Imperial College London

BEng Computing Individual Project

Imperial College London

Department of Computing

Is selfish routing really inefficient? Beyond worst-case price of anarchy

Author:

Lok Hin Chan

Supervisor:

Dr. Dario Paccagnan

Alex, Carver

Date: June 2021

Abstract

In this project, we would consider the problem of traffic routing, one of the most studied problems in Algorithmic Game Theory. One of the key metrics is the Price of Anarchy (PoA), which measures the inefficiency caused by changing centralised optimal routing to distributed selfish routing.

In particular, this project mainly focuses on the Price of Anarchy for Games having latency function of the form $a + bx^4$ suggested by the Bureau of Public Roads. Most of the synthetic data models the traffic flow with this function so that we can compare the result developed from this project against real-life driven synthetic data. Previously, researchers have proved a worst PoA upper bound of 2.151 for this particular set-up.

This project aims to answer a question: Can we achieve a tighter PoA estimate beyond the worst case scenario, possibly by utilising more information in the traffic network? In the project, we explore a couple of different approaches and show that we could attain a tighter PoA upper bound estimate by knowing how each agent acts selfishly in a particular traffic network with extra computational effort.

Apart from establishing a sound theoretical ground, the results are also applied to real-world Driven Data to show that the estimation indeed produces an improved upper bound. For instance, in the famous example of Sioux Falls, we could reduce the PoA estimate from 2.151 to around 1.15.

Contents

1 Introduction

Under the context of traffic routing, selfish decision is often inefficient. As a driver, we would choose the route which shortens the travel time for self-interest, but it might not yield the shortest travel time overall from the perspective of a system. This scenario interests researchers: Can we design a network to minimise the inefficiency caused by selfish routing?

The loss from changing centralised optimal routing to distributed selfish routing, also known as the "Price of Anarchy" (PoA), has been defined and quantified by Papadimitriou in 2001.[1]. A network with PoA close to 1 would mean that the network is efficient in the sense that the total cost from selfish routing is similar to that from optimal routing.

Formally, PoA is defined by the ratio of cost induced by Nash Equilibrium Flow and by Optimal Flow. Here the cost is a measure of loss when an agent travels through a road segment, for example travel time or gas consumed. The Nash Equilibrium Flow is achieved when each agents choose their travel routes based on their self-interest, whereas the Optimal Flow is achieved when each agents choose the travel routes which minimises the total cost imposed to the whole system.

Researchers such as Roughgarden and Terdos have shown upper bounds for PoA, which is also the worst-case analysis, for certain configurations of traffic networks.[2] Once we know the PoA from a certain traffic network, we could decrease PoA, which is the loss due to selfish routing, by rerouting the network or to change the cost travelling on some path by marginal cost taxes, or tolls.[3]

However, recent studies involving real-world data show a huge discrepancy between data-driven PoA and the theoretical upper bound as suggested. In particular, a study on traffic network in Singapore shows that the actual PoA would be 1.34, compared to the upper bound of 2.151. This discrepancy would translate into a loss of over 730,000 hours per day for commuting in Singapore.[4] Another study focused on real-world traffic network in Eastern Massachusetts concludes that the average PoA in a specific day's afternoon would be 1.5522, compared to the upper bound of 3.299.[5][6]

In practice, most of the real-world and synthetic data assumes that the traffic flow follows the Bureau of Public Roads functions[7] which has a cost of the form $a + bx^4$, therefore the theories developed in this report also assumes traffic routing with this particular configuration.

Since PoA calculated from real-world data represents an average case, where the PoA from theory gives the worst-case upper bound, a natural question would arise: What are the fundamental differences between theory and data? In other words, would it be possible to lower the upper bound of theoretical PoA in order to reduce the discrepancy between theoretical and exact PoA? This project aims to answer these questions.

In section 2, we first present technical preliminaries which also serves as the literature review of previous results. We then develop a sound theoretical framework which allows further study on real-world data-driven inefficiency analysis from section 3 to section 6, and in section 7, results developed from the previous sections would be implemented into real-world and simulated data and with the data we can investigate the gap between theoretical and data-driven PoA.

2 Preliminaries and Models

This section aims to provide a solid background framework via basic definitions and results which would be useful for the later part of the project. Through presenting the Model, Equilibrium and Optimal Flow, we could define notion of Price of Anarchy, as well as stating known results from the literature.

2.1 The Model

In this project, we mainly consider the case of non-atomic routing game, which means there are an infinite amount of agents who controls a negligible amount of the traffic. This assumption would be realistic on a big network or traffic on a highway.

We define the routing game as follows, with slight modifications from Colini-Baldeschi et al.[8]: The model is represented by a directed multi-graph $\mathcal{G} \equiv (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are finite vertex set and finite edge set respectively. We further assume a finite set of *origin-destination* (O/D) pairs indexed by $i \in \mathcal{I}$, and each index i is associated with a *traffic demand* $m^i \geq 0$, which represents the total traffic from vertex $O^i \in \mathcal{V}$ to vertex $D^i \in \mathcal{V}$. This implies that the total traffic inflow, $M = \sum_{i \in \mathcal{I}} m^i \geq 0$.

As a part of routing configuration we also define $\mathcal{P}^i$, the set of paths that joins from O^i to D^i, and for each $p \in \mathcal{P}^i$, p is a sequence of edges. By setting $\mathcal{P} \equiv \bigcup_{i \in \mathcal{I}} \mathcal{P}^i$, we can then define the feasible set of *routing flows* with

$$\mathcal{F} = \{f : \mathcal{P} \to \mathbb{R}^+ \mid \forall i \in \mathcal{I} : \sum_{p \in \mathcal{P}^i} f_p = m_i\}$$

Each $f \in \mathcal{F}$ would induce a *load* x_e on each edge $e \in \mathcal{E}$, and $x_e = \sum_{p \ni e} f_p \geq 0$, with *load profile* defined as $x = (x_e)_{e \in \mathcal{E}}$. Last but not least we define *latency functions* $c_e : \mathbb{R}^+ \to \mathbb{R}^+$ for each edge $e \in \mathcal{E}$, where c_e is a continuous, non-decreasing and differentiable function. Combining the latency functions and load profile in a routing game, we can conclude that the latency on edge $e \in \mathcal{E}$ is $c_e(x_e)$. By overloading the notation c, we could define the latency function along a path $c_p : \mathcal{F} \to \mathbb{R}^+$

for each path $p \in \mathcal{P}$, as well as the latency of a path with $c_p(f) = \sum_{e \in \mathcal{P}} c_e(f_e)$.

Combining the components discussed above, an non-atomic instance of routing game is defined by the tuple

$$\Gamma = (\mathcal{G}, \mathcal{I}, \{m^i\}_{i \in \mathcal{I}}, \{\mathcal{P}^i\}_{i \in \mathcal{I}}, \{c_e\}_{e \in \mathcal{E}})$$

with the overall latency $C : \mathcal{F} \to \mathbb{R}^+$ is defined as

$$C(f) = \sum_{p \in \mathcal{P}} c_p(f) f_p = \sum_{e \in \mathcal{E}} c_e(x_e) x_e$$

.

To understand how the definition above models traffic routing problem, the following table summarises the purpose of each component in an instance Γ:

Component	Representation under traffic routing context
$\mathcal{G}$	A directed graph representing road connection between junctions
$\mathcal{I}$	Index for each pair of destination and origin (O/D pairs)
m^i	Traffic flow demand for different O/D pairs representing demand for commuting from origin A to destination B
$\mathcal{P}^i$	Paths for different O/D pairs representing the possible routes commuting from origin A to destination B
c_e	Latency function for edges represents the cost when an agent passes through the road. Some possible measure of cost includes the time spent travelling through the road segment or the toll of the road.
$\mathcal{F}$	Feasible set of routing flow represents a traffic flow that satisfies traffic flow demands of all agents.

2.2 Nash Equilibrium and Optimal Flow

Informally, a Nash Equilibrium is achieved in a routing game when each agents do not have the incentive to change its path. In other words, each agent would choose the path with the smallest latency from their own perspective.

Definition 2.1 (Nash Equilibrium [3])**.** *A feasible flow* $f \in \mathcal{F}$ *for* Γ *is at Nash Equilibrium if and only if* $\forall i \in \mathcal{I}$ *and* $\forall p_1, p_2 \in \mathcal{P}^i$ *with* $f_{p_1} > 0$, $c_{p_1}(f) \leq c_{p_2}(f)$.

As for optimal flow, the overall cost $C(f)$ achieves an local minimum if any of the agents could not lower the overall cost by changing their paths. Described in economic terms, the increase in marginal cost of switching to another path has to be higher than the decrease in marginal benefit in staying in the same path for all possible paths. Moreover the local and global minimum of $C(f)$ are the same since the local and global minimum of a convex function in convex set are the same.

Formally we could characterise optimal flow of Γ with the following minisation problem:

Definition 2.2 (Optimal Flow from non-linear program)**.**

$$\begin{aligned} \text{minimise} \quad & \sum_{e \in \mathcal{E}} c_e(x_e) x_e \\ \text{subject to:} \quad & x_e = \sum_{p \ni e} f_p \\ & f \in \mathcal{F} \end{aligned} \tag{NLP-OF}$$

For convenience we define $\ell_p(f) = c_p(f) f_p$, which is the total latency along path p suffered by relevant agents, and analogously $\ell_e(x_e) = c_e(x_e) x_e$, the total latency on edge e, associated with the derivative $\ell'_p(f) = \sum_{e \in p} \ell'_e(x_e)$, and $\frac{d}{dx_e} \ell_e(x_e) = \ell'_e(x_e)$. Then similar to Definition 2.1, we could express optimal flow in terms of $\ell'_p(f)$:

Theorem 2.3 (Optimal Flow [9])**.** *A feasible flow* $f \in \mathcal{F}$ *for* Γ *is optimal if and only if* $\forall i \in \mathcal{I}$ *and* $\forall p_1, p_2 \in \mathcal{P}^i$ *with* $f_{p_1} > 0$, $\ell'_{p_1}(f) \leq \ell'_{p_2}(f)$.

The following Corollary follows from Definition 2.1 and Theorem 2.3:

Corollary 2.4 ([9])**.** *A flow f is feasible for $\Gamma = (\mathcal{G}, \mathcal{I}, \{m^i\}_{i\in\mathcal{I}}, \{\mathcal{P}^i\}_{i\in\mathcal{I}}, \{c_e\}_{e\in\mathcal{E}})$ is optimal if and only if it is at Nash Equilibrium for $\Gamma' = (\mathcal{G}, \mathcal{I}, \{m^i\}_{i\in\mathcal{I}}, \{\mathcal{P}^i\}_{i\in\mathcal{I}}, \{\ell'_e\}_{e\in\mathcal{E}})$, and $\ell_e(e_x) = c_e(x_e)x_e$ is a convex function for all $e \in \mathcal{E}$.*

With Corollary 2.4 and Definition 2.2, we can also define Nash Equilibrium as follows:

Corollary 2.5 (Nash Equilibrium from non-linear program[9])**.**

$$\begin{aligned} \text{minimise} \quad & \sum_{e\in\mathcal{E}} \int_0^{x_e} c_e(t)\,dt \\ \text{subject to:} \quad & x_e = \sum_{p\ni e} f_p \\ & f \in \mathcal{F} \end{aligned} \tag{NLP-NE}$$

The formula stated above, combined with Definition 2.2, gives us a systematic way to compute the load profile under Nash Equilibrium and Optimal Flow through solving non-linear program. Next theorem states the existence and uniqueness of Nash Equilibria, which is crucial for the definition of Price of Anarchy introduced in section 2.3.

Theorem 2.6 ([2][9])**.** *An instance Γ admits a feasible flow at least one Nash Equilibrium. Moreover, if f, $\tilde{f}$ are feasible flows at Nash Equilibrium, then $C(f) = C(\tilde{f})$.*

Next theorem is also known as the Variational Inequality Characterisation suggested by Smith. This theorem, combined with Definition 2.8, are vital for the proof presented in the rest of the report.

Theorem 2.7 (Variational Inequality Characterisation[10])**.** *If an instance Γ have Equilibrium load profile of x^{EQ} and Optimal Flow load profile of x^{OPT}, then*

$$\sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} \leq \sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{OPT}$$

2.3 Price of Anarchy

In this section, the notion of Price of Anarchy is defined. The Price of Anarchy, PoA(Γ), is the ratio of overall cost induced by Nash Equilibrium Flow f and overall

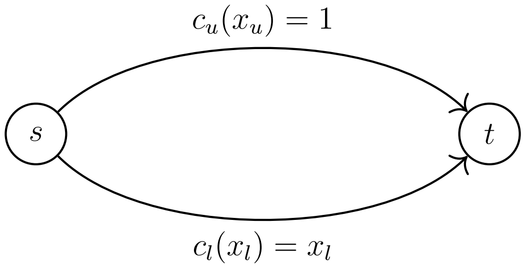


Figure 2.1: Pigou's Example

cost induced by Optimal Flow f^* derived from instance Γ defined in Section 2.1. Formally,

$$\text{PoA}(\Gamma) = \frac{C(f)}{C(f^*)}$$

And from Theorem 2.6, existence and uniqueness of Nash Equilibria implies the existence and uniqueness of the Price of Anarchy.

