Computational Aspects of the Braess's Paradox

Master thesis of:
Kyriakos Sergis
MPLA,
Department of Mathematics,
University of Athens (UOA).

Supervisor:
Dimitris Fotakis,
Lecturer,
School of Electrical and Computer Engineering,
National Technical University of Athens.
March 2, 2015

To my wife, Anthi
and sons, the twins Constantine and Rafael.

Contents

Abstract

In this thesis, we investigate the Braess's Paradox from a computational viewpoint. The motivation is to provide simple ways of improving network performance by exploiting the essence of the Braess's Paradox, namely the fact the network performance at equilibrium can be improved by edge removal. We first present approximation algorithms for the best subnetwork problem in random networks with linear latencies and polynomially many paths, each of polylogarithmic length. Moreover, we improve on the best known running time for the best subnetwork problem in certain classes of networks.

Chapter 1

Preliminaries

In this chapter, we present some basic notions of selfish routing which will be necessary for our study. We will then try to define the concept of *Nash equilibrium* (aka a *Wardrop equilibrium*) and make a distinction between the corresponding *Nash equilibrium flow* and the *Optimal flow*. Then, we will give a definition of the famous *Price of Anarchy (PoA)*, giving some more details about the correlation of this term to the network performance.

After all these, we will be ready to move on and talk about a special network, a rather peculiar instance, which is the main subject of this thesis. This phenomenon is called the *Braess's Paradox*.

For an elementary introduction to selfish routing the reader is referred to [1] and [2]. For a detailed reference to the *PoA* and the *Braess's Paradox*, he is asked to look at [3], [4] and [5].

1.1 Introduction

Routing games, are a special case of congestion games, where the major question is how to route traffic in a large network, like the internet, where there is no central authority that controls the flow of the links, in order to avoid congestion and consequently delays.

Here, we deal with one of the two widely known models of routing games, the *nonatomic selfish routing*. This model assumes that there is a huge amount of players, with each one of them controlling a negligible portion of the total traffic (the other model, called *atomic* deals with users that control a non-negligible amount of traffic).

The network is given by a directed graph G, with a vertex set V and an edge set E, $G = (V, E)$. It has also source-sink vertex pairs $(s_1, t_1), (s_2, t_2)$, $\ldots$, (s_k, t_k), where $s_i \neq t_i$, called *commodities*. Each source s_i is connected with sink t_i via a set of paths $\mathcal{P}_i$. We define $\mathcal{P} = \sum_{i=1}^{k} \mathcal{P}_i$. The network may have parallel edges, and each vertex may participate in more than one path.

As we mentioned earlier, each player controls a negligible portion of the total traffic. The total traffic r, is referred to as the *traffic rate*, and the portion of r that the player uses, is symbolized as r_i. Each r_i travels through the paths, via $flows$. The flow travelled on a path $P \in \mathcal{P}_i$ is symbolized as f_P, whereas the sum of the flows that travel through edge e is symbolized by f_e, $f_e = \sum_{P \in \mathcal{P}: e \in P} f_P$.

A flow f is called $feasible$ for r if it routes all of the traffic, which means that for each $i \in \{1, 2, \ldots, k\}$, $\sum_{P \in \mathcal{P}_i} f_P = r_i$.

Each edge of the network has a *cost function*, $l_e : \mathbb{R}^+ \to \mathbb{R}^+$, which is always considered nonnegative, continuous and nondecreasing. Also, l_e is considered semiconvex, that is $x \cdot l_e(x)$ is convex on $[0, \infty)$. We will call such latency functions as $standard$.

The cost of a path P with respect to a flow f, is the sum of the costs of the constituent edges:

$$l_P(f) = \sum_{e \in P} l_e(f_e) \tag{1.1}$$

Since we have defined the cost of a path or the cost of an edge, it's time to define the *cost* of a flow f as:

$$C(f) = \sum_{P \in \mathcal{P}} l_P(f) f_P \tag{1.2}$$

Using (1.1), the relation (1.2) becomes equivalently:

$$C(f) = \sum_{e \in E} l_e(f_e) f_e \tag{1.3}$$

We define an *instance* of the nonatomic selfish routing model or a *nonatomic selfish routing game*, as $\mathcal{G} = (G, r, l)$.

1.2 Nash equilibrium flow

Having in mind all the above, it is time to define the *Nash equilibrium flow* for an instance $\mathcal{G}$.

Definition 1.1. Let f be a feasible flow for an instance $\mathcal{G}$. The flow f is a *Nash equilibrium flow* if for every commodity $i \in \{1, 2, \ldots, k\}$ and every pair $P, \tilde{P} \in \mathcal{P}_i$ of $s_i - t_i$ paths with $f_P > 0$:

$$l_P(f) \leq l_{\tilde{P}}(f)$$

This means that the player has nothing to gain by changing his route from path P to path $\tilde{P}$, since path's P cost is lower.

If we would like to formalize this idea, we can use the following definition:

Definition 1.2. A flow f feasible for an instance $\mathcal{G}$ is at *Nash equilibrium* if for all $i \in \{1, 2, \ldots, k\}$, P_1, $P_2 \in \mathcal{P}_i$ and $\delta \in [0, f_{P_1}]$, we have $l_{P_1}(f) \leq l_{P_2}(\tilde{f})$, where:

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

Letting δ tend to 0, continuity and monotonicity of the edge latency functions, entail the above definition 1.1 of a flow at *Nash equilibrium*, occasionally called a *Wardrop equilibrium* (WE), due to an influential paper of Wardrop [6].

We conclude that if f is at *Nash equilibrium* then all $s_i - t_i$ paths share equal latency, say $L_i(f)$. Using relation (1.2) we conclude that $C(f) = \sum_{i=1}^{k} L_i(f) r_i$.

For a *single commodity instance*, we have a common latency for all paths, namely $L_{eq}(G)$, and also the following relation:

$$C(f) = L_{eq}(G) \cdot r \tag{1.4}$$

1.3 Existence and Uniqueness of WE flows

A very nice property of convex functions on a convex set is that the local and global optima coincide to a unique optimum value. Let's for now define the following term:

$$\Phi(f) = \sum_{e \in E} \int_0^{f_e} l_e(x)\,\mathrm{d}x$$

Since the latency functions are standard, $\Phi(f)$ is a convex function wrt path and edge flows. By convexity, $\Phi(f)$ attains a unique minimum value $\Phi(f_{min})$, let's say at a flow f_{min} (although there can be many "minimum" flow values). Having in mind definition 1.2, if a portion $\delta > 0$ leaves path P_1 to enter P_2, inducing flow $\tilde{f}_{min}$, it is relatively easy for the reader [1], [2], to check the following equivalence:

$$\Phi(f_{min}) \leq \Phi(\tilde{f}_{min}) \Leftrightarrow l_{P_1}(f_{min}) \leq l_{P_2}(\tilde{f}_{min})$$

Function $\Phi(\cdot)$ is called *Potential*, since *Nash equilibrium flows* are the global minimizers of it. So, we reach to the point of defining the following *Nonlinear Program* (NLP), which solution gives the corresponding *Nash equilibrium flows*:

(NLP 1):

$$\min \sum_{e \in E} \int_0^{f_e} l_e(x)\,\mathrm{d}x \quad \big(= \min \Phi(f)\big)$$

subject to

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

$$f_e = \sum_{P \in \mathcal{P}: e \in P} f_P$$

$$f_P \geq 0$$

Now, that we have proved the *Existence of equilibrium flows*, we may move on to the following theorem which refers to the *Uniqueness of equilibrium flows.* For more details the reader is advised to visit [1] and corollaries 2.6.2, 2.6.4 of [5]:

Theorem 1.3. *Let $\mathcal{G}$ be a nonatomic instance. Then, if f and $\tilde{f}$ are equilibrium flows for $\mathcal{G}$ then $l_e(f_e) = l_e(\tilde{f}_e)$ for every edge e.*

WE flows f and $\tilde{f}$ minimize the potential function $\Phi(\cdot)$. If we consider all convex combinations of f and $\tilde{f}$, $\lambda f + (1-\lambda)\tilde{f}$ in particular, where $\lambda \in [0, 1]$, these are feasible flows too. Since function $\Phi(\cdot)$ is convex, we have that $\Phi(\lambda f + (1-\lambda)\tilde{f}) \leq \lambda\Phi(f) + (1-\lambda)\Phi(\tilde{f})$. But f and $\tilde{f}$ are global minima, which means that the latter relation must hold with equality. This means that every summand $\int_0^x l_e(y)\,\mathrm{d}y$ must be linear between the values f_e and $\tilde{f}_e$. The latter implies that every cost function l_e is *constant* between f_e and $\tilde{f}_e$. Since there is no way to have increased cost values, under the same factor for every edge, but with the same total traffic r as input, this leads us to the fact that the edge costs must be equal.

Now, if the latency functions are *strictly increasing*, the following lemma is straightforward:

Lemma 1.4. *Let $\mathcal{G}$ be a nonatomic instance with standard strictly increasing latency functions. If f and $\tilde{f}$ are equilibrium flows for $\mathcal{G}$ then $f_e = \tilde{f}_e$ for every edge e.*

1.4 Characterization of Optimal flows

Apparently, *optimal flows* minimize the total cost of a flow $C(f)$, defined in (1.2), (1.3). Then, by using the relation (1.3), it is obvious that the following (NLP) finds the minimum-latency feasible flows, named *optimal*:

(NLP 2):

$$\min \sum_{e \in E} l_e(f_e) f_e$$

subject to

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

$$f_e = \sum_{P \in \mathcal{P}: e \in P} f_P$$

$$f_P \geq 0$$

Optimal flows should not be confused with *Nash equilibrium flows*. The former are flows that minimize the total latency, with some players suffering probably by long delays in favor of the rest who may travel faster to their destination. This means, that there *might* (most of the instances *should*) be players who have a serious motive to change their path to another one with lesser delays. Obviously, this is a description of a state that is unstable. The latter are selfish-optimum flows, where everyone suffers such a delay, that has no incentive in changing his route towards his destination to another, reaching an equilibrium state for all players.

Looking more carefully at the objective functions of the two (NLP)s, explicitly (NLP 1) and (NLP 2), if at the place of (NLP 1) objective functions's $l_e(x)$ we put $l_e^*(x) = \frac{d(x \cdot l_e(x))}{dx}$, then we've got the objective function of (NLP 2).

So, we come to the following corollary, which shows the *equivalence of equilibrium and optimal flows*:

Corollary 1.5. *Let $\mathcal{G}$ be a nonatomic instance such that, for every edge e, the function $x \cdot l_e(x)$ is convex and continuously differentiable. Then o is an optimal flow for $\mathcal{G} = (G, r, l)$ if and only if it is an equilibrium flow for (G, r, l^*).*

The next lemma is a straightforward consequence of theorem 1.3 and corollary 1.5:

Lemma 1.6. *Let $\mathcal{G}$ be a nonatomic instance such that, for every edge e, the function $x \cdot l_e(x)$ is convex and continuously differentiable. Then o is an optimal flow for $\mathcal{G} = (G, r, l)$ if and only if for every commodity $i \in \{1, 2, \ldots, k\}$ and every pair $P, \tilde{P} \in \mathcal{P}_i$ of $s_i - t_i$ paths with $o_P > 0$:*

$$l_P^*(o) \leq l_{\tilde{P}}^*(o)$$

Function $l^*(\cdot)$ is called *marginal cost function*. The reason is that when analyzing $l_e^*(x) = \frac{d(x \cdot l_e(x))}{dx} = l_e(x) + x \cdot \frac{d(l_e(x))}{dx}$, one term of it describes the latency incurred by the flow, while the second one describes the "anomaly" or increased congestion caused by that flow.

Existence and uniqueness of optimal flows is now obvious, for the same reasons described in section 1.3, since $l^*(\cdot)$ is a convex function. It is relatively easy to prove that the following lemmas are valid:

Lemma 1.7. *Let $\mathcal{G}$ be a nonatomic instance. Then, if o and $\tilde{o}$ are optimal flows for $\mathcal{G}$ then $l_e(o_e) = l_e(\tilde{o}_e)$ for every edge e.*

Lemma 1.8. *Let $\mathcal{G}$ be a nonatomic instance with standard strictly increasing latency functions. If o and $\tilde{o}$ are optimal flows for $\mathcal{G}$ then $o_e = \tilde{o}_e$ for every edge e.*

Having in mind relation (1.2), and since the latency incurred by the players is not the same, then the term *average* latency makes sense. So, we can conclude that $C(f) = \sum_{i=1}^{k} L^i_{opt}(f) r_i$, where L^i_{opt} is the *average* latency per commodity $s_i - t_i$.

For a *single commodity instance*, all players may suffer the same *average* latency $L_{opt}(G)$. We then have the following relation:

$$C(o) = L_{opt}(G) \cdot r \tag{1.5}$$

Before moving further on to the next section, it is time to present some useful lemmas that are straightforward proofs of the facts that we have already available in our hands:

Lemma 1.9. *Let $\mathcal{G}$ be a nonatomic instance with standard latency functions. A feasible flow f is a Nash flow if and only if for any feasible flow g:*

$$\sum_{e \in E} l_e(f_e) f_e \leq \sum_{e \in E} l_e(f_e) g_e \tag{1.6}$$

Also, a feasible flow o is an optimal flow if and only if for any feasible flow g:

$$\sum_{e \in E} l^*_e(o_e) o_e \leq \sum_{e \in E} l^*_e(o_e) g_e \tag{1.7}$$

Lemma 1.10. *Let $\mathcal{G}$ be a nonatomic instance with constant latency functions or latency functions of the form $l_e(x) = a_e \cdot x$ only. A feasible flow f is optimal if and only if it is at Nash equilibrium.*

1.5 Time Complexity

In 1984, Minoux [7] showed that the scaling technique of Edmonds and Karp [8], which for *linear cost* flows gives a polynomial time bound for the out-of-kilter method, can be also applied to quadratic separable min cost flows. Thus, we have the following lemma:

Lemma 1.11. *Let $\mathcal{G}$ be a nonatomic instance with linear latency functions $l_e(f_e) = a_e \cdot f_e + b_e \geq 0$ with nonnegative rational coefficients a_e, b_e. Then, (NLP 1) and (NLP 2) are solved in polynomial time.*

Hochbaum and Shanthikumar [9] proved that solving non-quadratic convex separable min-cost flow problems is not much harder than linear optimization. Their algorithm computes optimal solution to any specified accuracy in polynomial time (where optimal solution is the one that minimizes the objective function of (NLP 1), (NLP 2)). More precisely the algorithm finds a feasible solution that is ε-accurate from an optimal solution. The algorithm's running time is polynomial in $\log\left(\frac{1}{\varepsilon}\right)$ and the input size.

1.6 Price of Anarchy (PoA)

It is time to present a term that was introduced at first by Koutsoupias and Papadimitriou [10] as a way to measure the (in)efficiency of a network G. This term is called *Price of Anarchy (PoA)* and is defined as the following ratio:

$$PoA_G = \frac{\text{cost of worst WE of selfish routing on G}}{\text{cost of optimal routing on G}}$$

Since the cost of optimal routing could be at most the cost of worst WE of selfish routing on G, we have that $PoA_G \geq 1$. Having in mind lemma 1.10, there may be cases where the optimal and WE costs are equal, leading to $PoA_G = 1$, as we will see later in this thesis.

Using relations (1.4) and (1.5), for a *single commodity instance* we have that:

$$PoA_G = \frac{L_{eq}(G)}{L_{opt}(G)} \tag{1.8}$$

1.6.1 Pigou's example

Consider the single commodity instance in Fig. 1.1, referred to as *Pigou's example.*

We consider total traffic rate $r = 1$. At the case where $l_1(x) = 1$ and $l_2(x) = x$, it is easy for the reader to prove that at the equilibrium state all traffic is routed through the lower link, which suffers delay $l_2(1) = 1$, while the upper link is visited by no player. The total cost is $C(f) = 1 \cdot 1 + 0 \cdot 1 = 1$. The optimal case, is when half of the traffic uses the lower link, while the other half uses the upper one, causing total cost $C(o) = \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot 1 = \frac{3}{4}$. Consequently, the Price of Anarchy is $PoA = \frac{C(f)}{C(o)} = \frac{4}{3}$.

The inefficiency of selfish routing could be more severe. Consider the nonlinear case of the Pigou's example, with $r = 1$ and $l_1(x) = 1, l_2(x) = x^p$,

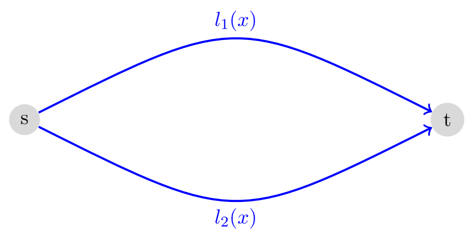


Figure 1.1: Pigou's example.

where $p \geq 2$. Again, at the equilibrium state all traffic is routed through the lower link, which suffers delay $l_2(1) = 1$, while the upper link is visited by no player. The total cost is the same, $C(f) = 1 \cdot 1 + 0 \cdot 1 = 1$. However, the optimal case is different. The marginal cost functions are $l_1^*(x) = 1$ and $l_2^*(x) = (p+1) \cdot x^p$ for upper and lower links respectively, which consequently means that $(p+1)^{-\frac{1}{p}}$ users will prefer the lower link, while the rest $1-(p+1)^{-\frac{1}{p}}$ users will prefer the upper link. The total cost of the optimal case turns out to be $C(o) = 1 - p \cdot (p+1)^{-\frac{p+1}{p}}$ and the Price of Anarchy is $PoA = (1 - p \cdot (p+1)^{-\frac{p+1}{p}})^{-1} = \Theta(\frac{p}{\ln p})$. What happens if p is too large, $p \to \infty$? Then the the optimal cost gets too small, $C(o) \to 0$, while the equilibrium cost is too big, $C(f) = 1$, causing $PoA \to \infty$.

All the above were mentioned at first, in order to see how PoA captures the inefficiency of the network, and secondly because the following term, that we will describe shortly, formalizes a bound on the PoA. This term is called *Pigou Bound* and was introduced by Roughgarden in [1] and [3].

Definition 1.12. Let $\mathcal{C}$ be a nonempty set of cost functions. The *Pigou Bound* $a(\mathcal{C})$ for $\mathcal{C}$ is:

$$a(\mathcal{C}) = \sup_{l \in \mathcal{C}} \sup_{x,r \geq 0} \frac{r \cdot l(r)}{x \cdot l(x) + (r-x) \cdot l(r)}$$

with the understanding that $\frac{0}{0} = 1$.

In [3], Roughgarden calls it *Anarchy Value*, but it is defined slightly differently, than in [1]. In [3], Roughgarden gives a proof that the inefficiency of the network, described by the factor PoA, is bounded by the *Anarchy Value*. In other words, network's inefficiency does not depend on the topology of the network itself rather, but on the links' latencies. It is proven, for example, that for polynomial latency functions that contain all constant cost functions, the worst equilibrium to optimal cost ratio is the one described in the case of the *linear* or *nonlinear Pigou's example*. For the *linear* case, the reader is also advised to study [11]. So, having also in mind lemma 1.10, we have that no nonatomic instance with *linear* or *nonlinear* latency functions has PoA larger than $\frac{4}{3}$ or $(1 - p \cdot (p+1)^{-\frac{p+1}{p}})^{-1} = \Theta(\frac{p}{\ln p})$, where $p \geq 2$ is the degree of the polynomial, respectively.

Here, we present a shorter proof of the fact that PoA is bounded by the *Pigou Bound* ([1]).

Lemma 1.13. *Let $\mathcal{C}$ be a set of cost functions that contains all the constant cost functions. Then the PoA can be lower bounded by $a(\mathcal{C})$.*

At first, if $x \geq r$ then $a(\mathcal{C}) \leq 1$. So, since $PoA \geq 1$ the lemma holds. Assuming now that $x < r$, then by the Pigou's example, by setting the traffic rate to r, $l_1(y) = r$, $l_2(y) = l(y)$ and $l_2(r) = r$, there is an equilibrium state where all traffic will prefer the lower link, yielding a cost $C_{PG}(f) = r \cdot l(r)$, where PG stands for the Pigou's graph. On the other hand, routing x units of traffic on the lower link and $r - x$ on the upper link, this is a feasible flow with cost $C_{PG}(g) = x \cdot l(x) + (r - x) \cdot l(r)$. Clearly, $C_{PG}(g) \leq C_{PG}(f)$ and $PoA \geq \frac{C_{PG}(f)}{C_{PG}(g)} \geq a(\mathcal{C})$.

The hypothesis of lemma 1.13 that $\mathcal{C}$ must contain the constant cost functions, can be relaxed at first by the hypothesis that the class must be *diverse*, which means that $\forall c > 0$ there is a latency function $l \in \mathcal{C}$ such that $l(0) = c$, and secondly by an even weaker condition that the class must be *inhomogenius*, in the sense that $\exists l \in \mathcal{C}$ such that $l(0) > 0$. For more details the reader is advised to study [3].

Lemma 1.14. *Let $\mathcal{C}$ be a set of cost functions. Then the PoA can be upper bounded by $a(\mathcal{C})$.*

Let f and o be equilibrium and optimal flows respectively. Then, if at the definition of the *Pigou Bound* we let $x = o_e$ and $r = f_e$, we have that $l_e(o_e)o_e = \frac{l_e(f_e)f_e}{a(\mathcal{C})} + (o_e - f_e)l_e(f_e)$. But, by the definition of optimal and equilibrium costs, we have that $C(o) = \frac{C(f)}{a(\mathcal{C})} + \sum_{e\in E} l_e(f_e)(o_e - f_e)$. But, since o is a feasible flow, by relation (1.6) letting $g_e = o_e$, we have that $\sum_{e\in E} l_e(f_e)(o_e - f_e) \geq 0$. The lemma now follows easily.

Before closing this chapter, we should mention a nice *bicriteria result* that came out of the joint work of Roughgarden and Tardos. For more details the reader is advised to visit [11]. This theorem says that the total latency incurred by a flow at Nash equilibrium is at most that of a feasible flow forced to route twice as much traffic between each source - destination pair.

Theorem 1.15. *If f is a Nash equilibrium flow for the instance (G, r, l) and g is feasible for the instance $(G, 2r, l)$, then $C(f) \leq C(g)$.*

Now we are ready to move on to the main study of this work, the *Braess's Paradox.*

Chapter 2

Braess's Paradox

2.1 The Paradox

A paradox is a statement or group of statements that leads to a contradiction or a situation which (if true) defies logic or reason.

Imagine a network, where adding an extra link with extra capacity to it, this can lead to a reduction of the overall performance of this network. The paradox gets more intensive if the link causes nearly zero delays for everyone who uses it. One could expect that the players will reach their destination faster. Put it differently, removing a link from the network, instead of creating obstacles for the users on their route to their destination, their life becomes easier, where everyone individually enjoys fewer delays. The performance of the network gets improved than the opposite.

Consider the single commodity network $s-t$, that is shown in Fig. 2.1.

The total traffic rate $r=1$ and the latencies are x for links (s,u), (w,t) and 1 for links (s,w), (u,t). Link (u,w) has 0 latency. Fig. 2.2 shows the equilibrium state.

At the equilibrium state, all traffic is routed via links (s,u), (u,w), (w,t) since every player, leaving source s, will prefer link (s,u) to (s,w) because the latter suffers bigger latency 1 than the former, then will choose link (u,w) since it has 0 latency, significantly smaller than link's (u,t) latency 1. Finally, will end his trip following the remaining link (w,t). Since this a

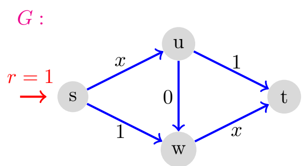


Figure 2.1: The original Braess's paradox instance.

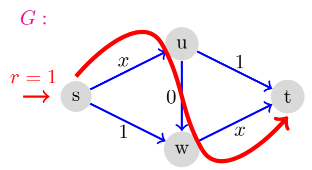


Figure 2.2: Equilibrium state.

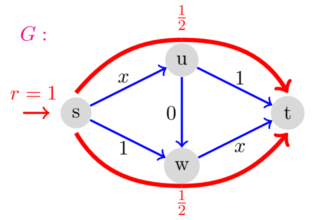


Figure 2.3: Optimal flow case.

single commodity instance, then the total latency is $L_{eq}(G) = 1+0+1 = 2$ and the total cost is given by relation (1.4) and is $C(f) = 2 \cdot 1 = 2$.

In Fig. 2.3 the optimal flow case is shown.

The reason is that according to corollary 1.5, the optimal flows of $(G, 1, l)$ are the equilibrium flows of $(G, 1, l^*)$. In Fig. 2.4 the corresponding marginal costs and edge flows above them are presented.

Since link's (u, w) users enjoy 0 latencies, $(2\epsilon)+1 = (2\epsilon)+2(1-\epsilon+\lambda) = 1+2(1-\epsilon+\lambda)$ must hold. Solving this simple equation, we have that $\lambda = 0$, $\epsilon = \frac{1}{2}$.

So, since the upper links's $\big($specifically links (s, u), $(u, t)\big)$ total latency equals the lower links's $\big($specifically links (s, w), $(w, t)\big)$ total latency, to $\epsilon + 1 = \frac{1}{2} + 1$, the total latency is $L_{opt}(G) = \frac{1}{2} + 1 = \frac{3}{2}$ per link (to be more specific we should talk about *average* latency according to section 1.4, but at this case the *average* latency coincides with the latency of the upper or lower links). The total cost is given by relation (1.5) and is $C(o) = \frac{3}{2} \cdot 1 = \frac{3}{2}$.

The *Price of Anarchy* is given by relation (1.8) and is therefore $PoA_G =$

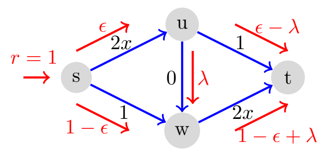


Figure 2.4: Marginal costs and edge flows.

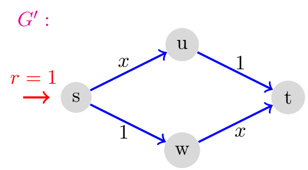


Figure 2.5: Subgraph $G' = (V', E')$ - Link (u, w) removal.

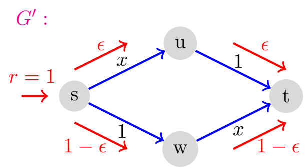


Figure 2.6: New Equilibrium state.

$\frac{L_{eq}(G)}{L_{opt}(G)} = \frac{4}{3}$.

If we now remove link (u, w), we are going to have a subnetwork of the original one (Fig. 2.5), constituting of the subgraph $G' = (V', E')$, where $V' = V$ and $E' = E \setminus \{(u, w)\}$.

At the equilibrium state, half of the total traffic rate will use the upper links while the other half will use the lower links (see Fig. 2.6).

The upper links' overall latency should be the same with the lower link's. So, equation $\epsilon + 1 = 1 + (1 - \epsilon)$ must hold. Solving this simple equation, we have that $\epsilon = \frac{1}{2}$. The total latency turns out to be $L_{eq}(G') = \frac{1}{2} + 1 = \frac{3}{2}$ and the total cost is given by relation (1.4) and is $C(f) = \frac{3}{2} \cdot 1 = \frac{3}{2}$.

At the optimal flow case, according again to corollary 1.5, the optimal flows of $(G', 1, l)$ are the equilibrium flows of $(G', 1, l^*)$. In Fig. 2.7 the corresponding marginal costs and edge flows above them are shown.

Again, the upper links' overall latency should be the same with the lower link's. So, equation $(2\epsilon) + 1 = 1 + 2(1 - \epsilon)$ must hold. Solving this simple equation, we have that $\epsilon = \frac{1}{2}$, the same flow with the equilibrium one! The total latency turns out to be $L_{opt}(G') = \frac{1}{2} + 1 = \frac{3}{2}$ and the total

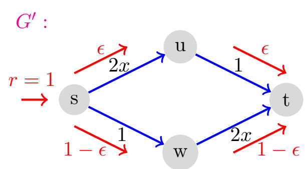


Figure 2.7: New Optimal flow case.

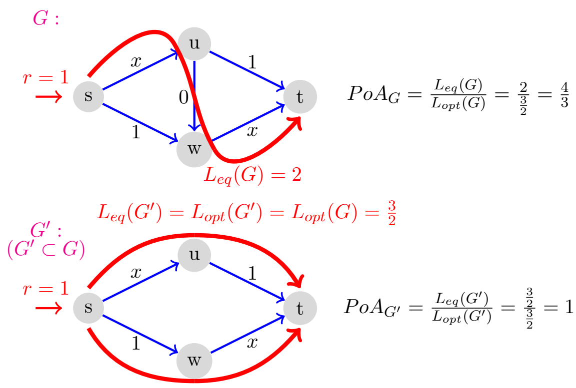


Figure 2.8: Original Braess's Paradox.

cost is given by relation (1.5) and is $C(o) = \frac{3}{2} \cdot 1 = \frac{3}{2}$, which means that $L_{opt}(G') = L_{eq}(G')$ and consequently $C(o) = C(f)$!

This means that the *Price of Anarchy* turns out to be $PoA_{G'} = \frac{L_{eq}(G')}{L_{opt}(G')} = 1$!

The great surprise, and consequently the heart of the paradox, arises from the fact that removing a link with 0 latency from a *partly inefficient* network, described by the $PoA = \frac{4}{3} > 1$, we get a new network that instead of being *more inefficient* (since the link removed has 0 latency), it turns out to be "*perfectly efficient*" in someway, since its $PoA = 1$!

Fig. 2.8 summarizes the paradox.

The paradox was first observed and presented by Braess [12] and history says that this phenomenon occurs quite frequently. For more details the reader is kindly requested to visit [2] and its references.

2.2 Paradox-Free vs Paradox-Ridden Networks

Having defined the paradox, it is time to define *paradox-ridden* networks.

What we have just proved in the paradox above, is that by deleting an edge from the network G, we get a subnetwork $H \subset G$, where $L_{eq}(H) = L_{opt}(G) = \frac{L_{eq}(G)}{PoA_G}$, and as a consequence, $PoA_H = 1$. Let's generalize this fact:

Definition 2.1. An instance $\mathcal{G}$ is *paradox-ridden* if there is a subnetwork H of G, obtained by edge removals, such that $L_{eq}(H) = L_{opt}(G) = \frac{L_{eq}(G)}{PoA_G}$.

So, *paradox-ridden* instances are networks, such that by edge removals their PoA get's better, to such a degree, that the PoA of the residual graph turns out to be 1.

But there are instances that do not suffer from the paradox. These are called *paradox-free*, such that by edge removals the residual network H has equilibrium latency $L_{eq}(H) \geq L_{eq}(G)$. So, we have the following definition:

Definition 2.2. An instance $\mathcal{G}$ is *paradox-free* if for every subnetwork H of G, $L_{eq}(H) \geq L_{eq}(G)$.

Apparently, there might be instances that are neither *paradox-ridden* nor *paradox-free*. If we remove edges from these instances, then for the corresponding network H, $L_{opt}(G) < L_{eq}(H) < L_{eq}(G)$ should hold.

So, from all the above, we understand that if we take all the residual networks H of G, there should be a *best subnetwork*, H^B, which will minimize the equilibrium latency among all subnetworks of G. Namely, $(\forall H \subset G)(\exists H^B \subset G)[L_{eq}(H^B) \leq L_{eq}(H)]$.

So, knowing the *best subnetwork*, is a crucial fact since if $L_{eq}(H^B) \geq L_{eq}(G)$ then the network is *paradox-free*, if $L_{opt}(G) < L_{eq}(H^B) < L_{eq}(G)$ then the network can be "reduced" to a network more efficient than the initial network G, and finally if $L_{opt}(G) = L_{eq}(H^B)$ then the network is *paradox-ridden* and may be reduced to an efficient network with $PoA = 1$.

2.3 Hardness of Detecting Paradox-Ridden Networks

Since the answer to the question "given a network instance $\mathcal{G}$, what is the *best subnetwork* H^B of G?" is critical, the definitions for the following optimization problems should not surprise us:

Definition 2.3. LINEAR LATENCY NETWORK DESIGN PROBLEM: Given a *single commodity instance* with *linear* latency functions, find the best subnetwork H^B.

Definition 2.4. GENERAL LATENCY NETWORK DESIGN PROBLEM: Given a *single commodity instance* with *standard* latency functions, find the best subnetwork H^B.

In [4], [5] Roughgarden presents strong evidences that detecting the *best subnetwork* is *not* a simple problem. He proved for example, that detecting *paradox-ridden* networks is as hard as solving the $P = NP$ problem.

Before moving on to the central theorems of this section, we should give the following definition:

Definition 2.5. A c - approximation algorithm for a minimization problem, runs in polynomial time and returns a solution no more than c times as costly as an optimal solution. The value c is the *approximation ratio* of the algorithm.

2.3.1 Linear Latency Functions - An Approximability Threshold of $\frac{4}{3}$

From section 1.6 we have ended to the conclusion that for *linear* latencies the *PoA* cannot be larger than $\frac{4}{3}$, or equivalently, $\frac{C(f)}{C(o)} = \frac{L_{eq}(G)}{L_{opt}(G)} \leq \frac{4}{3}$. Then, since $L_{eq}(H^B) \geq L_{opt}(G)$, we have that $\frac{L_{eq}(G)}{L_{eq}(H^B)} \leq \frac{L_{eq}(G)}{L_{opt}(G)} \leq \frac{4}{3}$. So, by definition 2.5 the latter means that the *trivial* algorithm that returns the same network G, is a $\frac{4}{3}$ - approximation algorithm for the LINEAR LATENCY NETWORK DESIGN PROBLEM, since $L_{eq}(G) \leq \frac{4}{3} \cdot L_{eq}(H^B)$.

So we come to the following conclusion:

Theorem 2.6. *The trivial algorithm is a $\frac{4}{3}$ - approximation algorithm for the LINEAR LATENCY NETWORK DESIGN PROBLEM.*

Now, the question that arises, is whether we could find a *less than* $\frac{4}{3}$ - approximation algorithm for the LINEAR LATENCY NETWORK DESIGN PROBLEM. The following theorem by Roughgarden [4], [5] clears the picture:

Theorem 2.7. *For networks with linear latency functions, and for every $\epsilon > 0$, there is no $(\frac{4}{3} - \epsilon)$ - approximation algorithm that finds the Best Subnetwork H^B $\big($unless $P = NP\big)$. Equivalently, for every $\epsilon > 0$, there is no $(\frac{4}{3} - \epsilon)$ - approximation algorithm for the LINEAR LATENCY NETWORK DESIGN PROBLEM $\big($unless again $P = NP\big)$.*

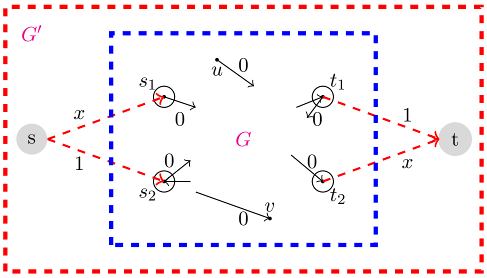
Proof. At first we assume that the graph $G = (V, E)$ has at least 4 vertices. That is, $|V| \geq 4$.

The proof depends on a reduction from a problem, known to be NP-complete, the Two Directed Disjoint Paths (2DDP) problem. This problem states that "given a directed graph $G = (V, E)$ and distinct vertices $s_i, t_i \in V$, where $i \in \{1, 2\}$, are there two $s_i - t_i$ paths P_i, such that P_1 and P_2 share no vertex?"

The reduction is polynomial in time, since we add two more vertices s and t, four edges more, $(s, s_1), (s, s_2), (t_1, t), (t_2, t)$, and assign latencies equal to 0 for all the edges in E, plus latencies equal to $x, 1, 1, x$ for the just mentioned four edges respectively. The total traffic rate is assigned to 1, as input in source s. The new graph is named $G' = (V', E')$, where $V' = V \cup \{s, t\}$, $E' = E \cup \{(s, s_1), (s, s_2), (t_1, t), (t_2, t)\}$ and the reduction is shown in Fig. 2.9.

The 2DDP instance is applied to the big network G'. So, the problem should be restated as follows: "given a directed graph $G' = (V', E')$ and distinct vertices $s_i, t_i \in V'$, where $i \in \{1, 2\}$, are there two $s_i - t_i$ paths P_i, such that P_1 and P_2 share no vertex?"

The proof is by contradiction. Suppose that there exists an algorithm $\mathcal{A}$ that approximates the Best Subnetwork H'^B with approximation ratio



Blue Dotted Area: Arbitrary Network G
Red Dotted Area: Network G'

Figure 2.9: Reduction from the 2DDP problem.

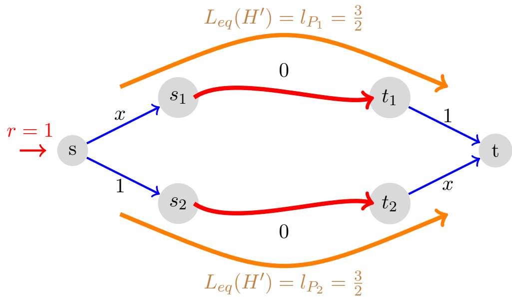


Figure 2.10: $L_{eq}(H'_{\mathcal{A}}) \geq 2$ - Assuming 2DDP instance is 'Yes'.

$\frac{4}{3} - \epsilon$, $\epsilon > 0$. That is, the algorithm $\mathcal{A}$ returns subgraph $H'_{\mathcal{A}}$ with common latency $L_{eq}(H'_{\mathcal{A}})$, such that $\frac{L_{eq}(H'_{\mathcal{A}})}{L_{eq}(H'^{B})} \leq \frac{4}{3} - \epsilon$. We claim that the algorithm decides the 2DDP problem.

- $L_{eq}(H'_{\mathcal{A}}) \geq 2$:

  We prove that the 2DDP instance is 'No'. So, let's assume by contradiction that the 2DDP instance is 'Yes'.

  This means that there exist two paths, namely the $s_1 - t_1$ path P_1 and the $s_2 - t_2$ path P_2, that share no vertex. It can be shown easily that there exists a subnetwork H' with $L_{eq}(H') = \frac{3}{2}$ (see Fig. 2.10).

  This means that $\frac{L_{eq}(H'_{\mathcal{A}})}{L_{eq}(H'^{B})} \geq \frac{2}{L_{eq}(H'^{B})} \geq \frac{2}{L_{eq}(H')} = \frac{2}{\frac{3}{2}} = \frac{4}{3}$. But the latter contradicts the hypothesis which says that $\frac{L_{eq}(H'_{\mathcal{A}})}{L_{eq}(H'^{B})} \leq \frac{4}{3} - \epsilon$.

- $L_{eq}(H'_{\mathcal{A}}) < 2$:

  We prove that the 2DDP instance is 'Yes'. So, let's assume by contradiction that the 2DDP instance is 'No'.

This means that there do not exist two paths, namely the $s_1 - t_1$ path P_1 and the $s_2 - t_2$ path P_2, that share no vertex.

So we have that either (*i*) there are no paths, or that (*ii*) there is only one path $\Big[$ namely (*iia*) path $s_1 - t_1$, or (*iib*) path $s_2 - t_2$, or (*iic*) path $s_1 - t_2$, or (*iid*) path $s_2 - t_1$ $\Big]$, or that (*iii*) there are two paths $s_1 - t_2$ and $s_2 - t_1$ which do not share any vertex, or that (*iv*) there are two paths $s_1 - t_1$ and $s_2 - t_2$ which share at least one common vertex. It can be shown easily that for every case and for every subnetwork H', $L_{eq}(H') \geq 2$. There is also an extra case, where there are two paths $s_1 - t_2$ and $s_2 - t_1$ that share at least one common vertex. But apparently, this case is equivalent to case (*iv*).

Case (*i*) follows easily, since if there are no paths that connect s_1, s_2 with t_1, t_2, then there is no way that traffic leaving s_1 or s_2, could ever reach t_1 or t_2. This means that for every subnetwork H', $L(H') \to \infty$, or $L_{eq}(H') \to \infty$, so $L_{eq}(H') > 2$.

For case (*iia*) there is only one path $s \to s_1 \to t_1 \to t$ with respective latencies x, 0, 1. Since $r = 1$, we have that for every subnetwork H', $L(H') = 2$ or $L_{eq}(H') = 2$.

For case (*iib*) there is only one path $s \to s_2 \to t_2 \to t$ with respective latencies 1, 0, x. Since $r = 1$, we have that for every subnetwork H', $L(H') = 2$ or $L_{eq}(H') = 2$.

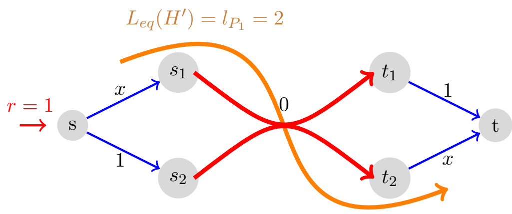
For case (*iic*) there is only one path $s \to s_1 \to t_2 \to t$ with respective latencies x, 0, x. Since $r = 1$, we have that for every subnetwork H', $L(H') = 2$ or $L_{eq}(H') = 2$.

