-restricted strategies, this result applies to them.

We then introduced *fair Stackelberg games*, which come with an external parameter $\beta \geq 1$ and require the Stackelberg leader not to put players on edges with latency more than $\beta$-times the latency on any other edge. The unfairness of any fair Stackelberg strategy is not allowed to be higher than $\beta$. We showed that the Price of Anarchy of *any* fair Stackelberg strategy is at least $\max\{\frac{4}{3+\alpha}, (\beta - \frac{\beta^2}{4})^{-1}\}$ for linear latencies and $\max\{[d(d+1)^{-(d+1)/d}(\alpha-1)+1]^{-1}, [\beta(1-\frac{d}{d+1}(\frac{\beta}{d+1})^{1/d})]^{-1}\}$ for polynomial latency functions. Lastly, we presented some ideas for algorithms that are fair Stackelberg strategies.

Fairness not being studied before in Stackelberg routing, this work served as an introduction to the topic. Aim was to get an overall understanding of the implications of fairness demands on Stackelberg routing. This work focussed on parallel-edge networks. Some results required linear or polynomial latency functions. A natural question for further work is whether the results we gained hold for more general networks and latency functions. In particular, we ask the following questions: Is the unfairness of optimal flow and any opt-restricted Stackelberg strategy also bounded for arbitrary latency functions, or can the unfairness be arbitrarily bad? Can the lower bounds on the Price of Anarchy for any fair Stackelberg strategy with linear or polynomial latency functions be generalised to arbitrary latencies? More importantly, can these bounds be sharpened? What are the respective upper bounds?

We further proved that for any $\alpha > 0$ and linear or polynomial latencies, 2 and $d+1$ are not only upper bounds on the unfairness of the LLF strategy, but also lower bounds, making these bounds tight. For Scale, we only proved that 2 and $d+1$ are upper bounds on the unfairness, whether these bounds are tight or if the lower bound on the unfairness of Scale is in fact strictly lower remains unclear. If we were to conjecture, we would think that for all $\alpha < 1$, Scale leads to unfairness which is strictly lower than the unfairness of the optimum. Related to this, we presumed, but were not able to prove, that the unfairness of the Scale strategy is not higher than the unfairness of the LLF strategy for a given instance $(G, l, \alpha)$. We therefore encourage the reader to examine this question themselves, as an answer would give further insights into the connection between unfairness and efficiency of strategies.

We further presented fair Stackelberg strategies which guarantee to put players not too worse off. For one of them, called $\beta$-approximate LLF, we were able to show performance guarantees. Do other, better fair Stackelberg strategies exist? Further, how difficult is the problem of finding the optimal fair Stackelberg strategy? [Rou04] showed that finding the optimal Stackelberg strategy without fairness considerations is NP-hard. Does this apply to to finding the best fair Stackelberg strategies as well, or is the complexity of this problem in a different complexity class?

As the reader can see, there remain a lot of questions open. We encourage the reader to regard this work as a starting point for further investigation of unfairness in Stackelberg routing. We are optimistic that further insights will yield interesting results. On this note, we leave the reader to it, and would be delighted to see readers being encouraged to do own work in this field.

# A. Appendix

## A.1. Mathematical program for general networks

$$
\begin{array}{llcll}
\min & \displaystyle\sum_{e\in E}(g_e+h_e)l_e(g_e+h_e) & & & \\
\text{s.t.} & \displaystyle\sum_{P\in\mathcal{P}_i} g_P & \leq & \alpha r_i & \forall i\in\{1,\dots,k\} \quad (1)\\
& \displaystyle\sum_{P\in\mathcal{P}_i} h_P & = & (1-\displaystyle\sum_{P\in\mathcal{P}_i} g_P)r_i & \forall i\in\{1,\dots,k\} \quad (2)\\
& \displaystyle\sum_{P\in\mathcal{P}_i} x_P & = & (1-\displaystyle\sum_{P\in\mathcal{P}_i} g_P)r_i & \forall i\in\{1,\dots,k\} \quad (3)\\
& \displaystyle\sum_{e\in E} h_e l_e(g_e+h_e) & \leq & \displaystyle\sum_{e\in E} x_e l_e(g_e+h_e) & \forall e\in E \quad (4)\\
& l(P_{g>0}) & \leq & \beta\cdot l(P') & \forall P_{g>0}, P'\in\mathcal{P}_i, \forall i\in\{1,\dots,k\} \quad (5)\\
& l(P_{h>0}) & \leq & 1\cdot l(P') & \forall P_{h>0}, P'\in\mathcal{P}_i, \forall i\in\{1,\dots,k\} \quad (6)\\
& g_e, h_e, x_e & \geq & 0 & \forall e\in E \quad (7)\\
& \displaystyle\sum_{P\in\mathcal{P}:e\in P} g_P & = & g_e & \forall e\in E \quad (8)\\
& \displaystyle\sum_{P\in\mathcal{P}:e\in P} h_P & = & h_e & \forall e\in E \quad (9)
\end{array}
$$