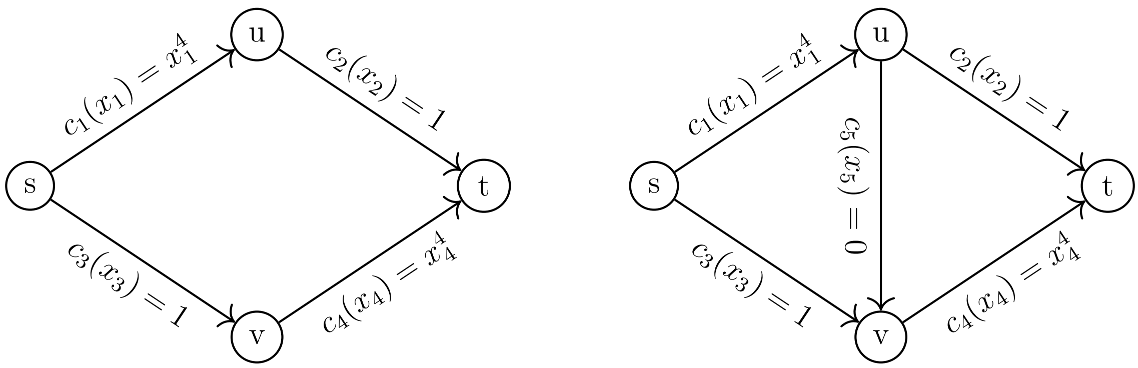
2.4 Examples

This section explores two classic examples proposed by Pigou[11] and Braess[12] respectively, and they serve as case studies for this section.

2.4.1 Pigou's Example

Consider the configuration on Figure 2.1 with $\Gamma = (\mathcal{G}, \mathcal{I}, \{m^i\}_{i\in\mathcal{I}}, \{\mathcal{P}^i\}_{i\in\mathcal{I}}, \{c_e\}_{e\in\mathcal{E}})$, where $\mathcal{G} = (\mathcal{V}, \mathcal{E}) = (\{s,t\}, \{u,l\})$, $\mathcal{I} = \{1\}$, $m^1 = 1$, $\mathcal{P}^1 = \{(u), (l)\}$ and $\{c_e\}_{e\in\mathcal{E}} = \{c_u, c_l\}$.

Solving Γ with Corollary 2.5, at Nash Equilibrium $x_l = 1$ and $x_u = 0$, inducing a total cost $C(f) = 1 * 0 + 1 * 1 = 1$. On the other hand solving Γ with Definition 2.2 yields $x_l = x_u = \frac{1}{2}$ at Optimal , with the total cost of $C(f^*) = 1 * \frac{1}{2} + \frac{1}{2} * \frac{1}{2} = \frac{3}{4}$. Therefore $\text{POA}(\Gamma) = \frac{4}{3}$. In fact the Price of Anarchy for any instance with linear latency function is upper-bounded by $\frac{4}{3}$,[2] and the result is summerised in the next section.



(a) Initial Configuration

(b) Augmented Configuration

Figure 2.2: Braess's Paradox

2.4.2 Braess's Paradox

Another famous example is the Braess's Paradox with slight modification. This paradox shows that adding a "helpful" link might actually negatively affect the traffic network by increasing the Price of Anarchy. For the configuration in Figure 2.2a the Nash Equilibrium and Optimal Flow have the same load profile x, with $x_1 = x_2 = x_3 = x_4 = \frac{1}{2}$ and $C(f) = 1.0625$. Since the cost under Nash Equilibrium and Optimal flow are the same, PoA(Γ) = 1.

Now we add a new new edge between vertex u and v with the latency of 0, as shown in Figure 2.2b. Intuitively adding the new edge would not increase the flow for Nash Equilibrium since agents can travel from u to v freely without incurring any additional cost. However if we derive the Nash Equilibrium and Optimal Flow from Corollary 2.5 and Definition 2.2 respectively, we obtain load profile $x_1 = x_4 = x_3 = 1$, $x_2 = x_3 = 0$ and total cost $C(f) = 2$ for Nash Equilibrium, and for Optimal Flow load profile would be $x_1 = x_4 = 0.2^{0.25}, x_3 = x_2 = 1 - 0.2^{0.25}, x_5 = 2 \cdot 0.2^{0.25} - 1$. Combining the result yields PoA(Γ) ≈ 2.151, which means that adding the new edge with 0 latency increases the total cost at Nash Equilibrium and hence the Price of Anarchy, contrary to our intuition. Note that the Price of Anarchy computed is exactly the upper bound for quartic functions as shown in section 2.5.2.

2.5 Bounds for Price of Anarchy

Bounding the Price of Anarchy is one of the main issue in traffic routing problem. Researchers had proposed different upper bounds for specific sets of latency functions, and in this section the upper bounds are summerised, while providing the lower bound for completeness.

2.5.1 Lower Bound

The lower bound is trivial: for any Γ, $\text{PoA}(\Gamma) \geq 1$ since by definition, total cost $C(f^*)$ induced by optimal flow $f^* \in \mathcal{F}$ is minimal by Definition 2.2. In other words for any $f \in \mathcal{F}$, $C(f) \geq C(f^*)$, and the result follows.

2.5.2 General Upper Bounds

Roughgarden[13] shows that the upper bound for Price of Anarchy does not depend on the topology of the network graph configuration, but instead on the class of the latency functions in the game. Also the analytical formula for upper bounds are derived in the same paper, summerised in the following table[13]:

Latency Function Class	Representation	Worst Case PoA
Linear	$ax + b$	$\frac{4}{3} \approx 1.333$
Quadratic	$ax^2 + bx + c$	$\frac{3\sqrt{3}}{3\sqrt{3}-2} \approx 1.626$
Cubic	$ax^3 + bx^2 + cx + d$	$\frac{4\sqrt[3]{4}}{4\sqrt[3]{4}-3} \approx 1.896$
Quartic	$ax^4 + bx^3 + cx^2 + dx + e$	$\frac{5\sqrt[4]{5}}{5\sqrt[4]{5}-4} \approx 2.151$
Polynomials of degree $\leq$ p	$\sum_{i=0}^{p} a_i x^i$	$\frac{(p+1)\sqrt[p]{p+1}}{(p+1)\sqrt[p]{p+1}-p} = \Theta(\frac{p}{\ln p})$
M/M/1 Delay Functions	$(u - x)^{-1}$	$\frac{1}{2}(1 + \sqrt{\frac{u_{min}}{u_{min}-R_{max}}})$

2.5.3 Another perspective of Upper Bound: Smooth Game Condition

Roughgarden has also proposed the notion of Smooth Game Condition, which leads the way to compute the upper bound of Price of Anarchy with the following definition:

Definition 2.8 (Smooth Game Condition[14]). *A game is (λ, μ)-smooth if the following inequality holds for optimal flow load profile x^{OPT} and Nash Equilibrium load profile x^{EQ} for some $0 \leq \lambda \leq 1$, $0 \leq \mu < 1$.*

$$\sum_{e \in \mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} \leq \lambda \sum_{e \in \mathcal{E}} c_e(x_e^{OPT})x_e^{OPT} + \mu \sum_{e \in \mathcal{E}} c_e(x_e^{EQ})x_e^{EQ}$$

Definition 2.9 (PoA under Smooth Condition[14]). *The Price of Anarchy Γ under Smoothing Condition defined above is*

$$inf\{\frac{\lambda}{1-\mu} : (\lambda, \mu) \text{ such that the game is } (\lambda, \mu) \text{ smooth}\}$$

Definition 2.9 follows directly from definition 2.8, which whenever (λ, μ) is satisfied in definition 2.8 then

$$\text{PoA}(\Gamma) = \frac{\sum_{e \in \mathcal{E}} c_e(x_e^{EQ})x_e^{EQ}}{\sum_{e \in \mathcal{E}} c_e(x_e^{OPT})x_e^{OPT}} \leq \frac{\lambda}{1-\mu}$$

Smooth Game Condition would be one of most important inequality that we would investigate and bound in order to improve the upper bound of Price of Anarchy in the rest of the report.

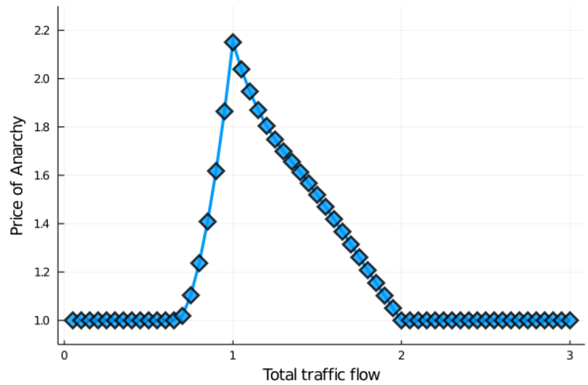


Figure 2.3: Price of Anarchy against traffic flow in Braess's Paradox

2.6 Real-world Driven Data: Empirical Analysis

In this section we will utilise the tool `TrafficAssignment.jl` developed by Kwon[15], which computes the Nash Equilibrium of a routing game condfiguration with polynomial latency function. Since $\ell_e(e_x) = c_e(x_e)x_e$ is a convex function for all polynomial latency function c_e, we could use Corollary 2.4 and computes the Optimal Flow by providing the derivative $\ell'_e(e_x)$.

In the rest of this section, two synthetic examples being shown are generated from the modified version of `TrafficAssignment.jl`, as well as the data set provided from the Transportation Network Test Problem (TNTP)[16].

2.6.1 Braess's Paradox

In this example, we would consider the setting of Braess's Paradox, i.e. section 2.4.2 and figure 2.2. Figure 2.3 plots Price of Anarchy against different total traffic inflow and individual traffic demand from node s to node t.

To make sure the graph matches the Price of Anarchy derived on section 2.4.2, observe that when the total traffic flow in Braess's Paradox is 1 the Price of Anarchy approximately equals to 2.151, which matches the result expected. Moreover when the total traffic flow is very small or very large, the Price of Anarchy is close to 1. This characteristic regarding to the total traffic flow has been formalised by

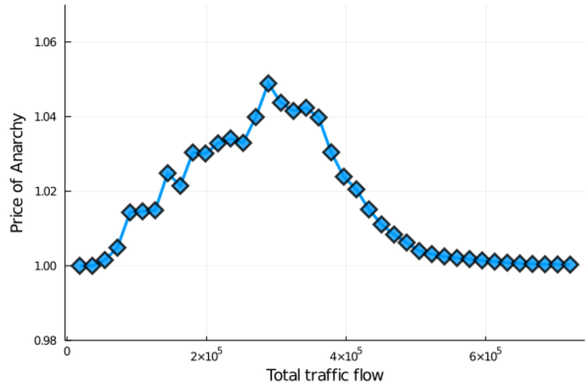


Figure 2.4: Price of Anarchy against traffic flow in Sioux Falls

Colini-Baldeschi et al,[8].

2.6.2 Sioux Falls

This example is a real-world data set from Sioux Falls, USA, which is widely studied in other papers since it is realistic enough but not too complicated. In this example there are 24 nodes and 76 edges, and the complete graph is shown in Appendix A. Same as Braess's Paradox, a graph of Price of Anarchy against different traffic flow is plotted as Figure 2.4.

Once again, it can be confirmed that when the traffic is small or large, the Price of Anarchy tends to 1. Another observation is that the maximum Price of Anarchy is around 1.05, meanwhile the latency function are quartic functions with the form of $c_e(x_e) = a_e + b_e x^4$ with $a_e, b_e \geq 0$ [16], hence having a theoretical upper bound of 2.151 derived from section 2.5.2. This shows that the gap of theoretical upper bound and real-world driven data is still huge (2.151 compared to 1.05).

3 Upper Bound with Smooth Game Condition: Without prior knowledge

In the rest of the report we would assume the traffic flow follows the cost function suggested by the Bureau of Public Roads[7], i.e. $c_e(x_e) = a_e + b_e x_e^4$, since most of the real-world model uses this set of cost function as their configuration, we could then evaluate the theories developed under this assumption against real-world data.

In this section we aim to show that under Smooth Game Condition defined in Definition 2.8 we can achieve the same upper bound of Price of Anarchy in Section 2.5.2, i.e. $\text{PoA}(\Gamma) \leq 1/(1 - \frac{4}{5^{1.25}}) = \gamma$, without using extra information. This result matches our expectation since we have constructed an example where $\text{PoA}(\Gamma)$ is exactly equal to γ in section 2.4.2. We would also show that $\text{PoA}(\Gamma) < 1/(1 - \frac{4}{5^{1.25}}) = \gamma$ under certain choice of cost functions.

Before showing the main results, we shall prove the following inequality as a lemma.