For case (*iid*) there is only one path $s \to s_2 \to t_1 \to t$ with respective latencies 1, 0, 1. Since $r = 1$, we have that for every subnetwork H', $L(H') = 2$ or $L_{eq}(H') = 2$.

For case (*iii*) there is one path, namely $P_1 : s \to s_1 \to t_2 \to t$, with respective latencies x, 0, x. Also, there is one additional path, namely $P_2 : s \to s_2 \to t_1 \to t$, with respective latencies 1, 0, 1. Since at the equilibrium state, all players will prefer the minimum latency path P_1, all of them will use this path, causing overall latency equal to 2. So, we have that for every subnetwork H', $L_{eq}(H') = 2$.

Now, for case (*iv*), the network resembles the equilibrium case of the Braess's original network presented in section 2.1, before the link (u, w) is removed (see Fig. 2.11). At that case, $L_{eq}(H') = 2$, for every subnetwork H'.

Thus, for every case and for every subnetwork H', $L_{eq}(H') \geq 2$. But this leads us to contradiction, since by the hypothesis, there exists a subnetwork $H'_{\mathcal{A}}$ returned by the algorithm, such that $L_{eq}(H'_{\mathcal{A}}) < 2$.



Case (iv): Two Adjoint Paths

Figure 2.11: $L_{eq}(H'_{\mathcal{A}}) < 2$ - Assuming 2DDP instance is 'No'.

Now, since subnetworks H', $H'_{\mathcal{A}}$, H'^{B} are applied to the big network G', it is relatively easy to apply the results of the proof to the subnetworks H, $H_{\mathcal{A}}$, H^B of the original network G respectively, by just removing vertices s, t, and edges $(s, s_1), (s, s_2), (t_1, t), (t_2, t)$. The proof is complete. □

2.3.2 Two Useful Lemmas

Before moving on to the next two theorems, we should plug into this section two useful lemmas.

The first lemma [4], [13], [14] gives as the lengths of all vertices, with respect to shortest paths at the equilibrium state.

Lemma 2.8. *Let f be a feasible flow for (G, r, l). For a vertex v in G, let $d_s(v)$ denote the length, with respect to edge lengths $\{l_e(f_e)\}_{e\in E}$, of a shortest $s - v$ path in G. Then f is at Nash equilibrium if and only if*

$$d_s(w) - d_s(v) \leq l_e(f_e) \tag{2.1}$$

for all edges $e = (v, w)$, with equality holding whenever $f_e > 0$.

Proof. Relation (2.1) holds by the "triangle inequality", for all kinds of flows, and by the definition of shortest path labels d. Let $P \in \mathcal{P}$ be an $s - t$ path. Then:

$$\begin{aligned} l_P(f) &= \sum_{e\in P} l_e(f_e) \\ &\geq \sum_{e=(v,w)\in P} d_s(w) - d_s(v) \\ &= d_s(t) - d_s(s) = d_s(t) \end{aligned}$$

But, since $d_s(t)$ is the minimum latency of the path P, we have that $l_P(f) = d_s(t)$. Thus, in relation (2.1), the $\geq$ sign should be $=$, which is true

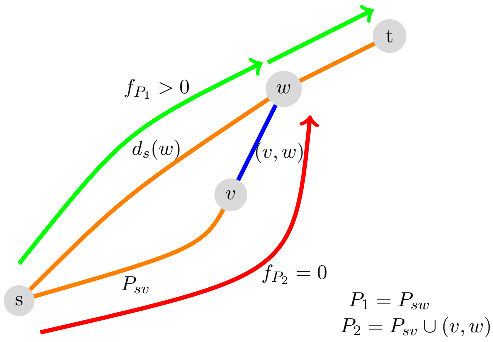


Figure 2.12: Case $d_s(w) - d_s(v) < l_e(f_e)$, where $e = (v, w)$.

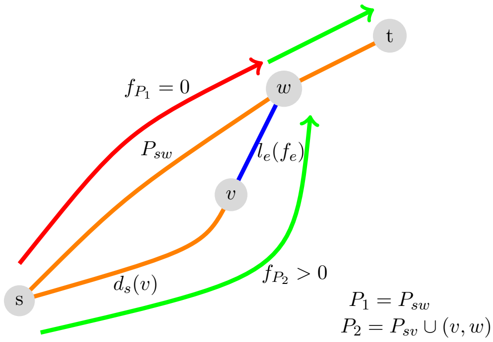


Figure 2.13: Case $d_s(w) - d_s(v) = l_e(f_e)$, where $e = (v, w)$.

if and only if for every link e that belongs to the path P:

$$d_s(w) - d_s(v) = l_e(f_e)$$

The lemma then follows easily from the definition 1.1 of the *Nash equilibrium flows*, which says that minimum latencies occur to paths or equivalently links with *positive* equilibrium flows. □

The following two pictures, give us an intuitive notion of the outcomes of lemma 2.8.

In Fig. 2.12, the equilibrium flow prefers the upper path P_1. So, we have that $d_s(w) < d_s(v) + l_e(f_e)$, or equivalently $d_s(w) - d_s(v) < l_e(f_e)$, where $e = (v, w)$.

In Fig. 2.13, the equilibrium flow prefers the lower path P_2. So, we have that $d_s(w) = d_s(v) + l_e(f_e)$, or equivalently $d_s(w) - d_s(v) = l_e(f_e)$.

The second lemma [4] states that there should always be *acyclic* Nash flows.

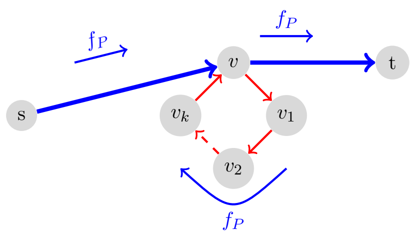


Figure 2.14: A cycle $C : v \to v_1 \to v_2 \to \ldots \to v_k$.

Lemma 2.9. *An instance* (G, r, l) *admits an acyclic flow at Nash equilibrium.*

Proof. In section 1.3 we proved the existence of *Nash equilibrium flows.* But these flows may not be acyclic. So, let's consider a non - acyclic Nash flow. We will first show that the flow edges of a cycle must have zero latency, and then we will show how to remove such cycles.

- A cycle must contain only zero - latency links:

  At first, the previous lemma 2.8 states that for every $e = (v, w)$ with $f_e > 0$, we must have that $d_s(w) = d_s(v) + l_{(v,w)}(f_{(v,w)})$.

  So, let's consider a path P, consisting of a sub-path $s \to v$, links (v, v_1), (v_1, v_2), ..., (v_k, v), $k \geq 2$, and a sub-path $v \to t$.

  Let's also consider a *cycle* C in P, with $C = (V_C, E_C)$, where $V_C = \{v, v_1, v_2, \ldots, v_k\}$ and $E_C = \{(v, v_1), (v_1, v_2), \ldots, (v_k, v)\}$ (see Fig. 2.14), and let's assume for simplicity, that the links' flows of that cycle do not participate in another path with positive equilibrium flow.

  The equilibrium flow of the path P is $f_P > 0$.

  By lemma 2.8, since $f_P > 0$, the following equations must hold:

$$
\begin{aligned}
d_s(v_1) &= d_s(v) + l_{(v,v_1)}(f_{(v,v_1)}) \\
d_s(v_2) &= d_s(v_1) + l_{(v_1,v_2)}(f_{(v_1,v_2)}) \\
&\cdot \\
&\cdot \\
&\cdot \\
d_s(v_k) &= d_s(v_{k-1}) + l_{(v_{k-1},v_k)}(f_{(v_{k-1},v_k)}) \\
d_s(v) &= d_s(v_k) + l_{(v_k,v)}(f_{(v_k,v)})
\end{aligned}
$$

  The sum of all the equations' lhs should be equal to the sum of all the equations' rhs.

So we have that $0 = 0 + \sum_{e \in E_C} l_e(f_e)$, or equivalently, $\sum_{e \in E_C} l_e(f_e) = 0$.

The latter means that $d_s(v) = d_s(v_1) = d_s(v_2) = \ldots = d_s(v_k)$ and that $(\forall e \in E_C)[l_e(f_e) = 0]$.

- Removing zero - latency flow cycles:

  As we have just mentioned, removing zero - latency flow cycles is not entirely trivial because the different edges of a flow cycle may have different flow paths, as they may participate in more than one cycle.

  We should always have in mind that f is defined as a function on paths, rather than on edges, and that by section 1.1, $f_e = \sum_{P \in \mathcal{P}: e \in P} f_P$.

  So, by the previous bullet, since these edges have zero latency, we may subtract flows from every link of every cycle, obtaining new edge flows for every edge of a cycle, and consequently obtaining new path flows. This process is called path decomposition and may be found in [15].

  After all path decompositions, from every original flow f_e per edge we should obtain flow $\tilde{f}_e$, and the following should hold:

  - ⋄ $f_e = \tilde{f}_e$, if e does not belong to a cycle.
  - ⋄ $f_e > \tilde{f}_e$, with $l_e(f_e) = 0$, if e belongs to a cycle. But $l_e(\cdot)$ is standard, which means that it is nondecreasing. So, $l_e(\tilde{f}_e) = 0$.

  From all the above, we have proved that for every case, $l_e(f_e) = l_e(\tilde{f}_e)$, for every edge $e = (v, w)$ of the network.

  But $d_s(w) = d_s(v) + l_e(f_e)$. So, by induction on every vertex v that carries flow f and consequently flow $\tilde{f}$, and with base case the vertex s, we obtain that $d_s(v) = d'_s(v)$, where $d'_s(v)$ we define the length of the same $s - v$ path induced by f, but now with respect to $\tilde{f}$.

  Since f is an equilibrium flow, the values of $d'_s(\cdot)$ are the minimal once. So $\tilde{f}$ is an equilibrium flow too, and this completes the proof. □

2.3.3 General Latency Functions - An Approximability Threshold of $\lfloor \frac{n}{2} \rfloor$

2.3.3.1 Trivial Algorithm - An $\lfloor \frac{n}{2} \rfloor$ - Approximation Algorithm

Now we are ready to move on to the main results of this subsection.

Theorem 2.10. *Let (G, r, l) be an instance, with $G = (V, E)$ and $|V| = n$. Then the trivial algorithm is a $\lfloor \frac{n}{2} \rfloor$ - approximation algorithm for the GENERAL LATENCY NETWORK DESIGN PROBLEM.*

Proof. We will prove that in the general case, $L_{eq}(G) \leq \lfloor \frac{n}{2} \rfloor \cdot L_{eq}(H)$ for every subnetwork H, which means equivalently that $L_{eq}(G) \leq \lfloor \frac{n}{2} \rfloor \cdot L_{eq}(H^B)$. So, let H be a subgraph of G.

Let f and f^* be the equilibrium flows for G and H respectively. Let also lemma's 2.8 $d_s(\cdot)$ values be with respect to equilibrium flow f. The length of the shortest $s-v$ path with respect to f^* is given by $d^*_s(v)$. By lemma 2.9 we may also assume that f is acyclic, and that every vertex of G is incident to a flow-carrying edge. The latter is obviously the worst case scenario.

Now, we can order every vertex of G, with respect to $d_s(\cdot)$ values. Let the order be non-decreasing, we may call it $\leq_o$, and let $R = \{v_0, v_1, v_2, \ldots, v_{n-1}, v_n\}$ be the outcome of that order, where $v_0 = s$ and $v_n = t$. It should be clear that $v \leq_o u \Leftrightarrow d_s(v) \leq d_s(u)$. If there are vertices with equal $d_s(\cdot)$ values, we may break ties arbitrarily, and place one of them in front of the other in that order.

Let's assume wlog that n is odd (if n is even, we may subdivide some edge and create a new vertex). We will show by induction, that $d_s(v_{2i}) \leq i \cdot L_{eq}(H)$.

- Base case, $i = 0$:

  This is trivial since $d_s(s) = 0$.

- Induction hypothesis, $d_s(v_{2(i-1)}) \leq (i-1) \cdot L_{eq}(H)$:

  We assume that $d_s(v_{2(i-1)}) \leq (i-1) \cdot L_{eq}(H)$ holds, where $i \in \{1, 2, \ldots, \frac{n-1}{2}\}$, and we shall prove that also $d_s(v_{2i}) \leq i \cdot L_{eq}(H)$ holds.

- Prove that $d_s(v_{2i}) \leq i \cdot L_{eq}(H)$:

  Let's now call an arbitrary edge e *light*, if $f_e \leq f^*_e$ and $f^*_e > 0$. Then, since latency functions are standard, we have that $l_e(f_e) \leq l_e(f^*_e) \leq L_{eq}(H)$.

  Consider now an $s-t$ cut, $S \subset R$ with $S = \{s = v_0, v_1, v_2, \ldots, v_k\}$ and $d_s(v_k) < d_s(t)$, that consists of *consecutive* vertices of R. It should be obvious, that $(\forall v \in R)(\forall u \in S)\big[\big(v \leq_o u\big) \Rightarrow \big(v \in S\big)\big]$, since all vertices are sorted with respect to f.

  Let's call $\delta^+(S)$ the set of edges that their *tail* is a vertex of S, but their *head* is a vertex that does not belong to S. On the contrary, $\delta^-(S)$ is exactly the set of edges that their *head* is a vertex of S, but their *tail* is a vertex that does not belong to S.

  Obviously, if g is a feasible flow, since s is included in S but not t, then $\sum_{e\in\delta^+(S)} g_e - \sum_{e\in\delta^-(S)} g_e = r$ should always hold.

  Apparently, since S is an $s-t$ cut, and all vertices in S are sorted with respect to flow f, we should have that $\sum_{e\in\delta^+(S)} f_e = r$, since there are no incoming flows $\big(\sum_{e\in\delta^-(S)} f_e = 0\big)$. But f^* is also a feasible flow.

So, having in mind that the vertices of S are sorted with respect to f and not to f^*, this means that $\sum_{e\in\delta^-(S)} f_e^* \geq 0$, or equivalently that $\sum_{e\in\delta^+(S)} f_e^* \geq r$.

We conclude that there should be at least one edge $e \in \delta^+(S)$ such that $f_e^* - f_e \geq 0$ with $f_e^* > 0$ or that e is a *light* edge.

Now, consider $s - t$ cut $S' = \{s, v_1, \ldots, v_{2(i-1)}\}$. Then, after all the above, there should be an integer $j \leq 2(i-1)$, and an integer $k > 2(i-1)$ with the property that it is the *largest integer* such that there is a *path of light edges* from v_j to v_k (and there should be one consisting of at least one *light* edge as we have already proved).

But $k \neq 2i-1$, because if $k = 2i-1$, then since v_k is the end of this maximal path of light edges, *all* light edges of $\delta^+(S')$ should end at (or have head) $v_k = v_{2i-1}$. But no light edge begins at (or has tail) v_k, because then k should not be the maximum such integer. So, for the set $S'' = S' \cup \{v_{2i-1}\}$ there are no light edges in $\delta^+(S'')$, and since $S'' = \{s, v_1, \ldots, v_{2(i-1)}, 2i-1\}$ is also an $s-t$ cut, we lead to contradiction. So, $k \geq 2i$.

What we have just proved, is that there is a path of light edges from v_j to v_k with $j \leq 2(i-1)$ and k maximal, such that $k \geq 2i$. So, by lemma 2.8 and induction hypothesis, we have that $d_s(v_j) \leq d_s(v_{2(i-1)}) \leq (i-1) \cdot L_{eq}(H)$.

Since $v_j \leq_o v_{2(i-1)} \leq_o v_{2i} \leq_o v_k$, what remains to be proven is that $d_s(v_k) - d_s(v_j) \leq L_{eq}(H)$, because thereafter we easily have that $d_s(v_{2i}) - d_s(v_{2(i-1)}) \leq d_s(v_k) - d_s(v_j)$, which equivalently means that $d_s(v_{2i}) \leq d_s(v_{2(i-1)}) + L_{eq}(H)$, and by induction hypothesis the result follows.

But v_j and v_k are vertices that participate in a path of light edges, so they induce equilibrium flow f^*. Then, by lemma 2.8, we should have that $0 = d_s^*(s) \leq d_s^*(v_j) \leq d_s^*(v_k) \leq d_s^*(t) = L_{eq}(H)$.

Let $v_{j+1}, v_{j+2}, \ldots, v_{k-1}$ be all the intermediate vertices of this light - edge path that begins from v_j and ends to v_k. Then, since every edge $e_{(v_m, v_{m+1})}$ with $m \in [j, k-1]$ is light, and by lemma 2.8, we should have the following:

$$\begin{aligned} d_s(v_k) - d_s(v_j) &= d_s(v_k) - d_s(v_{k-1}) + \ldots + d_s(v_{j+1}) - d_s(v_j) \\ &\leq d_s^*(v_k) - d_s^*(v_{k-1}) + \ldots + d_s^*(v_{j+1}) - d_s^*(v_j) \\ &= d_s^*(v_k) - d_s^*(v_j) \leq L_{eq}(H) \end{aligned}$$

The proof is complete.

□

2.3.3.2 Braess Graphs

Theorem 2.10 gives us an upper bound on the performance of the trivial algorithm. But there are some special graphs [4] that the trivial algorithm performs poorly, in the sense that it approximates the optimal solution with a ratio of *at least* the upper bound of $\lfloor \frac{n}{2} \rfloor$. These graphs are called *Braess graphs.*

For every positive integer k, there is a special graph called the *k-th Braess graph*, $B^k = (V_k, E_k)$, that has $2k+2$ vertices, $V_k = \{s, t, v_1, v_2, \ldots, v_k, w_1, w_2, \ldots, w_k\}$, and $4k+1$ edges, $E_k = \{(s, v_i), (v_i, w_i), (w_i, t) : 1 \leq i \leq k\} \cup \{(v_i, w_{i-1}) : 2 \leq i \leq k\} \cup \{(s, w_k), (v_1, t)\}$.

We next define edge latencies $l_e^k(\cdot)$, as follows:

- ◇ if $e = (v_i, w_i)$, then $l_e^k(x) = 0$, for $i \in [1, k]$,
- ◇ if $e = (v_i, w_{i-1})$ or (s, w_k) or (v_1, t), then $l_e^k(x) = 1$, for $i \in [2, k]$,
- ◇ if $e = (w_i, t)$ or $e = (s, v_{k-i+1})$, then $l_e^k(x)$ is a standard function that satisfies $l_e^k(\frac{k}{k+1}) = 0$ and $l_e^k(1) = i$, for $i \in [1, k]$.

Obviously, the B^1 graph is the original paradox graph (section 1.1), where vertices u, w of section's 1.1 graph play now the role of vertices w_1, v_1 respectively, but with one crucial difference. The edges (w_1, t) and (s, v_1) have the latencies defined above and not latency equal to x as in section 1.1., and that is the reason why the results of the next lemma are different if the same methodology is applied to the original paradox graph.

Lemma 2.11. *For any integer $n \geq 2$, there is an instance with n vertices for which the trivial algorithm produces a solution which is at least $\lfloor \frac{n}{2} \rfloor$ times larger than the optimal solution.*

Proof. For $n = 2$, consider the simplest graph G with two vertices s and t, and one edge linking them. Since there is only one subgraph that links vertices s and t too, that is subgraph $H = H^B = G$, then we have that $L_{eq}(G) = 1 \cdot L_{eq}(H^B)$. For $n = 3$, consider again the previous graph, but with the edge subdivided in order to produce one vertex more. Here again we have the same results.

For $n \geq 4$, the instance (G, r, l) is the one with $G = B^k$, traffic rate $r = k$, and edge latencies $l_e^k(\cdot)$.

For $n \geq 4$ even, the k-th Braess graph is the one with $n = 2k+2$ vertices. For $n \geq 4$ odd however, imagine a k-th Braess graph such that $n-1 = 2k+2$, but one edge of this graph is subdivided to produce one more vertex.

Let the traffic rate be k. Let also be the following paths $P_i : s \to v_i \to w_i \to t$ where $i \in [1, k]$, $Q_1 : s \to v_1 \to t$, $Q_i : s \to v_i \to w_{i-1} \to t$ where $i \in [2, k]$ and $Q_{k+1} : s \to w_k \to t$.

Routing one unit of flow on each of the $P_1, P_2, \ldots, P_k$ paths, gives a Nash equilibrium flow with $L_{eq}(G) = k + 1$.

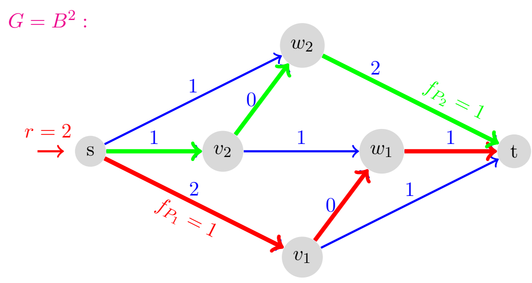


Figure 2.15: The 2-nd Braess Graph B^2.

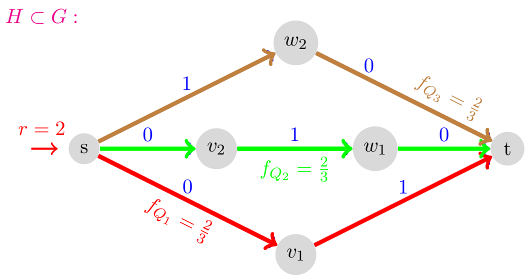


Figure 2.16: The Subnetwork H of B^2.

Let H be the subnetwork that is produced from B^k by deleting all edges of the form (v_i, w_i). If we route $\frac{k}{k+1}$ units of flow on each of the $Q_1, Q_2, \ldots, Q_k, Q_{k+1}$ paths, gives a Nash equilibrium flow with $L_{eq}(H) = 1$.
Thus we have that:

$$\frac{L_{eq}(G)}{L_{eq}(H^B)} \geq \frac{L_{eq}(G)}{L_{eq}(H)} = k + 1 = \lfloor \frac{n}{2} \rfloor$$

What follows now, is a demonstration of the proof for $n = 4$. Obviously $k = 2$, and the graph is the 2-nd Braess graph, $G = B^2$ (see Fig. 2.15).

At the equilibrium state above, the $r = 2$ traffic rate is divided equally to one unit of flow per each path, $P_1 : s \to v_1 \to w_1 \to t$ (red path) and $P_2 : s \to v_2 \to w_2 \to t$ (green path). Thus, $L_{eq}(G) = l_{P_1} = l_{P_2} = 2+0+1 = 1+0+2 = 3$.

If H is the subnetwork that is produced from B^2 by deleting edges (v_1, w_1), (v_2, w_2), then at the equilibrium state below, $\frac{2}{3}$ units of flow are routed through each path, $Q_1 : s \to v_1 \to t$ (red path), $Q_2 : s \to v_2 \to w_1 \to t$ (green path) and $Q_3 : s \to w_2 \to t$ (brown path). Thus, $L_{eq}(H) = l_{Q_1} = l_{Q_2} = l_{Q_3} = 0 + 1 = 0 + 1 + 0 = 1 + 0 = 1$. Fig. 2.16 shows this instance.

Thus, $\frac{L_{eq}(G)}{L_{eq}(H^B)} \geq \frac{L_{eq}(G)}{L_{eq}(H)} = 3 = \lfloor \frac{6}{2} \rfloor$. □

Until now, we have found an upper bound, namely $\lfloor \frac{n}{2} \rfloor$, on the performance of the trivial algorithm. We have also found some special graphs, namely the Braess graphs, that the network's equilibrium latency to the best subnetwork's equilibrium latency rate exceeds the $\lfloor \frac{n}{2} \rfloor$ upper bound.

2.3.3.3 General Latency Functions - An Approximability Threshold of $\lfloor \frac{n}{2} \rfloor$

Consequently, the question that arises, is whether we could find a *less than* $\lfloor \frac{n}{2} \rfloor$ - approximation algorithm for the GENERAL LATENCY NETWORK DESIGN PROBLEM. The following theorem by Roughgarden [4] clears the picture:

Theorem 2.12. *For every $\epsilon > 0$, there is no $(\lfloor \frac{n}{2} \rfloor - \epsilon)$ - approximation algorithm for the GENERAL LATENCY NETWORK DESIGN PROBLEM $\big($unless $P = NP\big)$.*

Proof. At first we assume that the graph $G = (V, E)$ has at least 4 vertices. That is, $n \geq 4$. Also, as in lemma 2.11, we may assume that n is even.

The proof depends on a reduction from a problem, known to be NP-hard, the Partition problem. This problem states that "given a set of p positive integers $a_1, a_2, \ldots, a_p$, is there a subset $S \subseteq \{1, 2, \ldots, p\}$, such that $\sum_{j \in S} a_j = \frac{1}{2} \sum_{j=1}^{p} a_j$?"

Here again, the graph G is the k-th Braess graph, $G = B^k$, with $n = 2k + 2$.

The reduction is polynomial in time, since we replace every edge of the form (v_i, w_i) by p parallel edges, where p is the total number of the positive integers $a_1, a_2, \ldots, a_p$ of the Partition problem. Let's denote these edges as $e_i^1, e_i^2, \ldots, e_i^p$.

We will now introduce a new property that characterizes an edge, called *capacity*. The capacity of an edge, is the maximum amount of flow that the edge can handle. So, if the flow that enters the link is below its capacity, then the flow will face acceptable delays. On the contrary, if the flow exceeds the link's capacity then the price paid will be by no means negligible. Thus, we will correlate the capacity of an edge with its latency. Roughly speaking, if the flow that passes through an edge e, exceeds its capacity by at least a small portion δ, then it becomes *oversaturated* and the latency grows too big, lets say by a very large constant M.

The capacity of an edge of the form e_i^j, where $j \in [1, p]$, is defined as the Partition problem's positive integer a_j. Wlog we may assume that each a_j is even. Let $A = \sum_{j=1}^{p} a_j$, and the traffic rate $r = k\frac{A}{2} + k + 1$. Let also δ be a sufficiently small constant, $\delta = \frac{1}{p+k}$, and M be a large positive integer, $M = \frac{n}{2}$.

We next define edge latencies $l_e(\cdot)$ with respect to edge capacities h_e, as follows:

◇ if $e = (v_i, w_{i-1})$ or (s, w_k) or (v_1, t), $i \in [2, k]$ then:

- $l_e(x) = 1$, for $x \leq 1$,
- $l_e(x) = M$, for $x \geq 1 + \delta$.

The capacity therefore is $h_e = 1$.

◇ if $e = (w_i, t)$ or $e = (s, v_{k-i+1})$, $i \in [1, k]$ then:

- $l_e(x) = 0$, for $x \leq \frac{1}{2}A + 1$,
- $l_e(x) = i$, for $x = \frac{1}{2}A + \frac{k+1}{k}$,
- $l_e(x) = M$, for $x \geq \frac{1}{2}A + \frac{k+1}{k} + \delta$.

The capacity therefore is $h_e = \frac{1}{2}A + \frac{k+1}{k}$.

◇ if $e = e_i^j$, $i \in [1, k]$ and $j \in [1, p]$ then:

- $l_e(x) = 0$, for $x \leq a_j - \delta$,
- $l_e(x) = 1$, for $x = a_j$,
- $l_e(x) = M$, for $x \geq a_j + \delta$.

The capacity is $h_e = a_j$.

The proof then follows from the following approach. If there is a solution to the Partition problem, then we can find this subset S, remove the unnecessary edges and get a subnetwork H that has equilibrium latency in reasonable numbers. But if there is no solution to the Partition problem, then we can either remove too few edges and have links where the excess of their capacity leads to flows at the limits of those capacities causing delays, or remove too many edges causing insufficient capacity to some links and thus oversaturated edges.

We are ready now to move on to the main part of the proof. The proof is by contradiction, and thus suppose that there exists an algorithm $\mathcal{A}$ that approximates the Best Subnetwork H^B with approximation ratio $\lfloor \frac{n}{2} \rfloor - \epsilon$, $\epsilon > 0$. That is, the algorithm $\mathcal{A}$ returns subgraph $H_{\mathcal{A}}$ with common latency $L_{eq}(H_{\mathcal{A}})$, such that $\frac{L_{eq}(H_{\mathcal{A}})}{L_{eq}(H^B)} \leq \lfloor \frac{n}{2} \rfloor - \epsilon$. We claim that the algorithm decides the Partition problem.

- $L_{eq}(H_{\mathcal{A}}) \geq \lfloor \frac{n}{2} \rfloor$:

  We prove that the Partition instance is 'No'. So, let's assume by contradiction that the Partition instance is 'Yes'.

  Then, there is a subset $S \subseteq \{1, 2, \ldots, p\}$, such that $\sum_{j \in S} a_j = \frac{1}{2}A$. Consider all edges e_i^j, $j \in [1, p]$. We will keep those e_i^j edges, where $j \in S$ and destroy the rest, that is destroy e_i^j edges for every $i \in [1, k]$,

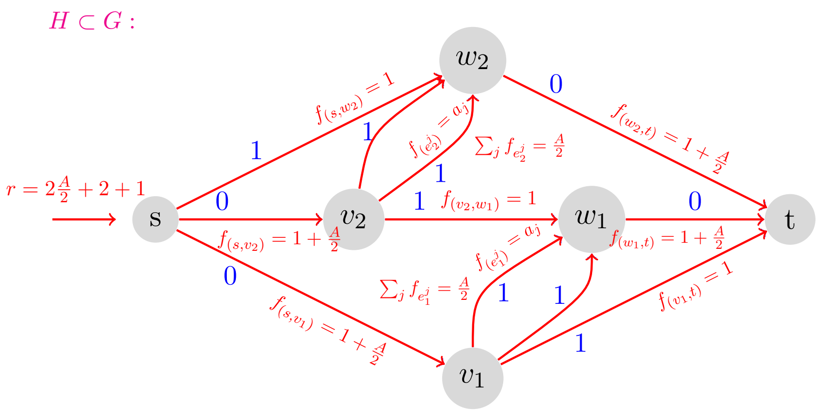


Figure 2.17: $L_{eq}(H_{\mathcal{A}}) \geq \lfloor \frac{n}{2} \rfloor$ - Assuming Partition instance is 'Yes'.

where $j \in \{1, 2, \dots, p\} \setminus S$. The new graph obtained, H, is a subgraph of G, $H \subset G$.

Consider again the $Q_1, Q_2, \dots, Q_{k+1}$ paths, as they were defined in lemma 2.11. Consider also a feasible flow f such that a_j units of flow are routed on each unique path containing edge e_i^j, and 1 unit of flow is routed on each path Q_i, $i \in [1, k+1]$. Then it is not hard to show that this is a Nash equilibrium flow for the subgraph H, with $L_{eq}(H) = 1$.

This means that $\frac{L_{eq}(H_{\mathcal{A}})}{L_{eq}(H^B)} \geq \frac{\lfloor \frac{n}{2} \rfloor}{L_{eq}(H)} = \frac{\lfloor \frac{n}{2} \rfloor}{1} = \lfloor \frac{n}{2} \rfloor$. But the latter contradicts the hypothesis which says that $\frac{L_{eq}(H_{\mathcal{A}})}{L_{eq}(H^B)} \leq \lfloor \frac{n}{2} \rfloor - \epsilon$.

What follows now, is a demonstration of this case for $n = 4$, which shows that $L_{eq}(H) = 1$. Obviously $k = 2$, and the graph is the 2-nd Braess graph, $G = B^2$.

If H is the subnetwork that is produced from B^2 by sustaining only those edges e_i^j, $j \in S$, then at the equilibrium state below, from the total traffic rate $r = 2\frac{A}{2} + 2 + 1$, a_j units of flow are routed on each unique path containing edge e_i^j, and 1 unit of flow is routed on each path Q_1, Q_2, Q_3. Thus, every path has equal equilibrium latency, $L_{eq}(H) = 1$.

In Fig. 2.17, the blue colored numbers are the latencies of each link, while the red ones are the corresponding edge flows.

- $L_{eq}(H_{\mathcal{A}}) < \lfloor \frac{n}{2} \rfloor$:

  We prove that the Partition instance is 'Yes'. So, let's assume by contradiction that the Partition instance is 'No'. This means that for every subset $S \subseteq \{1, 2, \dots, p\}$, we have that $\sum_{j \in S} a_j \neq \frac{1}{2}A$. So, we have two cases to consider:

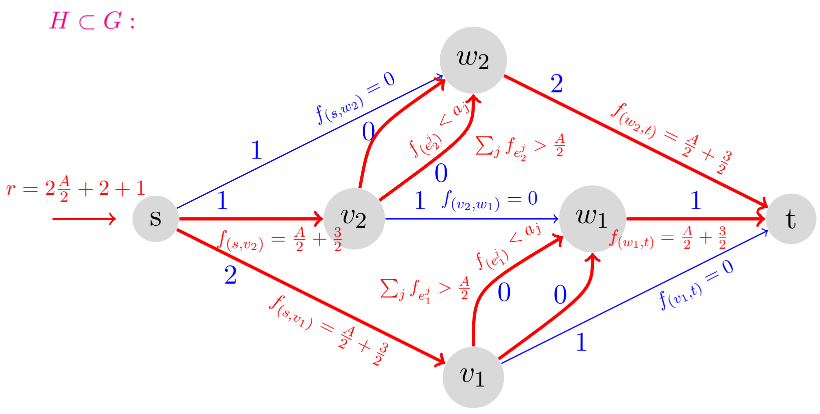


Figure 2.18: $L_{eq}(H_{\mathcal{A}}) < \lfloor \frac{n}{2} \rfloor$ - Assuming Partition instance is 'No'. (a) $\sum_{j \in S} a_j > \frac{1}{2}A$.

- $\diamond$ $\sum_{j \in S} a_j > \frac{1}{2}A$:
  Since every a_j is even, then we have that $\sum_{j \in S} a_j \geq \frac{1}{2}A + 2$. We will keep again, those e_i^j edges, where $j \in S$ and destroy the rest, that is destroy e_i^j edges, where $j \in \{1, 2, \ldots, p\} \setminus S$. The new graph obtained, H, is a subgraph of G, $H \subset G$.
  Obviously, since $\sum_{j \in S} a_j \geq \frac{1}{2}A + 2$, then $\frac{a_j}{\sum_{j \in S} a_j}(\frac{A}{2} + \frac{k+1}{k}) < a_j$ should hold.
  So, if we route $\frac{a_j}{\sum_{j \in S} a_j}(\frac{A}{2} + \frac{k+1}{k})$ units of flow along the unique $s - t$ path containing edge e_i^j, path $P_i^j : (s, v_i)$, e_i^j, (w_i, t), then we have an equilibrium flow with common equilibrium latency, $L_{eq}(H) = l_{P_i^j} = k + 1 = \lfloor \frac{n}{2} \rfloor$.
  Here again, is a demonstration of this case for $n = 4$, or $k = 2$.
  If H is the subnetwork that is produced from B^2 by sustaining only those edges e_i^j, $j \in S$, then at the equilibrium state below, from the total traffic rate $r = 2\frac{A}{2} + 2 + 1$, $\frac{a_j}{\sum_{j \in S} a_j}(\frac{A}{2} + \frac{3}{2})$ units of flow are routed on each unique path containing edge e_i^j. Thus, every path $P_i^j : (s, v_i)$, e_i^j, (w_i, t) has equal equilibrium latency, $L_{eq}(H) = l_{P_i^j} = 2 + 1 = \lfloor \frac{6}{2} \rfloor$.
  In Fig. 2.18, the red colored paths are the ones that route equilibrium flow, while the blue colored edges carry no flow at all. If we sum up each red path's latencies (blue numbers), we get the common equilibrium latency.
- $\diamond$ $\sum_{j \in S} a_j < \frac{1}{2}A$:
  Since every a_j is even, then we have that $\sum_{j \in S} a_j \leq \frac{1}{2}A - 2$. We

will keep again, those e_i^j edges, where $j \in S$ and destroy the rest, that is destroy e_i^j edges, where $j \in \{1, 2, \dots, p\} \setminus S$. The new graph obtained, H, is a subgraph of G, $H \subset G$.
This is the case where we will exploit the property that every edge has, that of capacity.
Since each e_i^j has capacity a_j (meaning that if the flow passed through that edge is bigger than the 'maximum' flow that it can carry, this then causes latency equal to M), all graph's e_i^j edges have total capacity $\sum_{j \in S} a_j$, which is at most $\frac{A}{2} - 2$, per each $i \in [1, k]$. But every (v_i, w_{i-1}), $i \in [2, k]$ or (v_1, t) edge has capacity equal to 1. This equivalently means that the total flow that can be entered in every vertex v_i cannot be more than $\frac{A}{2} - 1$ units, or else at least one edge, e_i^j or (v_i, w_{i-1}) for $i \in [2, k]$, e_1^j or (v_1, t) for $i = 1$, will be oversaturated.
However, we have k vertices of type v_i in H, so the overall capacity will be $k\frac{A}{2} - k$ units of flow. Now, having in mind that the total traffic rate is $k\frac{A}{2} + k + 1$, what only remains is that $2k + 1$ units of flow should be routed on edge (s, w_k). Since edge (s, w_k) has capacity 1, this means that it will be oversaturated, causing path latency at least $l_P = M \geq \lfloor \frac{n}{2} \rfloor$.
The meaning of all these that we have already described, is that we cannot avoid at least one edge of being oversaturated.
Here again, is a demonstration of this case for $n = 4$, or $k = 2$.
At the first case, edge (s, w_2) gets flow equal to its capacity, the worst case scenario. As we can see, the only solution that will cause none of the edges (s, v_1) or (s, v_2) to be oversaturated, is that the rest of the traffic be equally shared to $\frac{A}{2} + 1$. But this will oversaturate e_1^j or (v_1, t) for $i = 1$, e_2^j or (v_2, w_1) for $i = 2$.
In Fig. 2.19, the paths in red are those that have at least one link oversaturated. Here, paths $P_1 : s \to v_2 \to w_1 \to t$ and P_2^j : (s, v_1), e_i^j, (w_1, t), for every $j \in S$, have latency $l_{P_1} = l_{P_2^j} = M$.
At the second and final case as it is shown in Fig. 2.20, our efforts will be to avoid edges e_1^j or (v_1, t) for $i = 1$, e_2^j or (v_2, w_1) for $i = 2$ of being oversaturated, as we have described earlier. But this will cause extra flow on edge (s, w_2), beyond its capacity, and thus it gets oversaturated. In that case, path $P : s \to w_2 \to t$ (red path) has latency $l_P = M$.
Since for every feasible flow, at least one link will be oversaturated, we come to the conclusion that at the equilibrium flow, all paths' latencies will be at least M. Thus, $L_{eq}(H) \geq M \geq \lfloor \frac{n}{2} \rfloor$.

Thus, for every case and for every subnetwork H, $L_{eq}(H) \geq \lfloor \frac{n}{2} \rfloor$. But this leads us to contradiction, since by the hypothesis, there exists a

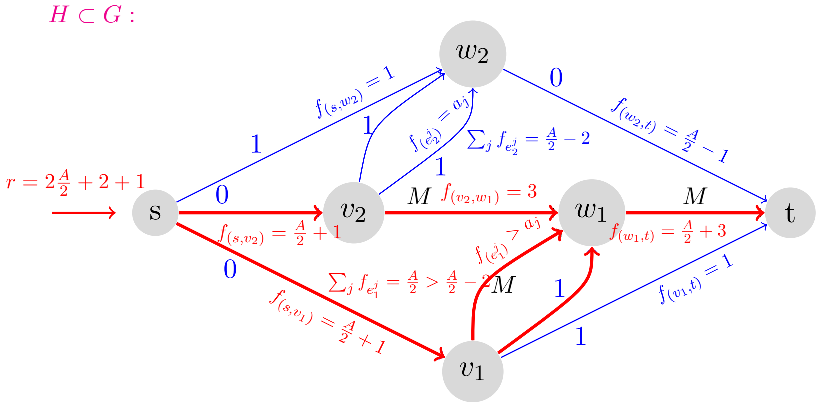


Figure 2.19: $L_{eq}(H_{\mathcal{A}}) < \lfloor \frac{n}{2} \rfloor$ - Assuming Partition instance is 'No'. (b) $\sum_{j \in S} a_j < \frac{1}{2}A$. (b1) Edge (s, w_2) gets flow equal to its capacity (worst case scenario). The paths in red are those that have at least one link oversaturated.