Table A.1.: Mathematical program for general multi-commodity networks

This is the mathematical programme of Table 4.1 applied to general networks. We now allow for an arbitrary demand $r$ and multiple commodities. As paths can consist of multiple edges, the notion is adapted to paths instead of edges. Further, constraint (8) states that the flow routed by the Stackelberg leader over a specific edge must equal the Stackelberg flow of all paths on this edge. Constraint (9) demands the same for the induced selfish flow.

# Bibliography


[Ans+03] Elliot Anshelevich, Anirban Dasgupta, Éva Tardos, and Tom Wexler. "Near-optimal network design with selfish agents". In: *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, STOC'03*. San Diego, CA, USA, 2003, pp. 511–520.

[BHS10] Vincenzo Bonifaci, Tobias Harks, and Guido Schäfer. "Stackelberg routing in arbitrary networks". In: *Mathematics of Operations Research* 35 (2 2010), pp. 330–346.

[Biy+18] Erdem Biyik, Daniel Lazar, Ramtin Pedarsani, and Dorsa Sadigh. "Altruistic autonomy: beating congestion on shared roads". In: *Proceedings of the 13th International Workshop on the Algorithmic Foundations of Robotics, WAFR'18*. Vol. 14. SPAR. 2018, pp. 887–904.

[BMW56] Martin Beckmann, C. B. McGuire, and Christopher B. Winsten. *Studies in the economics of transportation*. Yale University Press, 1956.

[Bra68] Dietrich Braess. "Über ein Paradoxon der Verkehrsplanung". In: *Unternehmensforschung* 12 (1968), pp. 258–268.

[BV19] Vittorio Bilò and Cosimo Vinci. "On Stackelberg strategies in affine congestion games". In: *Theory of Computing Systems* 63 (2019), pp. 1228–1249.

[CKS11] George Christodoulou, Elias Koutsoupias, and Paul G. Spirakis. "On the performance of approximate equilibria in congestion games". In: *Algorithmica* 61 (1 2011), pp. 116–140.

[CS07] José R. Correa and Nicolás E. Stier-Moses. *Stackelberg routing in atomic network games*. Columbia Working Paper No. DRO-2007-03. Columbia Business School, 2007.

[CSS07] José R. Correa, Andreas S. Schulz, and Nicolás E. Stier-Moses. "Fast, fair, and efficient flows in networks". In: *Operations Research* 55 (2 2007), pp. 215–225.

[Fot10] Dimitris Fotakis. "Stackelberg strategies for atomic congestion games". In: *Theory of Computing Systems* 47 (1 2010), pp. 218–249.

[Jah+05] Olaf Jahn, Rolf H. Möhring, Andreas S. Schulz, and Nicolás E. Stier-Moses. "System-optimal routing of traffic flows with user constraints in networks with congestion". In: *Operations Research* 53 (3 2005), pp. 600–616.

[KK09] George Karakostas and Stavros G. Kolliopoulos. "Stackelberg strategies for selfish routing in general multicommodity networks". In: *Algorithmica* 53 (1 2009), pp. 132–153.

[KLO97] Yannis A. Korilis, Aurel A. Lazar, and Ariel Orda. "Achieving network optima using Stackelberg routing strategies". In: *IEEE/ACM Transactions on Networking* 5 (1 1997), pp. 161–173.