Lemma 3.1. *For a game with cost functions* $c_e(x_e) = a_e + b_e x_e^4$,

$$(c_e(x_e^{EQ}) - c_e(x_e^{OPT}))x_e^{OPT} \leq \frac{4}{5^{1.25}} c_e(x_e^{EQ}) x_e^{EQ}$$

Proof. Consider the following 4 cases:

Case 1: The inequality holds when $x_e^{EQ} = 0$ since it implies that $c_e(x_e^{EQ}) \leq c_e(x_e^{OPT})$ and hence

$$(c_e(x_e^{EQ}) - c_e(x_e^{OPT}))x_e^{OPT} \leq 0 = \frac{4}{5^{1.25}} c_e(x_e^{EQ}) x_e^{EQ}$$

Case 2: The inequality also holds when $c_e(x_e^{EQ}) = 0$, which implies that

$$-c_e(x_e^{OPT}) x_e^{OPT} \leq 0 = \frac{4}{5^{1.25}} c_e(x_e^{EQ}) x_e^{EQ}$$

Case 3: The inequality holds when $b_e = 0$:

$$(a_e - a_e) x_e^{OPT} = 0 \leq \frac{4}{5^{1.25}} a_e x_e^{EQ}$$

Case 4: Assume that $c_e(x_e^{EQ})x_e^{EQ} \neq 0$ and $b_e \neq 0$, showing the above inequality is equivalent to show that

$$\frac{(c_e(x_e^{EQ}) - c_e(x_e^{OPT}))x_e^{OPT}}{c_e(x_e^{EQ})x_e^{EQ}} \leq \frac{4}{5^{1.25}} \tag{1}$$

Now we would like to maximise the numerator with respect to x_e^{EQ} in order to bound the expression on the left. Consider the following expression:

$$\begin{aligned} A &= (c_e(x_e^{EQ}) - c_e(x_e^{OPT}))x_e^{OPT} \\ &= ((a_e + b_e(x_e^{EQ})^4 - (a_e + b_e(x_e^{OPT})^4)x_e^{OPT} \\ &= (b_e(x_e^{EQ})^4 - b_e(x_e^{OPT})^4)x_e^{OPT} \end{aligned}$$

Differentiate A with respect to x_e^{OPT} yields

$$\frac{dA}{dx_e^{OPT}} = b_e(x_e^{EQ})^4 - 5b_e(x_e^{OPT})^4 \tag{2}$$

Setting (2) to 0 yields the extrema of:

$$x_e^{OPT*} = \frac{x_e^{EQ}}{5^{0.25}} \tag{3}$$

assuming that $b_e \neq 0$. To show that x_e^{OPT*} attains minimum, consider the second derivative of A:

$$\left(\frac{d}{dx_e^{OPT}}\right)^2 A\bigg|_{x_e^{OPT}=x_e^{OPT*}} = -20b_e(x_e^{OPT*})^3 \leq 0$$

Hence the maximum of A is

$$A' = \frac{4}{5^{1.25}} b_e(x_e^{EQ})^5$$

And the maximum value of the left side expression in (1) is

$$\frac{4}{5^{1.25}} \frac{b_e(x_e^{EQ})^5}{(a_e + b_e(x_e^{EQ})^4)x_e^{EQ}} \leq \frac{4}{5^{1.25}}$$

since $b_e(x_e^{EQ})^5/(a_e + b_e(x_e^{EQ})^4)x_e^{EQ} \leq 1$ for all $a_e, x_e^{EQ} \geq 0$, $b_e > 0$. ■

The following theorem utilises the definition of Smooth Game Condition (Definition 2.8), which shows that the upper bound of Price of Anarchy of cost functions in the form of $c_e(x_e) = a_e + b_e x_e^4$.

Theorem 3.2. *For a game with cost functions* $c_e(x) = a_e + b_e x_e^4$*,* $PoA(\Gamma) \leq \gamma \approx 2.151$ *under Smooth Game Condition (Definition 2.8).*

Proof.

$$\begin{aligned}
\sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} &= \lambda \sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT} + \sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} - \lambda \sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT} \\
&= \lambda \sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT} + (1-\lambda) \sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} \\
&\qquad + \lambda(\sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} - \sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT}) \\
&\leq \lambda \sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT} + (1-\lambda) \sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} \\
&\qquad + \lambda(\sum_{e\in\mathcal{E}} (c_e(x_e^{EQ}) - c_e(x_e^{OPT}))x_e^{OPT}) \qquad (4) \\
&\leq \lambda \sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT} + (1-\lambda) \sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} \\
&\qquad + \frac{4}{5^{1.25}}\lambda \sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} \qquad (5) \\
&= \lambda \sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT} + (1-\lambda+\frac{4}{5^{1.25}}\lambda) \sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ}
\end{aligned}$$

$$\begin{aligned}
&\Rightarrow \lambda(1-\frac{4}{5^{1.25}}) \sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} \leq \lambda \sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT} \\
&\Rightarrow (1-\frac{4}{5^{1.25}}) \sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} \leq \sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT} \\
&\Rightarrow \mathrm{PoA}(\Gamma) = \frac{\sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ}}{\sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT}} \leq \frac{1}{1-\frac{4}{5^{1.25}}} = \gamma \approx 2.151
\end{aligned}$$

Where inequality (4) arrives from theorem 2.7, and inequality (5) from lemma 3.1.

■

Notice that Pigou's Example with one edges having cost function $c_e(x_e) = 1$ and another with $c_e(x_e) = x_e^4$ leads to the Price of Anarchy of 2.151, which matches the upper bound shown above.

The following corollary shows that if we exclude constant cost functions in the game, i.e. cost function in the form of $c_e(x) = a_e$, then Price of Anarchy $\mathrm{PoA}(\Gamma) < \gamma \approx 2.515$.

Corollary 3.3. *For a game with cost functions $c_e(x_e) = a_e + b_e x_e^4$ excluding constant cost functions $c_e(x_e) = a_e$, i.e. $b_e \neq 0$ for all $e \in \mathcal{E}$, $PoA(\Gamma) < \gamma$ under Smooth Game Condition (Definition 2.8).*

Proof. By way of contradiction, assume that there are some games which excludes the constant cost functions but having $\mathrm{PoA}(\Gamma) = \gamma \approx 2.151$, then from Theorem 3.2 we have to show equality instead of inequality in (4) and (5).

In particular consider equality (5), which is we would like to prove

$$\sum_{e\in\mathcal{E}} (c_e(x_e^{EQ}) - c_e(x_e^{OPT}))x_e^{OPT} = \frac{4}{5^{1.25}} \sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ}$$

But then it implies to prove

$$(c_e(x_e^{EQ}) - c_e(x_e^{OPT}))x_e^{OPT} = \frac{4}{5^{1.25}} c_e(x_e^{EQ})x_e^{EQ} \tag{6}$$

individually since from lemma 3.1 $(c_e(x_e^{EQ}) - c_e(x_e^{OPT}))x_e^{OPT} \leq \frac{4}{5^{1.25}} c_e(x_e^{EQ})x_e^{EQ}$ for all $e \in \mathcal{E}$.

To show equality (6) we can divide it into cases analogous to lemma 3.1:

Case 1: When $x_e^{EQ} = 0$, equality (6) then becomes

$$-b_e(x_e^{OPT})^5 = 0$$

by substituting $c_e(x_e) = a_e + b_e(x_e)^4$ and $x_e^{EQ} = 0$, hence in this case $x_e^{OPT} = 0$ since $b_e \neq 0$ from the assumption.

Case 2: When $c_e(x_e^{EQ}) = 0$, then there are two possibilities:

Case 2.1: $c_e(x_e) = 0$ for all $x_e \geq 0$, but this is not possible from the assumption $b_e \neq 0$.

Case 2.2: $c_e(x_e) = b_e(x_e)^4$ and $x_e^{EQ} = 0$, then it is the same as Case 1.

Case 3: $b_e = 0$ is not possible from the assumption. Note that if $b_e = 0$ is allowed then $(c_e(x_e^{EQ}) - c_e(x_e^{OPT}))x_e^{OPT} = 0$ for all $x_e^{OPT} \geq 0$, which implies we could choose any x_e^{OPT} with $x_e^{EQ} = 0$ and it still satisfy the equality.

Case 4: A necessary condition for equality (6) to hold is when x_e^{OPT} attains maximum in $(c_e(x_e^{EQ}) - c_e(x_e^{OPT}))x_e^{OPT}$, i.e. the condition in equation (3). In other words, $x_e^{OPT*} = x_e^{EQ}/5^{0.25}$.

Combining conditions in Case 1 and Case 4,

$$x_e^{OPT} = \frac{x_e^{EQ}}{5^{0.25}} \tag{7}$$

for all $e \in \mathcal{E}$ would be the necessary condition in order to achieve equality (6). But this constraint leads to a contradiction. Notice that equation (7) implies that $x_e^{OPT} < x_e^{EQ}$ when $x_e^{EQ} \neq 0$ and $x_e^{OPT} = x_e^{EQ} = 0$ when $x_e^{EQ} = 0$.

By the definition of Nash Equilibrium, for all paths $p_1, p_2 \in \mathcal{P}^i$ with flow $f_{p_1} > 0$, $c_{p_1}(f) \leq c_{p_2}(f)$, which is,

$$\sum_{e \in p_1} (a_e + b_e(x_e^{EQ})^4) \leq \sum_{e \in p_2} (a_e + b_e(x_e^{EQ})^4)$$

Also by the definition of Optimal Flow, for all paths $p_1, p_2 \in \mathcal{P}^i$ with flow $f_{p_1} > 0$, $\ell'_{p_1}(f) \leq \ell'_{p_2}(f)$, which is,

$$\sum_{e \in p_1} (a_e + 5b_e(x_e^{OPT})^4) \leq \sum_{e \in p_2} (a_e + 5b_e(x_e^{OPT})^4) \tag{8}$$

Substituting x_e^{OPT} from equation (7) into equation (8),

$$\sum_{e \in p_1} (a_e + 5b_e(\frac{x_e^{EQ}}{5^{0.25}})^4) \leq \sum_{e \in p_2} (a_e + 5b_e(\frac{x_e^{EQ}}{5^{0.25}})^4)$$
$$\Rightarrow \sum_{e \in p_1} (a_e + b_e(x_e^{EQ})^4) \leq \sum_{e \in p_2} (a_e + b_e(x_e^{EQ})^4)$$

which is exactly the equation for Nash Equilibrium. This implies that flow x_e^{OPT} is also a valid Nash Equilibrium flow, and since $x_e^{OPT} < x_e^{EQ}$, that contradicts the definition of Nash Equilibrium on x_e^{EQ} since x_e^{EQ} is not the minimum flow for Nash Equilibrium. ■

4 Smooth Game Condition: Upper Bounded by minimum Equilibrium Flow with constraints

In this section we aim to build a tighter bound with the information of the normalised minimum Equilibrium Flow x'^{EQ}_{min}, together with certain constraints ("the constraints") under Smooth Game Condition (Definition 2.8). Combining these two elements we could construct an improved upper bound. This idea is originally presented by Correa et al. [17] and this section generalises the proof to Smooth Game Condition.

In particular the constraints are

- constant cost functions of form $c_e(x_e) = a_e$ are not allowed, except the function $c_e(x_e) = 0$.
- edges with equilibrium flow $x_e^{EQ} = 0$ are not allowed, except the case where $x_e^{OPT} = 0$ or the cost function is in the form of $c_e(x_e) = b_e x_e^4$.

And normalised Equilibrium Flow x'_e is defined as follows:

Definition 4.1 (Normalised Equilibrium Flow)**.** *For all edges $e \in \mathcal{E}$ with cost functions $c_e(x_e) = a_e + b_e x_e^4$, the normalised Equilibrium Flow x'_e is*

$$x'_e = \frac{x_e}{a_e'^{0.25}}$$

where

$$a'_e = \frac{a_e}{b_e}$$

Note that in certain case x'_e can be 0 or undefined.

Before showing the main results, we shall prove the following inequality as a lemma.

Lemma 4.2. *For a game with cost functions* $c_e(x_e) = a_e + b_e x_e^4$ *with the constraints described above,*

$$(c_e(x_e^{EQ}) - \lambda c_e(x_e^{OPT}))x_e^{OPT} \leq \frac{4}{5^{1.25}\lambda^{0.25}} c_e(x_e^{EQ})x_e^{EQ}$$

for all λ *satisfying* $0.2 < 1 - \{[(1 + (x_e'^{EQ})^4)x_e'^{EQ}]^{0.8} - (x_e'^{EQ})^4\} \leq \lambda \leq 1$*, where* $x_e'^{EQ} = x_e^{EQ}/a_e'^{0.25}$*, and* $a_e' = a_e/b_e$*. Otherwise if* $x_e'^{EQ}$ *is undefined or* $x_e'^{EQ} = 0$*, then the above inequality is satisfied for all* $0 < \lambda \leq 1$*.*

Proof. Consider the following 4 cases:

Case 1: $x_e^{EQ} = 0$ is not allowed from the constraints. On the contrary if we allow $x_e^{EQ} = 0$ to exist then we have to prove

$$(a_e(1 - \lambda) - \lambda b_e(x_e^{OPT})^4)x_e^{OPT} \leq 0$$

which does not always hold unless $\lambda = 1$, $x_e^{OPT} = 0$ or $a_e = b_e = 0$.

Case 2: When $c_e(x_e^{EQ}) = 0$, then there are two possibilities:

Case 2.1: $c_e(x_e) = 0$ for all $x_e \geq 0$, then

$$0 = (0 - 0)x_e^{OPT} \leq \frac{4}{5^{1.25}\lambda^{0.25}} 0 * x_e^{EQ} = 0$$

which holds for all λ. In this case $x_e'^{EQ}$ is undefined.