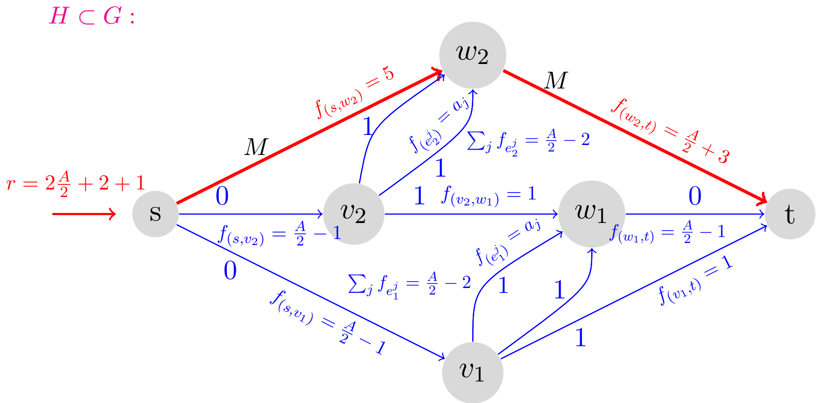


Figure 2.20: $L_{eq}(H_{\mathcal{A}}) < \lfloor \frac{n}{2} \rfloor$ - Assuming Partition Instance is 'No'. (b) $\sum_{j \in S} a_j < \frac{1}{2}A$. (b2) Avoid edges e_1^j or (v_1, t) for $i = 1$, e_2^j or (v_2, w_1) for $i = 2$ of being oversaturated. This will cause extra flow on edge (s, w_2), beyond its capacity, and thus it gets oversaturated.

subnetwork $H_{\mathcal{A}}$ returned by the algorithm, such that $L_{eq}(H_{\mathcal{A}}) < \lfloor \frac{n}{2} \rfloor$. This completes the proof.

□

2.4 Recognizing Paradox-Ridden Instances

We have just proved that detecting paradox-ridden networks with standard general or even linear latency functions is as hard as solving the $P = NP$ problem.

But if we relax our assumptions a little, by having latencies linear but strictly increasing, then things become easier. More details on these, in [16] and [2].

Theorem 2.13. *Paradox-ridden instances can be detected in polynomial time if the latency functions are strictly increasing and linear.*

Proof. Let's assume that we have an instance (G, r, l) with $G = (V, E)$.

In lemma 1.11 we have proved that given an instance with linear latency functions, then we can find its optimum flows in polynomial time. But by lemma 1.8, since the latencies are strictly increasing, then this optimum solution should be unique, say o.

Now, by definition 2.1, if the instance is paradox-ridden then there exists a subnetwork $H \subset G$, such that $L_{eq}(H) = L_{opt}(G) = \frac{L_{eq}(G)}{PoA_G}$. Put it differently, flow o of G should be equilibrium flow for H.

But lemma 2.8 gives us a convenient way of checking whether a feasible flow is an equilibrium one. Just verify if for every link $e = (v, w)$ with $o_e > 0$, $d_s(w) - d_s(v) = l_e(o_e)$ holds.

Apparently, this verification can be accomplished in polynomial time, since $E = \mathcal{O}(n^2)$, if $|V| = n$. Also, if the network is paradox-ridden, then $H = (V_H, E_H)$, where $V_H = V$ and $E_H = E \setminus \{e : o_e = 0\}$. □

Now, imagine an instance (G, r, l), with $G = (V, E)$, where latency functions are linear, $l_e(x) = a_e \cdot x + b_e$, where $a_e \geq 0$. This means that there could be links with strictly increasing latencies, $a_e > 0$, and links with constant latencies, $a_e = 0$. Let $E^c \subseteq E$, where $E^c = \{e \in E : a_e = 0\}$, the set of edges with constant latencies, and $E^i = E \setminus E^c$ the set of edges with strictly increasing latencies.

By lemmas 1.7 and 1.8, it should be clear that links with strictly increasing latencies induce the same optimal flow, while different optimal flows may travel through links with constant latencies. Thus, let's assume that we have in our hands an optimal flow o. The following linear program then, gives us the set of all optimal flows:

(LP):

$$\min C(f) = \min \sum_{e\in E^c} l_e(f_e)f_e = \min \sum_{e\in E^c} b_e \cdot f_e$$

subject to

$$\begin{aligned}
&\sum_{u:(v,u)\in E^i} o_{(v,u)} + \sum_{u:(v,u)\in E^c} f_{(v,u)} \\
&= \sum_{u:(u,v)\in E^i} o_{(u,v)} + \sum_{u:(u,v)\in E^c} f_{(u,v)}, && \forall v \in V \setminus \{s,t\} \\
&\sum_{u:(s,u)\in E^i} o_{(s,u)} + \sum_{u:(s,u)\in E^c} f_{(s,u)} = r, && flow-out \quad s\in V \\
&\sum_{u:(u,t)\in E^i} o_{(u,t)} + \sum_{u:(u,t)\in E^c} f_{(u,t)} = r, && flow-in \quad t\in V \\
&f_e \geq 0, && \forall e \in E^c
\end{aligned}$$

In particular, it should be clear that the solutions to the above (LP) are feasible flows, since they are flows that come out of s and end into t, summing up to the traffic rate r.

Moreover, a solution to the (LP) agrees with the given optimal flow o on all edges in E^i. Two different solutions of the (LP) differ only on edges belonging in E^c. Additionally, they minimize the objective function, which includes only the term $\sum_{e\in E^c} b_e \cdot f_e$, and does not include the term $\sum_{e\in E^i} a_e \cdot o_e^2 + b_e \cdot o_e$, since it is fixed for every solution to the (LP) above. Thus, every solution corresponds to an optimal flow.

Conversely, consider an optimal flow $\tilde{o}$, other than the optimal flow o given. It should be clear that it induces the same latency with o, for all $e \in E^i$. It only causes different $b_e \cdot f_e$ values, for every $e \in E^c$, than o does (although the sum will be the same). That is, optimal flow $\tilde{o}$ minimizes the objective function. It also satisfies all (LP)'s conditions, at first since it is a feasible flow, and secondly since it differs only on edges with constant latencies. Thus, it is a solution to (LP).

But we can find this optimal flow o by solving section's 1.4 (NLP 2), which by lemma 1.11 can be computed in polynomial time. Moreover, if (LP) has a unique optimal solution, this by [17] theorem 2, can be computed also in polynomial time. Thus we have the following theorem:

Theorem 2.14. *For instances with linear latency functions, the problem of detecting paradox-ridden networks can be decided in polynomial time, where (LP) has a unique optimal solution.*

In fact, if (LP) has a small number of feasible solutions, that means a polynomial number of feasible solutions, the problem of detecting paradox-ridden networks can also be decided in polynomial time. After all, if o is a feasible optimal solution that is a Nash equilibrium at the same time, then any other optimal solution o' with $\{e : o'_e > 0\} \subseteq \{e : o_e > 0\}$ is also a Nash equilibrium.

A network with a small number of constant latency functions, is such an example, which is usually the case, since in real networks there are not so many constant latency links.

In section 1.5 (Hochbaum, Shanthikumar [9]) we have referred to a way of finding ϵ - accurate optimum solution to the (NLP), in time polynomial in $\log(\frac{1}{\epsilon})$ and the input size. This means that, given an instance (G, r, l) where $G = (V, E)$, we can compute rather efficiently an optimal flow of G, within any specified accuracy level.

Let's assume now that the latency functions are polynomials. That is, $l_e(x) = \sum_{i=0}^{d} a_{e,i} x^i$, where $a_{e,i} \geq 0$, for every $e \in E$.

So, having computed optimal flow o, it should be rather tempting, due to monetary reasons, to try to modify coefficients $a_{e,i}$ in such a way that optimal flow o is turned into a Nash flow!

This modification should be the minimum one, which means that the Euclidean distance of each edge's e new coefficients $\tilde{a}_{e,i}$ with the original ones $a_{e,i}$, should be the minimum possible. Also, the equilibrium latency of every path P, with $o_P > 0$, should be equal to $L_{opt}(G)$.

This problem is called *MINIMUM LATENCY MODIFICATION PROBLEM.*

Theorem 2.15. *For instances with polynomial latency functions, the MINIMUM LATENCY MODIFICATION PROBLEM can be solved in polynomial time.*

Proof. Let's assume that the optimum flow is traversed through the network $G_o = (V, E_o)$, where $G_o \subseteq G$.

After all the above mentioned, the following quadratic program (QP) should give solution to the problem:

(QP):

$$\min \sum_{e \in E_o} \sum_{i=0}^{d} (a_{e,i} - \tilde{a}_{e,i})^2$$

subject to

$$\sum_{e \in P} \sum_{i=0}^{d} \tilde{a}_{e,i} o_e^i = L_{opt}(G), \forall P \in \mathcal{P}_{G_o}$$

$$\tilde{a}_{e,i} \geq 0, \qquad \forall e \in E_o, \forall i \in \{0, 1, \ldots, d\}$$

Since o is computed efficiently, by the first equality constraint of (QP), all positive flow driven paths have equal latency $L_{opt}(G)$.

Also, (LP) always admits a feasible solution. Indeed, by corollary 1.5, optimum flow o is a Nash flow on (G_o, r, l^*).

But $l_e^*(x) = \frac{d(x \cdot l_e(x))}{dx} = \sum_{i=0}^{d}(i+1)a_{e,i}x^i$.

This means that there exists a common equilibrium latency $\Lambda > 0$ for all paths in G_o. So, if we scale every edge's e coefficient $(i+1)a_{e,i}$ by a factor $\frac{L_{opt}(G)}{\Lambda}$, then we get the (QP)'s first equality constraint.

But (QP) is a convex separable quadratic program, that can be solved in polynomial time within any specified accuracy (for more on this, the reader is advised to visit [19]), and this completes the proof. □

Chapter 3

Braess's Paradox in Large Random Graphs

3.1 Braess Ratio

Having described all the above about the Braess's Paradox, it is time to address the following fundamental question: "Is Braess's paradox a widespread phenomenon, or is it such a rare case that can be ignored in practice?" This question has been addressed by [20], [21], [22], and is answered in the affirmative.

Before getting into the details of that answer, it is time to introduce a factor, called the *Braess ratio*. The Braess ratio is the largest factor by which the equilibrium latency of all traffic can be decreased by edge removals. More formally:

Definition 3.1. The Braess ratio $\beta(G, r, l)$ of an instance $\mathcal{G} = (G, r, l)$ is defined as:

$$\beta(G, r, l) = \max_{H \subseteq G} \min_{i=1}^{k} \frac{L^i_{eq}(G, r, l)}{L^i_{eq}(H, r, l)}$$

where H ranges over all the subnetworks of G that contain an $s_i - t_i$ path for each i.

In a single commodity network, definition 3.1 simplifies to:

$$\beta(G, r, l) = \max_{H \subseteq G} \frac{L_{eq}(G, r, l)}{L_{eq}(H, r, l)} \tag{3.1}$$

where $L_{eq}(H, r, l)$ denotes the equilibrium latency of the subnetwork H with traffic rate r and cost functions $l(\cdot)$.

The following three lemmas show the Braess ratio's relationship with the PoA_G factor, and the paradox-free, paradox-ridden instances. The proofs are quick and easy.

Lemma 3.2. *If G is a single commodity network then:*

$$\beta(G, r, l) \leq PoA_G$$

Proof. By relations (1.8) and (3.1) we have:

$$\beta(G, r, l) = \max_{H \subseteq G} \frac{L_{eq}(G, r, l)}{L_{eq}(H, r, l)} \leq \frac{L_{eq}(G, r, l)}{L_{opt}(G, r, l)} = PoA_G$$

□

Lemma 3.3. *If a single commodity network G is **paradox-free** then:*

$$\beta(G, r, l) \leq 1$$

Proof. By relations (1.8), (3.1) and definition 2.2 we have:

$$\beta(G, r, l) = \max_{H \subseteq G} \frac{L_{eq}(G, r, l)}{L_{eq}(H, r, l)} \leq 1$$

since $\forall H \subseteq G : L_{eq}(H, r, l) \geq L_{eq}(G, r, l)$. □

Lemma 3.4. *If a single commodity network G is **paradox-ridden** then:*

$$\beta(G, r, l) = PoA_G$$

Proof. By relations (1.8), (3.1) and definition 2.1 we have:

$$\beta(G, r, l) = \max_{H \subseteq G} \frac{L_{eq}(G, r, l)}{L_{eq}(H, r, l)} = \frac{L_{eq}(G, R, l)}{L_{opt}(G, R, l)} = PoA_G$$

since $\exists G' \subseteq G$, such that $\forall H \subseteq G$:
$L(G', r, l) \leq L(H, r, l)$ and $L(G', r, l) = L_{opt}(G, r, l)$. □

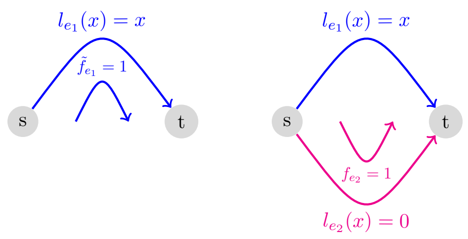
3.1.1 Two Useful Lemmas

Before moving on, we should plug into this section two useful lemmas, lemma 3.5 and lemma 3.8.

Lemma 3.5. *If f is a Nash equilibrium flow for (G, r, l), then for every vertex v we have that:*

$$d_s(v) + d_t(v) \geq L_{eq}(G, r, l)$$

where $d_t(v)$ denotes the length, with respect to edge lengths $\{l_e(f_e)\}_{e \in E}$, of the shortest $v - t$ path in G, with equality holding if v is a flow carrying vertex.



(a) Original graph (b) Added lower link

Figure 3.1: $(f, \tilde{f})$-light and heavy edges, $(f, \tilde{f})$-alternating paths. e_1 is a forward $(f, \tilde{f})$-light edge and alternating path, e_2 is a forward $(f, \tilde{f})$-heavy edge.

Proof. If $v \equiv s$ or $v \equiv t$, then this case is trivial.

It is easy to see, that the above relation holds by the "triangle inequality", for all kinds of flows, and by the definition of shortest path labels d.

Now, if v is a flow carrying vertex, then it cannot be the case $d_s(v) + d_t(v) > L_{eq}(G, r, l)$, because this contradicts the concept of the equilibrium flow, since there would be a path with latency bigger than that of $L_{eq}(G, r, l)$.

But if we accept this fact for that specific path, then all of the rest paths, should suffer from the same equilibrium overall latency $d_s(v) + d_t(v) = L'$. This means that all paths have the same latency $L' > L_{eq}(G, r, l)$, which contradicts the meaning of the equilibrium latency $L_{eq}(G, r, l)$.

□

Lemma 3.8 states that the equilibrium latency is a non-decreasing or strictly increasing function of the traffic rate, if the latencies are non-decreasing or strictly increasing respectively. In order to prove that, we have to define the idea of an "alternating path", which I guess it's worth mentioning.

Definition 3.6. Let f and $\tilde{f}$ be flows feasible for the instances (G, r, l) and $(G, \tilde{r}, l)$, respectively:

- An edge e of G is $(f, \tilde{f})$-light if $f_e \le \tilde{f}_e$ and $\tilde{f}_e > 0$, $(f, \tilde{f})$-heavy if $f_e > \tilde{f}_e$, and $(f, \tilde{f})$-null if $f_e = \tilde{f}_e = 0$.
- An undirected path is $(f, \tilde{f})$-alternating if it comprises only forward light edges and / or backward heavy edges.

Consider the instance of Fig. 3.1.

It is obvious that adding the lower link, the overall latency decreases from 1 to 0. The upper link is then a forward light edge, since $f_{e_1} = 0 < 1 = \tilde{f}_{e_1}$, while the lower link is a forward heavy edge, since $f_{e_2} = 1 > 0 = \tilde{f}_{e_2}$. The

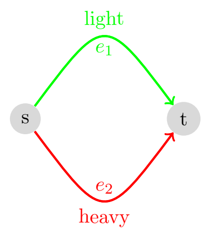
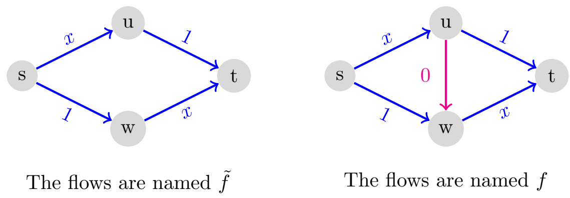


Figure 3.2: An overall demonstration - e_1 is a forward $(f, \tilde{f})$-light edge and alternating path, e_2 is a forward $(f, \tilde{f})$-heavy edge.



(a) Original graph

(b) Added (u, w) link

Figure 3.3: Original Braess's paradox instance - $(f, \tilde{f})$-light and heavy edges, $(f, \tilde{f})$-alternating paths.

undirected upper path is apparently $(f, \tilde{f})$-alternating. Fig. 3.2 summarizes this fact.

Now, consider the original Braess's paradox instance in Fig. 3.3.

As we have already mentioned, adding the (u, w) link, the overall latency now increases from 1 to 2. The (u, t) and (s, w) links are then forward light edges, since $f_{(u,t)} = f_{(s,w)} = 0 < \frac{1}{2} = \tilde{f}_{(u,t)} = \tilde{f}_{(s,w)}$, while (s, u), (u, w) and (w, t) links are forward heavy edges, since $f_{(s,u)} = f_{(w,t)} = 1 > \frac{1}{2} = \tilde{f}_{(s,u)} = \tilde{f}_{(w,t)}$ and $f_{(u,w)} = 1 > 0 = \tilde{f}_{(u,w)}$. Since (u, w) is a forward heavy edge, (w, u) is then a backward heavy edge. The undirected path (s, w), (w, u) and (u, t) is apparently $(f, \tilde{f})$-alternating. Fig. 3.4 summarizes this fact.

The idea behind this schema is that, according to [23], along an $(f, \tilde{f})$-alternating path, call it P, with no added edges, we can prove by induction that $d_s(v) \leq \tilde{d}_s(v), \forall v \in P$, like the first instance above, whereas if the alternating path contains the added edge, like the original Braess's paradox instance does, $d_s(v) \geq \tilde{d}_s(v), \forall v \in P$.

It can be proven, that an $s - t$ alternating path always exists ([13]):

Lemma 3.7. *Let f and $\tilde{f}$ be flows feasible for the single-commodity instances (G, r, l) and $(G, \tilde{r}, l)$, respectively, with $r \leq \tilde{r}$. Let also $G = (V, E)$. Then there is an $(f, \tilde{f})$-alternating $s - t$ path. Moreover, if f is directed acyclic, then every such path begins and ends with an $(f, \tilde{f})$-light edge.*

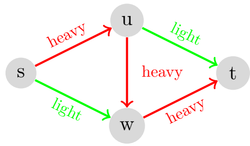


Figure 3.4: Original Braess's paradox instance - (a) (u,t) and (s,w) are forward $(f,\tilde{f})$-light edges. (b) (s,u), (u,w) and (w,t) are forward $(f,\tilde{f})$-heavy edges. (c) The undirected path (s,w), (w,u) and (u,t) is $(f,\tilde{f})$-alternating.

Proof. The proof is by contradiction. Suppose that there is no $(f,\tilde{f})$-alternating $s-t$ path. Let S denote the maximal set of nodes reachable from s via $(f,\tilde{f})$-alternating paths. This means that S contains s, but it does not contain t, since then there would be at least one $(f,\tilde{f})$-alternating $s-t$ path. Apparently, S is an $s-t$ cut. Thus:

$$\sum_{e\in\delta^+(S)} f_e - \sum_{e\in\delta^-(S)} f_e = r \tag{3.2}$$

$$\sum_{e\in\delta^+(S)} \tilde{f}_e - \sum_{e\in\delta^-(S)} \tilde{f}_e = \tilde{r} \tag{3.3}$$

where $\delta^+(S)$ and $\delta^-(S)$ are similar sets of edges defined in theorem 2.10.

Edges that exit S cannot be light, since then the corresponding vertices that these edges have as head, should belong to S, which contradicts the hypothesis of S's maximality. This means that all edges that belong to $\delta^+(S)$, should be $(f,\tilde{f})$-heavy or null.

For the same reason, edges that enter S cannot be heavy, since then the corresponding vertices that these edges have as tail, should belong to S, which contradicts the hypothesis of S's maximality. So, all edges that belong to $\delta^-(S)$, should be $(f,\tilde{f})$-light or null.

Also, at least one edge that belongs to $\delta^+(S)$, should be $(f,\tilde{f})$-heavy. That is, because if all edges were $(f,\tilde{f})$-null, then by relations (3.2) and (3.3) we should have that $0-\sum_{e\in\delta^-(S)} f_e = r$, $0-\sum_{e\in\delta^-(S)} \tilde{f}_e = \tilde{r}$ or that $r \leq 0$, $\tilde{r} \leq 0$ respectively. Then, the only one valid case would be that $r=\tilde{r}=0$, which is a trivial case.

So, $\sum_{e\in\delta^+(S)} f_e > \sum_{e\in\delta^+(S)} \tilde{f}_e$, since edges that belong to $\delta^+(S)$ are $(f,\tilde{f})$-heavy or null, with at least one edge being $(f,\tilde{f})$-heavy. Also, $\sum_{e\in\delta^-(S)} f_e \leq \sum_{e\in\delta^-(S)} \tilde{f}_e$ or $-\sum_{e\in\delta^-(S)} f_e \geq -\sum_{e\in\delta^-(S)} \tilde{f}_e$, since edges that belong to $\delta^-(S)$ are $(f,\tilde{f})$-light or null. Thus, combining the last with the relations (3.2) and (3.3) above, we have that:

$$r = \sum_{e \in \delta^+(S)} f_e - \sum_{e \in \delta^-(S)} f_e > \sum_{e \in \delta^+(S)} \tilde{f}_e - \sum_{e \in \delta^-(S)} \tilde{f}_e = \tilde{r}$$

which contradicts our hypothesis.

Moreover, if f is directed acyclic, then it sends no flow into s or out of t. Thus, the first and last edges of every $(f, \tilde{f})$-alternating $s - t$ path are light. □

Lemma 3.8. *For any network G and continuous, **non-decreasing** or **strictly increasing** latency functions $l_e(\cdot)$, $L_{eq}(G, r, l)$ is continuous and **non-decreasing** or **strictly increasing** function of r, respectively.*

Proof. The proof of this lemma, is due to [13], and uses lemma 3.7.

Let's assume that $r \leq \tilde{r}$ and that the latency functions are non-decreasing. What we need to prove is that $L_{eq}(G, r, l) \leq L_{eq}(G, \tilde{r}, l)$ or equivalently that $d_s(t) \leq \tilde{d}_s(t)$, since $L_{eq}(G, r, l) = d_s(t)$, $L_{eq}(G, \tilde{r}, l) = \tilde{d}_s(t)$.

But we are going to prove a stronger result, that $d_s(v) \leq \tilde{d}_s(v)$ for every v that belongs to an $(f, \tilde{f})$-alternating $s-t$ path, which by lemma 3.7 always exists.

The proof is by induction.

The base case, $d_s(s) = \tilde{d}_s(s)$ is trivial.

So, let's suppose that $d_s(v) \leq \tilde{d}_s(v)$, which is the induction hypothesis, and assume that the next vertex on the $(f, \tilde{f})$-alternating $s - t$ path is w. Then there are only two possibilities:

- ◇ Edge $e = (v, w)$ is forward light:

  Then $f_e \leq \tilde{f}_e$, and since $l_e(\cdot)$ is a non-decreasing latency function, we have that $l_e(f_e) \leq l_e(\tilde{f}_e)$.

  But from lemma 2.8, inequality (2.1) and by the induction hypothesis, we have that:

  $$d_s(w) \leq d_s(v) + l_e(f_e) \leq \tilde{d}_s(v) + l_e(\tilde{f}_e) = \tilde{d}_s(w)$$

  since $\tilde{f}_e > 0$ (by lemma 2.8).

- ◇ Edge $e = (w, v)$ is backward heavy:

  Then $f_e > \tilde{f}_e$, and since $l_e(\cdot)$ is a non-decreasing latency function, we have that $l_e(f_e) > l_e(\tilde{f}_e)$.

  Since $f_e > \tilde{f}_e$, then $f_e > 0$, which by lemma 2.8 means that $d_s(v) = d_s(w) + l_e(f_e)$. Also, by the same lemma, we have that $\tilde{d}_s(v) \leq \tilde{d}_s(w) + l_e(\tilde{f}_e)$. Then, using also the induction hypothesis, we have that:

  $$d_s(w) = d_s(v) - l_e(f_e) < \tilde{d}_s(v) - l_e(\tilde{f}_e) \leq \tilde{d}_s(w)$$

The induction is complete.

Now, if we have strictly increasing latency functions, things are still simple. We just have to modify the base case, since $d_s(s) = \tilde{d}_s(s)$, and check that for the first node v of the $(f, \tilde{f})$-alternating $s - t$ path, $d_s(v) < \tilde{d}_s(v)$. This is simple to check using the same method as described above. Just replace relation $l_e(f_e) \leq l_e(\tilde{f}_e)$ with $l_e(f_e) < l_e(\tilde{f}_e)$ for forward light edge $e = (s, v)$, and conclude that $d_s(v) < \tilde{d}_s(v)$. The same thing applies for the main body of the induction.

The proof is complete. □

3.2 Graph Models

To show that the Braess's paradox is or is not likely to happen, papers [20], [21] make extent use of the **Erdös-Rényi** random graph model. Paper [22] uses the notion of *Expanders*, graphs with some special attributes.

The Erdös-Rényi random graph model is used to prove the existence of graphs satisfying various properties, or equivalently, to provide a rigorous definition of what it means for a property to hold for *almost all graphs*.

It is a model for generating $G \in \mathcal{G}(n, p)$ random graphs with:

- n nodes.
- equal probability p of setting an edge between each pair of nodes, independently of the other edges.

The Erdös-Rényi random graph model, has the following properties:

◇ $G \in \mathcal{G}(n, p)$ has $\binom{n}{2}p$ edges on average.

◇ The degree distribution is binomial [24]:

$$P\big(deg(v) = k\big) = \binom{n-1}{k} p^k (1-p)^{(n-1-k)}$$

◇ As n grows large enough, the degree distribution approximates Poisson distribution:

$$P\big(deg(v) = k\big) \longrightarrow \frac{(np)^k e^{(-np)}}{k!}$$

as $n \to \infty$ and $np = const$. Thus:

$$\mathbb{E}[deg(v)] = np \tag{3.4}$$

◇ $\frac{\ln n}{n}$ is a *sharp threshold* of the connectedness of a graph G in $\mathcal{G}(n, p)$ [25]:

- if $np = 1$ then a graph $G \in \mathcal{G}(n,p)$ will *almost surely* have a largest component whose size is of order $n^{\frac{2}{3}}$.
- If np tends to a constant $c > 1$, then a graph $G \in \mathcal{G}(n,p)$ will *almost surely* have a unique giant component containing a positive fraction of the vertices. No other component will contain more than $\mathcal{O}(\log n)$ vertices.
- if $np < (1-\epsilon)\ln n$ or better $p < (1-\epsilon)\frac{\ln n}{n}$, then $G \in \mathcal{G}(n,p)$ will *almost surely* contain isolated vertices, and thus be disconnected.
- if $np > (1+\epsilon)\ln n$ or better $p > (1+\epsilon)\frac{\ln n}{n}$, then $G \in \mathcal{G}(n,p)$ will *almost surely* be connected.

The (α,β)-*Expander*, $\alpha,\beta > 0$, is a special family of graphs $G = (E,V)$, where the *expansion factor* is constant:

$$\min_{U \subset V, |U| \leq \frac{|V|}{2}} \frac{|\Gamma(U)|}{|U|} = \Omega(1)$$

where $\Gamma(U)$ is the neighborhood of U. Formally speaking, what follows is the definition of the (α,β)-Expander graphs:

Definition 3.9. (α,β)**-Expander graphs**:
A graph with n vertices is an $\big(\alpha,\beta\big)$-Expander if for every set of vertices U:

$$|\Gamma(U) \cup U| \geq \min\{(1+\alpha)|U|, (1+\beta)\frac{n}{2}\}$$

Intuitively, an Expander is a graph that has strong connectivity properties. It is a graph with a constant average degree, where beginning from s and moving towards t, every set of vertices (including the singleton set that contains only one vertex), expand themselves with a constant expansion factor α until they reach about half of the vertices V of the graph, precisely a factor $1+\beta$ of $\frac{|V|}{2}$. The same applies conversely, moving from t to s.

Many real world networks are (α,β)-Expanders. Networks with highly skewed degree distributions, such as *power-laws*, belong to this family.

For a fixed value $\gamma > 1$, we say that a graph is a power law graph with exponent γ, if the number of vertices of degree k is proportional to $k^{-\gamma}$.

The Internet is such a network. It has been proven that in the Internet, the frequency of nodes with degree d, f_d, is proportional to some power of d:

$$f_d \propto d^{\zeta}$$

where $\zeta < 0$.

For the Internet again, a node's degree d_v is proportional to some power of the node's rank r_v, where rank is the node's index, when we sort all nodes in decreasing order:

$$d_v \propto r_v^\eta$$

where $\eta < 0$.

More on this can be found in [26], [27], [28], [29] and [30].

It is now time to quote three more lemmas without proof. The first one is the expansion property of the Erdös-Rényi graphs, similar with the one of the (α, β)-Expanders (definition 3.9).The second one is a probability theory's useful tool that gives exponentially decreasing bounds on tail distributions of sums of independent random variables, while the third one is the *Law of Large Numbers*, which relates the mean with the expected value of an experiment, when it is being performed a large number of times.

Lemma 3.10. *There is some $c > 1$ such that, if $p > \frac{c \ln n}{n}$, then with high probability every subset U of G in $\mathcal{G}(n, p)$ is such that:*

$$|\Gamma(U)| \geq \big(\frac{e-1}{e} - o(1)\big) \min\{np|U|, n\}$$

where $\Gamma(U)$ is the neighborhood of U.

Now, since $\frac{e-1}{e} > \frac{3}{5}$, we have that:

$$|\Gamma(U)| \geq \big(\frac{3}{5} - o(1)\big) \min\{np|U|, n\}$$

Lemma 3.11. *Chernoff-Hoeffding Bound: Let X_i be a collection of independent random variables such that $a_i \leq X_i \leq b_i$. Define $X = \sum_i X_i$, then for all $\epsilon \in (0, 1)$:*

$$\mathbb{P}\big(|X - \mathbb{E}[X]| \geq \epsilon \mathbb{E}[X]\big) \leq 2e^{-\frac{2\epsilon^2 \mathbb{E}[X]^2}{\sum_i (b_i - a_i)^2}}$$

or equivalently:

$$\mathbb{P}\big((1-\epsilon)\mathbb{E}[X] \leq X \leq (1+\epsilon)\mathbb{E}[X]\big) \geq 1 - 2e^{-\frac{2\epsilon^2 \mathbb{E}[X]^2}{\sum_i (b_i - a_i)^2}} \tag{3.5}$$

If in addition the X_i are identically distributed with $X_i \in \{0, 1\}$, then:

$$\begin{aligned} \mathbb{P}\big(X - \mathbb{E}[X] \geq \epsilon \mathbb{E}[X]\big) &\leq e^{-\frac{\epsilon^2 \mathbb{E}[X]}{3}} \\ \mathbb{P}\big(X - \mathbb{E}[X] \leq -\epsilon \mathbb{E}[X]\big) &\leq e^{-\frac{\epsilon^2 \mathbb{E}[X]}{2}} \end{aligned}$$

or equivalently:

$$\mathbb{P}\big((1-\epsilon)\mathbb{E}[X] \leq X \leq (1+\epsilon)\mathbb{E}[X]\big) \geq 1 - e^{-\frac{\epsilon^2 \mathbb{E}[X]}{3}} \tag{3.6}$$

$$\mathbb{P}\big(X \leq (1+\epsilon)\mathbb{E}[X]\big) \geq 1 - e^{-\frac{\epsilon^2 \mathbb{E}[X]}{3}} \tag{3.7}$$

$$\mathbb{P}\big(X \geq (1-\epsilon)\mathbb{E}[X]\big) \geq 1 - e^{-\frac{\epsilon^2 \mathbb{E}[X]}{2}} \tag{3.8}$$

Lemma 3.12. *Law of Large Numbers: The average $\overline{X}_n$ of the results obtained from a large number of trials, n, should be close to the expected value $\mathbb{E}[X_n]$, and will tend to become closer as more trials are performed:*

$$\overline{X}_n \xrightarrow[n\to\infty]{} \mathbb{E}[X_n]$$

where $\overline{X}_n = \frac{X_1+X_2+\ldots+X_n}{n}$.

In [20] Valiant and Roughgarden proved that for graphs $G \in \mathcal{G}(n,p)$, where $p >> n^{-\frac{1}{2}+\zeta} > (1+\epsilon)\frac{\ln n}{n}$ for some $\zeta > 0$, which means by the above that G is not only *almost surely* connected, but is *dense* in some sense, then Braess's paradox occurs with high probability.

In [21] Chung and Young proved that for graphs $G \in \mathcal{G}(n,p)$, where $p \geq \frac{c}{\mathbb{P}(\mathcal{B}\leq\frac{\delta}{\ln(n)})}\frac{\ln n}{n} = (1+\epsilon)\frac{\ln n}{n}$, $c > 1$ and $\mathcal{B}$ is a *reasonable distribution* (as will be explained later on), which means by the above that G is *almost surely* connected, but is *sparse* in some sense, then Braess's paradox occurs with high probability.

In [22] Chung, Young and Zhao proved that for the family of (α,β)-Expander graphs, Braess's paradox occurs with high probability. It can be shown that the standard Erdös-Rényi graph with $p > \frac{(1+\epsilon)\ln n}{n}$ is a $(\frac{3}{5}np, \frac{1}{4})$-Expander.

3.3 Reasonable Distributions

In order to fully exploit the idea of a random graph, we have to define a way to make the values of the latency functions random.

In [20] and [21] the graphs $G \in \mathcal{G}(n,p)$ have affine latency functions, $l_e(x) = a_e \cdot x + b_e$, strictly increasing, while in [22] the (α,β)-Expander graphs have polynomial latency functions of the form $l_e(x) = h_e(x) + b_e$, where $h_e(x) = a_{1,e}x + a_{2,e}x^2 + \ldots + a_{k,e}x^k$ is a strictly increasing function too.

The randomness of the latency function values is succeeded via the following.

For the affine latency functions $l_e(x) = a_e \cdot x + b_e$, the factors a_e, b_e are independently drawn from distributions $\mathcal{A}$, $\mathcal{B}$ respectively, that satisfy some mild technical conditions which are:

- $\mathcal{A}$ has bounded support $[A_{\min}, A_{\max}]$ with $A_{\min} > 0$.
- a_e coefficients are dense in a closed interval $I_{\mathcal{A}}$, $|I_{\mathcal{A}}| > 0$, meaning that $\forall J \subseteq I_{\mathcal{A}}$ and $|J| > 0$, $\mathbb{P}(a_e \in J) > 0$.
- b_e coefficients are dense around zero in a closed interval $I_{\mathcal{B}} = [0,\eta]$, $\eta > 0$, meaning that $\forall J \subseteq I_{\mathcal{B}}$ and $|J| > 0$, $\mathbb{P}(b_e \in J) > 0$.

The distributions $\mathcal{A}$, $\mathcal{B}$ under the above restrictions are called *reasonable distributions*.

For the polynomial latency functions $l_e(x) = h_e(x) + b_e$ where $h_e(x) = a_{1,e}x + a_{2,e}x^2 + \ldots + a_{k,e}x^k$, (h_e, b_e) are drawn independently from distributions $(\mathcal{H}, \mathcal{B})$. We call $(\mathcal{H}, \mathcal{B})$ reasonable if:

- $\mathbb{P}(\mathcal{H} \in \mathcal{H}^*) = 1$, where $\mathcal{H}^*$ is the set of convex, strictly increasing functions h_e such that $h_e(0) = 0$.
- there exist $H_{min}, H_{max} \in \mathcal{H}^*$ such that $H_{max}(0) = H_{min}(0) = 0$ and $\mathbb{P}(H_{min} \leq \mathcal{H} \leq H_{max}) = 1$.
- there exist $H_1, H_2 \in \mathcal{H}^*$ where $H_1(0) = H_2(0) = 0$ and $H_1 < H_2$ otherwise, such that for all $\epsilon > 0$, $\mathbb{P}\big((1-\epsilon)H_1 \leq \mathcal{H} \leq H_1\big) > 0$ and $\mathbb{P}\big((1-\epsilon)H_2 \leq \mathcal{H} \leq H_2\big) > 0$.
- there is some interval $I_{\mathcal{B}} = [0, \nu]$ with $\nu > 0$ such that for every non-trivial subinterval $J \subseteq I_{\mathcal{B}}$, $\mathbb{P}[b_e \in J] > 0$.

Actually, $\mathcal{H}$ includes a large natural class of random convex functions, where the distribution over polynomials is a sub-class of it. Let $A_1, A_2, \ldots, A_k$ be continuously distributed positive random variables. Then the distribution over polynomials $A_1x + A_2x^2 + \ldots + A_kx^k$, satisfies the conditions for $\mathcal{H}$. Thus, these reasonable distributions can be viewed as a strict generalization of the reasonable distributions of the affine latency functions described just above.

We will also say that an instance (G, r, l) of an (α, β)-Expander graph G is δ-reasonable if:

- $\mathcal{B}$ is a continuous distribution except potentially at 0.
- $\alpha\mathbb{P}(\mathcal{B} \leq \frac{\delta}{\ln(n)}) > 4$.
- $\kappa = \min_{v \in \Gamma(s) \cup \Gamma(t)} deg(v) - 1$ is such that κ is $\omega(1)$.
- $\min\{deg(s), deg(t)\}$ is $o\Big(\alpha\kappa\mathbb{P}\big(\mathcal{B} \leq \delta\big)\mathbb{P}\big(\mathcal{B} \leq \frac{\delta}{\ln(n)}\big)\Big)$.
- $L(G, r, l)$ is $\mathcal{O}(1)$.

It is now time to give formal definitions of the three problems [20], [21], [22]:

Theorem 3.13. ***Braess's Paradox occurs in Large Dense Random Graphs with linear edge latency functions:***

Let G be an Erdös-Rényi random graph on n vertices with edge probability $p = \Omega(n^{-(\frac{1}{2})+\zeta})$, $\zeta > 0$. Let $\mathcal{A}$ and $\mathcal{B}$ be reasonable distributions and let all latency functions have the form $l_e(f_e) = a_ef_e + b_e$ where (a_e, b_e) is distributed *according to $\mathcal{A}\times\mathcal{B}$. There is a constant $\rho > 1$ such that, with high probability, network G admits a flow rate r such that the instance (G, r, l) has Braess's Ratio at least ρ.*

Recall section 3.2. Here, $p >> n^{-\frac{1}{2}+\zeta} > (1+\epsilon)\frac{\ln n}{n}$.

Theorem 3.14. ***Braess's Paradox occurs in Large Sparse Random Graphs with linear edge latency functions:***
Let G be an Erdös-Rényi random graph on n vertices with edge probability p. Let $\mathcal{A}$ and $\mathcal{B}$ be reasonable distributions and let all latency functions have the form $l_e(f_e) = a_e f_e + b_e$ where (a_e, b_e) is distributed according to $\mathcal{A}\times\mathcal{B}$. There are constants $\delta > 0$, $c > 1$, and $\rho > 1$ such that, if $\mathbb{P}\big(\mathcal{B} \leq \frac{\delta}{\ln(n)}\big)pn \geq c\ln(n)$, then there is a flow rate r such that the instance (G, r, l) has Braess's Ratio at least ρ with high probability.

Recall section 3.2. Here, $p \geq \frac{c}{\mathbb{P}(\mathcal{B}\leq\frac{\delta}{\ln(n)})}\frac{\ln n}{n} = (1+\epsilon)\frac{\ln n}{n}$, $c > 1$.

Theorem 3.15. ***Braess's Paradox occurs in (α, β)-Expanders with "polynomial edge latency" functions:***
Let G be an (α, β)-Expander, let s and t be a source and sink pair such that $deg(t) = \big(1+o(1)\big)deg(s)$, let $l = \{l_e\}_{e\in E(G)}$ be the latency functions for G and let r be a traffic rate. If for any sufficiently small fixed $\delta > 0$, (G, r, l) be a δ-reasonable instance, then with probability at least $1 - e^{-\Omega(\min\kappa, deg(s))}$ Braess's Paradox occurs.

3.4 Proof Idea

The proof of theorems 3.13, 3.14, 3.15 ([20], [21], [22]) is based on the following idea:

Let $G = (V, E)$ be a graph of a single commodity instance (G, r, l).

In order to prove that instance (G, r, l) is not paradox-free, by lemma 3.3 we have to prove that:

$$\beta(G, r, l) > 1$$

In the next few lines, we are going to show that in order to prove the above, it is adequate to show that:

$$r_l^{G'}\big(2B(1-\mu)\big) > \tilde{r}_l^{G}(2B)$$

where $0 < \mu < 1$ and $G' = (V', E')$ is a subgraph of G such that:

$$G' \subset G$$
$$V' = V$$
$$E' \subset E$$

$\tilde{r}_l^G(2B)$ is a way of symbolizing the traffic rate $\tilde{r}$, that induces a total Nash equilibrium latency $L_{eq}(G)$ equal to $2B$.