[KM02] V. S. Anil Kumar and Madhav V. Marathe. "Improved results for Stackelberg scheduling strategies". In: *Proceedings of the 29th International Colloquium on Automata, Languages and Programming, ICALP'02*. Málaga, Spain, 2002, pp. 776–787.

[KP99] Elias Koutsoupias and Christos Papadimitriou. “Worst-case equilibria”. In: *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science, STACS’99*. Vol. 1563. LNCS. Trier, Germany, 1999, pp. 404–413.

[KS09] Alexis C. Kaporis and Paul G. Spirakis. “The price of optimum in Stackelberg games on arbitrary single commodity networks and latency fuctions”. In: *Theoretical Computer Science* 410 (8–10 2009), pp. 745–755.

[Lin+11] Henry Lin, Tim Roughgarden, Éva Tardos, and Asher Walkover. “Stronger bounds on Braess’s paradox and the maximum latency of selfish routing”. In: *SIAM Journal on Discrete Mathematics* 25 (4 2011), pp. 1667–1686.

[Neu28] John von Neumann. “Zur Theorie der Gesellschaftsspiele. On the theory of games of strategy”. In: *Mathematische Annalen* 1 (1928), pp. 295–320.

[Nis+08] Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani (eds.) *Algorithmic Game Theory*. Cambridge University Press, 2008.

[Nis99] Noam Nisan. “Algorithms for selfish agents. Mechanism design for distributed computation”. In: *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science, STACS’99*. Vol. 1563. LNCS. Trier, Germany, 1999, pp. 1–15.

[Pap01] Christos Papadimitriou. “Algorithms, games, and the internet”. In: *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, STOC’01*. Hersonissos, Greece, 2001, pp. 749–753.

[Pig20] Arthur Cecil Pigou. *The economics of welfare*. Palgrave Macmillan, 1920.

[Rou02] Tim Roughgarden. “Selfish Routing”. PhD Thesis. Cornell University, 2002.

[Rou04] Tim Roughgarden. “Stackelberg scheduling strategies”. In: *SIAM Journal on Computing* 33 (2 2004), pp. 332–350.

[Rou05] Tim Roughgarden. *Selfish routing and the Price of Anarchy*. MIT Press, 2005.

[Rou06] Tim Roughgarden. “On the severity of Braess’s paradox: designing networks for selfish users is hard”. In: *Journal of Computer and System Sciences* 72 (5 2006), pp. 922–953.

[Rou07] Tim Roughgarden. “Selfish Routing and the Price of Anarchy”. In: *OPTIMA - Mathematical Programming Society Newsletter*. MPI 2007. May 15, 2007.

[Rou16] Tim Roughgarden. *Twenty lectures on Algorithmic Game Theory*. Cambridge University Press, 2016.

[RT02] Tim Roughgarden and Éva Tardos. “How bad is selfish routing?” In: *Journal of the ACM* 49 (2 2002), pp. 236–259.

[Smi79] Mike J. Smith. “The existence, uniqueness and stability of traffic equilibria”. In: *Transportation Research Part B: Methodological* 13 (4 1979), pp. 295–304.

[SS03] Andreas S. Schulz and Nicolàs Stier-Moses. “On the performance of user equilibria in traffic networks”. In: *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA’03*. Baltimore, Maryland, USA, 2003, pp. 86–87.

[Sta34] Heinrich von Stackelberg. *Marktform und Gleichgewicht*. 1934.

[SW09] Yogeshwer Sharma and David P. Williamson. “Stackelberg thresholds in network routing games or the value of altruism”. In: *Games and Economic Behavior* 67 (1 2009), pp. 174–190.

[Swa07] Chaitanya Swamy. “The effectiveness of Stackelberg strategies and tolls for network congestion games”. In: *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'07*. New Orleans, Louisiana, USA, 2007, pp. 1133–1142.

[War52] John Glen Wardrop. “Some theoretical aspects of road traffic research”. In: *Proceedings of the Institution of Civil Engineers*. Pt. II. Vol. 1. 1952, pp. 325–362.

[YH05] Hai Yang and Hai-Jun Huang. *Mathematical and economic theory of road pricing*. Emerald Publishing, 2005.