Case 2.2: $c_e(x_e) = b_e(x_e)^4$ and $x_e^{EQ} = 0$, then

$$-\lambda b_e(x_e^{OPT})^5 \leq \frac{4}{5^{1.25}\lambda^{0.25}} 0 * x_e^{EQ} = 0$$

which holds for all λ. In this case $x_e'^{EQ} = 0$.

Case 3: $b_e = 0$ is not allowed from the constraints. If we allow $b_e = 0$ then we have to prove

$$a_e(1 - \lambda)x_e^{OPT} \leq \frac{4}{5^{1.25}\lambda^{0.25}} b_e(x_e^{EQ})^5$$

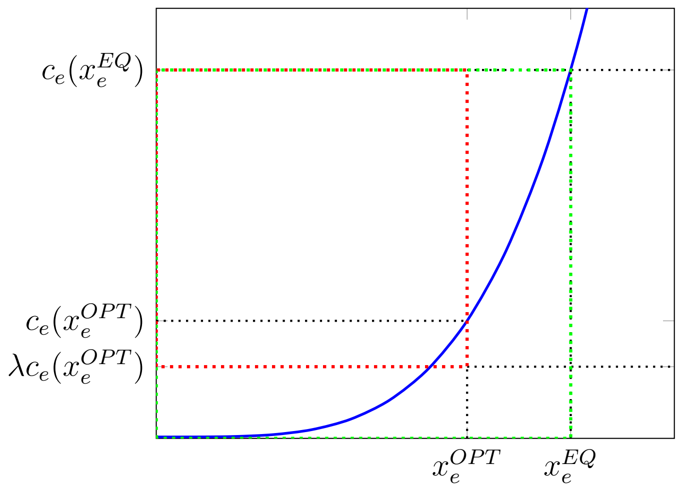


Figure 4.5: Illustration of Lemma 4.2

which does not hold generally, especially then $x_e^{OPT} >> x_e^{EQ}$.

Case 4: Otherwise assume that $c_e(x_e^{EQ})x_e^{EQ} \neq 0$ and $b_e \neq 0$, showing the above inequality is equivalent to show that

$$\begin{aligned}
&\frac{(c_e(x_e^{EQ}) - \lambda c_e(x_e^{OPT}))x_e^{OPT}}{c_e(x_e^{EQ})x_e^{EQ}} \leq \frac{4}{5^{1.25}\lambda^{0.25}} \\
\Leftrightarrow &\frac{[a_e(1-\lambda) + b_e(x_e^{EQ})^4 - \lambda b_e(x_e^{OPT})^4]x_e^{OPT}}{(a_e + b_e(x_e^{EQ})^4)x_e^{EQ}} \leq \frac{4}{5^{1.25}\lambda^{0.25}} \\
\Leftrightarrow &\frac{[a'_e(1-\lambda) + (x_e^{EQ})^4 - \lambda(x_e^{OPT})^4]x_e^{OPT}}{(a'_e + (x_e^{EQ})^4)x_e^{EQ}} \leq \frac{4}{5^{1.25}\lambda^{0.25}} \qquad (9)
\end{aligned}$$

where $a'_e = a_e/b_e$ and $b_e \neq 0$ from the assumption.

Using similar approach from lemma 3.1, we have to maximise the numerator with respect to x_e^{EQ}. Graphically, we would like to maximise the area of the red rectangle in Figure 4.5 with respect to the green rectangle. Now consider the following expression:

$$A = [a'_e(1-\lambda) + (x_e^{EQ})^4 - \lambda(x_e^{OPT})^4]x_e^{OPT}$$

Differentiate A with respect to x_e^{OPT} while fixing λ and x_e^{EQ} yields

$$\frac{dA}{dx_e^{OPT}} = a'_e(1-\lambda) + (x_e^{EQ})^4 - 5\lambda(x_e^{OPT})^4 \qquad (10)$$

Setting (10) to 0 yields the extrema of:

$$x_e^{OPT*} = \Big[a'_e(\frac{1-\lambda}{5\lambda}) + (\frac{1}{5\lambda})(x_e^{EQ})^4\Big]^{0.25} \tag{11}$$

To show that x_e^{OPT*} attains minimum, consider the second derivative of A:

$$\Big(\frac{d}{dx_e^{OPT}}\Big)^2 A\Big|_{x_e^{OPT}=x_e^{OPT*}} = -20\lambda(x_e^{OPT*})^3 \leq 0$$

Hence the maximum of A is

$$A' = \frac{4}{5^{1.25}\lambda^{0.25}}[a'_e(1-\lambda) + (x_e^{EQ})^4]^{1.25}$$

And the maximum value of the left side expression in (9) is

$$\frac{4}{5^{1.25}\lambda^{0.25}}\Big(\frac{[a'_e(1-\lambda) + (x_e^{EQ})^4]^{1.25}}{(a'_e + (x_e^{EQ})^4)x_e^{EQ}}\Big) \tag{12}$$

$$\Leftrightarrow \frac{4}{5^{1.25}\lambda^{0.25}}\Big(\frac{[(1-\lambda) + (x_e'^{EQ})^4]^{1.25}}{(1 + (x_e'^{EQ})^4)x_e'^{EQ}}\Big) \tag{13}$$

with $x_e'^{EQ} = x_e^{EQ}/a_e'^{0.25}$ and $a'_e \neq 0$.

Case 4.1: If $a'_e = 0$, expression (12) becomes

$$\frac{4}{5^{1.25}\lambda^{0.25}}\Big(\frac{(x_e^{EQ})^5}{(x_e^{EQ})^5}\Big) = \frac{4}{5^{1.25}\lambda^{0.25}}$$

In this case inequality (9) holds trivially and $x_e'^{EQ}$ is undefined.

Case 4.2: Otherwise combining (9) and (13), we need to show that

$$\frac{4}{5^{1.25}\lambda^{0.25}}\Big(\frac{[(1-\lambda) + (x_e'^{EQ})^4]^{1.25}}{(1 + (x_e'^{EQ})^4)x_e'^{EQ}}\Big) \leq \frac{4}{5^{1.25}\lambda^{0.25}} \tag{14}$$

In other words,

$$\underbrace{\Big(\frac{[(1-\lambda) + (x_e'^{EQ})^4]^{1.25}}{(1 + (x_e'^{EQ})^4)x_e'^{EQ}}\Big)}_{f(\lambda, x_e'^{EQ})} \leq 1$$

$$\Rightarrow \lambda \geq 1 - \{[(1 + (x_e'^{EQ})^4)x_e'^{EQ}]^{0.8} - (x_e'^{EQ})^4\} \tag{15}$$

From Figure 4.6, it can be observed that there is one root that satisfy the equation $f(\lambda, x_e'^{EQ}) = 1$ for all $0.2 < \lambda \leq 1$. Moreover if x_0 is the root of $f(\lambda, x_e'^{EQ}) = 1$ then $f(\lambda, x) < 1$ for all $x > x_0$. On the other hand, if $\lambda \leq 0.2$ then for all $x_e'^{EQ}$ we have $f(\lambda, x_e'^{EQ}) > 1$. ■

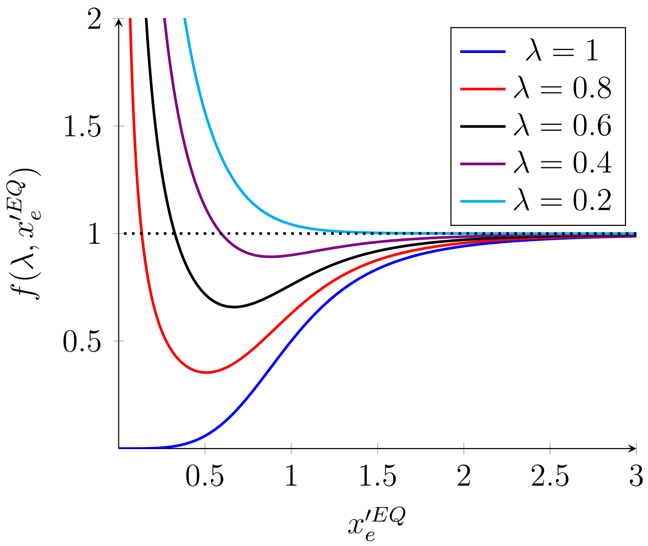


Figure 4.6: Graph of $f(\lambda, x_e'^{EQ})$ under different λ

Lemma 4.3. *For a game with cost functions $c_e(x_e) = a_e + b_e x_e^4$ with the constraints described above,*

$$\sum_{e\in\mathcal{E}}(c_e(x_e^{EQ}) - \lambda c_e(x_e^{OPT}))x_e^{OPT} \leq \frac{4}{5^{1.25}\lambda^{0.25}}\sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ}$$

for all λ satisfying $0.2 < 1 - \{[(1 + (x_{min}'^{EQ})^4)x_{min}'^{EQ}]^{0.8} - (x_{min}'^{EQ})^4\} \leq \lambda \leq 1$, where $x_{min}'^{EQ} = \min_e x_e'^{EQ}$, $x_e'^{EQ} = x_e^{EQ}/a_e'^{0.25}$ and $a_e' = a_e/b_e$, excluding the case where $x_e'^{EQ}$ is undefined or $x_e'^{EQ} = 0$.

Proof. The idea is to select a range of λ such that for all $e \in \mathcal{E}$ the inequality

$$(c_e(x_e^{EQ}) - \lambda c_e(x_e^{OPT}))x_e^{OPT} \leq \frac{4}{5^{1.25}\lambda^{0.25}} c_e(x_e^{EQ})x_e^{EQ} \tag{16}$$

holds. If we could find such range of λ, then

$$\sum_{e\in\mathcal{E}}(c_e(x_e^{EQ}) - \lambda c_e(x_e^{OPT}))x_e^{OPT} \leq \frac{4}{5^{1.25}\lambda^{0.25}}\sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} \tag{17}$$

also holds.

From lemma 4.2, inequality (16) holds when

$$\lambda_e \geq 1 - \{[(1 + (x_e'^{EQ})^4)x_e'^{EQ}]^{0.8} - (x_e'^{EQ})^4\} = g(x_e'^{EQ})$$

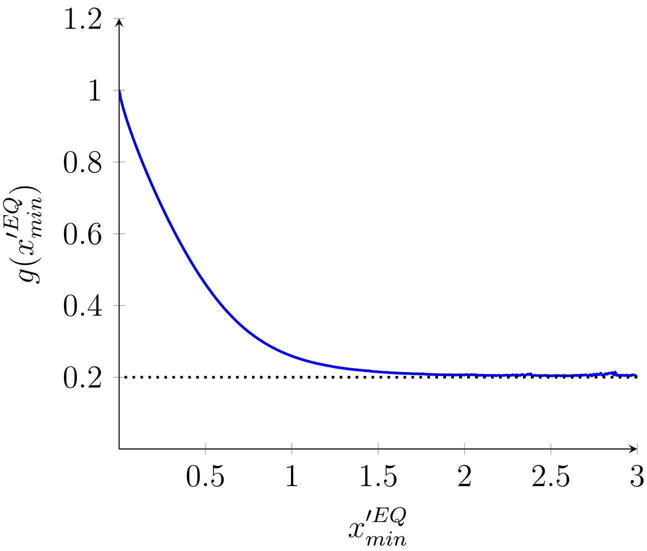


Figure 4.7: Graph of $g(x'^{EQ}_{min})$ against x'^{EQ}_{min}

for all $e \in \mathcal{E}$. Figure 4.7 plots the graph of $g(x'^{EQ}_{e})$ against x'^{EQ}_{e}. Since $g(x'^{EQ}_{e})$ is a decreasing function, we can choose x'^{EQ}_{min} so that

$$\lambda \geq 1 - \{[(1 + (x'^{EQ}_{min})^4)x'^{EQ}_{min}]^{0.8} - (x'^{EQ}_{min})^4\}$$

with this range of λ, inequality (16) holds for all $e \in \mathcal{E}$ and hence inequality (17) also holds.

Also from lemma 4.2 when x'^{EQ}_{e} is undefined or x'^{EQ}_{e} then inequality (16) holds for all λ, so it is excluded in finding bounds for λ.

■

Theorem 4.4. *For a game with cost function $c_e(x_e) = a_e + b_e x_e^4$ with the constraints described above,*

$$PoA(\Gamma) \leq \frac{\lambda^*}{1 - \frac{4}{5^{1.25}\lambda^{*0.25}}}$$

where $\lambda^* = 1 - \{[(1 + (x'^{EQ}_{min})^4)x'^{EQ}_{min}]^{0.8} - (x'^{EQ}_{min})^4$.