Indeed, the reason is the following:

Since $r_l^{G'}\big(2B(1-\mu)\big) > \tilde{r}_l^G(2B)$, then there exists a Nash equilibrium latency $\mathcal{L}_o$ of the instance $(G', \tilde{r}, l)$ such that:

$$r_l^{G'}\big(2B(1-\mu)\big) > \tilde{r}_l^{G'}(\mathcal{L}_o) = \tilde{r}_l^G(2B)$$

But, making use of lemma 3.8, since the latency functions are strictly increasing, if $r_l^{G'}\big(2B(1-\mu)\big) > \tilde{r}_l^{G'}(\mathcal{L}_o)$, then $2B(1-\mu) > \mathcal{L}_o$.

Thus, if by $L(G, r, l)$ we symbolize the Nash equilibrium latency $L_{eq}(G)$ of the instance (G, r, l), that is induced by the traffic rate r, then we have the following:

$$L(G, \tilde{r}, l) = L(G, \tilde{r}_l^G(2B), l) = 2B$$
$$L(H', \tilde{r}, l) \leq L(G', \tilde{r}, l) = L(G', \tilde{r}_l^{G'}(\mathcal{L}_o), l) = \mathcal{L}_o$$

where $H' \subseteq G$ such that $\forall H \subseteq G$, we have that $L_{eq}(H') \leq L_{eq}(H)$.

So we conclude that:

$$\begin{aligned}\beta(G, \tilde{r}, l) &= \max_{H \subseteq G} \frac{L(G, \tilde{r}, l)}{L(H, \tilde{r}, l)} \\ &= \frac{L(G, \tilde{r}, l)}{L(H', \tilde{r}, l)} \\ &\geq \frac{2B}{\mathcal{L}_o} > \frac{2B}{2B(1-\mu)} = \frac{1}{1-\mu} > 1\end{aligned}$$

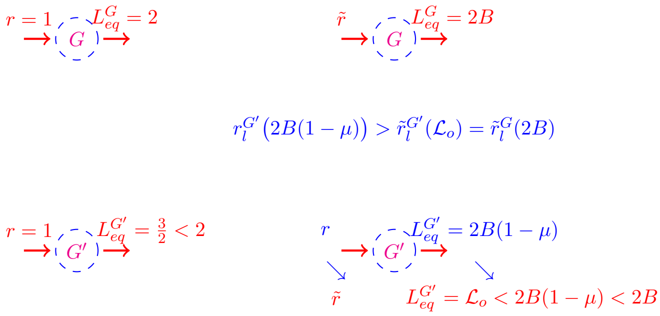
Thus, by lemma 3.3, G is *not paradox-free.*

Fig. 3.5 gives an intuitive approach of the proof idea.

On the left hand side is the original Braess's paradox, and on the right hand side the random network's Braess's paradox.

At the original Braess's paradox, the removal of the link (u, w), as fully described in section 2.1, leads to a subnetwork $G' \subset G$, such that for the same traffic rate the equilibrium latency is decreased from 2 to $\frac{3}{2}$. On the right hand side, we have the same basic idea.

By removing some links, that will be described shortly, we are leaded again to a subnetwork $G' \subset G$, such that for a bigger traffic rate r than the original one $\tilde{r}$, the equilibrium latency is less than $2B$ (equilibrium latency of G for traffic rate $\tilde{r}$), actually $2B(1-\mu)$. So, by lemma 3.8, and since the latency functions are strictly increasing, this is the same as saying that for a traffic rate $\tilde{r}$ less than r, the equilibrium latency is more decreased to $\mathcal{L}_o$ than that of $2B(1-\mu)$. Equivalently, this means that for the same traffic rate $\tilde{r}$ of the original network G, the decrement is bigger, from $2B$ to $\mathcal{L}_o$. Thus, G is not paradox-free.



(a) Original Braess's Paradox (b) Random Network's Braess's Paradox

Figure 3.5: An intuitive approach of the proof idea.

Let's now see what links should be removed in order to obtain the subnetwork $G' \subset G$.

Remember, that in the original Braess's paradox, in section 2.1, besides the (u, w) link that has 0 latency, we have two types of links. The links with latency 1, $(s, w), (u, t)$, and the links with latency x, $(s, u), (w, t)$. This fact guides us to the next definitions for the random graph G:

For the Erdös-Rényi random graphs ([20], [21]), let $I_{\mathcal{A}}$, $I_{\mathcal{B}}$ be defined as in section 3.3 and choose $A_1, A_2 \in I_{\mathcal{A}}$ such that $A_1 < A_2$, $B \in I_{\mathcal{B}}$. Also, fix $0 < \epsilon < 1$ so that $A_1 < (1-\epsilon)A_2$.

An edge e adjacent to s or t, with latency function of the form $l_e(x) = a_e x + b_e$, is called **1-type** if:

- $\diamond$ $a_e \leq A_1$
- $\diamond$ $b_e \in \big(B, (1+\epsilon)B\big)$

An edge e adjacent to s or t, with latency function of the form $l_e(x) = a_e x + b_e$, is called **X-type** if:

- $\diamond$ $a_e \in \big((1-\epsilon)A_2, A_2\big)$
- $\diamond$ $b_e \leq \epsilon B$

For the (α, β)-Expander graphs ([22]), let $(\mathcal{H}, \mathcal{B})$ be Reasonable Distributions, defined as in section 3.3 and fix $0 < \epsilon < 1$, c constant.

An edge e adjacent to s or t, with latency function of the form $l_e(x) = h_e(x) + b_e$, is called **1-type** if:

- $\diamond$ $(1-\epsilon)H_1 \leq h_e \leq H_1$
- $\diamond$ $c \leq b_e \leq (1+\epsilon)c$

An edge e adjacent to s or t, with latency function of the form $l_e(x) = h_e(x) + b_e$, is called **X-type** if:

- $\diamond$ $(1-\epsilon)H_2 \leq h_e \leq H_2$
- $\diamond$ $0 \leq b_e \leq \epsilon c$

Intuitively, 1-type edges have latencies of the form $\eta \cdot x + \theta$, where η is "small" enough while θ is "big" enough so as to resemble the 1 latency edges of the original Braess's paradox. On the contrary, X-type edges have latencies of the form $\eta \cdot x + \theta$, where η is "big" enough while θ is "small" enough so as to resemble the x latency edges of the original Braess's paradox.

What we are going to do, is group all the vertices of G in three groups.

The first group, $G_{1X} = (V_{1X}, E_{1X})$, contains all the 1-type edges adjacent to s and all the X-type edges adjacent to t. Thus, it has the corresponding vertices of these edges, plus the edges and vertices that connect them.

The second group, $G_{X1} = (V_{X1}, E_{X1})$, contains all the X-type edges adjacent to s and all the 1-type edges adjacent to t. Thus, it has the corresponding vertices of these edges, plus the edges and vertices that connect them.

The third group, $G_U = (V_U, E_U)$, contains all the rest edges adjacent to s and t, that do not belong to G_{1X} and to G_{X1}. Again, it has the corresponding vertices of these edges, plus the edges and vertices that connect them.

Now, if the probability of being a 1-type edge, call it p_1, is greater than the probability of being an X-type edge, call it p_X, we will randomly move 1-type edges to G_U so that the *expected* degrees of s and t are the same in G_{1X} and G_{X1}. The opposite process is followed when p_X is greater than p_1. So, it must be clear enough that the following two equations are valid (actually a formal proof will be presented using Chernoff-Hoeffding bounds):

$$\mathbb{E}[\Gamma_{G_{1X}}(s)] \simeq \mathbb{E}[\Gamma_{G_{X1}}(s)] \simeq \mathbb{E}[\Gamma_{G_{1X}}(t)] \simeq \mathbb{E}[\Gamma_{G_{X1}}(t)]$$
$$\mathbb{E}[\Gamma_{G_U}(s)] \simeq \mathbb{E}[\Gamma_{G_U}(t)]$$

Fig. 3.6 describes all the above and two more issues. The first is that for any internal vertices u and v, we have that $|d_s(u) - d_s(v)| \leq \delta$, where δ is a small positive number. This will be noted as the *δ-lemma*. The second is that the latencies of all internal vertices are balanced in such a way that $d_s(v) \lessapprox \frac{d_s(t)}{2}$. This will be noted as the *balance lemma*.

Now we are ready to answer the question given above, which asks what links should be removed in order to obtain the subnetwork $G' \subset G$.

$G' = (V', E')$ has the same vertices with $G = (V, E)$, $V' = V$, but does not have the edges, which link together the groups G_{1X}, G_{X1} and G_U. Thus, $E' \subset E$.

Fig. 3.7 explains the above.

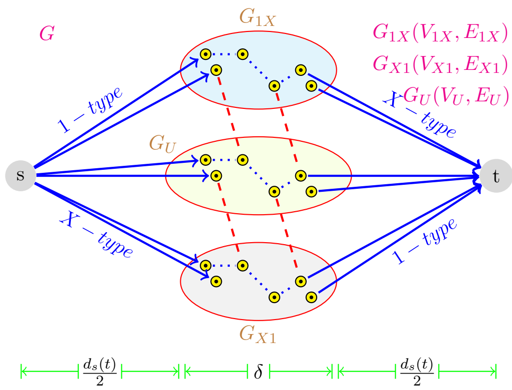


Figure 3.6: 1-type and X-type edges - G_{1X}, G_{X1}, G_U subgraphs - δ and *balance* lemmas.

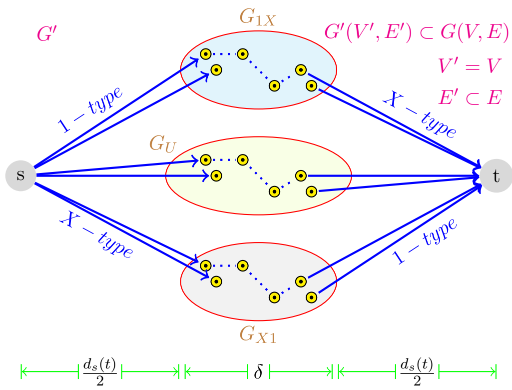


Figure 3.7: Resemblance with the original Braess's paradox - In the original Braess's paradox, the link being removed is the "middle" one (u, w). Here, the links being removed are again the "middle" ones, that link together G_{1X}, G_{X1} and G_U.

The resemblance with the original Braess's paradox is obvious. In the original Braess's paradox, the link being removed is the "middle" one (u, w), while in the random graph, the links being removed are again the "middle" ones, that link together G_{1X}, G_{X1} and G_U.

In the remaining lines of this chapter, we will prove that $r_l^{G'}\big(2B(1-\mu)\big) > \tilde{r}_l^G(2B)$.

3.5 Proof

The section 3.4 gave a brief description of the proof idea that papers [20], [21] and [22] present.

In this chapter we will focus on paper [21], that proves that Braess's paradox occurs in large random sparse graphs. We will try to approach the proofs by giving pictures that make things easier and more attractive. Of course, the proofs of the other two papers are not the same, but they are based on the same ideas, as was dictated earlier in this chapter. Any differences, will be explicitly made in order for the reader to have a good idea of these proofs.

3.5.1 Sizes of G_{1X}, G_{X1} and G_U

Under the same proof ideas described in the previous chapter, the following should be obvious:

Let's define $p^* = \min\{p_1, p_X\}$.

- ⋄ For each $v \in \Gamma(s) \setminus \{t\}$, if the edge (s, v) is 1-type assign v to V_{1X} with probability $\frac{p^*}{p_1}$. If it is X-type assign v to V_{X1} with probability $\frac{p^*}{p_X}$. Otherwise, assign v to V_U.
- ⋄ For each $v \in \Gamma(t) \setminus \{s\}$, if the edge (v, t) is 1-type assign v to V_{X1} with probability $\frac{p^*}{p_1}$. If it is X-type assign v to V_{1X} with probability $\frac{p^*}{p_X}$. Otherwise, assign v to V_U.
- ⋄ For each $v \notin \Gamma(s) \cup \Gamma(t) \setminus \{s, t\}$ assign v uniformly at random to one of V_{1X}, V_{X1} or V_U.

G_{1X}, G_{X1} and G_U are defined according to section 3.4, with the edge (s, t) assigned to G_U, if it exists.

Let's find the expected sizes of $\Gamma_{G_{1X}}(s), \Gamma_{G_{1X}}(t), \Gamma_{G_{X1}}(s), \Gamma_{G_{X1}}(t)$. What is in general the expected size of $\Gamma(v)$, $\mathbb{E}[\Gamma(v)]$, where $v \in V$? By relation (3.4), $\mathbb{E}[\Gamma(v)] = \mathbb{E}[deg(v)] = np$. Thus, by all the above we should have that:

$$\mathbb{E}[\Gamma_{G_{1X}}(s)] \simeq \mathbb{E}[\Gamma_{G_{X1}}(t)] \simeq \frac{p^*}{p_1} \cdot p_1 \cdot (np) = p^* pn$$
$$\mathbb{E}[\Gamma_{G_{X1}}(s)] \simeq \mathbb{E}[\Gamma_{G_{1X}}(t)] \simeq \frac{p^*}{p_X} \cdot p_X \cdot (np) = p^* pn$$

Similarly, the expected sizes of $\Gamma_{G_U}(s), \Gamma_{G_U}(t)$ should be:

$$\mathbb{E}[\Gamma_{G_U}(s)] \simeq \mathbb{E}[\Gamma_{G_U}(t)] \simeq \big(1 - (\frac{p^*}{p_1} \cdot p_1 + \frac{p^*}{p_X} \cdot p_X)\big) \cdot (np) = (1-2p^*)pn$$

Now, by relation (3.6), for $\Gamma_{G_{1X}}(s)$ (the same applies for $\Gamma_{G_{1X}}(t), \Gamma_{G_{X1}}(s), \Gamma_{G_{X1}}(t)$), we have that:

$$\mathbb{P}\big((1-\epsilon)\mathbb{E}[\Gamma_{G_{1X}}(s)] \leq |\Gamma_{G_{1X}}(s)| \leq (1+\epsilon)\mathbb{E}[\Gamma_{G_{1X}}(s)]\big) \geq 1 - e^{-\frac{\epsilon^2 \mathbb{E}[\Gamma_{G_{1X}}(s)]}{3}} \Rightarrow$$
$$\mathbb{P}\big((1-\epsilon)p^* pn \leq |\Gamma_{G_{1X}}(s)| \leq (1+\epsilon)p^* pn\big) \geq 1 - e^{-\frac{\epsilon^2 p^* pn}{3}}$$

Similarly, using relation (3.6) again, then for $\Gamma_{G_U}(s)$ we have that:

$$\mathbb{P}\big((1-\epsilon)(1-2p^*)pn \leq |\Gamma_{G_U}(s)| \leq (1+\epsilon)(1-2p^*)pn\big) \geq 1 - e^{-\frac{\epsilon^2(1-2p^*)pn}{3}}$$

The same applies for $\Gamma_{G_U}(t)$ too.

Now, since $1 - e^{-\frac{\epsilon^2 p^* pn}{3}}$ and $1 - e^{-\frac{\epsilon^2(1-2p^*)pn}{3}}$ are numbers very close to 1 as n grows large enough, this means that with *high probability* we have that:

$$(1-\epsilon)p^* pn \leq |\Gamma_{G_{1X}}(s)| \simeq |\Gamma_{G_{X1}}(s)| \simeq |\Gamma_{G_{1X}}(t)| \simeq |\Gamma_{G_{X1}}(t)| \leq (1+\epsilon)p^* pn \quad (3.9)$$

$$(1-\epsilon)(1-2p^*)pn \leq |\Gamma_{G_U}(s)| \simeq |\Gamma_{G_U}(t)| \leq (1+\epsilon)(1-2p^*)pn \quad (3.10)$$

3.5.1.1 Differences with Large Dense Random Graphs

In [20] the sets G_{1X}, G_{X1} and G_U differ in some way. Specifically, for the vertex sets V_{1X}, V_{X1} and V_U, we have:

$$V_{1X} = S_1 \cup T_2 \cup Q_1$$
$$V_{X1} = S_2 \cup T_1 \cup Q_2$$
$$V_U = U \cup Q_3$$

The sets $S_1, S_2, T_1, T_2, Q_1, Q_2, Q_3, U$ will be described just after we give the following definition of probabilities P_1, P_2.

If v is a vertex such that $v \in V \setminus \{s,t\}$, then this paper defines also the following probabilities:

- P_1 is the probability that edge (s,v) ((v,t) respectively) is 1-type, while edge (v,t) ((s,v) respectively) is either absent or not 1-type. Obviously:

$$P_1 = p \cdot p_1 \cdot (1 - p \cdot p_1)$$

- P_X is the probability that edge (s, v) $\big((v, t)$ respectively $\big)$ is X-type, while edge (v, t) $\big((s, v)$ respectively $\big)$ is either absent or not X-type. Obviously:
$$P_1 = p \cdot p_X \cdot (1 - p \cdot p_X)$$

Let's also define $P^* = \min\{P_1, P_X\}$.

Then, by defining the following, we have analogous behavior of the distributions of all the vertices to the groups $S_1 \cup T_2 \cup Q_1, S_2 \cup T_1 \cup Q_2, U \cup Q_3$, with the [21] distribution of the vertices to the sets V_{1X}, V_{X1} and V_U:

- ⋄ For each $v \in \Gamma(s)$, if the edge (s, v) is 1-type, while edge (v, t) is either absent or not 1-type, assign v to S_1 with probability $\frac{P^*}{P_1}$. If it is X-type, while edge (v, t) is either absent or not X-type, assign v to T_1 with probability $\frac{P^*}{P_X}$. Otherwise, assign v to U.
- ⋄ For each $v \in \Gamma(t)$, if the edge (v, t) is 1-type, while edge (s, v) is either absent or not 1-type, assign v to S_2 with probability $\frac{P^*}{P_1}$. If it is X-type, while edge (s, v) is either absent or not X-type, assign v to T_2 with probability $\frac{P^*}{P_X}$. Otherwise, assign v to U.
- ⋄ For each $v \notin \Gamma(s) \cup \Gamma(t)$ or if (s, v) or (v, t) has its b-coefficient at least $(1 + \epsilon)B$, assign v uniformly at random to one of Q_1, Q_2 or Q_3.
- ⋄ Otherwise, unclassified vertices are assigned to U.

Again, by Chernoff - Hoeffding bounds relation (3.6), we have that:

$$\begin{aligned}(1 - \epsilon)\Omega(pn) \le |S_1| \simeq |S_2| \simeq |T_1| \simeq |T_2| \\ \simeq |Q_1| \simeq |Q_2| \simeq |Q_3| \simeq |U| \le (1 + \epsilon)\Omega(pn)\end{aligned}$$

where $\epsilon = \frac{1}{(pn)^3}$, and $\Omega(pn)$ is the expected size of each of the above groups. Recall by section 3.2 that $\Omega(pn) >> (1 + \epsilon)\ln n$.

3.5.1.2 Differences with Expander Graphs

In [22] the sets G_{1X}, G_{X1} and G_U differ under the following manner.

The only thing that changes, is the way the vertices are distributed in the sets V_{1X}, V_{X1} and V_U.

By defining the following, we have analogous behavior of the distributions of all the vertices to the above groups, with the [21] distribution of the vertices to the corresponding sets V_{1X}, V_{X1} and V_U:

- ⋄ For each $v \in \big(\Gamma(s) \cup \Gamma(t)\big) \setminus \big(\Gamma(s) \cap \Gamma(t)\big)$:
  - If the edge (s, v) is 1-type, assign v to V_{1X} with probability p_X.
  - If the edge (s, v) is X-type, assign v to V_{X1} with probability p_1.

- If the edge (v, t) is 1-type, assign v to V_{X1} with probability p_X.
- If the edge (v, t) is X-type, assign v to V_{1X} with probability p_1.
- Otherwise assign v to V_U with the remaining probability.

◇ For each $v \in \Gamma(s) \cap \Gamma(t)$:

- If the edge (s, v) is 1-type and (v, t) is X-type, assign v to V_{1X}.
- If the edge (s, v) is X-type and (v, t) is 1-type, assign v to V_{X1}.
- Otherwise assign v to V_U with the remaining probability.

◇ The remaining vertices are assigned with probability $p_1 p_X$ to V_{1X}, with the same probability $p_1 p_X$ to V_{X1}, and with probability $1 - 2p_1 p_X$ to V_U.

What are the expected sizes of the sets V_{1X}, V_{X1} and V_U? Apparently they are as follows:

$$\mathbb{E}[|V_{1X}|] = \mathbb{E}[|V_{X1}|] = p_1 p_X n$$
$$\mathbb{E}[|V_U|] = (1 - 2p_1 p_X)n$$

The rest follow easily.

3.5.2 δ-Lemma

Now we are going to prove the δ-lemma, which says that for any internal vertices u and v, then $|d_s(u) - d_s(v)| \leq \delta$, where δ is a small positive number. At the beginning we prove the following lemma:

Lemma 3.16. *For any sufficiently small fixed $\delta > 0$, there are some constants $c > 1$ and $n_0 > 0$ such that, if $n > n_0$, $np > c \ln(n)$ and $\mathbb{P}\big(\mathcal{B} \leq \frac{\delta}{\ln(n)}\big)pn \geq 4$, then for* ***any two flow carrying vertices*** *u, v other than s, t in the instance $(G, \tilde{r}_l^G(2B), l)$, we have $|d_s(u) - d_s(v)| \leq 7\delta$ and $|d_t(u) - d_t(v)| \leq 7\delta$ with high probability.*

Proof. The proof is by induction.

Let v_s, v_t be $flow$ carrying vertices that minimize $d_s(v_s), d_t(v_t)$ respectively.

Since for any flow vertex v we have that $d_s(v) + d_t(v) = 2B$, it is adequate to prove that:

$$d_s(v_t) - d_s(v_s) \leq 7\delta$$

Obviously, the flow carrying edge $e_s = (s, v_s)$ has flow that is given by $f_{e_s} = \frac{l_{e_s} - b_{e_s}}{a_{e_s}} \leq \frac{2B}{A_{\min}}$. Fig. 3.8 describes this fact.

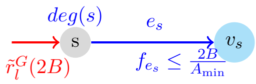


Figure 3.8: The flow of edge $e_s = (s, v_s)$.

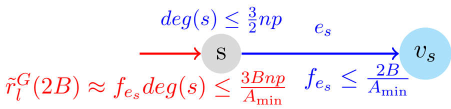


Figure 3.9: A bound of $\tilde{r}_l^G(2B)$.

By Chernoff - Hoeffding bounds relation (3.7), and since by relation (3.4) $\mathbb{E}[deg(s)] = np$ and $np > c\ln(n)$, then for $\epsilon = \frac{1}{2}$ we have that:

$$\mathbb{P}(deg(s) \leq \frac{3}{2}np) \geq 1 - e^{-\frac{\frac{1}{2}^2 np}{3}} > 1 - e^{-\frac{\frac{1}{2}^2 c\ln(n)}{3}}$$

But $1 - e^{-\frac{\frac{1}{2}^2 c\ln(n)}{3}}$ is a number close to 1 as n grows large enough. Thus, with *high probability*, $deg(s) \leq \frac{3}{2}np$. Then, since $\tilde{r}_l^G(2B) \approx f_{e_s} deg(s)$, we have that:

$$\tilde{r}_l^G(2B) \approx f_{e_s} \cdot deg(s) \leq \frac{2B}{A_{\min}} \cdot \frac{3}{2}np = \frac{3Bnp}{A_{\min}}$$

Fig. 3.9 shows the above.

Now, we are going to focus on the neighborhood $\Gamma(v_s)$ of v_s. From this set, there is a subset call it $\Gamma^\delta(v_s)$, where each $e = (v_s, v)$ with latency $a_e \cdot x + b_e$, has $b_e \leq \delta$.

Intuitively, by relation (3.4), $\mathbb{E}[\Gamma^\delta(v_s)] = \mathbb{P}(\mathcal{B} \leq \delta)np$.

Then, doing again the same job as above and since $np > c\ln(n)$, then for $\epsilon = \frac{1}{3}$, relation (3.8) gives that with high probability, $|\Gamma^\delta(v_s)| \geq \frac{2}{3}\mathbb{P}(\mathcal{B} \leq \delta)np$.

Fig. 3.10 shows the above.

Now, since the flow entering v_s is at most $\frac{2B}{A_{\min}}$, then <u>at most</u> half of the $\Gamma^\delta(v_s)$ links should have flow <u>at least</u> $\frac{\frac{2B}{A_{\min}}}{\frac{|\Gamma^\delta(v_s)|}{2}}$. This means that <u>at least</u> half of the $\Gamma^\delta(v_s)$ links should have flow <u>at most</u>:

$$\frac{\frac{2B}{A_{\min}}}{\frac{|\Gamma^\delta(v_s)|}{2}} \leq (\frac{2B}{A_{\min}})\frac{1}{\frac{1}{3}\mathbb{P}(\mathcal{B} \leq \delta)np}$$

Let's call the corresponding vertex set of these links U_0, and let's also pick a random vertex $v \in U_0$. Fig. 3.11 shows what we have just described.

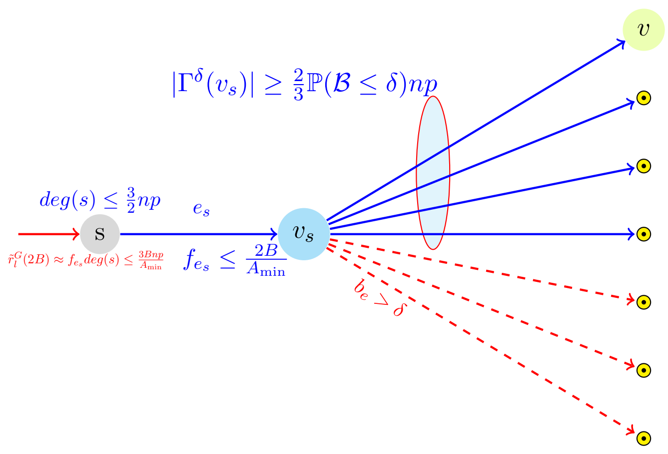


Figure 3.10: The subset of neighbors of v_s, $\Gamma^{\delta}(v_s)$, where each $e = (v_s, v)$ has latency $a_e \cdot x + b_e$, with $b_e \leq \delta$.

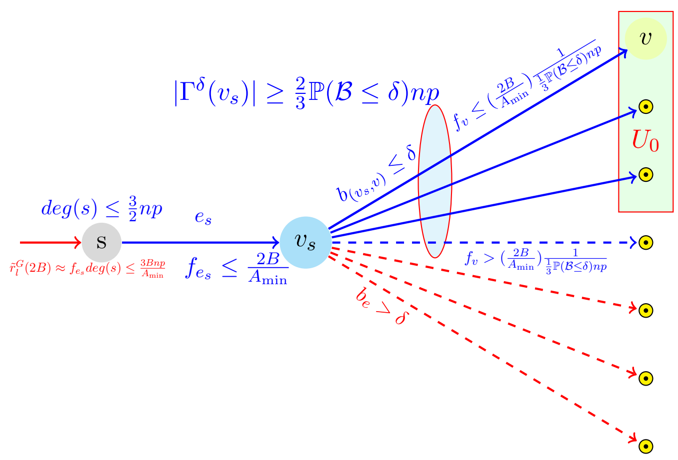


Figure 3.11: The flow of a link $e = (v_s, v)$ with latency $a_e \cdot x + b_e$ and with $b_e \leq \delta$. v belongs to the vertex subset U_0, with incoming flow at most $\left(\frac{2B}{A_{\min}}\right)\frac{1}{\frac{1}{3}\mathbb{P}(\mathcal{B}\leq\delta)np}$.

Let's define c_0 as:

$$c_0 = d_s(v_s) + A_{\max} \frac{6B}{A_{\min}\mathbb{P}(\mathcal{B} \leq \delta)np} + \delta \tag{3.11}$$

Then what is an upper bound of the minimum latency $d_s(v)$ of vertex v? It should be apparent that $d_s(v) \leq c_0$. Also, it should be obvious that $U_0 = \{v | d_s(v) \leq c_0\}$ and that:

$$|U_0| \geq \frac{|\Gamma^{\delta}(v_s)|}{2} = \frac{1}{3}\mathbb{P}(\mathcal{B} \leq \delta)np \tag{3.12}$$

The pair (U_0, c_0) is the base case of the induction.

Our induction hypothesis, will be the pair of the *flow carrying* set of vertices and its corresponding minimum latency, (U_i, c_i). Obviously $U_0 \subseteq U_i$ and $c_0 \leq c_i$.

We are going to prove that $U_i \subset U_{i+1}$ and that $c_i < c_{i+1}$. But this should not happen forever. Remember, by lemma 3.10, that this applies only while $\frac{3}{5}np|U_i| \leq |\Gamma(U_i)| < \frac{3n}{5}$. This sequence has to stop when $|\Gamma(U_i)| \geq \frac{3n}{5}$.

Now, let's call by $\bar{U}_i$ the complement of U_i, and by $U_i' \subseteq \bar{U}_i$ the set of vertices, where each e with tail in U_i and head in U_i' has $b_e \leq \gamma$, $\gamma = \frac{\delta}{\ln(n)}$, where edge's e latency is $a_e \cdot x + b_e$.

What is the expected size of U_i'?

Relation (3.4) gives the expected size of a single vertex neighbourhood, which is equal to np. Now, since U_i' is the neighborhood set of all the vertices belonging to U_i, it should be clear that $\mathbb{E}[U_i'] = \mathbb{P}(\mathcal{B} \leq \gamma)np|U_i|$.

Then, by Chernoff - Hoeffding bounds relation (3.8), and for $\epsilon = \frac{1}{2}$, since $np > c\ln(n)$, we should have that with *high probability*, $|U_i'| \geq \frac{1}{2}\mathbb{P}(\mathcal{B} \leq \gamma)np|U_i|$, as n grows large enough.

Now, since $(U_i, \bar{U}_i)$ is an $s - t$ cut, the flow crossing the cut should be at most $\frac{3Bnp}{A_{\min}}$. Then <u>at most</u> half of the $U_i - U_i'$ links should have flow <u>at least</u> $\frac{\frac{3Bnp}{A_{\min}}}{\frac{|U_i'|}{2}}$. This means that <u>at least</u> half of the $U_i - U_i'$ links should have flow <u>at most</u>:

$$\frac{\frac{3Bnp}{A_{\min}}}{\frac{|U_i'|}{2}} = \frac{6Bnp}{A_{\min}|U_i'|}$$

Let v be one of those vertices, belonging to this kind of $U_i - U_i'$ links, that have flow at most the above mentioned.

Fig. 3.12 shows all the above.

What is the minimum latency of the vertex v? It should be clear that $d_s(v) \leq c_i + A_{\max}\frac{6Bnp}{A_{\min}|U_i'|} + \gamma$.

But $|U_i'| \geq \frac{1}{2}\mathbb{P}(\mathcal{B} \leq \gamma)np|U_i|$. Thus:

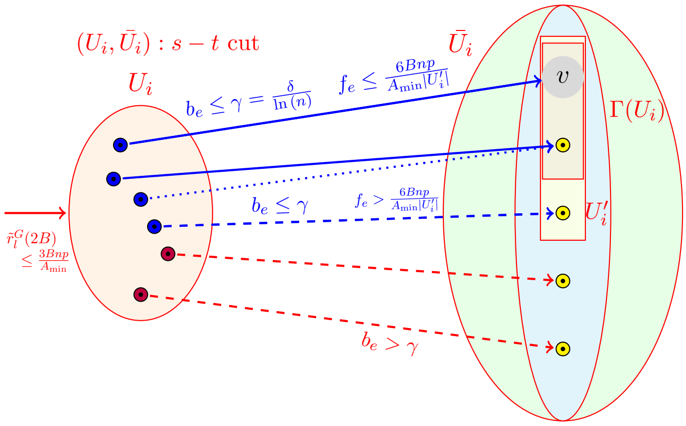


Figure 3.12: $(U_i, \bar{U}_i) : s-t$ cut. The flow crossing the cut should be at most the bound of $\tilde{r}_l^G(2B)$, $\frac{3Bnp}{A_{\min}}$. v belongs to the set U_i', where $b_e \le \gamma$ and the incoming flow is at most $\frac{6Bnp}{A_{\min}|U_i'|}$.

$$d_s(v) \le c_i + \frac{12A_{\max}B}{A_{\min}\mathbb{P}(\mathcal{B} \le \gamma)|U_i|} + \gamma$$

We define (U_{i+1}, c_{i+1}) as follows:

$$c_{i+1} = c_i + \frac{12A_{\max}B}{A_{\min}\mathbb{P}(\mathcal{B} \le \gamma)|U_i|} + \gamma \tag{3.13}$$
$$U_{i+1} = \{v|d_s(v) \le c_{i+1}\}$$

Also, since $\mathbb{P}\big(\mathcal{B} \le \frac{\delta}{\ln(n)}\big)pn \ge 4$, we have that:

$$|U_{i+1}| \ge |U_i'| \ge \frac{1}{2}\mathbb{P}(\mathcal{B} \le \gamma)np|U_i| \ge \big(\frac{1}{4}\mathbb{P}(\mathcal{B} \le \gamma)np + 1\big)|U_i|$$

Thus, doing all the math we have that:

$$|U_{i+1}| \ge \big(\frac{1}{4}\mathbb{P}(\mathcal{B} \le \gamma)np + 1\big)^i |U_0|$$

If i^* is the first i such that $|\Gamma(U_i)| \ge \frac{3n}{5}$ then, since $pn \ge \ln(n)$, we have that:

$$i^* \le \frac{\ln\big(\frac{|U_{i+1}|}{|U_0|}\big)}{\ln\big(\frac{1}{4}\mathbb{P}(\mathcal{B} \le \gamma)np + 1\big)} \le \frac{\ln\big(\frac{\frac{3n}{5}}{|U_0|}\big)}{\ln\big(\frac{1}{4}\mathbb{P}(\mathcal{B} \le \gamma)np + 1\big)}$$

Now, since $\mathbb{P}\big(\mathcal{B} \le \gamma\big)pn \ge c\ln(n)$, then by relation (3.12), it is relatively easy to check that the latter is not bigger than $\ln(n)$. So, we have that:

$$i^* \le \ln(n) \tag{3.14}$$

By relation (3.13), it is easy to conclude that:

$$c_{i^*} \le c_0 + \gamma i^* + \sum_{i=0}^{i^*} \frac{12 A_{\max} B}{A_{\min} \mathbb{P}(\mathcal{B} \le \gamma)|U_i|}$$

Then, by the above relation and relations (3.11), (3.14) we have that:

$$\begin{aligned} c_{i^*} &\le c_0 + \gamma i^* + \sum_{i=0}^{i^*} \frac{12 A_{\max} B}{A_{\min} \mathbb{P}(\mathcal{B} \le \gamma)|U_i|} \\ &\le c_0 + \frac{\delta}{\ln(n)} \ln(n) + \sum_{i=0}^{\ln(n)} \frac{12 A_{\max} B}{A_{\min} \mathbb{P}(\mathcal{B} \le \gamma)|U_i|} \\ &\le \ldots \le d_s(v_s) + 2\delta + \frac{A_{\max} B(6\mathbb{P}(\mathcal{B} \le \gamma) + 72)}{A_{\min} \mathbb{P}(\mathcal{B} \le \gamma)\mathbb{P}(\mathcal{B} \le \delta) np} \end{aligned}$$

So we have proved that:

$$\begin{aligned} c_{i^*} &\le d_s(v_s) + 2\delta + \frac{A_{\max} B(6\mathbb{P}(\mathcal{B} \le \gamma) + 72)}{A_{\min} \mathbb{P}(\mathcal{B} \le \gamma)\mathbb{P}(\mathcal{B} \le \delta) np} \\ U_{i^*} &= \{v | d_s(v) \le c_{i^*}\} \end{aligned} \tag{3.15}$$

Working in a similar way, but now beginning from the sink t, we should eventually come to the following result:

$$\begin{aligned} c'_{j^*} &\ge d_s(v_t) - 2\delta - \frac{A_{\max} B(6\mathbb{P}(\mathcal{B} \le \gamma) + 72)}{A_{\min} \mathbb{P}(\mathcal{B} \le \gamma)\mathbb{P}(\mathcal{B} \le \delta) np} \\ V_{j^*} &= \{v | d_s(v) \ge c'_{j^*}\} \end{aligned} \tag{3.16}$$

Without loss of generality, we may assume that:

$$\begin{aligned} c'_{j^*} - c_{i^*} \ge 0 \\ V_{j^*} \cap U_{i^*} = \varnothing \end{aligned}$$

Now, since $|\Gamma(U_{i^*})| + |\Gamma(V_{j^*})| \ge \frac{6n}{5}$ there are at least $\frac{n}{5}$ edge disjoint paths of length at most 2 between U_{i^*} and V_{j^*}.

Fig. 3.13 gives a good view.

Now, by Chernoff - Hoeffding bounds relation (3.8), and for $\mathbb{E}[Y] = \mathbb{P}(\mathcal{B} \le \delta)^2 \cdot \frac{n}{5}$, $\epsilon = \frac{7}{12}$, then, for sufficiently large n, with high probability there are at least $\frac{1}{12}\mathbb{P}(\mathcal{B} \le \delta)^2 n$ 2-hop paths, call them Y, where all edges e on that path have $b_e \le \delta$.

But since (U_{i^*}, V_{j^*}) is an $s - t$ cut, the flow crossing the cut should be $\tilde{r}_l^G(2B)$. Then at most half of the Y paths should have flow at least $\frac{\tilde{r}_l^G(2B)}{\frac{\frac{1}{12}\mathbb{P}(\mathcal{B} \le \delta)^2 n}{2}}$. This means that at least half of the Y paths, name them X, should have flow at most:

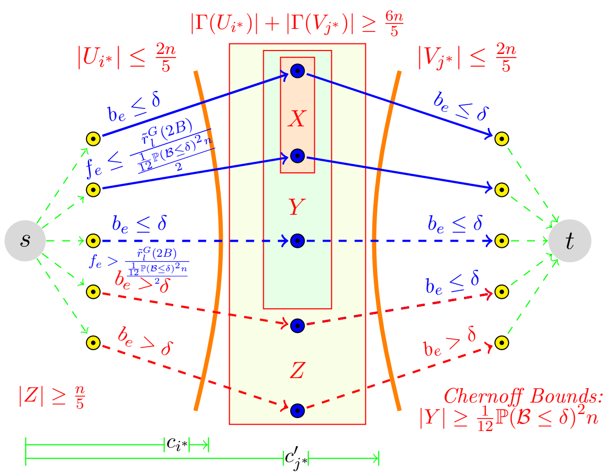


Figure 3.13: The "limit" sets U_{i^*} and V_{j^*}.

$$\frac{\tilde{r}_l^G(2B)}{\frac{\frac{1}{12}\mathbb{P}(\mathcal{B}\leq\delta)^2 n}{2}}$$

Then it should be obvious that:

$$c'_{j^*} - c_{i^*} \leq 2\big(\delta + A_{\max}\frac{\tilde{r}_l^G(2B)}{\frac{\frac{1}{12}\mathbb{P}(\mathcal{B}\leq\delta)^2 n}{2}}\big) \leq 2\delta + \frac{144 A_{\max} Bp}{A_{\min}\mathbb{P}(\mathcal{B} \leq \delta)^2}$$

So, by the above relation and relations (3.15), (3.16) we have that:

$$d_s(v_t) - d_s(v_s) \leq 6\delta + \frac{144 A_{\max} Bp}{A_{\min}\mathbb{P}(\mathcal{B} \leq \delta)^2} + 2\frac{A_{\max} B(6\mathbb{P}(\mathcal{B} \leq \gamma) + 72)}{A_{\min}\mathbb{P}(\mathcal{B} \leq \gamma)\mathbb{P}(\mathcal{B} \leq \delta) np}$$

which for large n, is at most 7δ. □

Completing the proof above, it is time to prove the δ-lemma:

Lemma 3.17. ***δ-lemma:***

For any sufficiently small fixed $\delta > 0$, there are some constants $c > 1$ and $n_0 > 0$ such that, if $n > n_0$, and $\mathbb{P}\big(\mathcal{B} \leq \frac{\delta}{\ln(n)}\big)pn \geq c\ln(n)$, then for ***any two vertices*** *u, v other than s, t in the instance $(G, \tilde{r}_l^G(2B), l)$, we have $|d_s(u) - d_s(v)| \leq 8\delta$ and $|d_t(u) - d_t(v)| \leq 8\delta$ with high probability.*

Proof. In order to prove the lemma, first of all we have to prove the following:

Let X be a connected set of vertices, that is a restriction of $G \setminus \{s, t\}$ to those edges e, where $b_e \leq \frac{\delta}{\ln(n)}$. If c is large enough, then with high probability:

$$|diam(X)| \leq \ln(n) \tag{3.17}$$

where $diam(X)$ is the diameter of X.

The proof is quite easy. What is the expected number of $|diam(X)|$? It is easy to see that:

$$\mathbb{E}[|diam(X)|] = \Big(\mathbb{P}\big(\mathcal{B} \leq \frac{\delta}{\ln(n)}\big)\Big)^{\mathcal{O}(n)} \mathcal{O}(n) \cdot p$$

Let $k = \mathcal{O}(n)$. Actually, it should be that $k \leq n-3$, since X does not contain s, t.

So we have that:

$$\mathbb{E}[|diam(X)|] = \Big(\mathbb{P}\big(\mathcal{B} \leq \frac{\delta}{\ln(n)}\big)\Big)^{k} \cdot kp$$

Then, by Chernoff - Hoeffding bounds relation (3.7), with high probability:

$$|diam(X)| \leq (1+\epsilon)\Big(\mathbb{P}\big(\mathcal{B} \leq \frac{\delta}{\ln(n)}\big)\Big)^{k} \cdot kp$$

where $\epsilon \in (0,1)$, as n grows large enough.

So, it is adequate to prove that:

$$(1+\epsilon)\Big(\mathbb{P}\big(\mathcal{B} \leq \frac{\delta}{\ln(n)}\big)\Big)^{k} \cdot kp \leq \ln(n)$$

which will yield to the result.

Since $\mathbb{P}\big(\mathcal{B} \leq \frac{\delta}{\ln(n)}\big)pn \geq c\ln(n)$, there is a $c' \geq c$ such that $\mathbb{P}\big(\mathcal{B} \leq \frac{\delta}{\ln(n)}\big)pn = c'\ln(n)$.

Combining all together, we have that:

$$c \leq c' \leq \frac{1}{\big(k(1+\epsilon)\big)^{\frac{1}{k}}} \cdot \big(\frac{p}{\ln(n)}\big)^{\frac{k-1}{k}} \cdot n$$

which is large enough, considering the size of n.

Now, let $P \in \mathcal{P}$ be a path from u to v. Suppose also that $d_s(u) \leq d_s(v)$. Then, we have two cases:

- *First Case*: P contains **at most one** flow carrying vertex:

  Then, **no** $e \in P$ carries flow. So, by lemma 2.8 and by relation (3.17)

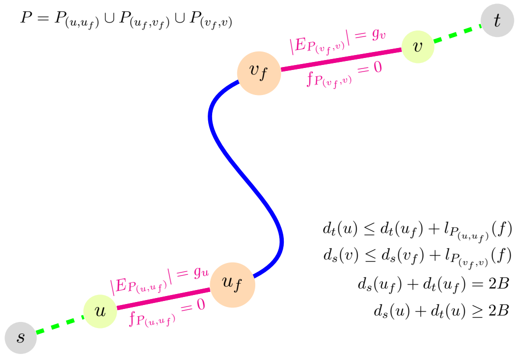


Figure 3.14: δ-lemma, the second case: path P contains **at least two** flow carrying vertices.

we have that:

$$\begin{aligned}
d_s(v) &\leq d_s(u) + l_{P_{(u,v)}}(f) \\
&= d_s(u) + \sum_{e \in P}(a_e \cdot f_e + \frac{\delta}{\ln(n)}) \\
&= d_s(u) + \sum_{e \in P}(a_e \cdot 0 + \frac{\delta}{\ln(n)}) \\
&\leq d_s(u) + \sum_{i=1}^{\ln(n)} \frac{\delta}{\ln(n)} = d_s(u) + \delta
\end{aligned}$$

- *Second Case*: P contains **at least two** flow carrying vertices:

  Let:

  - ◇ u_f be the closest flow carrying vertex to u on P.
  - ◇ v_f be the closest flow carrying vertex to v on P.
  - ◇ g_u be the number of edges between u and u_f on P.
  - ◇ g_v be the number of edges between v and v_f on P.

  Fiq. 3.14 describes the above.

  The following relations are valid:

  - ∘ $d_t(u) \leq d_t(u_f) + l_{P_{(u,u_f)}}(f)$
    It is valid since it is the analogous relation of that of lemma's 2.8.
  - ∘ $d_s(v) \leq d_s(v_f) + l_{P_{(v_f,v)}}(f)$
    It is valid, by lemma 2.8.

- $d_s(u_f) + d_t(u_f) = 2B$
  It is valid, by lemma 3.5.
- $d_s(u) + d_t(u) \geq 2B$
  It is valid, by lemma 3.5 again.

Thus we have:

$$\begin{aligned} d_s(u) &\geq 2B - d_t(u) \\ &\geq 2B - d_t(u_f) - l_{P_{(u,u_f)}}(f) \\ &= d_s(u_f) - \sum_{e \in P_{(u,u_f)}} (a_e \cdot f_e + \frac{\delta}{\ln(n)}) \\ &= d_s(u_f) - \sum_{e \in P_{(u,u_f)}} (a_e \cdot 0 + \frac{\delta}{\ln(n)}) \\ &= d_s(u_f) - \sum_{i=1}^{g_u} \frac{\delta}{\ln(n)} = d_s(u_f) - g_u \frac{\delta}{\ln(n)} \end{aligned}$$

So, by the relation just above, the relation (3.17) and lemma 3.16, then, because $d_s(v) \leq d_s(v_f) + l_{P_{(v_f,v)}}(f)$, we have:

$$\begin{aligned} d_s(v) - d_s(u) &\leq d_s(v_f) + l_{P_{(v_f,v)}}(f) - d_s(u) \\ &= d_s(v_f) - d_s(u) + \sum_{i=1}^{g_v} \frac{\delta}{\ln(n)} \\ &\leq d_s(v_f) - d_s(u_f) + (g_u + g_v) \frac{\delta}{\ln(n)} \\ &\leq 7\delta + \delta = 8\delta \end{aligned}$$

□

3.5.2.1 Differences with Large Dense Random Graphs

Recall by section 3.2 that $\Omega(pn) >> (1 + \epsilon) \ln n$, where $\Omega(pn)$ is the expected size of the sets $S_1, S_2, T_1, T_2, Q_1, Q_2, Q_3, U$. Generally, as relation (3.4) dictates, $\Omega(pn)$ is the expected size of $deg(v)$, for every $v \in V$.

As a consequence of the above is that the graph G should be connected in such a degree, that if we choose two random vertices u, v of the graph, then with high possibility, there exists a 2-hop path $P : u \to w \to v$ connecting the two vertices u and v, where w is a vertex belonging in Q_1, Q_2 or Q_3. Thus, by Chernoff - Hoeffding bounds, we can safely assume that the b-coefficients of both links (u, w) and (w, v) are at most γ, for $\gamma > 0$ small enough. This is property (P4) of [20].

The δ-lemma of [20], says that for both instances $(G, \tilde{r}_l^G(2B), l)$ and $(G', r_l^{G'}(2B(1-\mu)), l)$, we have that $|d_s(u) - d_s(v)| \leq 2\max\{\gamma, \delta\} = 2\delta$, where $\gamma < \delta$, with high probability (actually, in [20], Valiant and Roughgarden prove it only for the instance $(G', r_l^{G'}(2B(1-\mu)), l)$, since this is the only one necessary for the proof of theorem 3.13).

Valiant and Roughgarden prove at first that $|d_s(u) - d_s(v)| \leq \delta$, where u and v are flow carrying vertices. At the beginning they sort the flow carrying vertices in nondecreasing order of d-values, $s = v_1, v_2, \ldots, v_k = t$. The aim is to prove that $d_s(v_{k-1}) - d_s(v_2) \leq \delta$.

The proof uses property (P4), by assuming that the number of the 2-hop paths is κ. Consequently, strictly more than half of them carry at most $\frac{5B}{\kappa A_{\min}}$ units of flow. This means that $d_s(v_{k-1}) - d_s(v_2) \leq 2(A_{\max}\frac{5B}{\kappa A_{\min}}) + \gamma$. But by Chernoff - Hoeffding bounds and property (P4), $\kappa = \Omega(n^{2\zeta})$.

So, for $\gamma < \frac{\delta}{3}$ and n larger than a constant n_0, the right hand side is at most δ. This proof is the same for both instances.

Now, for any kind of vertices u and v, and this is the δ-lemma, $|d_s(u) - d_s(v)| \leq 2\delta$, for both instances $(G, \tilde{r}_l^G(2B), l)$ and $(G', r_l^{G'}(2B(1-\mu)), l)$.

Again, by property (P4) and Chernoff - Hoeffding bounds, any two random vertices in $S_1 \cup T_2 \cup Q_1$, $S_2 \cup T_1 \cup Q_2$ or $U \cup Q_3$ are connected by a 2-hop path with b-coefficients of both links at most γ. Now, since every link is bounded by δ, if the vertices are flow carrying, or by $\gamma < \delta$ in the opposite case, the proof of the lemma follows.

3.5.2.2 Differences with Expander Graphs

The proof of the δ-lemma is analogous to the [21] proof described above.

Chung, Young and Zhao prove first the lemma for two random flow carrying vertices u and v. They show that for the instance (G, r, l), we have that $|d_s(u) - d_s(v)| \leq 7\delta$ or equivalently $|d_t(u) - d_t(v)| \leq 7\delta$. The proof follows the same lines with lemma 3.16, but instead of using lemma 3.10 for the graph expansion it uses the expansion factor of section 3.2.

Now, if v' is a random vertex non flow carrying, then by using again the same proof lines as with lemma 3.16, they prove that $|d_s(v) - d_s(v')| \leq 5\delta$ or equivalently $|d_t(v) - d_t(v')| \leq 5\delta$, where v is flow carrying and is the "nearest", in terms of the latency, vertex to v'.

As a result they prove easily that with high probability, $|d_s(u') - d_s(v')| \leq 12\delta$ or $|d_t(u') - d_t(v')| \leq 12\delta$, where u', v' are random non flow carrying vertices.

3.5.3 Balance Lemma

Now, we are going to prove the balance lemma, which says that the latencies of all internal vertices are balanced in such a way that $d_s(v) \lessapprox \frac{d_s(t)}{2}$.

Lemma 3.18. ***Balance lemma:***

For any sufficiently small fixed $\delta > 0$, there are some constants $c > 1$ and $n_0 > 0$ such that, if $n > n_0$, and $\mathbb{P}\big(\mathcal{B} \leq \frac{\delta}{\ln(n)}\big)pn \geq c\ln(n)$, then for any vertex v other than s, t in the instance $(G, \tilde{r}_l^G(2B), l)$, we have $d_s(v) \leq B + 10\delta$ with high probability.

Proof. The proof is by contradiction.

Suppose that there exists a vertex v such that $d_s(v) > B + 10\delta$.

Then for every vertex u we have that $d_s(u) > B + 2\delta$. Indeed, this is true. We have the following two cases:

- *First Case*: $d_s(u) \leq d_s(v)$:

  Then by the delta lemma we have that:

  $$d_s(u) \geq d_s(v) - 8\delta > B + 10\delta - 8\delta = B + 2\delta$$

- *Second Case*: $d_s(u) \geq d_s(v)$:

  Then we have that:

  $$d_s(u) \geq d_s(v) > B + 10\delta > B + 2\delta$$

So, $d_s(u) > B + 2\delta$ is true under our assumption.

Now, for every flow carrying vertex w, and since $d_s(w) > B + 2\delta$, then by lemma 3.5 we have that:

$$d_t(w) = 2B - d_s(w) < 2B - B - 2\delta = B - 2\delta$$

Let $e = (s, v)$ be an edge adjacent to the sink s. Then $a_e f_e + b_e = d_s(v) > B + 2\delta$. Now, if $b_e \leq B + 2\delta$, then it should be that:

$$f_e > \frac{B + 2\delta - b_e}{a_e} > 0$$

Let E_s be the set of edges $E_s \subseteq E$, such that $E_s = \{e \in E | \exists u : e = (s, u) \wedge b_e \leq B + 2\delta\}$.

By Chernoff bounds relation (3.8), and by relation (3.4), then $|E_s| \geq (1 - \epsilon')\mathbb{P}\big(\mathcal{B} \leq B + 2\delta\big)pn$ with high probability, as n grows large enough.

Now, by the Law of Large Numbers lemma 3.12, we have that:

$$\frac{\sum_{e \in E_s} b_e}{|E_s|} \simeq \mathbb{E}[\mathcal{B}_{[0, B+2\delta]}]$$
$$\frac{\sum_{e \in E_s} \frac{1}{a_e}}{|E_s|} \simeq \mathbb{E}[\frac{1}{\mathcal{A}}]$$

Then, by the above two relations, we have:

$$\begin{aligned}\tilde{r}_l^G(2B) &\geq \sum_{e\in E_s} \frac{B+2\delta-b_e}{a_e} \\ &= (B+2\delta-\mathbb{E}[\mathcal{B}_{[0,B+2\delta]}]-o(1))\cdot \\ &\Big(\mathbb{E}[\frac{1}{\mathcal{A}}]+o(1)\Big)|E_s|\end{aligned}$$

So, the above in conjunction with $|E_s| \geq (1-\epsilon')\mathbb{P}\big(\mathcal{B} \leq B+2\delta\big)pn$, gives:

$$\tilde{r}_l^G(2B) \geq (1-\epsilon')\frac{B+2\delta-\mathbb{E}[\mathcal{B}_{[0,B+2\delta]}]-o(1)}{\mathbb{E}[\mathcal{A}]+o(1)} \cdot \mathbb{P}\big(\mathcal{B} \leq B+2\delta\big)pn \quad (3.18)$$

Working in a similar way with the edges adjacent to the tank t we have that:

$$\tilde{r}_l^G(2B) \leq (1+\epsilon')\frac{B-2\delta-\mathbb{E}[\mathcal{B}_{[0,B-2\delta]}]+o(1)}{\mathbb{E}[\mathcal{A}]-o(1)} \cdot \mathbb{P}\big(\mathcal{B} \leq B-2\delta\big)pn \quad (3.19)$$

Then, by the above relations (3.18) and (3.19) we have that:

$$\frac{\tilde{r}_l^G(2B)}{\tilde{r}_l^G(2B)} \geq \frac{(1-\epsilon')\frac{B+2\delta-\mathbb{E}[\mathcal{B}_{[0,B+2\delta]}]-o(1)}{\mathbb{E}[\mathcal{A}]+o(1)}\mathbb{P}\big(\mathcal{B} \leq B+2\delta\big)pn}{(1+\epsilon')\frac{B-2\delta-\mathbb{E}[\mathcal{B}_{[0,B-2\delta]}]+o(1)}{\mathbb{E}[\mathcal{A}]-o(1)}\mathbb{P}\big(\mathcal{B} \leq B-2\delta\big)pn}$$

It is not hard to see that by the choice of B and δ and the reasonableness of $\mathcal{B}$, the latter relation is greater than 1. So we have proven that $\tilde{r}_l^G(2B) > \tilde{r}_l^G(2B)$, which is a contradiction. □

3.5.3.1 Differences with Large Dense Random Graphs

The balance lemma says that for the instance $(G, \tilde{r}_l^G(2B), l)$ and for a flow carrying vertex v, $d_s(v) \leq B+2\delta$, whereas for the instance $(G', r_l^{G'}(2B(1-\mu)), l)$ and for every vertex $v \in U$, $d_s(v) \geq B(1-\mu)-4\delta$.

The proof is in a way similar with the above one of the sparse graphs.

Again, commenting for the instance $(G, \tilde{r}_l^G(2B), l)$, the proof is by contradiction. It uses again the δ-lemma and gives bounds for the flows of the vertices $v \in \Gamma(s)$ and $u \in \Gamma(t)$. Thus, it gives a lower bound for the factor $\frac{f_{(s,v)}}{f_{(u,t)}}$ that is a number $c > 1$.

By making use of the Chernoff - Hoeffding bounds, and by summing over all flows for the neighbors of s and t respectively, Valiant and Roughgarden prove that the total flow exiting the source s is at least a $\frac{c}{(1+(pn)^{-\frac{1}{3}})^2}$ factor times that entering the sink t. The only way that this could be valid, since the above total flows must be equal to the total traffic rate, is that the number of the vertices, n, should be bounded above by a constant, which is a contradiction.

The same idea stands for the instance $(G', r_l^{G'}(2B(1-\mu)), l)$ too, but limited to the subgraph $G' = U \cup Q_3 \cup \{s, t\}$. The proof is by contradiction for a flow carrying vertex $v \in U$ under similar lines, and then it uses property (P4) to prove the lemma for an arbitrary vertex in U.

3.5.3.2 Differences with Expander Graphs

For the Expander graphs, the balance lemma says that for the instance $(G, \tilde{r}_l^G(2B), l)$ and for any vertex v, $d_s(v) \leq c_s + 13\delta$, where $c_s \in [0, 2B]$. Also, for the instance $(G', r_l^{G'}(2B(1-\mu)), l)$ and for every vertex $v \in U$, $d_s(v) \geq c'_s - 13\delta$, where $c'_s \in [0, 2B(1-\mu)]$. c_s, c'_s are unique values that depend on the reasonable distributions $(\mathcal{H}, \mathcal{B})$, the degrees of s and t, and B.

The proof is in a way similar with the above one of the sparse graphs.

Again, the proof is by contradiction. The details of the proof are somehow technical, but the result is the same. The traffic rate that leaves sink s is bigger than that of the tank t, by a factor $\Omega\big(deg(s) + deg(t)\big)$.

3.5.4 Theorem's Main Proof

Working in a similar manner as with the lemmas 3.17 and 3.18, δ-lemma and balance lemma respectively, the following lemma holds for the instance $(G', r_l^{G'}(2B(1-\mu)), l)$:

Lemma 3.19. *For any sufficiently small fixed $\delta > 0$, there are some constants $c > 1$ and $n_0 > 0$ such that, if $n > n_0$, and $\mathbb{P}\big(\mathcal{B} \leq \frac{\delta}{\ln(n)}\big)pn \geq c\ln(n)$, and G_{1X}, G_{X1}, G_U are defined as above, then the instance $(G', r_l^{G'}(2B(1-\mu)), l)$ satisfies that:*

- *(δ-lemma) For any vertices u, v other than s, t both in one of G_{1X}, G_{X1} or G_U, $|d_s(v) - d_s(u)| \leq 8\delta$ with high probability.*
- *(Balance lemma) For any vertex v other than s, t in G_U, we have $B - 10\delta \leq d_s(v)$ with high probability.*

We are now ready to move on to the main proof of the theorem.

Proof. Remember that in section 3.4, we emphasized that the only thing that we have to prove is that $r_l^{G'}\big(2B(1-\mu)\big) > \tilde{r}_l^G(2B)$.

So, let f, f' be the equilibrium flows of G, G' respectively.

Suppose that e is adjacent to s in G_U. Then by the balance lemmas for G and G', lemmas 3.18 and 3.19 respectively, we have that:

$$a_e f_e + b_e \leq B + 10\delta$$
$$a_e f'_e + b_e \geq B - 10\delta$$

So:

$$f_e - f'_e \leq \frac{B + 10\delta - b_e}{a_e} - \frac{B - 10\delta - b_e}{a_e} = \frac{20\delta}{a_e} \leq \frac{20\delta}{A_{\min}}$$

Then by relation (3.10) we have that:

$$\tilde{r}_l^{G_U}(2B) - r_l^{G_U}(2B(1-\mu)) \leq (1+\epsilon^*)(1-2p^*)pn\frac{20\delta}{A_{\min}} \tag{3.20}$$

Now, let e_s be adjacent to s, e_t be adjacent to t in G_{1X}. Then by the lemmas (3.5) and (3.19) (δ-lemma) for the instance $(G', r_l^{G'}(2B(1-\mu)), l)$ we have:

$$\begin{aligned} 2B(1-\mu) &\leq a_{e_s} f'_{e_s} + b_{e_s} + 8\delta + a_{e_t} f'_{e_t} + b_{e_t} \\ &\leq A_1 f'_{e_s} + (1+\epsilon)B + 8\delta + A_2 f'_{e_t} + \epsilon B \end{aligned}$$

Thus,

$$B - 2\mu B - 2\epsilon B - 8\delta \leq A_1 f'_{e_s} + A_2 f'_{e_t}$$

Then by relation (3.9), and by summing over all choices of e_s, e_t we have:

$$\begin{aligned} r_l^{G_{1X}}\big(2B(1-\mu)\big) &\geq |\Gamma_{G_{1X}}(s)||\Gamma_{G_{1X}}(t)| \cdot \frac{(B - 2\mu B - 2\epsilon B - 8\delta)}{A_1|\Gamma_{G_{1X}}(t)| + A_2|\Gamma_{G_{1X}}(s)|} \\ &\geq \frac{(1-\epsilon^*)^2}{1+\epsilon^*} \cdot \frac{B - 2\mu B - 2\epsilon B - 8\delta}{A_1 + A_2} p^* pn \end{aligned}$$

Similarly, for G_{X1} we have the same:

$$r_l^{G_{X1}}\big(2B(1-\mu)\big) \geq \frac{(1-\epsilon^*)^2}{1+\epsilon^*} \cdot \frac{B - 2\mu B - 2\epsilon B - 8\delta}{A_1 + A_2} p^* pn$$

Let's consider now instance $(G, \tilde{r}_l^G(2B), l)$. By relation (3.9) and by lemma 3.18 (balance lemma) we have for G_{1X}:

$$\tilde{r}_l^{G_{1X}}(2B) \leq (1+\epsilon^*)pp^*n\frac{10\delta}{A_{\min}}$$

Similarly, for G_{X1} we have respectively:

$$\tilde{r}_l^{G_{X1}}(2B) \leq (1+\epsilon^*)pp^*n\frac{B + 10\delta}{(1-\epsilon)A_2}$$

By the above we have:

$$\begin{aligned} & r_l^{G_{1X} \cup G_{X1}}\big(2B(1-\mu)\big) - \tilde{r}_l^{G_{1X} \cup G_{X1}}(2B) \geq \\ & \geq (2\frac{(1-\epsilon^*)^2}{1+\epsilon^*} \cdot \frac{B - 2\mu B - 2\epsilon B - 8\delta}{A_1 + A_2} \\ & -(1+\epsilon^*)(\frac{10\delta}{A_{\min}} + \frac{B + 10\delta}{(1-\epsilon)A_2}))pp^*n \end{aligned} \tag{3.21}$$

So, by relations (3.20) and (3.21) we have that:

$$r_l^{G'}\big(2B(1-\mu)\big) - \tilde{r}_l^G(2B) = [r_l^{G_{1X}\cup G_{X1}}\big(2B(1-\mu)\big) - \tilde{r}_l^{G_{1X}\cup G_{X1}}(2B)] + [r_l^{G_U}\big(2B(1-\mu)\big) - \tilde{r}_l^{G_U}(2B)] \geq$$

$$\geq [(2\frac{(1-\epsilon^*)^2}{1+\epsilon^*}\cdot\frac{B-2\mu B-2\epsilon B-8\delta}{A_1+A_2} - (1+\epsilon^*)(\frac{10\delta}{A_{\min}}+\frac{B+10\delta}{(1-\epsilon)A_2}))pp^*n] - [(1+\epsilon^*)(1-2p^*)pn\frac{20\delta}{A_{\min}}]$$

Now, by letting $\epsilon, \epsilon^*, \delta, \mu \to 0$, the relation above gives:

$$r_l^{G'}\big(2B(1-\mu)\big) - \tilde{r}_l^G(2B) \geq (\frac{2B}{A_1+A_2} - \frac{B}{A_2})pp^*n > 0$$

Thus, $r_l^{G'}\big(2B(1-\mu)\big) > \tilde{r}_l^G(2B)$.

□

3.5.4.1 Differences with Large Dense Random Graphs

The resemblance of the theorem's 3.13 and 3.14 (section 3.5.4) main proofs is striking. It is sufficient for the reader to replace the above relations with the applicable ones for the dense random graphs, having in mind the outcomes of sections 3.5.1.1, 3.5.2.1 and 3.5.3.1.

3.5.4.2 Differences with Expander Graphs

The same things apply for the Expander graphs too. The main proofs of the theorems 3.15 and 3.14 (section 3.5.4) are again too close, but with different content. The reader is advised to visit [22].

Chapter 4

Best Subnetwork Approximation in Random Graphs

We have already seen how difficult it is to detect the best subnetwork. So, it should be always worthy to try to give an *approximation* of the best subnetwork, given the difficulty of detecting it.

We begin this chapter by proving that for networks with strictly increasing linear latencies, there is an upper bound of $\frac{4}{3}$ for the approximation ratio of the problem of finding the best subnetwork.

We end up, by proving two theorems that are an improvement of theorem 4 of [16], and provide approximating methods to random instances with polynomially many paths, each of polylogarithmic length, and linear latencies. By these theorems we may achieve quasipolynomial running times, for traffic rates of the size $\mathcal{O}(1)$ (or more generally $\mathcal{O}(\text{poly}(\ln\ln m))$) or even for traffic rates up to $\mathcal{O}(\text{poly}(\ln m))$, where m is the total number of the network's edges.

The following theorem is due to a joint work of Fotakis, Kaporis and Spirakis [16], [2].

Theorem 4.1. *For instances with strictly increasing linear latency functions, the problem of finding the best subnetwork can be approximated in polynomial time within a factor of* $(\frac{4}{3} - \delta)$*, where* $\delta > 0$ *depends on the instance itself.*

Proof. Wlog we may assume that $r = 1$. We have to distinguish the following two cases:

- ⋄ Paradox-ridden networks:

In theorem 2.13 we have proved that paradox ridden instances can be detected in polynomial time. Thus, for this case, the approximation ratio is 1.

◇ Networks not paradox-ridden:

Let G be the corresponding network. In section 1.6 we have shown that PoA_G is upper bounded by $\frac{4}{3}$.

Let f be a Nash flow on H^B, and o be an optimal flow of G. Since the instance is not paradox-ridden, $f \neq o$. By Taylor expansion for quadratic functions we have that:

$$\begin{aligned} C(f) &= \sum_{e\in E} l_e(f_e)f_e = \sum_{e\in E}(a_e f_e^2 + b_e f_e) \\ &= \sum_{e\in E}(a_e o_e^2 + b_e o_e) + \sum_{e\in E}(2a_e o_e + b_e)(f_e - o_e) + \sum_{e\in E} a_e(f_e - o_e)^2 \\ &= C(o) + \sigma \end{aligned}$$

where $\sigma = \sum_{e\in E}(2a_e o_e + b_e)(f_e - o_e) + \sum_{e\in E} a_e(f_e - o_e)^2$.

Apparently, $\sigma > 0$, since by relation (1.7), letting $g_e = f_e$ and $l_e^*(x) = \frac{d(x\cdot l_e(x))}{dx} = 2a_e x + b_e$, we have easily that $\sum_{e\in E}(2a_e o_e + b_e)(f_e - o_e) \geq 0$. Now, since $f \neq o$, there exists at least one $e \in E$ such that $f_e \neq o_e$. So, $\sum_{e\in E}(2a_e o_e + b_e)(f_e - o_e) > 0$.

Thus, by relations (1.4), (1.5) we have that:

$$\begin{aligned} L_{eq}(H^B) &= L_{opt}(G) + \sigma \Rightarrow \\ 1 &= \frac{L_{opt}(G)}{L_{eq}(H^B)} + \frac{\sigma}{L_{eq}(H^B)} \Rightarrow \\ \frac{L_{opt}(G)}{L_{eq}(H^B)} &= 1 - \frac{\sigma}{L_{eq}(H^B)} \end{aligned}$$

But, by relation (1.8) we have that:

$$L_{opt}(G) = \frac{L_{eq}(G)}{PoA_G}$$

So,

$$\begin{aligned} \frac{L_{eq}(G)}{L_{eq}(H^B)} \cdot \frac{1}{PoA_G} &= 1 - \frac{\sigma}{L_{eq}(H^B)} \Rightarrow \\ \frac{L_{eq}(G)}{L_{eq}(H^B)} &= PoA_G - PoA_G \cdot \frac{\sigma}{L_{eq}(H^B)} \end{aligned}$$

If we set $\delta = PoA_G \cdot \frac{\sigma}{L_{eq}(H^B)} > 0$, then the trivial algorithm that returns the whole network G is a $(\frac{4}{3} - \delta)$ - approximation algorithm for the problem of finding the best subnetwork H^B.

□

Before moving on, we should mention the following useful lemma, that is another way to define the Chernoff-Hoeffding bound described in lemma 3.11:

Lemma 4.2. *Chernoff-Hoeffding Bound: Let X_i, $i \in \{1, 2, \dots, k\}$, be a collection of random variables independently distributed in $[0, 1]$. Define $X = \frac{1}{k}\sum_{i=1}^{k} X_i$. Then, for all $\varepsilon > 0$:*

$$\mathbb{P}[|X - \mathbb{E}[X]| > \varepsilon] \leq 2e^{-2\varepsilon^2 k}$$

The next theorem is also a joint work of Fotakis, Kaporis and Spirakis [16], [2].

Given a network $G = (V, E)$, this lemma declares that any flow can be approximated by a "sparse" flow that traverses a number of paths, at most logarithmic in m, where $m = |E|$.

Specifically, let $(G, 1, l)$ be an instance, with linear latencies $l_e(x) = a_e \cdot x + b_e$ and rational coefficients $a_e, b_e \geq 0$. Let also $\alpha = \max_{e \in E} \{a_e\}$.

Let also $\mu = |\mathcal{P}| \leq m^{d_1}$ be the total number of the original network's paths, where $m = |E|$. Let also $|P| \leq \ln^{d_2} m$ for all $P \in \mathcal{P}$.

The proof is based on Althöfer's specification lemma [18].

Lemma 4.3. *Let $(G, 1, l)$ be an instance defined on a network $G = (V, E)$, and let f be any feasible flow. Then, for any $\epsilon > 0$, there exists a feasible flow $\tilde{f}$ that assigns traffic to at most $\left\lceil \frac{\ln(2m)}{2\epsilon^2} \right\rceil$ paths, such that $\left|\tilde{f}_e - f_e\right| \leq \epsilon$, for all edges e.*

Proof. First of all, the traffic rate r is normalized to 1. Then, every kind of a feasible flow, can be considered as a probability distribution among the paths that the flow selects to traverse.

Let $\mu = |\mathcal{P}|$ denote the total number of paths of the network. Then, every $s - t$ path can be identified by a number $j \in \{1, 2, \dots, \mu\}$.

Thus, if j is one of the μ $s - t$ paths, then, since $r = 1$, f_j can be viewed as the probability that j path is selected.

We will prove that selecting a specific number of κ paths, out of the total μ, uniformly at random with replacement according to the *unknown* (probability distribution) f, and assign to each one of the μ paths a flow $\tilde{f}_j$ equal to the number of times path j is selected divided by κ, we obtain a flow $\tilde{f}$ which is an ϵ - approximation to f with positive probability.

The number κ depends on a suitable logarithmic size of m. Also observe that nor f neither $\tilde{f}$ need to be known. In fact, it is necessary that distribution $\tilde{f}$ should be appropriately 'tuned' as to imply that $\mathbb{P}\left(\left|\tilde{f}_e - f_e\right| \leq \epsilon\right) > 0$. This probability is known as "the probability of a good event".

Having in mind all the above, we will give the proof of this lemma under the following approach:

- The number κ is defined as $\kappa := \left\lceil \frac{\ln(2m)}{2\epsilon^2} \right\rceil$.
- We define κ independent identically distributed random variables P_1, $P_2, \ldots, P_\kappa$ each taking an integer value in $\{1, 2, \ldots, \mu\}$, with repetition, according to f. Thus, $\mathbb{P}[P_i = j] = f_j$, for all $i \in \{1, 2, \ldots, \kappa\}$ and $j \in \{1, 2, \ldots, \mu\}$.
- Let F_j be a random variable defined as $F_j = \frac{|\{i \in \{1,2,\ldots,\kappa\} : P_i = j\}|}{\kappa}$.
- By linearity of expectation, $\mathbb{E}[F_j] = f_j$. Indeed, let's define $X_{i,j}$ as follows:

$$X_{i,j} = \begin{cases} 1 & \text{if } P_i = j, \\ 0 & \text{otherwise.} \end{cases}$$

  Then we have that:

$$\mathbb{E}[X_{i,j}] = 1 \cdot \mathbb{P}[X_{i,j} = 1] + 0 \cdot \mathbb{P}[X_{i,j} = 0] = \mathbb{P}[P_i = j] = f_j$$

  So,

$$\mathbb{E}[F_j] = \mathbb{E}[\frac{1}{\kappa} \sum_{i=1}^{\kappa} X_{i,j}] = \frac{1}{\kappa} \sum_{i=1}^{\kappa} \mathbb{E}[X_{i,j}] = \frac{1}{\kappa} \cdot \kappa \cdot f_j = f_j$$

- For each $e \in E$ and each P_i, $i \in \{1, 2, \ldots, \kappa\}$ we define an indicator variable $F_{e,i}$ that is:

$$F_{e,i} = \begin{cases} 1 & \text{if } e \in P_i, \\ 0 & \text{otherwise.} \end{cases}$$

- The random variables P_i are independent, for every edge e. Thus, $F_{e,i}$, $i \in \{1, 2, \ldots, \kappa\}$ are independent as well.
- Let F_e be a random variable defined as $F_e = \frac{1}{\kappa} \sum_{i=1}^{\kappa} F_{e,i}$. Then, by the definition of the random variables F_j we have that:

$$F_e = \sum_{j : e \in j} F_j = \frac{1}{\kappa} \sum_{i=1}^{\kappa} F_{e,i}$$

- Again, by linearity of expectation, $\mathbb{E}[F_e] = f_e$, for all edges e.

- Each F_j, F_e can be interpreted as the amount of flow traversing each path j, edge e respectively.

- Also, since $\sum_{j=1}^{\mu} F_j = 1$, then the random variables $F_1, F_2, \ldots, F_\mu$ define a feasible flow that assigns positive traffic to *at most* κ paths and agrees with f in expectation.

- By applying lemma 4.2's Chernoff-Hoeffding Bound, then by all the above, we obtain that for every edge e:

$$\begin{aligned}\mathbb{P}[|F_e - f_e| > \epsilon] &= \mathbb{P}[|F_e - \mathbb{E}[F_e]| > \epsilon] \\ &\leq 2e^{-2\epsilon^2\kappa} \\ &< 2e^{-2\epsilon^2[\frac{\ln(2m)}{2\epsilon^2}]} \\ &< \frac{1}{m}\end{aligned}$$

- By applying the union bound, we have that $\mathbb{P}[\exists e \in E : |F_e - f_e| > \epsilon] < m \cdot \frac{1}{m} = 1$.

- Thus, for any $\epsilon > 0$, there exists a feasible flow $\tilde{f} = F$ that assigns traffic to at most $\kappa = \left\lceil \frac{\ln(2m)}{2\epsilon^2} \right\rceil$ paths, such that with positive probability $\left|\tilde{f}_e - f_e\right| \leq \epsilon$, for all edges $e \in E$.

□

Lemma 4.3 guarantees that there exists an ϵ - approximation flow, $\tilde{f}$, to the best subnetwork's Nash flow f, that assigns traffic to at most $\left\lceil \frac{\ln(2m)}{2\epsilon^2} \right\rceil$ paths.

If at first the number of paths, and secondly all paths' latencies, are polynomial in size, then by exhaustive search, we can find this specific flow $\tilde{f}$ that is an $\frac{\epsilon}{2}$ - Nash flow on the best subnetwork H^B, with latency at most $L_{eq}(H^B) + \frac{\epsilon}{2}$, in subexponential time. More specifically:

Theorem 4.4. *Let $(G, 1, l)$ be an instance, with $G = (V, E)$ and linear latencies $l_e(x) = a_e \cdot x + b_e$. Let $\alpha = \max_{e \in E} \{a_e\}$, and let H^B be the Best Subnetwork of G. For some constants d_1, d_2, let $|\mathcal{P}| \leq m^{d_1}$ and $|P| \leq \ln^{d_2} m$, for all $P \in \mathcal{P}$. Then, for any $\epsilon > 0$, we can compute in time:*

$$m^{\mathcal{O}\left(\frac{d_1 \alpha^2 \ln^{2d_2+1}(2m)}{\epsilon^2}\right)} \tag{4.1}$$

a flow $\tilde{f}$ that is an $\frac{\epsilon}{2}$ - Nash flow on $G_{\tilde{f}}$ and satisfies $l_P(\tilde{f}) \leq L_{eq}(H^B) + \frac{\epsilon}{2}$, for all paths P in $G_{\tilde{f}}$.

Proof. Let $\epsilon > 0$ be any fixed constant. Also, let $\varepsilon = \frac{\epsilon}{2\alpha \ln^{d_2}(2m)}$, and f be a Nash flow on subnetwork H^B. By lemma 2.9, we may assume that f is acyclic, and let H^B be the network $G_f = (V, E_f)$. By applying the previous lemma to H^B with approximation parameter ε we get that there exists a feasible flow $\tilde{f}$ that assigns traffic to at most $\kappa = \left\lceil \frac{\ln(2m)}{2\varepsilon^2} \right\rceil = \left\lceil \frac{2\alpha^2 \ln^{2d_2+1}(2m)}{\epsilon^2} \right\rceil$ paths, such that with positive probability $\left|\tilde{f}_e - f_e\right| \leq \varepsilon$, for all edges $e \in H^B$. Since f is acyclic, $\tilde{f}$ is acyclic too.

Now, we have to distinguish the following two cases:

- $\diamond$ $\forall e \in H^B$, $\tilde{f}_e \leq f_e + \varepsilon$:
  By all the above we have that:

$$\begin{aligned}
l_P(\tilde{f}) &\leq \sum_{e \in P} \big(a_e(f_e + \varepsilon) + b_e\big) \\
&= \sum_{e \in P} (a_e \cdot f_e + b_e) + \sum_{e \in P} (a_e \cdot \varepsilon) \\
&= L_{eq}(H^B) + \sum_{e \in P} (a_e \cdot \varepsilon) \\
&\leq L_{eq}(H^B) + |P| \cdot \alpha \cdot \varepsilon \\
&\leq L_{eq}(H^B) + \ln^{d_2} m \cdot \alpha \cdot \frac{\epsilon}{2\alpha \ln^{d_2}(2m)} \\
&\leq L_{eq}(H^B) + \frac{\epsilon}{2}
\end{aligned}$$

- $\diamond$ $\forall e \in H^B$, $\tilde{f}_e \geq f_e - \varepsilon$:
  By all the above we have that:

$$\begin{aligned}
l_P(\tilde{f}) &\geq \sum_{e \in P} \big(a_e(f_e - \varepsilon) + b_e\big) \\
&= \sum_{e \in P} (a_e \cdot f_e + b_e) - \sum_{e \in P} (a_e \cdot \varepsilon) \\
&= L_{eq}(H^B) - \sum_{e \in P} (a_e \cdot \varepsilon) \\
&= L_{eq}(H^B) - \frac{\epsilon}{2\alpha \ln^{d_2}(2m)} \cdot \sum_{e \in P} a_e \\
&\geq L_{eq}(H^B) - \frac{\epsilon}{2\alpha \ln^{d_2}(2m)} \cdot \sum_{e \in P} \alpha \\
&\geq L_{eq}(H^B) - \frac{\epsilon}{2\alpha \ln^{d_2}(2m)} \cdot \ln^{d_2} m \cdot \alpha \\
&\geq L_{eq}(H^B) - \frac{\epsilon}{2}
\end{aligned}$$

So, there exists a feasible acyclic flow $\tilde{f}$ that assigns positive flow to at most κ paths, that is an $\frac{\epsilon}{2}$ - Nash flow on the best subnetwork H^B, with latency at most $L_{eq}(H^B) + \frac{\epsilon}{2}$.

Therefore,

- For every set of κ paths, out of the total μ, we find the lemma 4.3's corresponding flow g. Total number of paths, μ: less than m^{d_1}.
- If the corresponding network G_g is acyclic, we check whether g is an ϵ - Nash flow, by computing the minimum and maximum latency paths and then finding their difference. If they differ by at most ϵ, then g is an ϵ - Nash flow. This computation is polynomial in time.
- Among all acyclic ϵ - Nash flows, we return the one that has maximum latency, the minimum of the rest flows' maximum latencies. It must be clear that this latency cannot be more than $L_{eq}(H^B) + \frac{\epsilon}{2}$, since by exhaustive search we should encounter $\tilde{f}$ that has the above proven property.
- Since the total number of paths is less than m^{d_1}, there are at most $m^{d_1 \cdot \kappa}$ different sets of κ paths. Thus, the total computational time is $m^{\mathcal{O}(d_1 \cdot \kappa)}$.