Proof.

$$\begin{aligned}\sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} \leq& \sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{OPT}\\ =&\lambda\sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT} + \sum_{e\in\mathcal{E}}(c_e(x_e^{EQ}) - \lambda c_e(x_e^{OPT}))x_e^{OPT}\\ \leq&\lambda\sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT} + \frac{4}{5^{1.25}\lambda^{0.25}}\sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} \qquad (18)\end{aligned}$$

$$\Rightarrow (1 - \frac{4}{5^{1.25}\lambda^{0.25}})\sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} \leq \lambda\sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT}$$

$$\Rightarrow \text{PoA}(\Gamma) = \frac{\sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ}}{\sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT}} \leq \frac{\lambda}{1 - \frac{4}{5^{1.25}\lambda^{0.25}}} = h(\lambda) \qquad (19)$$

From lemma 4.3, inequality (18) holds if

$$0.2 < 1 - \{[(1 + (x'^{EQ}_{min})^4)x'^{EQ}_{min}]^{0.8} - (x'^{EQ}_{min})^4\} \leq \lambda \leq 1 \qquad (20)$$

We now need to find the minimal value of $h(\lambda)$ with constraint (20). Figure 4.8 shows the graph of $h(\lambda)$ against λ. Since $h(\lambda)$ is an increasing function, we have to take λ as small as possible while satisfying constraint (20). Hence $\lambda^* = 1 - \{[(1 + (x'^{EQ}_{min})^4)x'^{EQ}_{min}]^{0.8} - (x'^{EQ}_{min})^4$ yields the minimal $h(\lambda^*)$.

Substituting $\lambda = \lambda^*$ into inequality (19) gives the desired statement in the theorem. ■

The following corollary shows that on certain extra condition, Price of Anarchy $\text{PoA}(\Gamma) < \lambda/(1 - \frac{4}{5^{1.25}\lambda^{0.25}})$.

Corollary 4.5. *For a game with cost function $c_e(x_e) = a_e + b_e x_e^4$ with the constraints described above, if there exist a pair of $e_1, e_2 \in \mathcal{E}$, $e_1 \neq e_2$, satisfying the following condition:*

- $c_{e_1}(x_{e_1})x_{e_1} \neq 0$ *and* $c_{e_2}(x_{e_2})x_{e_2} \neq 0$
- $a_{e_1} \neq 0$ *and* $a_{e_2} \neq 0$
- $x'^{EQ}_{e_1} \neq x'^{EQ}_{e_2}$

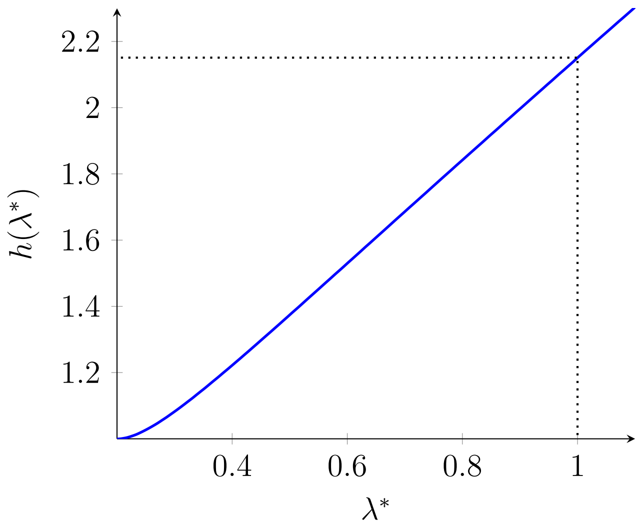


Figure 4.8: Graph of $h(\lambda)$ against λ

Then

$$PoA(\Gamma) < \frac{\lambda}{1 - \frac{4}{5^{1.25}\lambda^{0.25}}}$$

Proof. The first and second conditions alongside with the constraints excludes Case 1, 2, 3 in lemma 4.3, so we only have to consider case 4.

From Figure 4.6 we can see that $f(\lambda, x_{min}^{\prime EQ}) = 1$ only has one root for $0.2 < \lambda \leq 1$. Moreover for $x > x_{min}^{\prime EQ}$, $f(\lambda, x) < 1$. From condition 3 we know that at least one of the edges $x_e^{\prime EQ} > x_{min}^{\prime EQ}$. Therefore we can replace $\leq$ with $<$ in inequality (18) from Theorem 4.4. Hence inequality (19) becomes

$$\text{PoA}(\Gamma) < \frac{\lambda}{1 - \frac{4}{5^{1.25}\lambda^{0.25}}}$$

■

To conclude, under certain restrictions we could lower the upper bound to $\text{PoA}(\Gamma) \leq \lambda/(1 - \frac{4}{5^{1.25}\lambda^{0.25}})$ where λ depends on the normalised minimum flow in the game. An interesting observation is that the choice of λ according to minimum flow represents the worst case scenario, but for most of the case it is entirely possible to choose a lower λ and inequality (18) still holds. We would investigate this idea further in section 6. In next section we would try to find another way of bounding which depends on the maximum flow in the game.

5 Smooth Game Condition: Upper Bounded by maximum Equilibrium Flow with constraints

In this section we would like to find a lower bound that depends on the normalised maximum flow instead of normalised minimum flow as proved in the previous section. Different from last section, we do not impose any constraints on the type of cost function or equilibrium. In other words this result is applicable to all configuration of games.

Before getting to the main results, we would like to introduce a new notation η. This idea is originally presented by Correa et al. [17] and this section generalises the proof to Smooth Game Condition, and to select the optimal η which are not mentioned in Correa et al.'s paper.

Lemma 5.1. *For a game with cost function $c_e(x_e) = a_e + b_e x_e^4$, there exist some $0 \leq \eta \leq 1$ such that $c_e(0) \geq \eta c_e(x_e^{EQ})$. Then $(a_e + b_e(x_e^{EQ})^4)x_e^{EQ} \geq b_e(x_e^{EQ})^5/(1-\eta)$.*

Proof. Since $c_e(x_e^{EQ}) \leq c_e(0)$, it is easy to see that there exist some η such that $c_e(0) \geq \eta c_e(x_e^{EQ})$ and hence $a_e \geq \eta(a_e + b_e(x_e^{EQ})^4)$.
Then,

$$\begin{aligned}(a_e + b_e(x_e^{EQ})^4)x_e^{EQ} &= a_e x_e^{EQ} + b_e(x_e^{EQ})^5 \\ &\geq \eta(a_e + b_e(x_e^{EQ})^4)x_e^{EQ} + b_e(x_e^{EQ})^5\end{aligned}$$

$$\begin{aligned}&\Rightarrow (1-\eta)(a_e + b_e(x_e^{EQ})^4)x_e^{EQ} \geq b_e(x_e^{EQ})^5 \\ &\Rightarrow (a_e + b_e(x_e^{EQ})^4)x_e^{EQ} \geq \frac{b_e(x_e^{EQ})^5}{1-\eta}\end{aligned}$$

■

Lemma 5.2. *For a game with cost function* $c_e(x_e) = a_e + b_e x_e^4$ *with* $\eta \leq 1/(1+(x'^{EQ}_{max})^4)$, *then* $\sum_{e\in\mathcal{E}}(a_e + b_e(x_e^{EQ})^4)x_e^{EQ} \geq \sum_{e\in\mathcal{E}} b_e(x_e^{EQ})^5/(1-\eta)$, *where* $x'^{EQ}_{max} = \max_e x'^{EQ}_e$ *excluding the case when* x'^{EQ}_e *is undefined or* $x'^{EQ}_e = 0$, $x'^{EQ}_e = x_e^{EQ}/a'^{0.25}_e$ *and* $a'_e = a_e/b_e$.

Proof. Consider the following cases with the inequality $c_e(0) \geq \eta c_e(x_e^{EQ})$:

Case 1: When $b_e = 0$, then $a_e \geq \eta(a_e + 0(x_e^{EQ})^4)$ holds for all $0 < \eta \leq 1$, this case corresponds to the situation when x'^{EQ}_e is undefined.

Case 2: When $x'^{EQ}_e = 0$, then $a_e \geq \eta(a_e + b_e(0)^4)$ holds for all $0 < \eta \leq 1$, this case corresponds to the situation when $x'^{EQ}_e = 0$.

Case 3: Otherwise for each edge $e \in \mathcal{E}$,

$$\begin{aligned} a_e &\geq \eta_e(a_e + b_e(x_e^{EQ})^4) \\ \Rightarrow \eta_e &\leq \frac{a_e}{a_e + b_e(x_e^{EQ})^4} \\ \Rightarrow \eta_e &\leq \frac{1}{1+(x'^{EQ}_e)^4} \end{aligned}$$

Now we have to find η such that $\eta \leq 1/(1+(x'^{EQ}_e)^4)$ for all $e \in \mathcal{E}$. Since $1/(1+(x'^{EQ}_e)^4)$ decreases as x'^{EQ}_e increases, so using $\eta \leq 1/(1+(x'^{EQ}_{max})^4)$ as the bound would make sure that all edges satisfy the inequality.
From lemma 5.1, all edges $e \in \mathcal{E}$ satisfy the inequality

$$(a_e + b_e(x_e^{EQ})^4)x_e^{EQ} \geq \frac{b_e(x_e^{EQ})^5}{1-\eta}$$

and hence

$$\sum_{e\in\mathcal{E}}(a_e + b_e(x_e^{EQ})^4)x_e^{EQ} \geq \frac{\sum_{e\in\mathcal{E}} b_e(x_e^{EQ})^5}{1-\eta}$$

■

Lemma 5.3. *For a game with cost functions* $c_e(x_e) = a_e + b_e x_e^4$,

$$(c_e(x_e^{EQ}) - \lambda c_e(x_e^{OPT}))x_e^{OPT} \leq \frac{4}{5^{1.25}} b_e(x_e^{EQ})^5$$

where $\lambda = 1$.

Proof. Consider the following 3 cases:

Case 1: If $x_e^{EQ} = 0$ then

$$-b_e(x_e^{OPT})^4 x_e^{OPT} \leq \frac{4}{5^{1.25}}(b_e * (0)^5) = 0$$

which always hold.

Case 2: If $b_e = 0$ then

$$0 \leq \frac{4}{5^{1.25}}(0 * (x_e^{EQ})^5) = 0$$

which always hold.

Case 3: Otherwise we would use the same approach in case 4 of lemma 3.1, then eventually we need to show that

$$\frac{4}{5^{1.25}} \underbrace{\left(\frac{[(1-\lambda) + (x_e'^{EQ})^4]^{1.25}}{(x_e'^{EQ})^4 x_e'^{EQ}}\right)}_{f(\lambda, x_e'^{EQ})} \leq \frac{4}{5^{1.25}}$$

Note that we would like to find a λ such that $f(\lambda, x_e'^{EQ}) \leq 1$. But since for all $x_e'^{EQ} \in \mathbb{R}^+$, $f(\lambda, x_e'^{EQ}) = 1$ when $\lambda = 1$, and $f(\lambda, x_e'^{EQ}) > 1$ when $0 < \lambda < 1$, the only choice for λ is 1. ■

Theorem 5.4. *For a game with cost function* $c_e(x_e) = a_e + b_e x_e^4$*,*

$$PoA(\Gamma) \leq \frac{1}{1 - (1-\eta^*)\frac{4}{5^{1.25}}}$$

where $\eta^* = 1/(1 + (x_{max}'^{EQ})^4)$*.*

Proof.

$$\begin{aligned}
\sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} \leq & \sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{OPT} \\
= & \lambda \sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT} + \sum_{e\in\mathcal{E}} (c_e(x_e^{EQ}) - \lambda c_e(x_e^{OPT}))x_e^{OPT} \\
= & \lambda \sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT} + \frac{\sum_{e\in\mathcal{E}} (c_e(x_e^{EQ}) - \lambda c_e(x_e^{OPT}))x_e^{OPT}}{\sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ}} \sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} \\
\leq & \lambda \sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT} \\
& + (1-\eta)\frac{\sum_{e\in\mathcal{E}} (c_e(x_e^{EQ}) - \lambda c_e(x_e^{OPT}))x_e^{OPT}}{\sum_{e\in\mathcal{E}} b_e(x_e^{EQ})^5} \sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} \qquad (21) \\
\leq & \lambda \sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT} + +(1-\eta)\frac{4}{5^{1.25}} \sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} \qquad (22)
\end{aligned}$$

$$\Rightarrow (1-(1-\eta)\frac{4}{5^{1.25}}) \sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ} \leq \lambda \sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT}$$

$$\Rightarrow \text{PoA}(\Gamma) = \frac{\sum_{e\in\mathcal{E}} c_e(x_e^{EQ})x_e^{EQ}}{\sum_{e\in\mathcal{E}} c_e(x_e^{OPT})x_e^{OPT}} \leq \frac{1}{1-(1-\eta)\frac{4}{5^{1.25}}} = h(\eta)$$

where inequality (21) comes from lemma 5.2, and inequality (22) comes from lemma 5.3, with summing the inequality in the lemma term by term.

Next we have to find η such that it maximises the Price of Anarchy, meanwhile satisfying the condition in lemma 5.2, which is $0 \leq \eta \leq 1/(1+(x'^{EQ}_{max})^4)$. Since $h(\eta)$ is a decreasing function as illustrated in Figure 5.9, in order to minimise $h(\eta)$ we need to choose η as large as possible. Hence $\eta^* = 1/(1+(x'^{EQ}_{max})^4)$. ■

To conclude, if we know the normalised maximum flow then we could easily obtain a better lower bound as suggest in theorem 5.4. However similar to the conclusion in section 4, this choice of η represents a worst case scenario, where we could actually take another better η while the inequality (21) still holds. These ideas will be formally presented in the next section.