□

Before we move on to the main theorems of this section, we present the following two useful theorems:

Theorem 4.5. *McDiarmid's inequality: Let $Z_1, Z_2, \ldots, Z_m \in \mathcal{Z}$ be independent random variables and $f : \mathcal{Z}^m \to \mathbb{R}$ be a function of $Z_1, Z_2, \ldots, Z_m$. If for all $i \in [m]$ and for all $z_1, z_2, \ldots, z_m, z_i' \in \mathcal{Z}$ the function f satisfies:*

$$\left| f(z_1, z_2, \ldots, z_i, \ldots, z_m) - f(z_1, z_2, \ldots, z_i', \ldots, z_m) \right| \leq c_i$$

then for $t > 0$:

$$\mathbb{P}(|f - \mathbb{E}[f]| \geq t) \leq 2e^{\frac{-2t^2}{\sum_{i=1}^m c_i^2}}$$

Theorem 4.6. *Kahane's inequality: Let $r_1, r_2, \ldots, r_m$ be a sequence of identically and independently distributed (i.i.d.) Rademacher ± 1 random variables, i.e., $\mathbb{P}(r_i = \pm 1) = \frac{1}{2}$ for all $i \in [m]$. In addition, let $u_1, u_2, \ldots, u_m \in \mathbb{R}^d$ be a deterministic sequence of vectors. Then, for $2 \leq p < \infty$:*

$$\mathbb{E} \left\| \sum_{i=1}^m r_i u_i \right\|_p \leq \sqrt{p} \Big(\sum_{i=1}^m \|u_i\|_p^2 \Big)^{\frac{1}{2}}$$

Now, we prove the following theorem that is a generalization of Barman's theorem 4 in [31] for . In order to proceed to the proof, we use the following version of Hoeffding's inequality:

Lemma 4.7. *Hoeffding's inequality: Let $X_1, X_2, \dots, X_n$ be independent random variables with $\mathbb{P}\big(X_i \in [a_i, b_i]\big) = 1$. Then:*

$$\mathbb{P}\Big(\Big| \frac{1}{n} \sum_{i=1}^{n} X_i - \mathbb{E}[\frac{1}{n} \sum_{i=1}^{n} X_i] \Big| \geq \epsilon \Big) \leq 2e^{-\frac{2n^2 \epsilon^2}{\sum_{i=1}^{n} (b_i - a_i)^2}}$$

Theorem 4.8. *Let $X = \{x_1, x_2, \dots, x_n\} \subset \mathbb{R}^d$ be a set of vectors with $\gamma = \max_{x \in X} \|x\|_\infty$ and $\epsilon > 0$. Then, for every $\mu \in conv(X)$ there exists a $\left\lceil \frac{2\gamma^2 \cdot \ln n}{\epsilon^2} \right\rceil$ uniform vector $\tilde{\mu} \in conv(X)$ such that:*

$$\|\mu - \tilde{\mu}\|_\infty \leq \epsilon$$

Proof. Since $\mu \in \text{conv}(X)$, it can be expressed as a convex combination of the x_i's: $\mu = \sum_{i=1}^{n} \beta_i x_i$, where $\sum_{i=1}^{n} \beta_i = 1$, $\beta_i \geq 0, \forall i \in [n]$. Then $\beta = (\beta_1, \beta_2, \dots, \beta_n)$ corresponds to a probability distribution over vectors $x_1, x_2, \dots, x_n$. The latter means that under probability distribution β, vector x_i is drawn with probability β_i. The vector μ is the mean of this distribution. More specifically, the jth component of μ is the expected value of the random variable v_j that takes value $x_{i,j}$ with probability β_i, where $x_{i,j}$ is the jth component of vector x_i, $j \in [d]$. Specifically:

$$\mu_j = \mathbb{E}[v_j] = \sum_{i=1}^{n} \mathbb{P}(v_j = x_{i,j}) \cdot x_{i,j} = \sum_{i=1}^{n} \beta_i \cdot x_{i,j}$$

and

$$\mu = \mathbb{E}_{v \sim \beta}[v]$$

Let $v'_1, v'_2, \dots, v'_m$ be m i.i.d. draws from β. The sample mean vector is defined to be:

$$\tilde{\mu} = \frac{1}{m} \sum_{i=1}^{m} v'_i$$

Also, in expectation, the sample mean should be equal to μ. Specifically:

$$\mu = \mathbb{E}_{v'_1, v'_2, \dots, v'_m \sim \beta} \frac{1}{m} \sum_{i=1}^{m} v'_i$$

Thus, we have the following:

$$\mathbb{P}\big(\|\mu - \tilde{\mu}\|_\infty \geq \epsilon\big) =$$
$$\mathbb{P}\Bigg(\Bigg\|\tilde{\mu} - \Big(\mathbb{E}_{v_1',v_2',\dots,v_m' \sim \beta}\frac{1}{m}\sum_{i=1}^m v_i'\Big)\Bigg\|_\infty \geq \epsilon\Bigg) =$$
$$\mathbb{P}\Bigg(\Bigg\|\frac{1}{m}\sum_{i=1}^m v_i' - \Big(\mathbb{E}_{v_1',v_2',\dots,v_m' \sim \beta}\frac{1}{m}\sum_{i=1}^m v_i'\Big)\Bigg\|_\infty \geq \epsilon\Bigg)$$
$$\mathbb{P}\Bigg(\Bigg\|\frac{1}{m}\sum_{i=1}^m v_i' - \mathbb{E}\Big[\frac{1}{m}\sum_{i=1}^m v_i'\Big]\Bigg\|_\infty \geq \epsilon\Bigg) =$$
$$\mathbb{P}\Bigg(\Bigg\|\frac{1}{m}\sum_{i=1}^m v_i' - \frac{1}{m}\sum_{i=1}^m \mathbb{E}[v_i']\Bigg\|_\infty \geq \epsilon\Bigg) =$$
$$\mathbb{P}\Bigg(\Bigg\|\frac{1}{m}\Big(\sum_{i=1}^m v_i' - \sum_{i=1}^m \mathbb{E}[v_i']\Big)\Bigg\|_\infty \geq \epsilon\Bigg) =$$
$$\mathbb{P}\Bigg(\Bigg\|\frac{1}{m}\Big(\sum_{i=1}^m (v_{i,1}', v_{i,2}', \dots, v_{i,d}') - \sum_{i=1}^m (\mathbb{E}[v_{i,1}'], \mathbb{E}[v_{i,2}'], \dots, \mathbb{E}[v_{i,d}'])\Big)\Bigg\|_\infty \geq \epsilon\Bigg) =$$
$$\mathbb{P}\Bigg(\Bigg\|\frac{1}{m}\sum_{i=1}^m v_{i,1}' - \frac{1}{m}\sum_{i=1}^m \mathbb{E}[v_{i,1}'], \dots, \frac{1}{m}\sum_{i=1}^m v_{i,d}' - \frac{1}{m}\sum_{i=1}^m \mathbb{E}[v_{i,d}']\Bigg\|_\infty \geq \epsilon\Bigg) \quad (4.2)$$

Let's assume that for an $l \in [d]$:

$$\left|\frac{1}{m}\sum_{i=1}^m v_{i,l}' - \frac{1}{m}\sum_{i=1}^m \mathbb{E}[v_{i,l}']\right| =$$
$$\max\left\{\left|\frac{1}{m}\sum_{i=1}^m v_{i,1}' - \frac{1}{m}\sum_{i=1}^m \mathbb{E}[v_{i,1}']\right|, \left|\frac{1}{m}\sum_{i=1}^m v_{i,2}' - \frac{1}{m}\sum_{i=1}^m \mathbb{E}[v_{i,2}']\right|, \dots, \left|\frac{1}{m}\sum_{i=1}^m v_{i,d}' - \frac{1}{m}\sum_{i=1}^m \mathbb{E}[v_{i,d}']\right|\right\}$$

Then from the definition of the ∞ norm, relation (4.2) becomes:

$$\mathbb{P}\big(\|\mu - \tilde{\mu}\|_\infty \geq \epsilon\big) = \mathbb{P}\Bigg(\left|\frac{1}{m}\sum_{i=1}^m v_{i,l}' - \frac{1}{m}\sum_{i=1}^m \mathbb{E}[v_{i,l}']\right| \geq \epsilon\Bigg) \qquad (4.3)$$

But since $v' \sim \beta$, each $v_{i,l}'$ takes value $x_{k,l}$ with probability β_k, for a $k \in [n]$. Thus, $|v_{i,l}'| \leq \gamma = \max_{x \in X} \|x\|_\infty$ and $\mathbb{P}\big(v_{i,l}' \in [-\gamma, \gamma]\big) = 1$ for every $i \in [m], l \in [d]$.

Then, because $v_{1,l}', v_{2,l}', \dots, v_{m,l}'$ are independent random variables, then from Hoeffding's inequality, lemma 4.7, relation (4.3) becomes:

$$\mathbb{P}\big(\|\mu - \tilde{\mu}\|_\infty \geq \epsilon\big) =$$
$$\mathbb{P}\Bigg(\Bigg|\frac{1}{m}\sum_{i=1}^{m} v'_{i,l} - \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}[v'_{i,l}]\Bigg| \geq \epsilon\Bigg) \leq 2e^{-\frac{2m^2\epsilon^2}{\sum_{i=1}^{m}(2\gamma)^2}} = 2e^{-\frac{m\epsilon^2}{2\gamma^2}} \quad (4.4)$$

So, for $m \geq \frac{2\gamma^2 \cdot \ln n}{\epsilon^2}$, we have that $\mathbb{P}\big(\|\mu - \tilde{\mu}\|_\infty \geq \epsilon\big) \leq \frac{2}{n}$. So it suffices for m to be $m := \left\lceil \frac{2\gamma^2 \cdot \ln n}{\epsilon^2} \right\rceil$. The conclusion follows.

□

Having in mind the above three theorems 4.5, 4.6, 4.8 and due to the work of Barman in [31], we will prove the following two theorems:

Theorem 4.9. *Let $(G, 1, l)$ be an instance, with $G = (V, E)$ and linear latencies $l_e(x) = a_e \cdot x + b_e$, with rational coefficients $a_e, b_e \geq 0$ and let H^B be the Best Subnetwork of G. Let also $\mu = |\mathcal{P}| \leq m^{d_1}$ be the total number of paths, $|P| \leq \ln^{d_2} m$ for all $P \in \mathcal{P}$ and $\alpha = \max_{e\in E}\{a_e\}$. Then, there exist a flow $\tilde{f}$, to at most $k \leq \kappa$ paths P with $\tilde{f}_P = \frac{\rho}{\kappa}, \rho \in [\kappa]$, that is an $\frac{\epsilon}{2}$ - Nash flow on $G_{\tilde{f}}$ and satisfies $l_P(\tilde{f}) \leq L_{eq}(H^B) + \frac{\epsilon}{2}$, for all paths P in $G_{\tilde{f}}$. This flow can be computed in time:*

$$m^{\mathcal{O}\left(\frac{d_1 p \alpha^2 m^{\frac{2d_1}{p}} \ln^{2d_2} m}{\epsilon^2}\right)} \quad (4.5)$$

for p norm with $2 \leq p < \infty$, where the number of paths, k, is less than or equal to $\kappa = \left\lceil \frac{64 p \alpha^2 m^{\frac{2d_1}{p}} \ln^{2d_2} m}{\epsilon^2} \right\rceil$,
or:

$$m^{\mathcal{O}\left(\frac{d_1^2 \alpha^2 \ln^{2d_2+1} m}{\epsilon^2}\right)} \quad (4.6)$$

for the $p = \infty$ norm, where the number of paths, k, is less than or equal to $\kappa = \left\lceil \frac{8 d_1 \alpha^2 \ln^{2d_2+1} m}{\epsilon^2} \right\rceil$.

Proof. Let A be the following matrix:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1\mu} \\ a_{12} & a_{22} & a_{23} & \dots & a_{2\mu} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{1\mu} & a_{2\mu} & a_{3\mu} & \dots & a_{\mu\mu} \end{bmatrix}$$

with:

$$a_{ii} = \sum_{e \in P_i} a_e$$
$$a_{ij} = \sum_{e \in P_i, P_j} a_e$$

and B the following one:

$$B = \begin{bmatrix} b_{P_1} \\ b_{P_2} \\ \vdots \\ b_{P_\mu} \end{bmatrix}$$

where:

$$b_{P_i} = \sum_{e \in P_i} b_e$$

It should be clear that for every $a_{ij} \in A$ it should be:

$$a_{ij} = \sum_{e \in P_i, P_j} a_e = \sum_{e \in P_j, P_i} a_e = a_{ji}$$

Then, every path P_i's latency should be the following $\forall i \in [\mu]$:

$$l_{P_i}(f) = a_{1i} f_{P_1} + a_{2i} f_{P_2} + \ldots + a_{ii} f_{P_i} + a_{i+1i} f_{P_{i+1}} + \ldots + a_{\mu i} f_{P_\mu} + b_{P_i}$$

Now, let the Nash Equilibrium flow of the Best Subnetwork H^B be f_{eq}.

Then, there will be an $S \subseteq [\mu]$ such that $\forall i, j \in S : f_{eqP_i}, f_{eqP_j} > 0$ and $l_{P_i}(f_{eq}) = l_{P_j}(f_{eq}) = L_{eq}(H^B)$. If $S \subset [\mu]$, then $T = [\mu] - S$ and $\forall i \in T : f_{eqP_i} = 0$ and $l_{P_i}(f_{eq}) \neq L_{eq}(H^B)$.

Let $a_{max} = \max\{a_{ij} | a_{ij} \in A, \forall i, j \in [\mu]\}$. Then:

$$\begin{aligned} a_{max} &= \max\{a_{ij} | a_{ij} \in A, \forall i, j \in [\mu]\} \\ &\leq \sum_{l=1}^{\max\{|P|\}} \alpha \\ &\leq \alpha \cdot \ln^{d_2} m \end{aligned}$$

Now, let's define the following vector:

$$\nu = A^{(1)} \cdot f_{eqP_1} + A^{(2)} \cdot f_{eqP_2} + \ldots + A^{(\mu)} \cdot f_{eqP_\mu} \tag{4.7}$$

It should be obvious that $\nu \in \text{conv}(\{A^{(j)}\}_j)$.

Using McDiarmid's and Kahane's inequalities, we will find a vector $\tilde{\nu} \in \text{conv}(\{A^{(j)}\}_j)$ such that $\|\nu - \tilde{\nu}\|_p \leq \frac{\epsilon}{2}$.

Indeed. $f_{eq} = (f_{eqP_1}, f_{eqP_2}, \ldots, f_{eqP_\mu})$ could be interpreted as a probability distribution over vectors $A^{(1)}, A^{(2)}, \ldots, A^{(\mu)}$. That is, under probability distribution f_{eq} vector $A^{(i)}$ is drawn with probability $f_{eq_{P_i}}$. The vector ν is the mean of this distribution. Specifically, the jth component of ν is the expected value of the random variable that takes value a_{ij} with probability $f_{eq_{P_i}}$, where a_{ij} is the jth component of vector $A^{(i)}$. We succinctly express these component-wise equalities as follows:

$$\mathbb{E}_{v \sim f_{eq}}[v] = \nu$$

Let $v_1, v_2, \ldots, v_\kappa$ be κ i.i.d. draws from f_{eq}. The sample mean vector is defined to be $\frac{1}{\kappa}\sum_{i=1}^{\kappa} v_i$. Below we specify function $g : \{A^{(j)}\}_j^{\kappa} \to \mathbb{R}$ to quantify the p-norm distance between the sample mean vector and the ν.

$$g(v_1, v_2, \ldots, v_\kappa) := \left\| \frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i - \nu \right\|_p$$

where p is norm, with $2 \leq p < \infty$.

Now, for every $j \in [\mu]$ the following is valid:

$$\left\| A^{(j)} \right\|_p = \left(\sum_{i=1}^{\mu} a_{ij}^p \right)^{\frac{1}{p}} \leq \left(\sum_{i=1}^{\mu} a_{max}^p \right)^{\frac{1}{p}} = \mu^{\frac{1}{p}} \cdot a_{max} \leq \alpha \cdot m^{\frac{d_1}{p}} \cdot \ln^{d_2} m$$

We will use McDiarmid's inequality. In particular, we will establish that with positive probability the sample mean vector defined over $\kappa := \left\lceil \frac{64 p \alpha^2 m^{\frac{2d_1}{p}} \ln^{2d_2} m}{\epsilon^2} \right\rceil$ draws, is $\frac{\epsilon}{2}$ close to ν in p-norm. Hence, the stated claim is implied by the probabilistic method.

For any κ tuple $(v_1, v_2, \ldots, v_i, \ldots, v_\kappa) \in \{A^{(j)}\}_j^{\kappa}$ and $v_i' \in \{A^{(j)}\}_j$, we show that $|g(v_1, v_2, \ldots, v_i, \ldots, v_\kappa) - g(v_1, v_2, \ldots, v_i', \ldots, v_\kappa)|$ is no more than $\frac{2 \cdot \alpha \cdot m^{\frac{d_1}{p}} \cdot \ln^{d_2} m}{\kappa}$. We can assume without loss of generality that $g(v_1, v_2, \ldots, v_i, \ldots, v_\kappa) \geq g(v_1, v_2, \ldots, v_i', \ldots, v_\kappa)$, since the other case is symmetric.

Setting $u := \frac{1}{\kappa}\sum_{j \neq i} v_j - \nu$ we have:

$$\begin{aligned}
g(v_1, v_2, \ldots, v_i, \ldots, v_\kappa) - g(v_1, v_2, \ldots, v_i', \ldots, v_\kappa) &= \left\| u + \frac{1}{\kappa} v_i \right\|_p - \left\| u + \frac{1}{\kappa} v_i' \right\|_p \\
&\leq \|u\|_p + \frac{1}{\kappa}\|v_i\|_p - \|u\|_p + \frac{1}{\kappa}\|v_i'\|_p \\
&\leq \frac{1}{\kappa}\|v_i\|_p + \frac{1}{\kappa}\|v_i'\|_p \\
&\leq \frac{1}{\kappa}\max\{\left\|A^{(j)}\right\|_p\} + \frac{1}{\kappa}\max\{\left\|A^{(j)}\right\|_p\} \\
&= \frac{2}{\kappa}\max\{\left\|A^{(j)}\right\|_p\} \\
&\leq \frac{2}{\kappa}\alpha \cdot m^{\frac{d_1}{p}} \cdot \ln^{d_2} m
\end{aligned}$$

Given that g satisfies $|g(v_1, v_2, \ldots, v_i, \ldots, v_\kappa) - g(v_1, v_2, \ldots, v_i', \ldots, v_\kappa)| \leq \frac{2\alpha m^{\frac{d_1}{p}} \ln^{d_2} m}{\kappa}$, we can apply Mc-Diarmid's inequality, with $c_i = \frac{2\alpha m^{\frac{d_1}{p}} \ln^{d_2} m}{\kappa}$ for all $i \in [\kappa]$, to obtain:

$$\mathbb{P}(|g - \mathbb{E}[g]| \geq t) \leq 2e^{\frac{-2t^2\kappa}{4\alpha^2 m^{\frac{2d_1}{p}} \ln^{2d_2} m}} \tag{4.8}$$

At first, we are going to prove that $\mathbb{E}[g] \leq 2\mathbb{E}_{v_i, r_i} \left\| \sum_{i=1}^{\kappa} r_i \frac{v_i}{\kappa} \right\|_p$, where $r_1, r_2, \ldots, r_\kappa$ be a sequence of i.i.d. Rademacher ± 1 random variables.

Recall that in expectation the sampled mean is equal to ν:

$$\mathbb{E}_{v_1', v_2', \ldots, v_\kappa' \sim f_{eq}} \sum_{i=1}^{\kappa} v_i' = \nu$$

Hence, we have that:

$$\begin{aligned}
\mathbb{E}[g] &= \mathbb{E}_{v_1, v_2, \ldots, v_\kappa} \left\| \frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i - \nu \right\|_p \\
&= \mathbb{E}_{v_1, v_2, \ldots, v_\kappa} \left\| \frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i - \mathbb{E}_{v_1', v_2', \ldots, v_\kappa'} \frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i' \right\|_p \\
&= \mathbb{E}_{v_1, v_2, \ldots, v_\kappa} \left\| \mathbb{E}_{v_1', v_2', \ldots, v_\kappa'} \left(\frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i - \frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i' \right) \right\|_p
\end{aligned} \tag{4.9}$$

Since $\|\cdot\|$ is convex for $p \geq 1$, Jensen's inequality gives:

$$\begin{aligned}
\mathbb{E}_{v_1, v_2, \ldots, v_\kappa} \left\| \mathbb{E}_{v_1', v_2', \ldots, v_\kappa'} \left(\frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i - \frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i' \right) \right\|_p &\leq \mathbb{E}_{v_1, v_2, \ldots, v_\kappa} \mathbb{E}_{v_1', v_2', \ldots, v_\kappa'} \left\| \left(\frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i - \frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i' \right) \right\|_p \\
&= \frac{1}{\kappa} \mathbb{E}_{v_i, v_i'} \left\| \left(\sum_{i=1}^{\kappa} (v_i - v_i') \right) \right\|_p
\end{aligned} \tag{4.10}$$

Let $r_1, r_2, \ldots, r_\kappa$ be a sequence of i.i.d. Rademacher ± 1 random variables, i.e., $\mathbb{P}(r_i = \pm 1) = \frac{1}{2}$ for all $i \in [\kappa]$. Since, for all $i \in [\kappa]$, v_i and v_i' are i.i.d. copies we can write:

$$\begin{aligned}
\frac{1}{\kappa}\mathbb{E}_{v_i,v_i'}\left\|\left(\sum_{i=1}^{\kappa}(v_i-v_i')\right)\right\|_p &= \frac{1}{\kappa}\mathbb{E}_{v_i,v_i',r_i}\left\|\left(\sum_{i=1}^{\kappa}r_i(v_i-v_i')\right)\right\|_p \\
&\leq \frac{1}{\kappa}\mathbb{E}_{v_i,v_i',r_i}\left[\left\|\sum_{i=1}^{\kappa}r_iv_i\right\|_p+\left\|\sum_{i=1}^{\kappa}r_iv_i'\right\|_p\right] \\
&= \frac{1}{\kappa}\mathbb{E}_{r_i}\left[\mathbb{E}_{v_i,v_i'}\left(\left\|\sum_{i=1}^{\kappa}r_iv_i\right\|_p+\left\|\sum_{i=1}^{\kappa}r_iv_i'\right\|_p\middle|r_1,\dots,r_\kappa\right)\right] \\
&= \frac{1}{\kappa}\mathbb{E}_{r_i}\left[\mathbb{E}_{v_i}\left(\left\|\sum_{i=1}^{\kappa}r_iv_i\right\|_p\middle|r_1,\dots,r_\kappa\right)+\mathbb{E}_{v_i'}\left(\left\|\sum_{i=1}^{\kappa}r_iv_i'\right\|_p\middle|r_1,\dots,r_\kappa\right)\right] \\
&= \frac{1}{\kappa}\mathbb{E}_{r_i}\left[2\mathbb{E}_{v_i}\left(\left\|\sum_{i=1}^{\kappa}r_iv_i\right\|_p\middle|r_1,\dots,r_\kappa\right)\right] \\
&= 2\mathbb{E}_{v_i,r_i}\left\|\sum_{i=1}^{\kappa}r_i\frac{v_i}{\kappa}\right\|_p \qquad (4.11)
\end{aligned}$$

The penultimate equality follows from the following:

$$\mathbb{E}_{v_i}\left(\left\|\sum_{i=1}^{\kappa}r_iv_i\right\|_p\middle|r_1,\dots,r_\kappa\right)=\mathbb{E}_{v_i'}\left(\left\|\sum_{i=1}^{\kappa}r_iv_i'\right\|_p\middle|r_1,\dots,r_\kappa\right)$$

Overall, relations (4.9), (4.10), (4.11) imply that:

$$\mathbb{E}[g]\leq 2\mathbb{E}_{v_i,r_i}\left\|\sum_{i=1}^{\kappa}r_i\frac{v_i}{\kappa}\right\|_p \qquad (4.12)$$

Now, having proved that $\mathbb{E}[g]\leq 2\mathbb{E}_{v_i,r_i}\left\|\sum_{i=1}^{\kappa}r_i\frac{v_i}{\kappa}\right\|_p$, we will show that $\mathbb{P}(g\geq\varepsilon)\leq 2e^{-2}$.

Indeed, by applying Kahane's inequality with $u_i=\frac{v_i}{\kappa}$ we obtain:

$$\begin{aligned}
\mathbb{E}_{v_i,r_i}\left\|\sum_{i=1}^{\kappa}r_i\frac{v_i}{\kappa}\right\|_p &= \mathbb{E}_{v_i}\left[\mathbb{E}_{r_i}\left[\left\|\sum_{i=1}^{\kappa}r_i\frac{v_i}{\kappa}\right\|_p\middle|v_1\dots v_\kappa\right]\right] \\
&\leq \mathbb{E}_{v_i}\left[\sqrt{p}\left(\sum_{i=1}^{\kappa}\left\|\frac{v_i}{\kappa}\right\|_p^2\right)^{\frac{1}{2}}\right]\leq\mathbb{E}_{v_i}\left[\sqrt{p}\left(\sum_{i=1}^{\kappa}\frac{\left(\max\{\|A^{(j)}\|_p\}\right)^2}{\kappa^2}\right)^{\frac{1}{2}}\right] \\
&\leq \sqrt{p}\frac{\alpha m^{\frac{d_1}{p}}\ln^{d_2}m}{\sqrt{\kappa}} \qquad (4.13)
\end{aligned}$$

By relations (4.12) and (4.13) we have:

$$\mathbb{E}[g] \leq 2\sqrt{p}\frac{\alpha m^{\frac{d_1}{p}} \ln^{d_2} m}{\sqrt{\kappa}}$$

For sample size:

$$\kappa \geq 16 \cdot \frac{p\alpha^2 m^{\frac{2d_1}{p}} \ln^{2d_2} m}{\varepsilon^2} \tag{4.14}$$

where $\varepsilon > 0$, we have:

$$\begin{aligned}\mathbb{E}[g] &\leq 2\sqrt{p}\frac{\alpha m^{\frac{d_1}{p}} \ln^{d_2} m}{\sqrt{\kappa}} \\ &\leq 2\sqrt{p}\frac{\alpha m^{\frac{d_1}{p}} \ln^{d_2} m}{4 \cdot \frac{\sqrt{p}\alpha m^{\frac{d_1}{p}} \ln^{d_2} m}{\varepsilon}} \\ &\leq \frac{\varepsilon}{2}\end{aligned}$$

Setting $t = \frac{\varepsilon}{2}$ in relation (4.8) we have the following two cases:

- $g - \mathbb{E}[g] \geq 0$:

$$\begin{aligned}&\mathbb{P}(|g - \mathbb{E}[g]| \geq t) \leq 2e^{\frac{-2t^2\kappa}{4\alpha^2 m^{\frac{2d_1}{p}} \ln^{2d_2} m}} \Rightarrow \\ &\mathbb{P}(g - \mathbb{E}[g] \geq \frac{\varepsilon}{2}) \leq 2e^{\frac{-2\varepsilon^2\kappa}{16\alpha^2 m^{\frac{2d_1}{p}} \ln^{2d_2} m}} \Rightarrow \\ &\mathbb{P}(g \geq \mathbb{E}[g] + \frac{\varepsilon}{2}) \leq 2e^{\frac{-2\varepsilon^2 \frac{16p\alpha^2 m^{\frac{2d_1}{p}} \ln^{2d_2} m}{\varepsilon^2}}{16\alpha^2 m^{\frac{2d_1}{p}} \ln^{2d_2} m}} \Rightarrow \\ &\mathbb{P}(g \geq \mathbb{E}[g] + \frac{\varepsilon}{2}) \leq 2e^{-2p} \Rightarrow \\ &\mathbb{P}(g \geq \frac{\varepsilon}{2}) \leq 2e^{-2} \Rightarrow \\ &\mathbb{P}(g \geq \varepsilon) \leq 2e^{-2}\end{aligned}$$

- $g - \mathbb{E}[g] \leq 0$:

$$\begin{aligned}&|g - \mathbb{E}[g]| \geq t \Rightarrow \\ &\mathbb{E}[g] - g \geq t \Rightarrow \\ &g \leq \mathbb{E}[g] - t \Rightarrow \\ &g \leq \frac{\varepsilon}{2} - \frac{\varepsilon}{2} \Rightarrow \\ &g \leq 0\end{aligned}$$

which is a contradiction.

So, $\mathbb{P}(g \geq \varepsilon) \leq 2e^{-2}$ or $\mathbb{P}\Big(\left\|\frac{1}{\kappa}\sum_{i=1}^{\kappa} v_i - \nu\right\|_p \geq \varepsilon\Big) \leq 2e^{-2}$. Therefore, with positive probability:

$$\left\|\frac{1}{\kappa}\sum_{i=1}^{\kappa} v_i - \nu\right\|_p \leq \varepsilon$$

or for $\varepsilon = \frac{\epsilon}{2}$:

$$\left\|\frac{1}{\kappa}\sum_{i=1}^{\kappa} v_i - \nu\right\|_p \leq \frac{\epsilon}{2} \tag{4.15}$$

The latter means that we have found a vector $\tilde{\nu} := \frac{1}{\kappa}\sum_{i=1}^{\kappa} v_i$, such that $\tilde{\nu} \in \text{conv}(\{A^{(j)}\}_j)$ and $\|\nu - \tilde{\nu}\|_p \leq \frac{\epsilon}{2}$.

Since $v_1, v_2, \ldots, v_\kappa$ are κ i.i.d. draws from f_{eq}, vector $\tilde{\nu}$ could be expressed as:

$$\tilde{\nu} = A^{(1)} \cdot \tilde{f}_{P_1} + A^{(2)} \cdot \tilde{f}_{P_2} + \ldots + A^{(\mu)} \cdot \tilde{f}_{P_\mu} \tag{4.16}$$

$$= \frac{\kappa_1}{\kappa} \cdot A^{(i_{\kappa_1})} + \frac{\kappa_2}{\kappa} \cdot A^{(i_{\kappa_2})} + \ldots + \frac{\kappa_k}{\kappa} \cdot A^{(i_{\kappa_k})} \tag{4.17}$$

where $\kappa_1 + \kappa_2 + \ldots + \kappa_k = \kappa$ and $1 \leq \kappa_1, \kappa_2, \ldots, \kappa_k \leq \kappa$.

Taking into account relations (4.7) and (4.16), relation (4.15) could be analyzed as follows:

$$\|\nu - \tilde{\nu}\|_p \leq \frac{\epsilon}{2} \Rightarrow$$
$$\left\|A^{(1)} \cdot (f_{eq_{P_1}} - \tilde{f}_{P_1}) + A^{(2)} \cdot (f_{eq_{P_2}} - \tilde{f}_{P_2}) + \ldots + A^{(\mu)} \cdot (f_{eq_{P_\mu}} - \tilde{f}_{P_\mu})\right\|_p \leq \frac{\epsilon}{2} \Rightarrow$$
$$\left\|A^{(1)} \cdot (f_{eq_{P_1}} - \tilde{f}_{P_1}) + A^{(2)} \cdot (f_{eq_{P_2}} - \tilde{f}_{P_2}) + \ldots + A^{(\mu)} \cdot (f_{eq_{P_\mu}} - \tilde{f}_{P_\mu}) + B - B\right\|_p \leq \frac{\epsilon}{2} \Rightarrow$$
$$\left\|A \cdot (f_{eq} - \tilde{f}) + B - B\right\|_p \leq \frac{\epsilon}{2} \Rightarrow$$
$$\left\|(A \cdot f_{eq} + B) - (A \cdot \tilde{f} + B)\right\|_p \leq \frac{\epsilon}{2} \Rightarrow$$
$$\left\|\Big(l_{P_1}(f_{eq}) - l_{P_1}(\tilde{f}), l_{P_2}(f_{eq}) - l_{P_2}(\tilde{f}), \ldots, l_{P_\mu}(f_{eq}) - l_{P_\mu}(\tilde{f})\Big)\right\|_p \leq \frac{\epsilon}{2} \Rightarrow$$
$$\left|l_{P_1}(f_{eq}) - l_{P_1}(\tilde{f})\right|^p + \left|l_{P_2}(f_{eq}) - l_{P_2}(\tilde{f})\right|^p + \ldots + \left|l_{P_\mu}(f_{eq}) - l_{P_\mu}(\tilde{f})\right|^p \leq \frac{\epsilon^p}{2^p}$$

Thus, $\forall i \in S$:

$$\left|L_{eq}(H^B) - l_{P_i}(\tilde{f})\right| \leq \frac{\epsilon}{2}$$

or better:

$$l_{P_i}(\tilde{f}) \leq L_{eq}(H^B) + \frac{\epsilon}{2} \tag{4.18}$$

Now, since by relation (4.14) $\kappa \geq 16 \cdot \frac{p\alpha^2 m^{\frac{2d_1}{p}} \ln^{2d_2} m}{\varepsilon^2}$, then for $\varepsilon = \frac{\epsilon}{2}$ it suffices to consider only a $\kappa := \left\lceil \frac{64 p\alpha^2 m^{\frac{2d_1}{p}} \ln^{2d_2} m}{\epsilon^2} \right\rceil$ number of paths.

Therefore, since $\mu \leq m^{d_1}$ for some $d_1 > 0$, then by exhaustive search in time $T = m^{\mathcal{O}(d_1 \cdot \kappa)}$ we can find a subnetwork (the one that minimizes the latency of the maximum latency path) such that:

$$T = m^{\mathcal{O}(\frac{d_1 p\alpha^2 m^{\frac{2d_1}{p}} \ln^{2d_2} m}{\epsilon^2})}$$

Hence, by the relation above and relations (4.17), (4.18) we come to the theorem's conclusion for p norm with $2 \leq p < \infty$.

Now, for the $p = \infty$ case, we are going to use theorem 4.8.

More specifically, $f_{eq} = (f_{eqP_1}, f_{eqP_2}, \ldots, f_{eqP_\mu})$ could be interpreted again as a probability distribution over vectors $A^{(1)}, A^{(2)}, \ldots, A^{(\mu)}$.

Then, $\gamma = \max_{A^{(j)}} \left\|A^{(j)}\right\|_\infty = a_{max} \leq \alpha \cdot \ln^{d_2} m$.

So, since $\mu \leq m^{d_1}$, for $\kappa := \left\lceil \frac{2d_1\alpha^2 \ln^{2d_2+1} m}{\varepsilon^2} \right\rceil$ we have that:

$$\|\tilde{\nu} - \nu\|_\infty \leq \varepsilon$$

or for $\varepsilon = \frac{\epsilon}{2}$, we have for $\kappa := \left\lceil \frac{8d_1\alpha^2 \ln^{2d_2+1} m}{\epsilon^2} \right\rceil$ that:

$$\|\tilde{\nu} - \nu\|_\infty \leq \frac{\epsilon}{2}$$

where:

$$\begin{aligned} \nu &= A^{(1)} \cdot f_{eqP_1} + A^{(2)} \cdot f_{eqP_2} + \ldots + A^{(\mu)} \cdot f_{eqP_\mu} \\ \tilde{\nu} &= A^{(1)} \cdot \tilde{f}_{P_1} + A^{(2)} \cdot \tilde{f}_{P_2} + \ldots + A^{(\mu)} \cdot \tilde{f}_{P_\mu} \\ &= \frac{\kappa_1}{\kappa} \cdot A^{(i_{\kappa_1})} + \frac{\kappa_2}{\kappa} \cdot A^{(i_{\kappa_2})} + \ldots + \frac{\kappa_k}{\kappa} \cdot A^{(i_{\kappa_k})} \end{aligned}$$

and $\kappa_1 + \kappa_2 + \ldots + \kappa_k = \kappa$, $1 \leq \kappa_1, \kappa_2, \ldots, \kappa_k \leq \kappa$, as previously stated. Obviously $\nu, \tilde{\nu} \in \text{conv}(\{A^{(j)}\}_j)$.

Thus:

$$\|\nu - \tilde{\nu}\|_\infty \leq \frac{\epsilon}{2} \Rightarrow$$
$$\left\| A^{(1)} \cdot (f_{eq_{P_1}} - \tilde{f}_{P_1}) + A^{(2)} \cdot (f_{eq_{P_2}} - \tilde{f}_{P_2}) + \ldots + A^{(\mu)} \cdot (f_{eq_{P_\mu}} - \tilde{f}_{P_\mu}) \right\|_\infty \leq \frac{\epsilon}{2} \Rightarrow$$
$$\left\| A^{(1)} \cdot (f_{eq_{P_1}} - \tilde{f}_{P_1}) + A^{(2)} \cdot (f_{eq_{P_2}} - \tilde{f}_{P_2}) + \ldots + A^{(\mu)} \cdot (f_{eq_{P_\mu}} - \tilde{f}_{P_\mu}) + B - B \right\|_\infty \leq \frac{\epsilon}{2} \Rightarrow$$
$$\left\| A \cdot (f_{eq} - \tilde{f}) + B - B \right\|_\infty \leq \frac{\epsilon}{2} \Rightarrow$$
$$\left\| (A \cdot f_{eq} + B) - (A \cdot \tilde{f} + B) \right\|_\infty \leq \frac{\epsilon}{2} \Rightarrow$$
$$\left\| \Big(l_{P_1}(f_{eq}) - l_{P_1}(\tilde{f}), l_{P_2}(f_{eq}) - l_{P_2}(\tilde{f}), \ldots, l_{P_\mu}(f_{eq}) - l_{P_\mu}(\tilde{f}) \Big) \right\|_\infty \leq \frac{\epsilon}{2}$$

Hence, $\max \left\{ \left| l_{P_1}(f_{eq}) - l_{P_1}(\tilde{f}) \right|, \left| l_{P_2}(f_{eq}) - l_{P_2}(\tilde{f}) \right|, \ldots, \left| l_{P_\mu}(f_{eq}) - l_{P_\mu}(\tilde{f}) \right| \right\} \leq \frac{\epsilon}{2}$, so again we conclude at relation (4.18).

Therefore, since $\mu \leq m^{d_1}$ for some $d_1 > 0$, then by exhaustive search in time $T = m^{\mathcal{O}(d_1 \cdot \kappa)}$ we can find a subnetwork (the one that minimizes the latency of the maximum latency path) such that:

$$T = m^{\mathcal{O}(\frac{d_1^2 \alpha^2 \ln^{2d_2+1} m}{\epsilon^2})}$$

Hence, we come to the theorem's conclusion for p norm with $p = \infty$.

□

Next we proceed to a theorem which is a variation of the former, for different bounds of the flow's $\tilde{f}$ number of paths and corresponding time of finding them. The method is the same, though the idea differs.

Theorem 4.10. *Let $(G, 1, l)$ be an instance, with $G = (V, E)$ and linear latencies $l_e(x) = a_e \cdot x + b_e$, with rational coefficients $a_e, b_e \geq 0$ and let H^B be the Best Subnetwork of G. Let also $\mu = |\mathcal{P}| \leq m^{d_1}$ be the total number of paths and $\alpha = \max_{e \in E} \{a_e\}$. Then, there exist a flow $\tilde{f}$, to at most $k \leq \kappa$ paths P with $\tilde{f}_P = \frac{\rho}{\kappa}, \rho \in [\kappa]$, that is an $\frac{\epsilon}{2}$ - Nash flow on $G_{\tilde{f}}$ and satisfies $l_P(\tilde{f}) \leq L_{eq}(H^B) + \frac{\epsilon}{2}$, for all paths P in $G_{\tilde{f}}$. This flow can be computed in time:*

$$m^{\mathcal{O}(\frac{d_1 p \alpha^2 m^{\frac{2}{p}+2}}{\epsilon^2})} \tag{4.19}$$

for p norm with $2 \leq p < \infty$, where the number of paths, k, is less than or equal to $\kappa = \left\lceil \frac{64 p \alpha^2 m^{\frac{2}{p}+2}}{\epsilon^2} \right\rceil$,
or:

$$m^{\mathcal{O}(\frac{d_1^2 \alpha^2 m^2 \ln m}{\epsilon^2})} \tag{4.20}$$

for the $p = \infty$ norm, where the number of paths, k, is less than or equal to $\kappa = \left\lceil \frac{8 d_1 \alpha^2 m^2 \ln m}{\epsilon^2} \right\rceil$.

Proof. Again, let the Nash Equilibrium flow of the Best Subnetwork H^B be f_{eq}.

Then, there will be an $S \subseteq [\mu]$, where μ is the number of paths, such that $\forall i, j \in S : f_{eqP_i}, f_{eqP_j} > 0$ and $l_{P_i}(f_{eq}) = l_{P_j}(f_{eq}) = L_{eq}(H^B)$. If $S \subset [\mu]$, then $T = [\mu] - S$ and $\forall i \in T : f_{eqP_i} = 0$ and $l_{P_i}(f_{eq}) \neq L_{eq}(H^B)$.

Suppose that we have the following set of vectors $X = \{x_1, x_2, \ldots, x_\mu\}$. Every $x_i, i \in [\mu]$, is a vector of $m = |E|$ elements, and corresponds to the path P_i. The vector x_i's component x_{ij}, represents the presence of edge e_j at the corresponding path P_i. That is, if edge e_j belongs to the path P_i, then $x_{ij} = 1$, else $x_{ij} = 0$.

For example, if the edges e_1, e_2, e_4, e_8 belong to the path P_i, where $m = |E| = 10$, then:

$$x_i = (1, 1, 0, 1, 0, 0, 0, 1, 0, 0)$$

If f_{P_i} is the path P_i's corresponding flow, then $f_{P_i} \cdot x_i$ gives f_{P_i}'s contribution to each edge of that path. Specifically, for the example above $f_{P_i} \cdot x_i = (f_{P_i}, f_{P_i}, 0, f_{P_i}, 0, 0, 0, f_{P_i}, 0, 0)$.

Now, let's define the following vector:

$$\nu = f_{eqP_1} x_1 + f_{eqP_2} x_2 + \ldots + f_{eqP_\mu} x_\mu \tag{4.21}$$

Then, the following is valid:

$$\begin{aligned}
\nu &= f_{eqP_1} x_1 + f_{eqP_2} x_2 + \ldots + f_{eqP_\mu} x_\mu \\
&= (\sum_{P \in \mathcal{P}: e_1 \in P} f_{eq_P}, \sum_{P \in \mathcal{P}: e_2 \in P} f_{eq_P}, \ldots, \sum_{P \in \mathcal{P}: e_m \in P} f_{eq_P}) \\
&= (f_{eq_{e_1}}, f_{eq_{e_2}}, \ldots, f_{eq_{e_m}})
\end{aligned} \tag{4.22}$$

It should be obvious that $\nu \in \text{conv}(X)$.

Using McDiarmid's and Kahane's inequalities, we will find a vector $\tilde{\nu} \in \text{conv}(X)$ such that $\|\nu - \tilde{\nu}\|_p \leq \varepsilon$, where ε is a positive constant.

Indeed. $f_{eq} = (f_{eqP_1}, f_{eqP_2}, \ldots, f_{eqP_\mu})$ could be interpreted as a probability distribution over vectors $x_1, x_2, \ldots, x_\mu$. That is, under probability distribution f_{eq} vector x_i is drawn with probability f_{eqP_i}. The vector ν is the mean of this distribution. Specifically, the jth component of ν is the expected value of the random variable that takes value x_{ij} with probability f_{eqP_i}, where x_{ij} is the jth component of vector x_i. We succinctly express these component-wise equalities as follows:

$$\mathbb{E}_{v \sim f_{eq}}[v] = \nu$$

Now, let $v_1, v_2, \ldots, v_\kappa$ be κ i.i.d. draws from f_{eq}. The sample mean vector is defined to be $\frac{1}{\kappa}\sum_{i=1}^{\kappa} v_i$. Below we specify function $g : X^\kappa \to \mathbb{R}$ to quantify the p-norm distance between the sample mean vector and the ν.

$$g(v_1, v_2, \ldots, v_\kappa) := \left\| \frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i - \nu \right\|_p$$

where p is norm, with $2 \leq p < \infty$.

Also, for every $j \in [\mu]$ the following is valid:

$$\max_{x \in X} \|x\|_p \leq (\underbrace{1^p + 1^p + \ldots + 1^p}_{m=|E| \text{ times}})^{\frac{1}{p}} = m^{\frac{1}{p}}$$

We will use McDiarmid's inequality. In particular, we will establish that with positive probability the sample mean vector defined over $\kappa := \left\lceil \frac{64p\alpha^2 m^{\frac{2}{p}+2}}{\epsilon^2} \right\rceil$ draws, is $\varepsilon = \frac{\epsilon}{2m\alpha}$ close to ν in p-norm, where $\alpha = \max_{e \in E} \{a_e\}$. Hence, the stated claim is implied by the probabilistic method.

For any κ tuple $(v_1, v_2, \ldots, v_i, \ldots, v_\kappa) \in X^\kappa$ and $v_i' \in X$, we show that $|g(v_1, v_2, \ldots, v_i, \ldots, v_\kappa) - g(v_1, v_2, \ldots, v_i', \ldots, v_\kappa)|$ is no more than $\frac{2}{\kappa} \cdot m^{\frac{1}{p}}$. We can assume without loss of generality that $g(v_1, v_2, \ldots, v_i, \ldots, v_\kappa) \geq g(v_1, v_2, \ldots, v_i', \ldots, v_\kappa)$, since the other case is symmetric.

Setting $u := \frac{1}{\kappa}\sum_{j \neq i} v_j - \nu$ we have:

$$\begin{aligned}
g(v_1, v_2, \ldots, v_i, \ldots, v_\kappa) - g(v_1, v_2, \ldots, v_i', \ldots, v_\kappa) &= \left\| u + \frac{1}{\kappa} v_i \right\|_p - \left\| u + \frac{1}{\kappa} v_i' \right\|_p \\
&\leq \|u\|_p + \frac{1}{\kappa}\|v_i\|_p - \|u\|_p + \frac{1}{\kappa}\|v_i'\|_p \\
&\leq \frac{1}{\kappa}\|v_i\|_p + \frac{1}{\kappa}\|v_i'\|_p \\
&\leq \frac{1}{\kappa}\max\{\|x\|_p\} + \frac{1}{\kappa}\max\{\|x\|_p\} \\
&= \frac{2}{\kappa}\max\{\|x\|_p\} \\
&\leq \frac{2}{\kappa} \cdot m^{\frac{1}{p}}
\end{aligned}$$

Given that g satisfies $|g(v_1, v_2, \ldots, v_i, \ldots, v_\kappa) - g(v_1, v_2, \ldots, v_i', \ldots, v_\kappa)| \leq \frac{2}{\kappa} m^{\frac{1}{p}}$, we can apply Mc-Diarmid's inequality, with $c_i = \frac{2}{\kappa} m^{\frac{1}{p}}$ for all $i \in [\kappa]$, to obtain:

$$\mathbb{P}(|g - \mathbb{E}[g]| \geq t) \leq 2e^{\frac{-\kappa t^2}{2m^{\frac{2}{p}}}} \tag{4.23}$$

By using the same approach as in the proof of theorem 4.9 we can prove that:

$$\mathbb{E}[g] \leq 2\mathbb{E}_{v_i,r_i} \left\| \sum_{i=1}^{\kappa} r_i \frac{v_i}{\kappa} \right\|_p \tag{4.24}$$

where $r_1, r_2, \ldots, r_\kappa$ be a sequence of i.i.d. Rademacher ± 1 random variables.

Now, by applying Kahane's inequality with $u_i = \frac{v_i}{\kappa}$ we obtain:

$$\begin{aligned}
\mathbb{E}_{v_i,r_i} \left\| \sum_{i=1}^{\kappa} r_i \frac{v_i}{\kappa} \right\|_p &= \mathbb{E}_{v_i} \left[\mathbb{E}_{r_i} \left[\left\| \sum_{i=1}^{\kappa} r_i \frac{v_i}{\kappa} \right\|_p \middle| v_1 \ldots v_\kappa \right] \right] \\
&\leq \mathbb{E}_{v_i} \left[\sqrt{p} \left(\sum_{i=1}^{\kappa} \left\| \frac{v_i}{\kappa} \right\|_p^2 \right)^{\frac{1}{2}} \right] \\
&\leq \mathbb{E}_{v_i} \left[\sqrt{p} \left(\sum_{i=1}^{\kappa} \frac{\left(\max_{x \in X} \|x\|_p \right)^2}{\kappa^2} \right)^{\frac{1}{2}} \right] \\
&\leq \sqrt{p} \frac{m^{\frac{1}{p}}}{\sqrt{\kappa}}
\end{aligned} \tag{4.25}$$

By using relations (4.24) and (4.25) we have that:

$$\mathbb{E}[g] \leq 2\sqrt{p} \frac{m^{\frac{1}{p}}}{\sqrt{\kappa}}$$

Thus, for sample size:

$$\kappa \geq 16 \cdot \frac{p m^{\frac{2}{p}}}{\varepsilon^2} \tag{4.26}$$

we have $\mathbb{E}[g] \leq \frac{\varepsilon}{2}$.

Setting $t = \frac{\varepsilon}{2}$ in relation (4.23), and by following the same guidelines of theorem 4.9, we may prove that $\mathbb{P}(g \geq \varepsilon) \leq 2e^{-2}$ or equivalently that $\mathbb{P}\Big(\left\| \frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i - \nu \right\|_p \geq \varepsilon \Big) \leq 2e^{-2}$.

Therefore, with positive probability:

$$\left\| \frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i - \nu \right\|_p \leq \varepsilon \tag{4.27}$$

The latter means that we have found a vector $\tilde{\nu} := \frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i$, such that $\tilde{\nu} \in \text{conv}(X)$ and $\|\nu - \tilde{\nu}\|_p \leq \varepsilon$.

Since $v_1, v_2, \ldots, v_\kappa$ are κ i.i.d. draws from f_{eq}, vector $\tilde{\nu}$ could be expressed as:

$$\tilde{\nu} = x_1 \cdot \tilde{f}_{P_1} + x_2 \cdot \tilde{f}_{P_2} + \ldots + x_\mu \cdot \tilde{f}_{P_\mu} \tag{4.28}$$
$$= \frac{\kappa_1}{\kappa} \cdot x_{\kappa_1} + \frac{\kappa_2}{\kappa} \cdot x_{\kappa_2} + \ldots + \frac{\kappa_k}{\kappa} \cdot x_{\kappa_k} \tag{4.29}$$

where $\kappa_1 + \kappa_2 + \ldots + \kappa_k = \kappa$ and $1 \leq \kappa_1, \kappa_2, \ldots, \kappa_k \leq \kappa$.
Also, vector $\tilde{\nu}$ could be expressed in a form similar to (4.22) as follows:

$$\tilde{\nu} = (\tilde{f}_{e_1}, \tilde{f}_{e_2}, \ldots, \tilde{f}_{e_m}) \tag{4.30}$$

Taking into account relations (4.22) and (4.30), relation (4.27) could be analyzed as follows:

$$\begin{aligned}
&\|\nu - \tilde{\nu}\|_p \leq \varepsilon \Rightarrow \\
&\left\|\left(f_{eq_{e_1}} - \tilde{f}_{e_1}, f_{eq_{e_2}} - \tilde{f}_{e_2}, \ldots, f_{eq_{e_m}} - \tilde{f}_{e_m}\right)\right\|_p \leq \varepsilon \Rightarrow \\
&\sum_{i=1}^{m} \left|f_{eq_{e_i}} - \tilde{f}_{e_i}\right|^p \leq \varepsilon^p \Rightarrow \\
&\left|f_{eq_{e_i}} - \tilde{f}_{e_i}\right| \leq \varepsilon, \forall i \in [m]
\end{aligned} \tag{4.31}$$

Now, let's consider a path P_i with $f_{eq_{P_i}} > 0$. Suppose now that P_i's edges are $e_{z_1}, e_{z_2}, \ldots, e_{z_M}$. Then by relation (4.31) we have:

$$\begin{aligned}
\left|L_{eq}(H^B) - l_{P_i}(\tilde{f})\right| &= \left|a_{e_{z_1}}(f_{eq_{e_{z_1}}} - \tilde{f}_{e_{z_1}}) + \ldots + a_{e_{z_M}}(f_{eq_{e_{z_M}}} - \tilde{f}_{e_{z_M}})\right| \\
&\leq \alpha \sum_{i=1}^{M} \left|f_{eq_{e_{z_i}}} - \tilde{f}_{e_{z_i}}\right| \leq \alpha M \varepsilon \leq \alpha m \varepsilon
\end{aligned}$$

Thus, $\forall i \in S$, $\left|L_{eq}(H^B) - l_{P_i}(\tilde{f})\right| \leq \alpha m \varepsilon$. So, if we choose $\varepsilon = \frac{\epsilon}{2\alpha m}$ then:

$$\left|L_{eq}(H^B) - l_{P_i}(\tilde{f})\right| \leq \frac{\epsilon}{2}$$

or better:

$$l_{P_i}(\tilde{f}) \leq L_{eq}(H^B) + \frac{\epsilon}{2} \tag{4.32}$$

Now, since $\varepsilon = \frac{\epsilon}{2\alpha m}$, relation (4.26) should be:

$$\begin{aligned}
\kappa &\geq 16 \cdot \frac{pm^{\frac{2}{p}}}{\varepsilon^2} \\
&\geq 64 \cdot \frac{p\alpha^2 m^{\frac{2}{p}+2}}{\epsilon^2}
\end{aligned} \tag{4.33}$$

Thus, it suffices to consider only a $\kappa := \left\lceil \frac{64p\alpha^2 m^{\frac{2}{p}+2}}{\epsilon^2} \right\rceil$ number of paths.

Therefore, since $\mu \leq m^{d_1}$ for some $d_1 > 0$, then by exhaustive search in time $T = m^{\mathcal{O}(d_1 \cdot \kappa)}$ we can find a subnetwork (the one that minimizes the latency of the maximum latency path) such that:

$$T = m^{\mathcal{O}(\frac{d_1 p\alpha^2 m^{\frac{2}{p}+2}}{\epsilon^2})}$$

Hence, we come to the theorem's conclusion for p norm with $2 \leq p < \infty$.

Now, for the $p = \infty$ case, we are going to use theorem 4.8.

More specifically, $f_{eq} = (f_{eqP_1}, f_{eqP_2}, \ldots, f_{eqP_\mu})$ could be interpreted again as a probability distribution over vectors $x_1, x_2, \ldots, x_\mu$.

Then, $\gamma = \max_{x \in X} \|x\|_\infty = 1$.

So, since $\mu \leq m^{d_1}$, for $\kappa \geq \frac{2d_1 \ln m}{\varepsilon^2}$ we have that:

$$\|\tilde{\nu} - \nu\|_\infty \leq \varepsilon$$

where:

$$\begin{aligned}
\nu &= x_1 \cdot f_{eqP_1} + x_2 \cdot f_{eqP_2} + \ldots + x_\mu \cdot f_{eqP_\mu} \\
\tilde{\nu} &= x_1 \cdot \tilde{f}_{P_1} + x_2 \cdot \tilde{f}_{P_2} + \ldots + x_\mu \cdot \tilde{f}_{P_\mu} \\
&= \frac{\kappa_1}{\kappa} \cdot x_{\kappa_1} + \frac{\kappa_2}{\kappa} \cdot x_{\kappa_2} + \ldots + \frac{\kappa_k}{\kappa} \cdot x_{\kappa_k}
\end{aligned}$$

and $\kappa_1 + \kappa_2 + \ldots + \kappa_k = \kappa$, $1 \leq \kappa_1, \kappa_2, \ldots, \kappa_k \leq \kappa$.

Also, as previously stated:

$$\begin{aligned}
\nu &= (f_{eqe_1}, f_{eqe_2}, \ldots, f_{eqe_m}) \\
\tilde{\nu} &= (\tilde{f}_{e_1}, \tilde{f}_{e_2}, \ldots, \tilde{f}_{e_m})
\end{aligned}$$

Obviously $\nu, \tilde{\nu} \in \mathrm{conv}(X)$.

Thus:

$$\begin{aligned}
&\|\nu - \tilde{\nu}\|_\infty \leq \varepsilon \Rightarrow \\
&\left\| \left(f_{eqe_1} - \tilde{f}_{e_1}, f_{eqe_2} - \tilde{f}_{e_2}, \ldots, f_{eqe_m} - \tilde{f}_{e_m} \right) \right\|_\infty \leq \varepsilon \Rightarrow \\
&\max \left\{ \left| f_{eqe_i} - \tilde{f}_{e_i} \right|, i \in [m] \right\} \leq \varepsilon \Rightarrow \\
&\left| f_{eqe_i} - \tilde{f}_{e_i} \right| \leq \varepsilon, \forall i \in [m]
\end{aligned}$$

Now, by doing the same work as previously, $\forall i \in S, \left| L_{eq}(H^B) - l_{P_i}(\tilde{f}) \right| \leq \alpha m \varepsilon$. So, if we choose $\varepsilon = \frac{\epsilon}{2\alpha m}$ then:

$$l_{P_i}(\tilde{f}) \leq L_{eq}(H^B) + \frac{\epsilon}{2}$$

Again, since $\varepsilon = \frac{\epsilon}{2\alpha m}$, we should have:

$$\begin{aligned}\kappa &\geq \frac{2d_1 \ln m}{\varepsilon^2} \\ &\geq \frac{8d_1\alpha^2 m^2 \ln m}{\epsilon^2}\end{aligned}$$

Thus, it suffices to consider only a $\kappa := \left\lceil \frac{8d_1\alpha^2 m^2 \ln m}{\epsilon^2} \right\rceil$ number of paths.

Therefore, since $\mu \leq m^{d_1}$ for some $d_1 > 0$, then by exhaustive search in time $T = m^{\mathcal{O}(d_1 \cdot \kappa)}$ we can find a subnetwork (the one that minimizes the latency of the maximum latency path) such that:

$$T = m^{\mathcal{O}(\frac{d_1^2\alpha^2 m^2 \ln m}{\epsilon^2})}$$

Hence, we come to the theorem's conclusion for p norm with $p = \infty$.

□

The following lemma makes a comparison of the relevant computational times of theorem 4.9's $p = \infty$ and $2 \leq p < \infty$ norm. It demonstrates that the computational time of the $p = \infty$ norm is always less than the same theorem's computational time of the $2 \leq p < \infty$ norm:

Lemma 4.11. *Let $T^{4.9}_{p=\infty}$ and $T^{4.9}_{2\leq p<\infty}$ be the corresponding computational times of theorem 4.9's $p = \infty$ and $2 \leq p < \infty$ norm. Then, by assuming that the constant terms of the formulas $\mathcal{O}(\cdot)$ are all equal:*

$$T^{4.9}_{p=\infty} \leq T^{4.9}_{2\leq p<\infty}$$

Proof.

$$\begin{aligned}&T^{4.9}_{p=\infty} \leq T^{4.9}_{2\leq p<\infty} \Leftrightarrow \\ &m^{\mathcal{O}(\frac{d_1^2\alpha^2 \ln^{2d_2+1} m}{\epsilon^2})} \leq m^{\mathcal{O}(\frac{d_1 p\alpha^2 m^{\frac{2d_1}{p}} \ln^{2d_2} m}{\epsilon^2})} \Leftrightarrow \\ &d_1 \ln m \leq p m^{\frac{2d_1}{p}} \Leftrightarrow \\ &\frac{d_1}{p} \ln m \leq m^{2\frac{d_1}{p}}\end{aligned}$$

Since the above inequality, for a constant total number of edges $m > 1$, depends on the changeable factor $\frac{d_1}{p}$, it suffices to prove that:

$$f(x) \leq g(x), \forall x > 0$$

where $f(x) := x \ln m, g(x) := m^{2x}$, where variable $x := \frac{d_1}{p} > 0$.

At first it is simple to prove that when $x \to 0^+$, then $f(x) \leq g(x)$. So:

$$\lim_{x \to 0^+} f(x) = 0 \leq 1 = \lim_{x \to 0^+} g(x)$$

Then we show that both f, g are strictly increasing functions:

$$f'(x) = \ln m > 0$$

and

$$g'(x) = 2m^{2x} \ln m > 0$$

Moreover:

$$\begin{aligned}
&f'(x) = \ln m \leq 2m^{2x} \ln m = g'(x) \Rightarrow \\
&\lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \leq \lim_{h \to 0} \frac{g(x+h) - g(x)}{h} \Rightarrow \\
&\lim_{h \to 0} \frac{g(x) - f(x)}{h} \leq \lim_{h \to 0} \frac{g(x+h) - f(x+h)}{h} \Rightarrow \\
&g(x) - f(x) \leq g(x+h) - f(x+h) \qquad (4.34)
\end{aligned}$$

for $h > 0$ with $h \to 0$.

Now, since both f, g are strictly increasing and since for the 'base' case $(x \to 0^+)$, $f(x) \leq g(x)$, then from relation (4.34) we have:

$$f(x+h) \leq g(x+h) - g(x) + f(x) \leq g(x+h) - g(x) + g(x) \leq g(x+h)$$

Hence, for every $x > 0$, $f(x) \leq g(x)$. □

The following lemma makes a comparison of the relevant computational times of theorem 4.10's $p = \infty$ and $2 \leq p < \infty$ norm:

Lemma 4.12. *Let $T^{4.10}_{p=\infty}$ and $T^{4.10}_{2 \leq p < \infty}$ be the corresponding computational times of theorem 4.10's $p = \infty$ and $2 \leq p < \infty$ norm. Then, by assuming that the constant terms of the formulas $\mathcal{O}(\cdot)$ are all equal:*

- *If $\frac{d_1}{p} \leq \frac{m^{\frac{2}{p}}}{\ln m}$ (or less strictly, $\frac{d_1}{p} \leq m^{\frac{2}{p}-1}$) then:*

$$T^{4.10}_{p=\infty} \leq T^{4.10}_{2 \leq p < \infty}$$

- *If $\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}}}{\ln m}$ (or less strictly, $\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}+1}}{m-1}$) then:*

$$T^{4.10}_{p=\infty} \geq T^{4.10}_{2 \leq p < \infty}$$

The less strict constraints may be used when building an instance (they seem simpler in use).

Proof. The function $\ln(\cdot)$ is strictly increasing (since the logarithm's base is greater than 1). So, the following inequalities are valid:

$$\ln x < \ln(x+1) \leq (x+1) - 1 = x \tag{4.35}$$

Also,

$$\frac{x-1}{x} \leq \ln x \tag{4.36}$$

Now, by using inequality (4.35) for the first case, and inequality (4.36) for the second case we have lemma's conclusion.

□

Having in mind lemma 4.11 above, the following lemma makes a comparison between the minimum computational time of theorem 4.9, $T^{4.9}_{p=\infty}$ specifically, and 4.10's $T^{4.10}_{p=\infty}$ and $T^{4.10}_{2\leq p<\infty}$ computational times.

Lemma 4.13. *Let $T^{4.10}_{p=\infty}$ and $T^{4.10}_{2\leq p<\infty}$ be the corresponding computational times of theorem 4.10's $p = \infty$ and $2 \leq p < \infty$ norm. Let also $T^{4.9}_{p=\infty}$ be the computational time of theorem 4.9's $p = \infty$ norm. Then, by assuming that the constant terms of the formulas $\mathcal{O}(\cdot)$ are all equal:*

- *If $\ln m \geq m^{\frac{1}{d_2}}$ then:*
$$T^{4.9}_{p=\infty} \geq T^{4.10}_{p=\infty}$$
- *If $\ln m \leq m^{\frac{1}{d_2}}$ then:*
$$T^{4.9}_{p=\infty} \leq T^{4.10}_{p=\infty}$$
- *If $\frac{d_1}{p} \leq \frac{m^{\frac{2}{p}+2}}{\ln^{2d_2+1} m}$ (or less strictly, $\frac{d_1}{p} \leq m^{\frac{2}{p}-2d_2+1}$) then:*
$$T^{4.9}_{p=\infty} \leq T^{4.10}_{2\leq p<\infty}$$
- *If $\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}+2}}{\ln^{2d_2+1} m}$ (or less strictly, $\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}+2d_2+3}}{(m-1)^{2d_2+1}}$) then:*
$$T^{4.9}_{p=\infty} \geq T^{4.10}_{2\leq p<\infty}$$

The less strict constraints may be used when building an instance (they seem simpler in use).

Proof. The proof is simple. For the first two cases the conclusion follows easily. For the third and fourth case we use inequalities (4.35) and (4.36) respectively. The conclusion follows.

□

The next lemma takes into account theorem 4.4. It makes a comparison between theorem 4.4's computational time, $T^{4.4}$ specifically, and 4.10's $T^{4.10}_{p=\infty}$ and $T^{4.10}_{2\leq p<\infty}$ computational times.

Lemma 4.14. *Let $T^{4.10}_{p=\infty}$ and $T^{4.10}_{2\leq p<\infty}$ be the corresponding computational times of theorem 4.10's $p=\infty$ and $2 \leq p < \infty$ norm. Let also $T^{4.4}$ be the computational time of theorem 4.4. Then, by assuming that the constant terms of the formulas $\mathcal{O}(\cdot)$ are all equal:*

- *If $p \geq \frac{\ln^{2d_2+1}(2m)}{m^{\frac{2}{p}+2}}$ (or less strictly $p \geq \frac{2^{2d_2+1}}{m^{\frac{2}{p}-2d_2+1}}$) then:*

$$T^{4.4} \leq T^{4.10}_{2\leq p<\infty}$$

- *If $p \leq \frac{\ln^{2d_2+1}(2m)}{m^{\frac{2}{p}+2}}$ (or less strictly $p \leq \frac{1}{2^{d_2+1}} \cdot \frac{(2m-1)^{2d_2+1}}{m^{\frac{2}{p}+2d_2+3}}$) then:*

$$T^{4.4} \geq T^{4.10}_{2\leq p<\infty}$$

- *If $d_1 \geq \frac{\ln^{2d_2+1}(2m)}{m^2 \ln m}$ (or less strictly $d_1 \geq 2^{2d_2+1} \cdot \frac{m^{2d_2}}{m-1}$) then:*

$$T^{4.4} \leq T^{4.10}_{p=\infty}$$

- *If $d_1 \leq \frac{\ln^{2d_2+1}(2m)}{m^2 \ln m}$ (or less strictly $d_1 \leq \frac{1}{2^{2d_2+1}} \cdot \frac{(2m-1)^{2d_2+1}}{m^{2d_2+4}}$) then:*

$$T^{4.4} \geq T^{4.10}_{p=\infty}$$

The less strict constraints may be used when building an instance (they seem simpler in use).

Proof. The proof is simple too. For the first and second case we use inequalities (4.35) and (4.36) respectively. For the last two cases we use both the aforementioned inequalities. The conclusion follows easily.

□

The next lemma compares theorem 4.4's computational time, $T^{4.4}$ specifically, with the computational time of theorem 4.9's $p=\infty$ norm, $T^{4.9}_{p=\infty}$, which by lemma 4.11 is less than theorem 4.9's $2 \leq p < \infty$ norm.

Lemma 4.15. *Let $T^{4.4}$ be the computational time of theorem 4.4 and $T^{4.9}_{p=\infty}$ be the corresponding computational time of theorem 4.9's $p=\infty$ norm. Then, by assuming that the constant terms of the formulas $\mathcal{O}(\cdot)$ are all equal:*

- *If $d_1 \leq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$ then:*

$$T^{4.9}_{p=\infty} \leq T^{4.4}$$

- *If* $d_1 \geq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$ *then:*

$$T^{4.9}_{p=\infty} \geq T^{4.4}$$

Proof. The proof is straightforward.

□

Now, we are ready to present the following theorem, which makes a comparison of the relevant computational times of theorems 4.4, 4.9, and 4.10. We are going to use lemmas 4.11 to 4.15:

Theorem 4.16. *Let $(G, 1, l)$ be an instance, with $G = (V, E)$ and linear latencies $l_e(x) = a_e \cdot x + b_e$, with rational coefficients $a_e, b_e \geq 0$ and let H^B be the Best Subnetwork of G. Let also $\mu = |\mathcal{P}| \leq m^{d_1}$ be the total number of paths, $|P| \leq \ln^{d_2} m$ for all $P \in \mathcal{P}$ and $\alpha = \max_{e \in E}\{a_e\}$. Then, there exist a flow $\tilde{f}$, to at most $k \leq \kappa$ paths P with $\tilde{f}_P = \frac{\rho}{\kappa}, \rho \in [\kappa]$, that is an $\frac{\epsilon}{2}$ - Nash flow on $G_{\tilde{f}}$ and satisfies $l_P(\tilde{f}) \leq L_{eq}(H^B) + \frac{\epsilon}{2}$, for all paths P in $G_{\tilde{f}}$. This flow can be computed in the following <u>minimum</u> computational times:*

- $m^{\mathcal{O}\left(\frac{d_1^2 \alpha^2 \ln^{2d_2+1} m}{\epsilon^2}\right)}$ *where the number of paths, k, is less than or equal to $\kappa = \left\lceil \frac{8 d_1 \alpha^2 \ln^{2d_2+1} m}{\epsilon^2} \right\rceil$,*

  under the following strict constraints:

$$\frac{m^{\frac{2}{p}}}{\ln m} \leq \frac{d_1}{p} \leq \frac{m^{\frac{2}{p}+2}}{\ln^{2d_2+1} m}$$
$$\ln m \leq m^{\frac{1}{d_2}}$$
$$d_1 \leq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

  or less strict constraints:

$$\frac{m^{\frac{2}{p}+1}}{m-1} \leq \frac{d_1}{p} \leq m^{\frac{2}{p}-2d_2+1}$$
$$m^{2d_2} \leq m-1$$
$$d_1 \leq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

  <u>*or*</u>

  under the following strict constraints:

$$\frac{d_1}{p} \leq \frac{m^{\frac{2}{p}}}{\ln m}$$
$$\ln m \leq m^{\frac{1}{d_2}}$$
$$d_1 \leq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

or less strict constraints:

$$\frac{d_1}{p} \leq m^{\frac{2}{p}-1}$$
$$\ln m \leq m^{\frac{1}{d_2}}$$
$$d_1 \leq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

- $m^{\mathcal{O}\left(\frac{d_1 \alpha^2 \ln^{2d_2+1}(2m)}{\epsilon^2}\right)}$ *where the number of paths, k, is less than or equal to $\kappa = \left\lceil \frac{2\alpha^2 \ln^{2d_2+1}(2m)}{\epsilon^2} \right\rceil$,*
  under the following strict constraints:

$$\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}}}{\ln m}$$
$$p \geq \frac{\ln^{2d_2+1}(2m)}{m^{\frac{2}{p}+2}}$$
$$d_1 \geq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

or less strict constraints:

$$\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}+1}}{m-1}$$
$$p \geq \frac{2^{2d_2+1}}{m^{\frac{2}{p}-2d_2+1}}$$
$$d_1 \geq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

or

under the following strict constraints:

$$\frac{d_1}{p} \leq \frac{m^{\frac{2}{p}}}{\ln m}$$
$$d_1 \geq \max\left\{\frac{\ln^{2d_2+1}(2m)}{m^2 \ln m}, \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}\right\}$$

or less strict constraints:

$$\frac{d_1}{p} \leq m^{\frac{2}{p}-1}$$
$$d_1 \geq \max\left\{2^{2d_2+1} \cdot \frac{m^{2d_2}}{m-1}, \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}\right\}$$

- $m^{\mathcal{O}(\frac{d_1 p\alpha^2 m^{\frac{2}{p}+2}}{\epsilon^2})}$ *where the number of paths, k, is less than or equal to* $\kappa = \left\lceil \frac{64p\alpha^2 m^{\frac{2}{p}+2}}{\epsilon^2} \right\rceil$,
  under the following strict constraints:

$$p \leq \frac{\ln^{2d_2+1}(2m)}{m^{\frac{2}{p}+2}}$$
$$\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}+2}}{\ln^{2d_2+1} m}$$

  or less strict constraints:

$$p \leq \frac{1}{2^{d_2+1}} \cdot \frac{(2m-1)^{2d_2+1}}{m^{\frac{2}{p}+2d_2+3}}$$
$$\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}+2d_2+3}}{(m-1)^{2d_2+1}}$$

- $m^{\mathcal{O}(\frac{d_1^2 \alpha^2 m^2 \ln m}{\epsilon^2})}$ *where the number of paths, k, is less than or equal to* $\kappa = \left\lceil \frac{8d_1\alpha^2 m^2 \ln m}{\epsilon^2} \right\rceil$,
  under the following strict constraints:

$$d_1 \leq \frac{\ln^{2d_2+1}(2m)}{m^2 \ln m}$$
$$\ln m \geq m^{\frac{1}{d_2}}$$

  or less strict constraints:

$$d_1 \leq \frac{1}{2^{2d_2+1}} \cdot \frac{(2m-1)^{2d_2+1}}{m^{2d_2+4}})$$
$$\ln m \geq m^{\frac{1}{d_2}}$$

  where p is norm with $2 \leq p < \infty$.
  The less strict constraints may be used when building an instance (they seem simpler in use).

Proof. By lemma 4.11 we have proved that $T^{0.5}_{p=\infty} \leq T^{0.5}_{2\leq p<\infty}$. Thus, it is sufficient to compare the computational times of theorems 4.4 and 4.10, only with the $T^{4.9}_{p=\infty}$ case, taking into account the constraints of lemmas 4.11 to 4.15.

- First case, $T^{4.9}_{p=\infty} = \min\{T^{4.10}, T^{4.4}\}$:

◇ $T^{4.10}_{2\le p<\infty} \le T^{4.10}_{p=\infty}$:

This means that $T^{4.9}_{p=\infty} = \min\{T^{4.10}_{2\le p<\infty}, T^{4.4}\}$. Then, we have shown that $\frac{d_1}{p} \ge \frac{m^{\frac{2}{p}}}{\ln m}$ (or less strictly, $\frac{d_1}{p} \ge \frac{m^{\frac{2}{p}+1}}{m-1}$), $\frac{d_1}{p} \le \frac{m^{\frac{2}{p}+2}}{\ln^{2d_2+1} m}$ (or less strictly, $\frac{d_1}{p} \le m^{\frac{2}{p}-2d_2+1}$), $d_1 \le \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$.
By combining the first two strict inequalities we conclude that:

$$\frac{m^{\frac{2}{p}}}{\ln m} \le \frac{d_1}{p} \le \frac{m^{\frac{2}{p}+2}}{\ln^{2d_2+1} m} \Rightarrow$$
$$\frac{m^{\frac{2}{p}}}{\ln m} \le \frac{m^{\frac{2}{p}+2}}{\ln^{2d_2+1} m} \Rightarrow$$
$$\ln m \le m^{\frac{1}{d_2}}$$

Also, by combining the first two less strict inequalities we conclude that:

$$\frac{m^{\frac{2}{p}+1}}{m-1} \le \frac{d_1}{p} \le m^{\frac{2}{p}-2d_2+1} \Rightarrow$$
$$\frac{m^{\frac{2}{p}+1}}{m-1} \le m^{\frac{2}{p}-2d_2+1} \Rightarrow$$
$$m^{2d_2} \le m-1$$

So, for this case we conclude to the following restrictions:

$$\frac{m^{\frac{2}{p}}}{\ln m} \le \frac{d_1}{p} \le \frac{m^{\frac{2}{p}+2}}{\ln^{2d_2+1} m}$$
$$\ln m \le m^{\frac{1}{d_2}}$$
$$d_1 \le \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

or less strictly:

$$\frac{m^{\frac{2}{p}+1}}{m-1} \le \frac{d_1}{p} \le m^{\frac{2}{p}-2d_2+1}$$
$$m^{2d_2} \le m-1$$
$$d_1 \le \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

◇ $T^{4.10}_{p=\infty} \le T^{4.10}_{2\le p<\infty}$:

This means that $T^{4.9}_{p=\infty} = \min\{T^{4.10}_{p=\infty}, T^{4.4}\}$. Then, we have shown that $\frac{d_1}{p} \le \frac{m^{\frac{2}{p}}}{\ln m}$ (or less strictly, $\frac{d_1}{p} \le m^{\frac{2}{p}-1}$), $\ln m \le m^{\frac{1}{d_2}}$, $d_1 \le \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$.

So, for this case we conclude to the following restrictions:

$$\frac{d_1}{p} \leq \frac{m^{\frac{2}{p}}}{\ln m}$$
$$\ln m \leq m^{\frac{1}{d_2}}$$
$$d_1 \leq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

or less strictly:

$$\frac{d_1}{p} \leq m^{\frac{2}{p}-1}$$
$$\ln m \leq m^{\frac{1}{d_2}}$$
$$d_1 \leq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

- Second case, $T^{4.4} = \min\{T^{4.10}, T^{4.9}_{p=\infty}\}$:

  ◇ $T^{4.10}_{2\leq p<\infty} \leq T^{4.10}_{p=\infty}$:

  This means that $T^{4.4} = \min\{T^{4.10}_{2\leq p<\infty}, T^{4.9}_{p=\infty}\}$. Then, we have shown that $\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}}}{\ln m}$ (or less strictly, $\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}+1}}{m-1}$), $p \geq \frac{\ln^{2d_2+1}(2m)}{m^{\frac{2}{p}+2}}$ (or less strictly $p \geq \frac{2^{2d_2+1}}{m^{\frac{2}{p}-2d_2+1}}$), $d_1 \geq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$.
  So, for this case we conclude to the following restrictions:

$$\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}}}{\ln m}$$
$$p \geq \frac{\ln^{2d_2+1}(2m)}{m^{\frac{2}{p}+2}}$$
$$d_1 \geq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

  or less strictly:

$$\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}+1}}{m-1}$$
$$p \geq \frac{2^{2d_2+1}}{m^{\frac{2}{p}-2d_2+1}}$$
$$d_1 \geq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

◇ $T^{4.10}_{p=\infty} \leq T^{4.10}_{2\leq p<\infty}$:

This means that $T^{4.4} = \min\{T^{4.10}_{p=\infty}, T^{4.9}_{p=\infty}\}$. Then, we have shown that $\frac{d_1}{p} \leq \frac{m^{\frac{2}{p}}}{\ln m}$ (or less strictly, $\frac{d_1}{p} \leq m^{\frac{2}{p}-1}$), $d_1 \geq \frac{\ln^{2d_2+1}(2m)}{m^2 \ln m}$ (or less strictly $d_1 \geq 2^{2d_2+1} \cdot \frac{m^{2d_2}}{m-1}$), $d_1 \geq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$.
So, for this case we conclude to the following restrictions:

$$\frac{d_1}{p} \leq \frac{m^{\frac{2}{p}}}{\ln m}$$
$$d_1 \geq \max\left\{\frac{\ln^{2d_2+1}(2m)}{m^2 \ln m}, \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}\right\}$$

or less strictly:

$$\frac{d_1}{p} \leq m^{\frac{2}{p}-1}$$
$$d_1 \geq \max\left\{2^{2d_2+1} \cdot \frac{m^{2d_2}}{m-1}, \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}\right\}$$

- Third case, $T^{4.10}_{2\leq p<\infty} = \min\{T^{4.4}, T^{4.9}_{p=\infty}\}$:

We have shown that $p \leq \frac{\ln^{2d_2+1}(2m)}{m^{\frac{2}{p}+2}}$ (or less strictly $p \leq \frac{1}{2^{d_2+1}} \cdot \frac{(2m-1)^{2d_2+1}}{m^{\frac{2}{p}+2d_2+3}}$), $\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}+2}}{\ln^{2d_2+1} m}$ (or less strictly, $\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}+2d_2+3}}{(m-1)^{2d_2+1}}$).
So, for this case we conclude to the following restrictions:

$$p \leq \frac{\ln^{2d_2+1}(2m)}{m^{\frac{2}{p}+2}}$$
$$\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}+2}}{\ln^{2d_2+1} m}$$

or less strictly:

$$p \leq \frac{1}{2^{d_2+1}} \cdot \frac{(2m-1)^{2d_2+1}}{m^{\frac{2}{p}+2d_2+3}}$$
$$\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}+2d_2+3}}{(m-1)^{2d_2+1}}$$

- Fourth case, $T^{4.10}_{p=\infty} = \min\{T^{4.4}, T^{4.9}_{p=\infty}\}$:

We have shown that $d_1 \leq \frac{\ln^{2d_2+1}(2m)}{m^2 \ln m}$ (or less strictly $d_1 \leq \frac{1}{2^{2d_2+1}} \cdot \frac{(2m-1)^{2d_2+1}}{m^{2d_2+4}}$), $\ln m \geq m^{\frac{1}{d_2}}$.

So, for this case we conclude to the following restrictions:

$$d_1 \leq \frac{\ln^{2d_2+1}(2m)}{m^2 \ln m}$$
$$\ln m \geq m^{\frac{1}{d_2}}$$

or less strictly:

$$d_1 \leq \frac{1}{2^{2d_2+1}} \cdot \frac{(2m-1)^{2d_2+1}}{m^{2d_2+4}})$$
$$\ln m \geq m^{\frac{1}{d_2}}$$

□

Now, it is relatively easy to introduce traffic rates $r > 0$ (not only for the special case $r = 1$) to each proof of lemma 4.3 and of theorems 4.4, 4.9, 4.10, by applying new flows $f_P := \frac{f_P}{r}$ for every path $P \in \mathcal{P}$, and / or $f_e := \frac{f_e}{r}$ for every edge $e \in E$. Specifically, $\epsilon := \frac{\epsilon}{r}$ and thus both the bounds κ and the computational times should be multiplied by a factor of r^2.