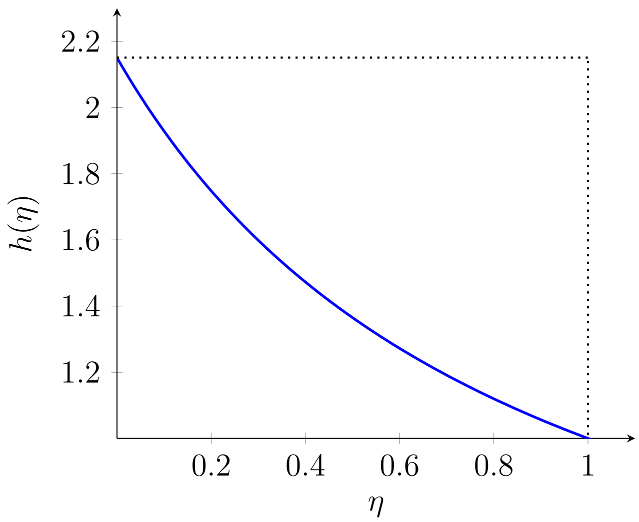


Figure 5.9: Graph of $h(\eta)$ against η

6 Smooth Game Condition: A better Upper Bound from knowing all Equilibrium Flows

On previous sections the upper bound of Price of Anarchy is determined by the maximum or minimum Equilibrium Flow, however in reality we have to compute Equilibrium Flows for all edges in order to determine to maximum and minimum flow and therefore we are interested in using Equilibrium Flow from all edges to build a tighter bound.

This section is separated into two parts: Bound improvement from Section 4 and improvement from Section 5. The main difference between these two subsections are that subsection 6.1 requires the constraint on the game introduced in section 4 but no extra constraint is needed in subsection 6.2.

6.1 Further improved bound from Section 4

The idea of Theorem 6.1 is to amortise inequality (18) so that instead of choosing the worst case λ which depends on the minimal (normalised) Equilibrium Flow, we now allows more flexibility for λ while inequality (18) still holds.

Theorem 6.1. *For a game with cost function $c_e(x_e) = a_e + b_e x_e^4$ with the constraints described in section 3,*

$$PoA(\Gamma) \leq \frac{\lambda^*}{1 - \frac{4}{5^{1.25}\lambda^{*0.25}}}$$

where λ^* is given by the following optimisation formula:

$$\begin{aligned} \min_{\lambda} \quad & \lambda \\ s.t. \quad & \sum_{e \in \mathcal{E}_3} \left(1 - \frac{[(1-\lambda) + (x_e'^{EQ})^4]^{1.25}}{(1 + (x_e'^{EQ})^4) x_e'^{EQ}}\right) c_e(x_e^{EQ}) x_e^{EQ} \geq 0 \\ & 0.2 \leq \lambda \leq 1 \end{aligned} \tag{23}$$

where edges $\mathcal{E}$ are partitioned in the following way:

- *$\mathcal{E}_1$ is the set of edges with $c_e(x_e^{EQ}) = 0$ (Case 2 in lemma 4.2)*
- *$\mathcal{E}_2$ is the set of edges with $c_e(x_e^{EQ}) \neq 0$, $b_e \neq 0$ and $a_e = 0$ (Case 4.1 in lemma 4.2)*
- *$\mathcal{E}_3$ is the set of edges with $c_e(x_e^{EQ}) \neq 0$, $b_e \neq 0$ and $a_e \neq 0$ (Case 4.2 in lemma 4.2)*

Proof. Consider a similar derivation from theorem 4.4. In particular consider inequality (18):

$$
\begin{aligned}
&\sum_{e\in\mathcal{E}}(c_e(x_e^{EQ}) - \lambda c_e(x_e^{OPT}))x_e^{OPT} \\
&\quad\leq \sum_{e\in\mathcal{E}_2\cup\mathcal{E}_3}(c_e(x_e^{EQ}) - \lambda c_e(x_e^{OPT}))x_e^{OPT} && (24)\\
&\quad= \sum_{e\in\mathcal{E}_2\cup\mathcal{E}_3}\frac{(c_e(x_e^{EQ}) - \lambda c_e(x_e^{OPT}))x_e^{OPT}}{c_e(x_e^{EQ})x_e^{EQ}}c_e(x_e^{EQ})x_e^{EQ}\\
&\quad= \sum_{e\in\mathcal{E}_2\cup\mathcal{E}_3}\frac{a_e(1-\lambda) + b_e(x_e^{EQ})^4 - \lambda b_e(x_e^{OPT})^4}{(a_e + b_e(x_e^{EQ})^4)x_e^{EQ}}c_e(x_e^{EQ})x_e^{EQ}\\
&\quad= \sum_{e\in\mathcal{E}_2\cup\mathcal{E}_3}\frac{a'_e(1-\lambda) + (x_e^{EQ})^4 - \lambda(x_e^{OPT})^4}{(a'_e + (x_e^{EQ})^4)x_e^{EQ}}c_e(x_e^{EQ})x_e^{EQ}\\
&\quad\leq \frac{4}{5^{1.25}\lambda^{0.25}}\Big(\sum_{e\in\mathcal{E}_3}\frac{[a'_e(1-\lambda) + (x_e^{EQ})^4]^{1.25}}{(a'_e + (x_e^{EQ})^4)x_e^{EQ}}c_e(x_e^{EQ})x_e^{EQ} + \sum_{e\in\mathcal{E}_2}c_e(x_e^{EQ})x_e^{EQ}\Big) && (25)\\
&\quad= \frac{4}{5^{1.25}\lambda^{0.25}}\Big(\sum_{e\in\mathcal{E}_3}\frac{[(1-\lambda) + (x_e'^{EQ})^4]^{1.25}}{(1 + (x_e'^{EQ})^4)x_e'^{EQ}}c_e(x_e^{EQ})x_e^{EQ} + \sum_{e\in\mathcal{E}_2}c_e(x_e^{EQ})x_e^{EQ}\Big)\\
&\quad\leq \frac{4}{5^{1.25}\lambda^{0.25}}\sum_{e\in\mathcal{E}_2\cup\mathcal{E}_3}c_e(x_e^{EQ})x_e^{EQ} && (26)\\
&\quad= \frac{4}{5^{1.25}\lambda^{0.25}}\sum_{e\in\mathcal{E}}c_e(x_e^{EQ})x_e^{EQ}
\end{aligned}
$$

Inequality (24) comes from Case 2 in 4.2, and inequality (25) arrives from the maximisation of the numerator shown from expression (12) as well as Case 4.1, and we would also like to make inequality (26) holds such that inequality (18) holds. Making (25) holds is equivalent to the following, noticing that $\mathcal{E}_2$ and $\mathcal{E}_3$ are disjoint sets:

$$
\begin{aligned}
\sum_{e\in\mathcal{E}_3}c_e(x_e^{EQ})x_e^{EQ} &\geq \sum_{e\in\mathcal{E}_3}\frac{[(1-\lambda) + (x_e'^{EQ})^4]^{1.25}}{(1 + (x_e'^{EQ})^4)x_e'^{EQ}}c_e(x_e^{EQ})x_e^{EQ}\\
\sum_{e\in\mathcal{E}_3}\Big(1 - \frac{[(1-\lambda) + (x_e'^{EQ})^4]^{1.25}}{(1 + (x_e'^{EQ})^4)x_e'^{EQ}}\Big)c_e(x_e^{EQ})x_e^{EQ} &\geq 0 \qquad (27)
\end{aligned}
$$

Also from Figure 4.6 and Smooth Game Condition, we know that $0.2 < \lambda \leq 1$ and Figure 4.8 we are minimising λ in order to lower the upper bound of Price of Anarchy, hence the optimisation formula follows.

Notice that the function

$$f_e(\lambda) = \Big(1 - \frac{[(1-\lambda) + (x_e'^{EQ})^4]^{1.25}}{(1+(x_e'^{EQ})^4)x_e'^{EQ}}\Big)c_e(x_e^{EQ})x_e^{EQ}$$

is strictly increasing with respect to λ and since the sum of increasing functions is also an strictly increasing function, therefore

$$f(\lambda) = \sum_{e\in\mathcal{E}_3}\Big(1 - \frac{[(1-\lambda) + (x_e'^{EQ})^4]^{1.25}}{(1+(x_e'^{EQ})^4)x_e'^{EQ}}\Big)c_e(x_e^{EQ})x_e^{EQ}$$

is an increasing function. Now the optimisation formula (23) turns into solving λ for the following equality:

$$f(\lambda) = \sum_{e\in\mathcal{E}_3}\Big(1 - \frac{[(1-\lambda) + (x_e'^{EQ})^4]^{1.25}}{(1+(x_e'^{EQ})^4)x_e'^{EQ}}\Big)c_e(x_e^{EQ})x_e^{EQ} = 0$$

Now since f(λ) is an strictly increasing function, with $f(0.2) < 0$ and $f(1) > 0$ shown from Figure 4.6, we can then find $f(\lambda^*) = 0$ for some unique $0.2 < \lambda^* \leq 1$ via Newton's method, with initial value of $\lambda = (1+0.2)/2 = 0.6$ and

$$f'(\lambda) = \sum_{e\in\mathcal{E}_3}\frac{[(1-\lambda) + (x_e'^{EQ})^4]^{0.25}}{(1+(x_e'^{EQ})^4)x_e'^{EQ}}c_e(x_e^{EQ})x_e^{EQ}$$

On the other hand if $\mathcal{E}_3 = \emptyset$, then optimisation formula (23) becomes minimising λ subject to $0 \leq 0$, which always holds. Therefore in this case we could select $\lambda = 0.2$ in order to push down upper bound for Price of Anarchy as much as possible according to Figure 4.8.

An interesting point to observe is that when $\lambda = 0.2$, PoA(Γ) ≤ 1. On the other hand if $\mathcal{E}_3 = \emptyset$, the only cost functions allowed are $c_e(x_e) = 0$ or $c_e(x_e) = b_e x_e^4$, and in this situation the exact PoA(Γ) is 1.

Finally the Price of Anarchy is given with the same equation same as from Theorem 4.4, i.e.

$$\text{PoA}(\Gamma) \leq \frac{\lambda^*}{1 - \frac{4}{5^{1.25}\lambda^{*0.25}}}$$

■

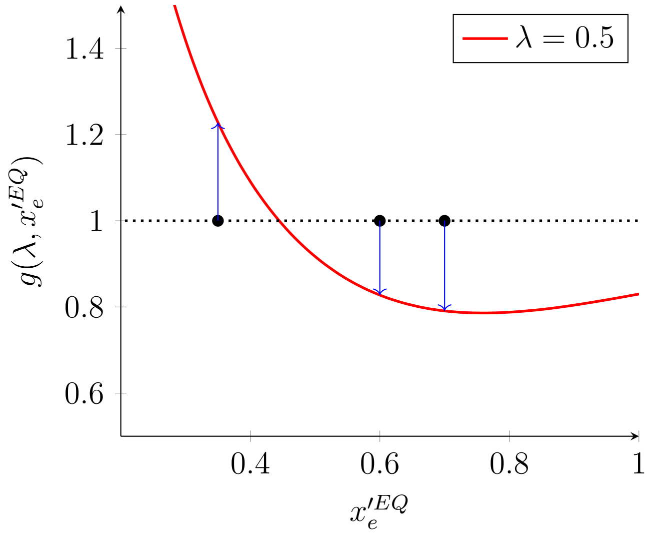


Figure 6.10: Illustration of optimisation formula (23)

As an illustration of optimisation formula (23), consider Figure 6.10 which plots the function

$$g(\lambda, x_e'^{EQ}) = \frac{[(1-\lambda) + (x_e'^{EQ})^4]^{1.25}}{(1 + (x_e'^{EQ})^4)x_e'^{EQ}}$$

and the optimisation formula aims to find an optimal value for λ such that the magnitude of the blue arrows cancel each other, where the magnitude has been scaled by a factor of $c_e(x_e)x_e$.

6.2 Further improved bound from Section 5

The following Theorem rephrases Theorem 5.4 with a replacement of worst case η with an exact value of η.

Theorem 6.2. *For a game with cost function* $c_e(x_e) = a_e + b_e x_e^4$,

$$PoA(\Gamma) \leq \frac{1}{1 - \alpha \frac{4}{5^{1.25}}}$$

where $\alpha = 1 - \eta = \sum_{e \in \mathcal{E}} b_e (x_e^{EQ})^5 / \sum_{e \in \mathcal{E}} (a_e + b_e (x_e^{EQ})^4) x_e^{EQ}$.

Proof. Replace inequality (21) with an equality by setting

$$1 - \eta = \frac{\sum_{e \in \mathcal{E}} b_e (x_e^{EQ})^5}{\sum_{e \in \mathcal{E}} (a_e + b_e (x_e^{EQ})^4) x_e^{EQ}} = \alpha$$

and the bound of PoA(Γ) comes from the rest of the proof 5.4. ∎

6.3 Summary

By combining section 6.1 and section 6.2, we can conclude the following:

Theorem 6.3. *For any game with cost function* $c_e(x_e) = a_e + b_e x_e^4$,

$$PoA(\Gamma) \leq \gamma_2$$

And for a game with cost function $c_e(x_e) = a_e + b_e x_e^4$ *with constraints in section 4 being satisfied, then*

$$PoA(\Gamma) \leq \min(\gamma_1, \gamma_2)$$

where γ_1 *is the estimate computed from Theorem 6.1 and* γ_2 *is the estimate computed from Theorem 6.2.*

7 Applying Upper Bound to Real-world Driven Data

In this section, we would apply the findings from Section 3 to Section 6 into Real-world Driven Data from the Transport Network Test Problem(TNTP)[16]. We will then compare the exact Price of Anarchy against the Upper Bound this report proposes, as well as the theoretical Upper Bound of 2.151. Last but not least, we would provide some analysis of the results.

7.1 Methodology

In order to estimate the upper bound the results in section 6, in particular Theorem 6.3 will be considered. The reason of not using results from Section 3 to Section 5 are as follows:

- Section 3 does not provide any improvement on the Upper Bound, i.e. the estimation is still 2.151.
- In order to calculate the minimum and maximum modified Equilibrium Flow required in Section 4 and 5 respectively, we need to know all Equilibrium Flow in the Game. Hence we can apply all Equilibrium Flows to Section 6 directly without extra computing work.
- Section 6 provides a better bound than Section 4 and Section 5 since Section 4 and 5 assumes a worst case scenario for the choice of λ and η

Next we would like to examine how upper bounds changes with different traffic flows under the same Game Configuration. Hence we will plot the estimate Price of Anarchy against different traffic inflow, which is scaled up or down proportionally to the original flow.

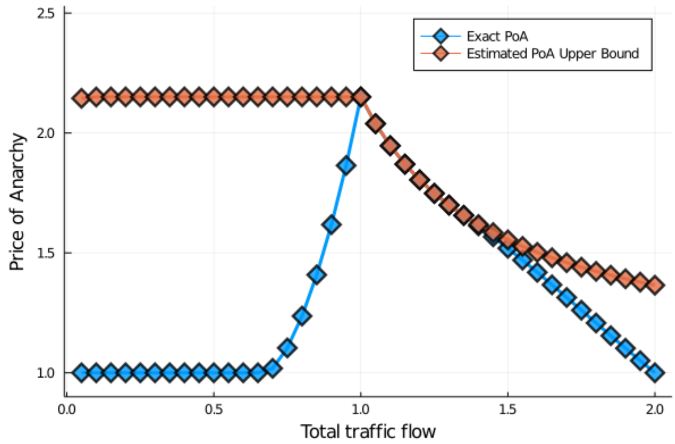


Figure 7.11: Exact and Estimated PoA against traffic flow in Braess's Paradox

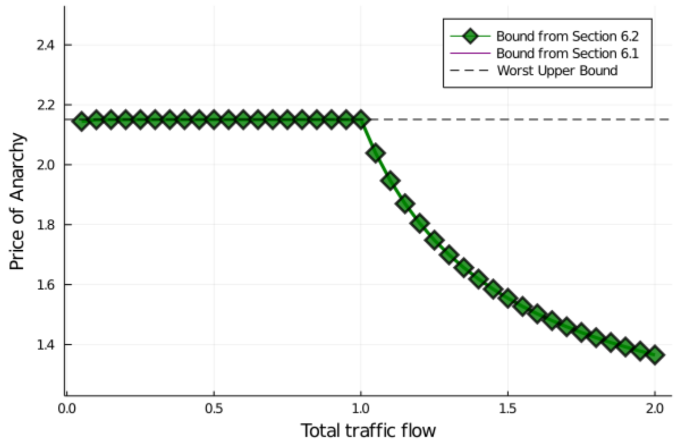


Figure 7.12: Estimated PoA with different approach of bounding in Braess's Paradox

7.2 Braess's Paradox revisited

Consider the example of Braess's Paradox with Augmented Configuration, as shown in Figure 2.2b. We would now apply the upper bound into this Configuration under different traffic flows, as shown in Figure 7.11. Note that we could not use the result γ_1 in Theorem 6.3 since the Configuration contains cost function $c_3(x_3) = c_2(x_2) = 1$, which violates first constraint stated in Section 4.

Observe that the upper bound computed from Section 6.2 stays flat when the total traffic flow ranges from 0 to 1, and decrease when the total traffic flow is greater than 1. To understand the reason, we can split it into two cases:

- When the total traffic flow ranges from 0 to 1, then all the Equilibrium flows from node s to t in Figure 2.2b travels in the following order: $s \to u \to v \to t$. This means that

$$\begin{aligned}\alpha &= \frac{\sum_{e\in\mathcal{E}} b_e(x_e^{EQ})^5}{\sum_{e\in\mathcal{E}}(a_e + b_e(x_e^{EQ})^4)x_e^{EQ}} \\ &= \frac{b_{e1}(x_{e1}^{EQ})^5 + b_{e5}(x_{e5}^{EQ})^5 + b_{e4}(x_{e4}^{EQ})^5}{(a_{e1} + b_{e1}(x_{e1}^{EQ})^4)x_{e1}^{EQ} + (a_{e1} + b_{e5}(x_{e5}^{EQ})^4)x_{e5}^{EQ} + (a_{e4} + b_{e4}(x_{e4}^{EQ})^4)x_{e4}^{EQ}} \\ &= \frac{k^5 + 0 + k^5}{k^5 + 0 + k^5} \\ &= 1\end{aligned}$$

  where k is the total traffic flow. This implies that the estimation for PoA(Γ) $\leq \gamma \approx 2.151$ from Theorem 6.2.

- When the total flow is greater than 1, then some of the Equilibrium Flows will travel through edge 2 and edge 3 in Figure 2.2b. In this case the numerator for α does not include flows for edge 2 and edge 3 since $b_e = 0$ for both of the edges. But on the other hand denominator for α includes flow for edge 2 and 3, hence causing α no longer equal to 1, and hence having a tighter upper bound.

Braess's Paradox is in fact an extreme example where the estimate PoA from section 6.2 decreases. For the next two cases we will see that the estimate PoA from section 6.2 increases with the total traffic flow.

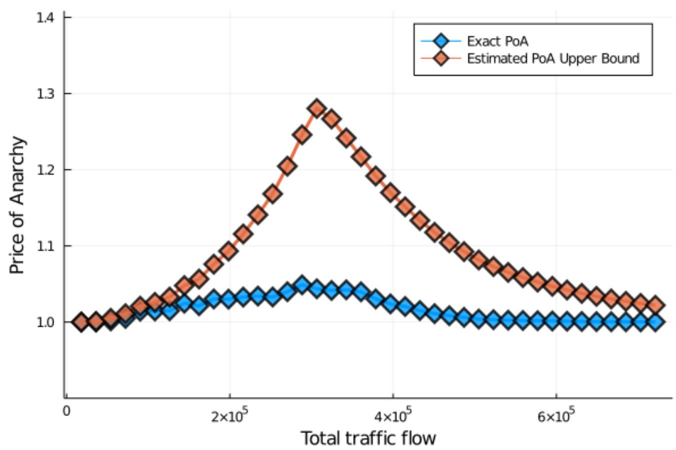


Figure 7.13: Exact and Estimated PoA against traffic flow in Sioux Falls

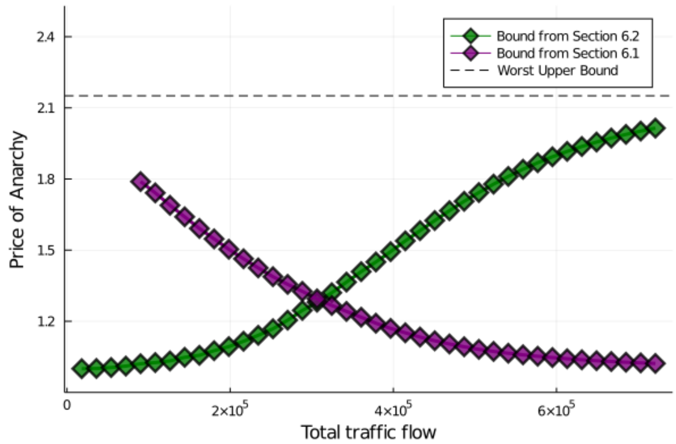


Figure 7.14: Estimated PoA with different approach of bounding in Sioux Falls

7.3 Sioux Falls revisited

Next we would revisit the example of Sioux Falls in Section 2.6.2. By applying estimated upper bound into the Configuration under various traffic flow we obtain the result shown in Figure 7.13. For some of the traffic flow the constraints stated in section 4 are satisfied, and hence we could use the estimate in Section 6.1 as shown in Figure 7.14.

Now observe that the estimate in Section 6.1 decreases as the total flow increases. This generally holds with the following argument:

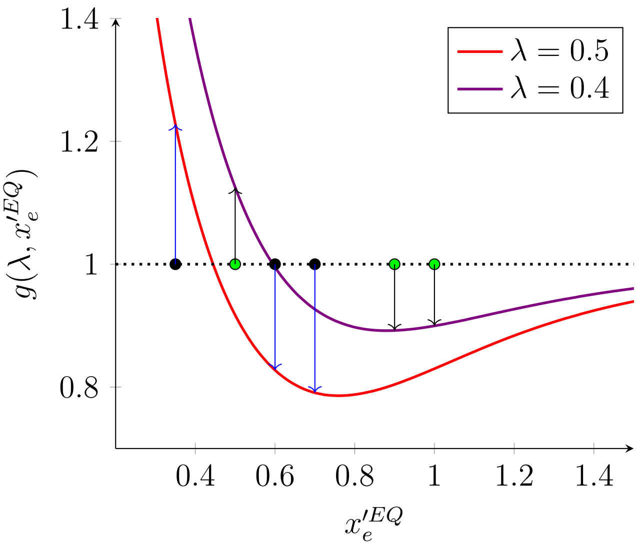


Figure 7.15: Choice of λ under different $x_e'^{EQ}$

When the total traffic flow increases, Equilibrium Flow of each edges also increases in general. In this case we can can pick a lower λ while constraint (23) still holds as illustrated in figure 7.15, which turns out generates a lower upper bound as shown in Figure 7.14.

On the other hand, the estimate in Section 6.2 increases as the total flow increases. This holds because

$$\begin{aligned}\alpha &= \frac{\sum_{e\in\mathcal{E}} b_e(x_e^{EQ})^5}{\sum_{e\in\mathcal{E}}(a_e + b_e(x_e^{EQ})^4)x_e^{EQ}} \\ &= \frac{\sum_{e\in\mathcal{E}} b_e(x_e^{EQ})^5}{\sum_{e\in\mathcal{E}}(a_e x_e^{EQ} + b_e(x_e^{EQ})^5)}\end{aligned}$$

and when the Equilibrium Flow of the edges are generally large, $b_e(x_e^{EQ})^5$ will dominate over the denominator and hence α would increase and tends to 1, hence an increase in the estimated upper bound.

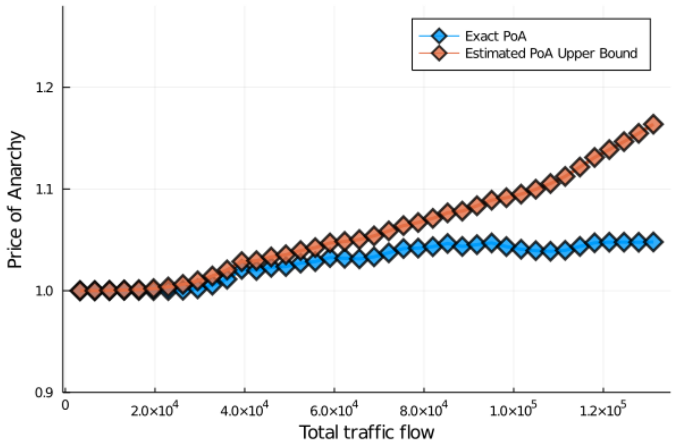


Figure 7.16: Exact and Estimated PoA against traffic flow in Eastern Massachusetts

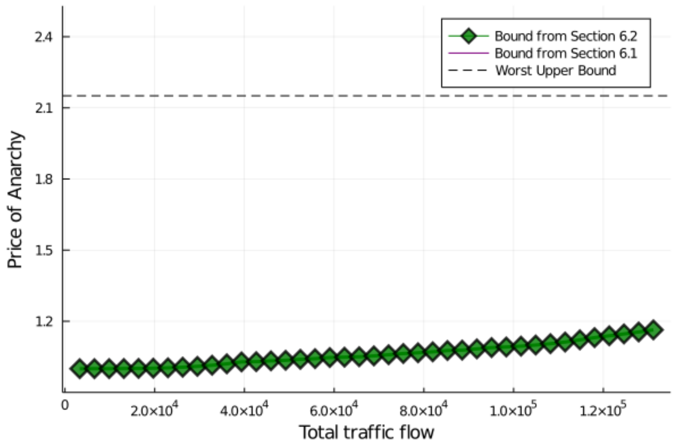


Figure 7.17: Estimated PoA with different approach of bounding in Eastern Massachusetts

7.4 Eastern Massachusetts

The last example to consider is the Eastern Massachusetts Transport Network[5]. From the TNTP dataset[16], we can see that there are some edge which has 0 Equilibrium Flow, for example edge flowing from node 5 to node 10 as shown in Appendix B. Since we do not know the Optimal Flow of this edge and this edge is not in the form of $c_e(x_e) = b_e x_e^4$, this edge clearly violates the second constraint in section 3.

By applying estimated upper bound into the Configuration under various traffic flow we obtain the result shown in 7.16. And since only the estimate from Section **??** can be used, the estimate and the worst upper bound are plotted in Figure 7.17.

Note that even we only used one of the two bounds, the estimated bound is still far from the worst upper bound 2.151, this shows that the bounding in Section 6 is in fact tight.

8 Ethical Considerations

One of the Ethical consideration would be the legal implications on modifying and using the source code. This project invloves running synthetic data with `TrafficAssignment.jl` on `https://github.com/chkwon/TrafficAssignment.jl`. This software is under MIT license, which means that the software is free to modify, distribute, sell, resell, copy, publish and sublicense if the license is included in these works. Graphs related to Price of Anarchy against traffic flow in Section 2 and Section 7 is computed and generated using this software and some modifications are made to compute Optimal Flow. The modified software have included the original MIT license in `TrafficAssignment.jl`.

This project uses synthetic data instead of real data, so there does not have any ethical implications with regards to personal data collection and processing.

9 Conclusion and Future Work

9.1 Conclusion

In this project we have addressed the issue of loose bound of Price of Anarchy which leads to a high discrepancy between actual and theoretical bound of Price of Anarchy. One of the contributions of this project is to tighten the upper bound with extra information on Equilibrium Flows. The project also proposes methods to find such an upper bound for cost functions in form of $c_e(x_e) = a_e + b_e x_e^4$.

In particular, two different approaches are proposed based on Smooth Game Condition. The first approach suggested in Section 4 upper bounds the Price of Anarchy via modified minimum Equilibrium Flow, where certain constraints are applicable to the Game. The second approach suggested in Section 5 upper bounds the Price of Anarchy via modified maximum Equilibrium Flow with no extra conditions required, which implies that the second approach can be applied to a wider range of cases than that of first approach.

In later part of the report, those two approaches suggested previously are extended such that the upper bound does not only depend on modified maximum or modified minimum Equilibrium Flow, but Equilibrium Flow from all edges in the game. This extended approach further tightens the estimated lower bound and hence a better estimate.

Last but not least the findings in the report are applied to different Real-world Driven Data Examples, and we have shown that the estimate lower bound proposed in this project gives a tighter upper bound with respect to the actual Price of Anarchy, compare to the worst case upper bound. As an example, Transport Network in Sioux Falls has an estimated upper bound of 1.217, compared to actual Price of Anarchy of 1.039 and worst upper bound of 2.151.

9.2 Future Work

In order to compute the estimated upper bound as suggested in the report, we often have to compute all Equilibrium Flows which is potentially computationally expensive. In fact we have only reduced around half of the computation effort compared to computing the exact Price of Anarchy, which also requires Optimal Flows. Hence it would be interesting whether we could obtain a good approximation on Equilibrium Flow without computing the whole Traffic Flow, or even to approach the problem without calculating Equilibrium Flow.

There are also a couple of ideas that are worth investigating but not has not been done in this project, such as:

- Merging the approaches in Section 4 and 5 into one approach with a "smooth" upper bound instead of having a spike in Figure 7.13. By finding a smooth upper bound around the spike likely implies a tighter upper bound around that region.
- Investigate the relation between x_e^{EQ} and x_e^{OPT*} in equation (11). From the relation we could have a better understanding on the behaviour on the inequality (13), and hence a chance to obtain a tighter bound.

References

[1] Christos H. Papadimitriou. Algorithms, games, and the internet. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors, *Automata, Languages and Programming*, pages 1–3, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. ISBN 978-3-540-48224-6.

[2] T. Roughgarden and É. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, 2002. doi: https://doi.org/10.1145/506147.506153.

[3] John Glen Wardrop. *Some theoretical aspects of road traffic research*. Institution of Civil Engineers, 1952.

[4] B. Monnot, F. Benita, and G. Piliouras. Routing games in the wild: Efficiency, equilibration and regret. In Nikhil R. Devanur and Pinyan Lu, editors, *Web and Internet Economics*, pages 340–353, Cham, 2017. Springer International Publishing. ISBN 978-3-319-71924-5.

[5] J. Zhang, S. Pourazarm, C. G. Cassandras, and I. C. Paschalidis. The price of anarchy in transportation networks by estimating user cost functions from actual traffic data. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 789–794, 2016. doi: 10.1109/CDC.2016.7798364.

[6] jingzbu. Inversevistraffic. URL `https://github.com/jingzbu/InverseVIsTraffic/tree/master/04_1_InverseVI_uni_class_cdc16`.

[7] *Traffic Assignment Manual*. Bureau of Public Roads, U.S. Department of Commerce, Urban Planning Division, Washington D.C., 1964.

[8] Riccardo Colini-Baldeschi, Roberto Cominetti, Panayotis Mertikopoulos, and Marco Scarsini. When is selfish routing bad? the price of anarchy in light and heavy traffic. *Oper. Res.*, 68(2):411–434, March 2020. ISSN 0030-364X. doi: 10.1287/opre.2019.1894. URL `https://doi.org/10.1287/opre.2019.1894`.

[9] Martin Josef Beckmann, Charles B. MacGuire, and Christopher B. Winsten. *Studies in the economics of transportation*. Yale University Press, 1959.

[10] M.J. Smith. The existence, uniqueness and stability of traffic equilibria. *Transportation Research Part B: Methodological*, 13(4), 1979. doi: 10.1016/0191-2615(79)90022-5.

[11] Arthur C. Pigou. *The economics of welfare*. Routledge, 2017.

[12] D. Braess. Über ein paradoxon aus der verkehrsplanung. *Unternehmensforschung*, 12:258–268, 1968. doi: https://doi.org/10.1007/BF01918335.

[13] Tim Roughgarden. The price of anarchy is independent of the network topology. *Journal of Computer and System Sciences*, 67(2):341 – 364, 2003. ISSN 0022-0000. doi: https://doi.org/10.1016/S0022-0000(03)00044-8. URL `http://www.sciencedirect.com/science/article/pii/S0022000003000448`. Special Issue on STOC 2002.

[14] Tim Roughgarden. Intrinsic robustness of the price of anarchy. *Journal of the ACM*, 62(5):1–42, 2015. doi: 10.1145/2806883.

[15] Changhyun Kwon. Trafficassignment.jl. URL `https://github.com/chkwon/TrafficAssignment.jl`.

[16] Transportation networks for research. URL `https://github.com/bstabler/TransportationNetworks`.

[17] José R. Correa, Andreas S. Schulz, and Nicolás E. Stier-Moses. A geometric approach to the price of anarchy in nonatomic congestion games. *Games and Economic Behavior*, 64(2):457–469, 2008. doi: 10.1016/j.geb.2008.01.001.

Appendices

Appendix A: Sioux Falls Graph

Graph obtained from [16]:

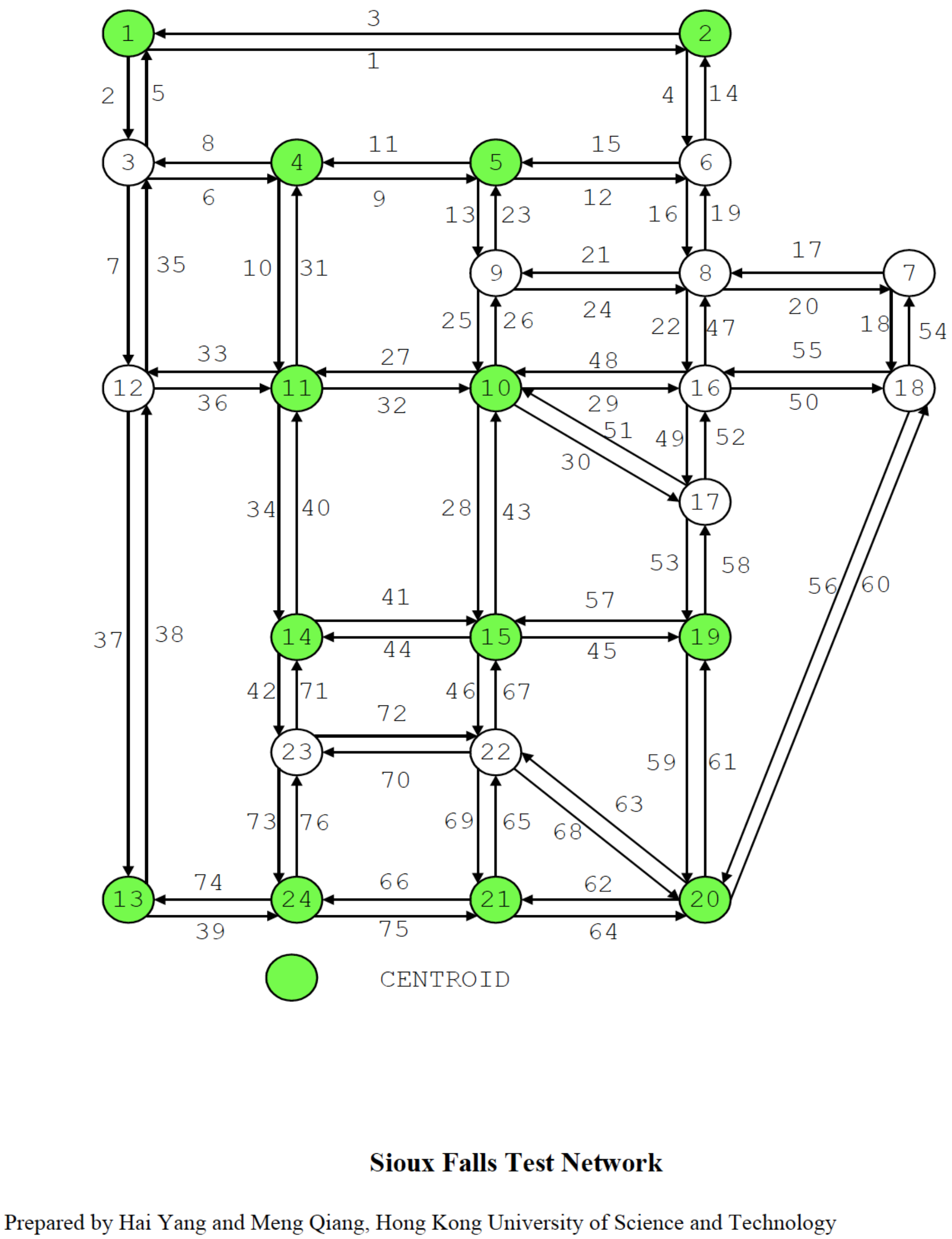


Figure 9.18: Graph of Sioux Falls Example

Appendix B: Eastern Massachusetts Graph

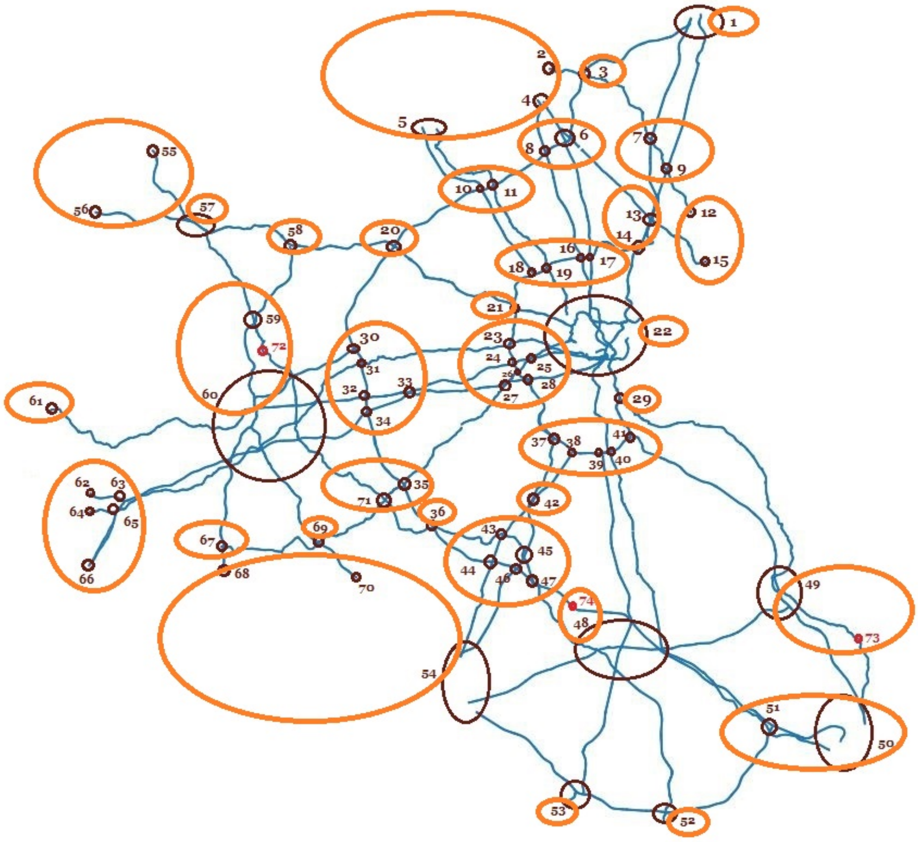


Figure 9.19: Graph of Eastern Massachusetts Traffic