Then, the following comparison theorem is valid:

Theorem 4.17. *First Comparison Theorem: Let (G, r, l) be an instance, $r > 0$, with $G = (V, E)$ and linear latencies $l_e(x) = a_e \cdot x + b_e$, with rational coefficients $a_e, b_e \geq 0$ and let H^B be the Best Subnetwork of G. Let also $\mu = |\mathcal{P}| \leq m^{d_1}$ be the total number of paths, $|P| \leq \ln^{d_2} m$ for all $P \in \mathcal{P}$ and $\alpha = \max_{e \in E}\{a_e\}$. Then, there exists a flow $\tilde{f}$, to at most $k \leq \kappa$ paths P with $\tilde{f}_P = \frac{\rho}{\kappa}, \rho \in [\kappa]$, that is an $\frac{\epsilon}{2}$ - Nash flow on $G_{\tilde{f}}$ and satisfies $l_P(\tilde{f}) \leq L_{eq}(H^B) + \frac{\epsilon}{2}$, for all paths P in $G_{\tilde{f}}$. This flow can be computed in the following minimum computational times:*

- $m^{\mathcal{O}(\frac{d_1^2 \alpha^2 r^2 \ln^{2d_2+1} m}{\epsilon^2})}$ *where the number of paths, k, is less than or equal to $\kappa = \left\lceil \frac{8 d_1 \alpha^2 r^2 \ln^{2d_2+1} m}{\epsilon^2} \right\rceil$,*

  under the following strict constraints:

$$\frac{m^{\frac{2}{p}}}{\ln m} \leq \frac{d_1}{p} \leq \frac{m^{\frac{2}{p}+2}}{\ln^{2d_2+1} m}$$
$$\ln m \leq m^{\frac{1}{d_2}}$$
$$d_1 \leq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

  or less strict constraints:

$$\frac{m^{\frac{2}{p}+1}}{m-1} \leq \frac{d_1}{p} \leq m^{\frac{2}{p}-2d_2+1}$$
$$m^{2d_2} \leq m - 1$$
$$d_1 \leq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

or

under the following strict constraints:

$$\frac{d_1}{p} \leq \frac{m^{\frac{2}{p}}}{\ln m}$$
$$\ln m \leq m^{\frac{1}{d_2}}$$
$$d_1 \leq \Big(\frac{\ln(2m)}{\ln m}\Big)^{2d_2+1}$$

or less strict constraints:

$$\frac{d_1}{p} \leq m^{\frac{2}{p}-1}$$
$$\ln m \leq m^{\frac{1}{d_2}}$$
$$d_1 \leq \Big(\frac{\ln(2m)}{\ln m}\Big)^{2d_2+1}$$

- $m^{\mathcal{O}\left(\frac{d_1\alpha^2 r^2 \ln^{2d_2+1}(2m)}{\epsilon^2}\right)}$ *where the number of paths, k, is less than or equal to* $\kappa = \left\lceil \frac{2\alpha^2 r^2 \ln^{2d_2+1}(2m)}{\epsilon^2} \right\rceil$,

  under the following strict constraints:

$$\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}}}{\ln m}$$
$$p \geq \frac{\ln^{2d_2+1}(2m)}{m^{\frac{2}{p}+2}}$$
$$d_1 \geq \Big(\frac{\ln(2m)}{\ln m}\Big)^{2d_2+1}$$

or less strict constraints:

$$\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}+1}}{m-1}$$
$$p \geq \frac{2^{2d_2+1}}{m^{\frac{2}{p}-2d_2+1}}$$
$$d_1 \geq \Big(\frac{\ln(2m)}{\ln m}\Big)^{2d_2+1}$$

or

under the following strict constraints:

$$\frac{d_1}{p} \leq \frac{m^{\frac{2}{p}}}{\ln m}$$
$$d_1 \geq \max\Big\{\frac{\ln^{2d_2+1}(2m)}{m^2 \ln m}, \Big(\frac{\ln(2m)}{\ln m}\Big)^{2d_2+1}\Big\}$$

or less strict constraints:

$$\frac{d_1}{p} \leq m^{\frac{2}{p}-1}$$
$$d_1 \geq \max\left\{2^{2d_2+1} \cdot \frac{m^{2d_2}}{m-1}, \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}\right\}$$

- $m^{\mathcal{O}(\frac{d_1 p \alpha^2 r^2 m^{\frac{2}{p}+2}}{\epsilon^2})}$ *where the number of paths, k, is less than or equal to* $\kappa = \left\lceil \frac{64 p \alpha^2 r^2 m^{\frac{2}{p}+2}}{\epsilon^2} \right\rceil$,
under the following strict constraints:

$$p \leq \frac{\ln^{2d_2+1}(2m)}{m^{\frac{2}{p}+2}}$$
$$\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}+2}}{\ln^{2d_2+1} m}$$

or less strict constraints:

$$p \leq \frac{1}{2^{d_2+1}} \cdot \frac{(2m-1)^{2d_2+1}}{m^{\frac{2}{p}+2d_2+3}}$$
$$\frac{d_1}{p} \geq \frac{m^{\frac{2}{p}+2d_2+3}}{(m-1)^{2d_2+1}}$$

- $m^{\mathcal{O}(\frac{d_1^2 \alpha^2 r^2 m^2 \ln m}{\epsilon^2})}$ *where the number of paths, k, is less than or equal to* $\kappa = \left\lceil \frac{8 d_1 \alpha^2 r^2 m^2 \ln m}{\epsilon^2} \right\rceil$,
under the following strict constraints:

$$d_1 \leq \frac{\ln^{2d_2+1}(2m)}{m^2 \ln m}$$
$$\ln m \geq m^{\frac{1}{d_2}}$$

or less strict constraints:

$$d_1 \leq \frac{1}{2^{2d_2+1}} \cdot \frac{(2m-1)^{2d_2+1}}{m^{2d_2+4}})$$
$$\ln m \geq m^{\frac{1}{d_2}}$$

where p *is norm with* $2 \leq p < \infty$.
The less strict constraints may be used when building an instance (they seem simpler in use).

Having a thorough look at the values of p that minimize the expression $f(p) = p \cdot m^{\frac{2}{p}}$ (that appears explicitly in the 3rd case above), it could be easily shown that it takes the minimum value when $p = 2\ln m$. After that, theorem 4.17 is modified as follows:

Theorem 4.18. *Second Comparison Theorem: Let (G, r, l) be an instance, $r > 0$, with $G = (V, E)$ and linear latencies $l_e(x) = a_e \cdot x + b_e$, with rational coefficients $a_e, b_e \geq 0$ and let H^B be the Best Subnetwork of G. Let also $\mu = |\mathcal{P}| \leq m^{d_1}$ be the total number of paths, $|P| \leq \ln^{d_2} m$ for all $P \in \mathcal{P}$ and $\alpha = \max_{e \in E}\{a_e\}$. Then, there exists a flow $\tilde{f}$, to at most $k \leq \kappa$ paths P with $\tilde{f}_P = \frac{\rho}{\kappa}, \rho \in [\kappa]$, that is an $\frac{\epsilon}{2}$ - Nash flow on $G_{\tilde{f}}$ and satisfies $l_P(\tilde{f}) \leq L_{eq}(H^B) + \frac{\epsilon}{2}$, for all paths P in $G_{\tilde{f}}$. This flow can be computed in the following minimum computational times:*

- $m^{\mathcal{O}(\frac{d_1^2 \alpha^2 r^2 \ln^{2d_2+1} m}{\epsilon^2})}$ *where the number of paths, k, is less than or equal to $\kappa = \left\lceil \frac{8 d_1 \alpha^2 r^2 \ln^{2d_2+1} m}{\epsilon^2} \right\rceil$,*

  under the following strict constraints:

$$2m^{\frac{1}{\ln m}} \leq d_1 \leq \frac{2m^{\frac{1}{\ln m}+2}}{\ln^{2d_2} m}$$
$$\ln m \leq m^{\frac{1}{d_2}}$$
$$d_1 \leq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

  or less strict constraints:

$$\frac{m^{\frac{1}{\ln m}+1}}{m-1} \leq \frac{d_1}{2\ln m} \leq m^{\frac{1}{\ln m}-2d_2+1}$$
$$m^{2d_2} \leq m-1$$
$$d_1 \leq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

  <u>or</u>

  under the following strict constraints:

$$d_1 \leq 2m^{\frac{1}{\ln m}}$$
$$\ln m \leq m^{\frac{1}{d_2}}$$
$$d_1 \leq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

  or less strict constraints:

$$d_1 \leq 2m^{\frac{1}{\ln m}-1}\ln m$$
$$\ln m \leq m^{\frac{1}{d_2}}$$
$$d_1 \leq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$

- $m^{\mathcal{O}\left(\frac{d_1\alpha^2 r^2 \ln^{2d_2+1}(2m)}{\epsilon^2}\right)}$ *where the number of paths, k, is less than or equal to* $\kappa = \left\lceil \frac{2\alpha^2 r^2 \ln^{2d_2+1}(2m)}{\epsilon^2} \right\rceil$,
under the following strict constraints:
$$d_1 \geq 2m^{\frac{1}{\ln m}}$$
$$m^{\frac{1}{\ln m}+2} \geq \frac{\ln^{2d_2+1}(2m)}{2\ln m}$$
$$d_1 \geq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$
or less strict constraints:
$$\frac{d_1}{2\ln m} \geq \frac{m^{\frac{1}{\ln m}+1}}{m-1}$$
$$2\ln m \geq \frac{2^{2d_2+1}}{m^{\frac{1}{\ln m}-2d_2+1}}$$
$$d_1 \geq \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}$$
<u>*or*</u>
under the following strict constraints:
$$d_1 \leq 2m^{\frac{1}{\ln m}}$$
$$d_1 \geq \max\left\{\frac{\ln^{2d_2+1}(2m)}{m^2\ln m}, \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}\right\}$$
or less strict constraints:
$$d_1 \leq 2m^{\frac{1}{\ln m}-1}\ln m$$
$$d_1 \geq \max\left\{2^{2d_2+1}\cdot\frac{m^{2d_2}}{m-1}, \left(\frac{\ln(2m)}{\ln m}\right)^{2d_2+1}\right\}$$

- $m^{\mathcal{O}\left(\frac{d_1\alpha^2 r^2 m^{\frac{1}{\ln m}+2}\ln m}{\epsilon^2}\right)}$ *where the number of paths, k, is less than or equal to* $\kappa = \left\lceil \frac{128\alpha^2 r^2 m^{\frac{1}{\ln m}+2}\ln m}{\epsilon^2} \right\rceil$,
under the following strict constraints:
$$m^{\frac{1}{\ln m}+2} \leq \frac{\ln^{2d_2+1}(2m)}{2\ln m}$$
$$d_1 \geq \frac{2m^{\frac{1}{\ln m}+2}}{\ln^{2d_2} m}$$
or less strict constraints:
$$2^{d_2+2}\cdot\ln m \leq \frac{(2m-1)^{2d_2+1}}{m^{\frac{1}{\ln m}+2d_2+3}}$$
$$\frac{d_1}{2\ln m} \geq \frac{m^{\frac{1}{\ln m}+2d_2+3}}{(m-1)^{2d_2+1}}$$

- $m^{\mathcal{O}(\frac{d_1^2\alpha^2r^2m^2\ln m}{\epsilon^2})}$ *where the number of paths, k, is less than or equal to* $\kappa = \left\lceil \frac{8d_1\alpha^2r^2m^2\ln m}{\epsilon^2} \right\rceil$,
  under the following strict constraints:

$$d_1 \leq \frac{\ln^{2d_2+1}(2m)}{m^2\ln m}$$
$$\ln m \geq m^{\frac{1}{d_2}}$$

  or less strict constraints:

$$d_1 \leq \frac{1}{2^{2d_2+1}} \cdot \frac{(2m-1)^{2d_2+1}}{m^{2d_2+4}})$$
$$\ln m \geq m^{\frac{1}{d_2}}$$

  The less strict constraints may be used when building an instance (they seem simpler in use).

We summarize our conclusions. In this section we proved two theorems which are an improvement of theorem 4 of [16]. They provide approximating methods to random instances with polynomially many paths, each of polylogarithmic length, and linear latencies.

If the traffic rate is of the size $\mathcal{O}(1)$ (or more generally $\mathcal{O}(\text{poly}(\ln\ln m))$), then quasipolynomial computational running times may be achieved. The same result is valid also for traffic rates up to $\mathcal{O}(\text{poly}(\ln m))$, where $m = |E|$.

Chapter 5

Resolving Braess's Paradox in Good Random Networks

In this chapter we study the approximation of the best subnetwork, given a large random network that satisfies some "good" properties, defined later on. It is based on Chapter 3's studies by [20], [21], [22] on the likeliness of the Paradox on large random graphs, and on the joint work of Fotakis, Kaporis, Lianeas and Spirakis in [32].

A polynomial time approximation preserving reduction to a simplified network is presented in this chapter, where all neighbors of s and t are directly connected by 0 latency edges.

Building on this, an approximation scheme is obtained such that for any constant $\epsilon > 0$ and with high probability, it computes a subnetwork and an ϵ-Nash flow with maximum latency at most $(1+\epsilon)L_{eq}(H^B) + \epsilon$, where $L_{eq}(H^B)$ is the equilibrium latency of the best subnetwork.

This thesis contribution is an improvement on the best known running time for approximating the best subnetwork and its equilibrium latency for this kind of networks, which was originally presented in [32].

Overall, the approximation scheme runs in polynomial time if the random network has average degree $\mathcal{O}(poly(\ln n))$ and the traffic rate is $\mathcal{O}(poly(\ln \ln n))$ and in quasipolynomial time for average degrees up to $o(n)$ and traffic rates of $\mathcal{O}(poly(\ln n))$, where $n \equiv |V|$.

5.1 Problem Specific Definitions

Best Subnetwork Equilibrium Latency Problem. An approximation scheme of the Best Subnetwork is the main output of this chapter. In the *Best Subnetwork Equilibrium Latency* problem, or BestSubEL in short, we are given an instance (G, r, l), and seek for the best subnetwork H^B of (G, r, l) and its equilibrium latency $L_{eq}(H^B)$.

Good Networks. We restrict our attention to undirected $s - t$ networks $G(V, E)$. We let $n \equiv |V|$ and $m \equiv |E|$. For any vertex v, we let $\Gamma(v) = \{u \in V : \{u, v\} \in E\}$ denote the set of v's neighbors in G. Similarly, for any non-empty $S \subseteq V$, we let $\Gamma(S) = \bigcup_{v \in S} \Gamma(v)$ denote the set of neighbors of the vertices in S, and let $G[S]$ denote the subnetwork of G induced by S. For convenience, we let $V_s \equiv \Gamma(s)$, $E_s \equiv \{\{s, u\} : u \in V_s\}$, $V_t \equiv \Gamma(t)$, $E_t \equiv \{\{v, t\} : v \in V_t\}$, and $V_m \equiv V \setminus (\{s, t\} \cup V_s \cup V_t)$. We also let $n_s = |V_s|$, $n_t = |V_t|$, $n_+ = \max\{n_s, n_t\}$, $n_- = \min\{n_s, n_t\}$, and $n_m = |V_m|$. We sometimes write $V(G)$, $n(G)$, $V_s(G)$, $n_s(G)$, $\ldots$, if G is not clear from the context.

It is convenient to think that the network G has a layered structure consisting of s, the set of s's neighbors V_s, an "intermediate" subnetwork connecting the neighbors of s to the neighbors of t, the set of t's neighbors V_t, and t. Then, any $s-t$ path starts at s, visits some $u \in V_s$, proceeds either directly or through some vertices of V_m to some $v \in V_t$, and finally reaches t. Thus, we refer to $G_m \equiv G[V_s \cup V_m \cup V_t]$ as the *intermediate subnetwork* of G. Depending on the structure of G_m, we say that:

- G is a *random* $\mathcal{G}(n, p)$ network if (i) n_s and n_t follow the binomial distribution with parameters n and p, and (ii) if any edge $\{u, v\}$, with $u \in V_m \cup V_s$ and $v \in V_m \cup V_t$, exists independently with probability p. Namely, the intermediate network G_m is an Erdös-Rényi random graph with $n - 2$ vertices and edge probability p, except from the fact that there are no edges in $G[V_s]$ and in $G[V_t]$.

- G is *internally bipartite* if the intermediate network G_m is a bipartite graph with independent sets V_s and V_t. G is *internally complete bipartite* if every neighbor of s is directly connected by an edge to every neighbor of t.

- G is 0-*latency simplified* if it is internally complete bipartite and every edge e connecting a neighbor of s to a neighbor of t has latency function $l_e(x) = 0$.

The 0-*latency simplification* G_0 of a given network G is a 0-latency simplified network obtained from G by replacing $G[V_m]$ with a set of 0-latency edges directly connecting every neighbor of s to every neighbor of t. Moreover, we say that a 0-latency simplified network G is *balanced*, if $|n_s - n_t| \le 2n_-$.

We say that a network $G(V, E)$ is (n, p, k)-*good*, for some integer $n \le |V|$, some probability $p \in (0, 1)$, with $pn = o(n)$, and some constant $k \ge 1$, if G satisfies that:

1. The maximum degree of G is at most $3np/2$, i.e., for any $v \in V$, $|\Gamma(v)| \le 3np/2$.

Algorithm 1: Approximation Scheme for BestSubEL in Good Networks

```
Input: Good network G(V,E), rate r > 0, approximation guarantee ε > 0
Output: Subnetwork H of G and ε-Nash flow g in H with L_g(H) ≤ (1+ε)L_eq(H^B) + ε
if L_eq(G) < ε, return G and a Nash flow of (G, r, l)
create the 0-latency simplification G_0 of G
if r ≥ (B_max n_+)/(ε A_min), then let H_0 = G_0 and let f be a Nash flow of (G_0, r)
else, let H_0 be the subnetwork and f the ε/6-Nash flow of Thm. 5.6 applied with error ε/6
let H be the subnetwork and let g be the ε-Nash flow of Lemma 5.8 starting from H_0 and f
return the subnetwork H and the ε-Nash flow g
```

2. G is an *expander graph*, namely, for any set $S \subseteq V$, $|\Gamma(S)| \geq \min\{np\,|S|, n\}/2$.

3. The edges of G have random reasonable latency functions distributed according to $\mathcal{A} \times \mathcal{B}$, and for any constant $\eta > 0$, $\mathbb{P}[\mathcal{B} \leq \eta/\ln n]np = \omega(1)$.

4. If $k > 1$ and we randomly partition V_m into k sets $V_m^1, \ldots, V_m^k$, each of cardinality $|V_m|/k$, all the induced subnetworks $G[\{s,t\} \cup V_s \cup V_m^i \cup V_t]$ are $(n/k, p, 1)$-good, with a possible violation of the maximum degree bound by s and t.

If G is a *random* $\mathcal{G}(n,p)$ network, with n sufficiently large and $p \geq ck\frac{\ln n}{n}$ for some large enough constant $c > 1$ (which by section 3.2 means that it will almost surely be connected), then G is a (n,p,k)-good network with high probability (see e.g., [33]), provided that the latency functions satisfy condition (3) above. Similarly, the random instances considered in [34] are good with high probability. Also note that the 0-latency simplification of a good network is balanced, due to conditions (1) and (2).

5.2 The Approximation Scheme and Outline of the Analysis

In this section, we describe the main steps of the approximation scheme (see also Algorithm 1), and give an outline of its analysis. We let $\varepsilon > 0$ be the approximation guarantee, and assume that $L_{eq}(G) \geq \varepsilon$. Otherwise, any Nash flow of (G, r, l) suffices.

Algorithm 1 is based on an approximation-preserving reduction of BestSubEL for a good network G to BestSubEL for the 0-latency simplification G_0 of G. The first step of our approximation-preserving reduction is to show that the equilibrium latency of the best subnetwork does not increase when we consider the 0-latency simplification G_0 of a network G instead of G itself. Since decreasing the edge latencies (e.g., decreasing $l_{(u,w)}(x) = 1$ to $l_{(u,w)}(x) = 0$ in Fig. 2.1) may trigger Braess's paradox, we need lemma 5.2 and its careful proof to make sure that zeroing out the latency of the intermediate subnetwork does not cause an abrupt increase in the equilibrium latency.

Next, we focus on the 0-latency simplification G_0 of G (step 2 in Algorithm 1). We show that if the traffic rate is large enough, i.e., if $r = \Omega(n_+/\varepsilon)$, the paradox has a marginal influence on the equilibrium latency. Thus, any Nash flow of (G_0, r, l) is an $(1+\varepsilon)$-approximation of BestSubEL (lemma 5.3, step 4). If $r = \mathcal{O}(n_+/\varepsilon)$, we use an approximate version of Caratheodory's theorem (see theorem 3 of [31] and also its wide use in chapter 4) and obtain an $\varepsilon/6$-approximation of BestSubEL for (G_0, r, l) (theorem 5.6, step 4).

We now have a subnetwork H_0 and an $\varepsilon/6$-Nash flow f that comprise a good approximate solution to BestSubEL for the simplified instance (G_0, r, l). The next step of our approximation-preserving reduction is to extend f to an approximate solution to BestSubEL for the original instance (G, r, l). The intuition is that due to the expansion and the reasonable latencies of G, any collection of 0-latency edges of H_0 used by f to route flow from V_s to V_t can be "simulated" by an appropriate collection of low-latency paths of the intermediate subnetwork G_m of G. We first prove this claim for a small part of H_0 consisting only of neighbors of s and neighbors of t with approximately the same latency under f (lemma 5.7, the proof draws ideas from [21]'s lemma 5). Then, using a careful latency-based grouping of the neighbors of s and of the neighbors of t in H_0, we extend this claim to the entire H_0 (lemma 5.8). Thus, we obtain a subnetwork H of G and an ε-Nash flow g in H such that $L_g(H) \leq (1+\varepsilon)L_{eq}(H^B) + \varepsilon$ (step 5).

We summarize our main result. The proof follows by combining lemma 5.2, theorem 5.6, and lemma 5.8 in the way indicated by Algorithm 1 and the discussion above.

Theorem 5.1. *Let $G(V, E)$ be (n, p, k)-good network, where $k \geq 1$ is a large enough constant, let $r > 0$ be any traffic rate, and let H^B be the best subnetwork of (G, r, l). Then, for any $\varepsilon > 0$, Algorithm 1 computes in time $n_+^{\mathcal{O}(r^2 A_{\max}^2/\varepsilon^2)}\mathrm{poly}(|V|)$, a flow g and a subnetwork H of G such that with high probability, wrt the random choice of the latency functions, g is an ε-Nash flow of (H, r, l) and has equilibrium latency $L_g(H) \leq (1+\varepsilon)L_{eq}(H^B) + \varepsilon$.*

By the definition of reasonable latencies, $A_{\max}$ is a constant. Also, by lemma 5.3, r affects the running time only if $r = \mathcal{O}(n_+/\varepsilon)$. In fact, previous work on selfish network design assumes that $r = \mathcal{O}(1)$, see e.g., [5]. Thus, if $r = \mathcal{O}(1)$ (or more generally, if $r = \mathcal{O}(\text{poly}(\ln\ln n))$) and $pn = \mathcal{O}(\text{poly}(\ln n))$, in which case $n_+ = \mathcal{O}(\text{poly}(\ln n))$, theorem 5.1 gives a randomized polynomial-time approximation scheme for BestSubEL in good networks. Moreover, the running time is quasipolynomial for traffic rates up to $\mathcal{O}(\text{poly}(\ln n))$ and average degrees up to $o(n)$, i.e., for the entire range of p in [20], [21].

The next sections are devoted to the proofs of lemmas 5.2 and 5.8, and of theorem 5.6.

5.3 Network Simplification

We first show that the equilibrium latency of the best subnetwork does not increase when we consider the 0-latency simplification G_0 of a network G instead of G itself.

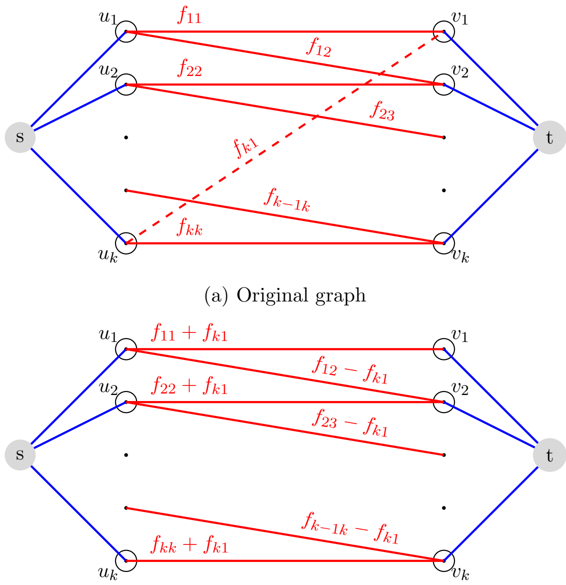
Lemma 5.2. *Let G be any network, let $r > 0$ be any traffic rate, and let H^B be the best subnetwork of (G, r, l). Then, there is a subnetwork H' of the 0-latency simplification of H^B (and thus, a subnetwork of G_0) with $L_{eq}(H') \leq L_{eq}(H^B)$.*

Proof. Throughout the proof, we assume wlog that all the edges of H^B are used by the H^B's equilibrium flow f (otherwise, we can remove all unused edges from H^B). The proof is constructive, and at the conceptual level, proceeds in two steps.

For the first step, given the equilibrium flow f of the best subnetwork H^B of G, we construct a simplification H_1 of H^B that is internally bipartite and has constant latency edges connecting $\Gamma(s)$ to $\Gamma(t)$. H_1 also admits f as an equilibrium flow, and thus $L_{eq}(H_1) = L_{eq}(H^B)$. We also show how to further simplify H_1 so that its intermediate bipartite subnetwork becomes acyclic.

To construct the simplification H_1 of H^B, we let f be the equilibrium flow of H^B, and let $L \equiv L_{eq}(H^B)$. For each $u_i \in \Gamma(s)$ and $v_j \in \Gamma(t)$, we let $f_{ij} = \sum_{p=(s,u_i,\ldots,v_j,t)} f_p$ be the flow routed by f from u_i to v_j. The network H_1 is obtained from H^B by replacing the intermediate subnetwork of H^B with a bipartite subnetwork connecting $\Gamma(s)$ and $\Gamma(t)$ with constant latency edges. More specifically, instead of the intermediate subnetwork of H^B, for each $u_i \in \Gamma(s)$ and $v_j \in \Gamma(t)$ with $f_{ij} > 0$, we have an edge (u_i, v_j) of constant latency $b_{ij} = L - (a_{(s,u_i)} f_{(s,u_i)} + b_{(s,u_i)}) - (a_{(v_j,t)} f_{(v_j,t)} + b_{(v_j,t)})$ (the corresponding $a_{(u_i,v_j)}$ is set to 0). If $f_{ij} = 0$, u_i and v_j are not connected in H_1. We note that by construction, H_1 admits f as an equilibrium flow, and thus $L_{eq}(H_1) = L$.

Furthermore, we modify H_1 by deleting some edges from its intermediate subnetwork so that the induced bipartite subgraph $H_1[\Gamma(s) \cup \Gamma(t)]$ becomes



(a) Original graph

(b) New graph

Figure 5.1: In (a), we have a cycle $C = (u_1, v_2, u_2, \ldots, v_k, u_k, v_1, u_1)$ in the intermediate subnetwork $H_1[\Gamma(s) \cup \Gamma(t)]$. We assume that f_{k1} is the minimum amount flow through an edge of C in the equilibrium flow f. In (b), we remove the edge e_{k1} and we show the corresponding change in the amount of flow on the remaining edges of C. Since the latency functions of the edges in C are constant, the change in the flow does not affect equilibrium.

acyclic. Therefore, in the resulting network, for each $u_i \in \Gamma(s)$ and each $v_j \in \Gamma(t)$, there is at most one (s, u_i, v_j, t) path in H_1. Hence, the resulting network admits a unique equilibrium flow with a unique path decomposition.

To this end, let us assume that there is a cycle $C = (u_1, v_2, u_2, \ldots, v_k, u_k, v_1, u_1)$ in the intermediate subnetwork $H_1[\Gamma(s) \cup \Gamma(t)]$. We let $e_{k1} = (u_k, v_1)$ be the edge of C with the minimum amount of flow in f, and let f_{k1} be the flow through e_{k1} (see also Fig. 5.1). Then, removing e_{k1}, and updating the flows along the remaining edges of C so that $f'_{ii} = f_{ii} + f_{k1}$, $1 \leq i \leq k$, and $f'_{i(i+1)} = f_{i(i+1)} - f_{k1}$, $1 \leq i \leq k-1$, we "break" the cycle C, by eliminating the flow in e_{k1}, and obtaining a new equilibrium flow f' of the same rate r and with the same latency L as that of f. Applying this procedure repeatedly to all cycles, we end up with an internally bipartite network H_1 with an acyclic intermediate subnetwork that includes constant latency edges only. Moreover, H_1 admits an equilibrium flow f of latency L. This concludes the first part of the proof.

The second part of the proof is to show that we can either remove some of the intermediate edges of H_1 or zero their latencies, and obtain a subnetwork

H' of the 0-latency simplification of H^B with $L_{eq}(H') \leq L_{eq}(H^B)$. To this end, we describe a procedure where in each step, we either remove some intermediate edge of H_1 or zero its latency, without increasing the latency of the equilibrium flow.

Let us focus on an edge $e_{kl} = (u_k, v_l)$ with $b_{kl} > 0$, and attempt to set its latency function to $b'_{kl} = 0$. We have also to change the equilibrium flow f to a new flow f' that is an equilibrium flow of latency at most L in the modified network with $b'_{kl} = 0$. We let r_p be the amount of flow moving from an $s-t$ path $p = (s, u_i, v_j, t)$ to the path $p_{kl} = (s, u_k, v_l, t)$ during this change. We note that r_p may be negative, in which case, $|r_p|$ units of flow actually move from p_{kl} to p. Thus, r_p's define a rerouting of f to a new flow f', with $f'_p = f_p - r_p$, for any $s-t$ path p other than p_{kl}, and $f'_{kl} = f_{kl} + \sum_p r_p$.

Next, we show how to compute r_p's so that f' is an equilibrium flow of cost at most L in the modified network (where we want to set $b'_{kl} = 0$). We let $\mathcal{P} = \mathcal{P}_{H_1} \setminus \{p_{kl}\}$ denote the set of all $s-t$ paths in H_1 other than p_{kl}. We let $\vec{F}$ be the $|\mathcal{P}| \times |\mathcal{P}|$ matrix, indexed by the paths $p \in \mathcal{P}$, where $\vec{F}[p_1, p_2] = \sum_{e \in p_1 \cap p_2} a_e - \sum_{e \in p_1 \cap p_{kl}} a_e$, and let $\vec{r}$ be the vector of r_p's. Then, the p-th component of $\vec{F}\vec{r}$ is equal to $l_p(f) - l_p(f')$. In the following, we consider two cases depending on whether $\vec{F}$ is singular or not.

If $\vec{F}$ is non-singular, the linear system $\vec{F}\vec{r} = \varepsilon\,\vec{1}$ has a unique solution $\vec{r}_\varepsilon$, for any $\varepsilon > 0$. Moreover, due to linearity, for any $\alpha \geq 0$, the unique solution of the system $\vec{F}\vec{r} = \alpha\,\varepsilon\,\vec{1}$ is $\alpha\,\vec{r}_\varepsilon$. Therefore, for an appropriately small $\varepsilon > 0$, the linear system $Q_\varepsilon = \{\vec{F}\vec{r} = \varepsilon\,\vec{1}, f_p - r_p \geq 0 \;\; \forall p \in \mathcal{P}, f_{kl} + \sum_p r_p \geq 0, l_{p_{kl}}(f') \leq L + b_{kl} - \varepsilon\}$ admits a unique solution $\vec{r}$.

We keep increasing ε until one of the inequalities of Q_ε becomes tight. If it first becomes $r_p = f_p$ for some path $p = (s, u_i, v_j, t) \in \mathcal{P}$, we remove the edge (u_i, v_j) from H_1 and adjust the constant latency of e_{kl} so that $l_{p_{kl}}(f') = L - \varepsilon$. Then, the flow f' is an equilibrium flow of cost $L - \varepsilon$ for the resulting network, which has one edge less than the original network H_1. If $\sum_p r_p < 0$ and it first becomes $\sum_p r_p = -f_{kl}$, we remove the edge e_{kl} from H_1. Then, f' is an equilibrium flow of cost $L - \varepsilon$ for the resulting network, which again has one edge less than H_1. If $\sum_p r_p > 0$ and it first becomes $l_{p_{kl}}(f') = L + b_{kl} - \varepsilon$, we set the constant latency of the edge e_{kl} to $b'_{kl} = 0$. In this case, f' is an equilibrium flow of cost $L - \varepsilon$ for the resulting network that has one edge of 0 latency more than the initial network H_1.

If $\vec{F}$ is singular, proceeding similarly, we compute r_p's so that f' is an equilibrium flow of cost L in a modified network that includes one edge less than the original network H_1.

When $\vec{F}$ if singular, the homogeneous linear system $\vec{F}\vec{r} = \vec{0}$ admits a nontrivial solution $\vec{r} \neq \vec{0}$. Moreover, due to linearity, for any $\alpha \in \mathbb{R}$, $\alpha\,\vec{r}$ is also a solution to $\vec{F}\vec{r} = \vec{0}$. Therefore, the linear system $Q_0 = \{\vec{F}\vec{r} = \vec{0}, f_p - r_p \geq 0 \;\; \forall p \in \mathcal{P}, f_{kl} + \sum_p r_p \geq 0\}$ admits a solution $\vec{r} \neq \vec{0}$ that makes at least one of the inequalities tight.

We recall that the p-th component of $\vec{F}\vec{r}$ is equal to $l_p(f) - l_p(f')$. Therefore, for the flow f' obtained from the particular solution $\vec{r}$ of Q_0, the latency of any path $p \in \mathcal{P}$ is equal to L.

If $\vec{r}$ is such that $r_p = f_p$ for some path $p = (s, u_i, v_j, t) \in \mathcal{P}$, we remove the edge (u_i, v_j) from H_1 and adjust the constant latency of e_{kl} so that $l_{p_{kl}}(f') = L$. Then, the flow f' is an equilibrium flow of cost L for the resulting network, which has one edge less than the original network H_1.

If $\vec{r}$ is such that $\sum_p r_p = -f_{kl}$, we remove the edge e_{kl} from H_1. Then, f' is an equilibrium flow of cost L for the resulting network, which again has one edge less than H_1.

Each time we apply the procedure above either we decrease the number of edges of the intermediate network by one or we increase the number of 0-latency edges of the intermediate network by one, without increasing the latency of the equilibrium flow. Moreover, if p_{kl} is disjoint to the paths $p \in \mathcal{P}$, $\vec{F}$ is non-singular (next paragraph) and the procedure above leads to a decrease in the equilibrium latency, and eventually to setting $b'_{kl} = 0$. So, by repeatedly applying these steps, we end up with a subnetwork H' of the 0-latency simplification of H^B with $L_{eq}(H') \leq L_{eq}(H^B)$.

To show that if p_{kl} is disjoint to the paths $p \in \mathcal{P}$, $\vec{F}$ is non-singular we show that the matrix $\vec{F}$ is positive definite (which implies that $\vec{F}$ is non-singular[1]). We first note that if p_{kl} is disjoint to all $p \in \mathcal{P}$, then for all $p_1, p_2 \in \mathcal{P}$, $\vec{F}[p_1, p_2] = \sum_{e \in p_1 \cap p_2} a_e$.

Hence, for all $\vec{x} \in \mathbb{R}^{|\mathcal{P}|}$, $\vec{x}^T \vec{F} \vec{x} = \sum_{e \in E(\mathcal{P})} a_e x_e^2 \geq 0$, where $E(\mathcal{P})$ denotes the set of edges included in the paths of $\mathcal{P}$ and $x_e = \sum_{p: e \in p} x_p$. Since the intermediate network of H_1 is acyclic and any flow in H_1 has a unique path decomposition, if $\vec{x}$ has one or more non-zero components, there is at least one edge e adjacent to either s or t such that $x_e > 0$, and thus $\vec{x}^T \vec{F} \vec{x} > 0$. Otherwise, the difference of the flow defined by $\vec{x}$ with the trivial flow defined by $\vec{0}$ would indicate the existence of a cycle in the intermediate subnetwork of H_1. This is a contradiction, since by the first part of the proof, the intermediate part of H_1 is acyclic.

□

5.4 Approximating the Best Subnetwork of Simplified Networks

We proceed to show how to approximate the BestSubEL problem in a balanced 0-latency simplified network G_0 with reasonable latencies. We may always regard G_0 as the 0-latency simplification of a good network G. We

[1] or else there exists a vector $\vec{x} \neq 0$ such that $\vec{F}\vec{x} = 0$, or $\vec{x}^T \vec{F} \vec{x} = 0$, which means that $\vec{F}$ is not positive definite.

first prove two useful lemmas (lemmas 5.3 and 5.4) about the maximum traffic rate r up to which BestSubEL remains interesting, and about the maximum amount of flow routed on any edge / path in the best subnetwork.

We first show that for 0-latency simplified instances (G_0, r, l), we can assume, essentially wlog, that the traffic rate $r = \mathcal{O}(n_+/\varepsilon)$. Otherwise, a Nash flow f of (G_0, r, l) is an $(1+\varepsilon)$-approximation of the BestSubEL problem in (G_0, r, l):

Lemma 5.3. *Let G_0 be any 0-latency simplified network, let $r > 0$, and let H_0^* be the best subnetwork of (G_0, r, l). For any $\varepsilon > 0$, if $r > \frac{B_{\max} n_+}{A_{\min}\varepsilon}$, then $L_{eq}(G_0) \le (1+\varepsilon) L_{eq}(H_0^*)$.*

Proof. We assume that $r > \frac{B_{\max} n_+}{A_{\min}\varepsilon}$ and we let f be a Nash flow of (G_0, r, l). We consider how f allocates r units of flow to the edges of $E_s \equiv E_s(G_0)$ and to the edges $E_t \equiv E_t(G_0)$. For simplicity, we let $L \equiv L_{eq}(G_0)$ denote the equilibrium latency of G_0, and let $A_s = \sum_{e \in E_s} 1/a_e$ and $A_t = \sum_{e \in E_t} 1/a_e$.

Since G_0 is a 0-latency simplified network and f is a Nash flow of (G_0, r, l), there are $L_1, L_2 > 0$, with $L_1 + L_2 = L$, such that all used edges incident to s (resp. to t) have latency L_1 (resp. L_2) in the Nash flow f. Since $r > \frac{B_{\max} n_+}{A_{\min}}$, then by an averaging argument, $L_1, L_2 > B_{\max}$ and all edges in $E_s \cup E_t$ are used by f. Moreover, by an averaging argument again, we have that there is an edge $e \in E_s$ with $a_e f_e \le r/A_s$, and that there is an edge $e \in E_t$ with $a_e f_e \le r/A_t$. Therefore, $L_1 \le (r/A_s) + B_{\max}$ and $L_2 \le (r/A_t) + B_{\max}$, and thus, $L \le \frac{r}{A_s} + \frac{r}{A_t} + 2B_{\max}$.

On the other hand, if we ignore the additive terms b_e of the latency functions, the optimal average latency of the players is $r/A_s + r/A_t$, which implies that $L_{eq}(H_0^*) \ge r/A_s + r/A_t$. Therefore, $L \le L_{eq}(H_0^*) + 2B_{\max}$. Moreover, since $r > \frac{B_{\max} n_+}{A_{\min}\varepsilon}$, $A_s \le n_s/A_{\min}$, and $A_t \le n_t/A_{\min}$, we have that:

$$
\begin{aligned}
L_{eq}(H_0^*) &\ge \frac{r}{A_s} + \frac{r}{A_t} \\
&\ge \frac{B_{\max} n_s}{A_{\min}\varepsilon}\frac{A_{\min}}{n_s} + \frac{B_{\max} n_t}{A_{\min}\varepsilon}\frac{A_{\min}}{n_t} \\
&\ge 2B_{\max}/\varepsilon
\end{aligned}
$$

Therefore, $2B_{\max} \le \varepsilon L_{eq}(H_0^*)$, and $L \le (1+\varepsilon) L_{eq}(H_0^*)$.

□

Now, since we have shown that for traffic rates $r = \Omega(n_+/\varepsilon)$, the paradox has a minimal influence on the equilibrium latency (step 2 in Algorithm 1), then wlog, we may focus on traffic rates $r = \mathcal{O}(n_+/\varepsilon)$:

Lemma 5.4. *Let G_0 be a balanced 0-latency simplified network with reasonable latencies, let $r > 0$ with $r \le \frac{B_{\max} n_+}{A_{\min}\varepsilon}$, and let f be a Nash flow of the best* *subnetwork of* (G_0, r, l)*. For any* $\varepsilon > 0$*, if* $\mathbb{P}[\mathcal{B} \leq \varepsilon/4] \geq \delta$*, for some constant* $\delta > 0$*, there exists a constant* $\rho = \frac{24A_{\max}B_{\max}}{\delta\varepsilon A_{\min}^2}$ *such that with probability at least* $1 - \mathrm{e}^{-\delta n_-/8}$*,* $f_e \leq \rho$*, for all edges* e*.*

Proof. We proceed to show that in a 0-latency simplified instance (G_0, r, l), the best subnetwork Nash flow routes $\mathcal{O}(r/n_+)$ units of flow on any edge and on any $s-t$ path with high probability (where the probability is with respect to the random choice of the latency function coefficients). Intuitively, we show that in the best subnetwork Nash flow, with high probability, all used edges and all used $s-t$ paths route a volume of flow not significantly larger than their fair share. We first prove the following technical lemma:

Lemma 5.5. *Let* G_0 *be a balanced* 0*-latency simplified network with reasonable latencies, let* $r > 0$ *be any traffic rate, and let* f *be any Nash flow of the best subnetwork of* (G_0, r, l)*. For any* $\varepsilon > 0$*, if* $L_{eq}(G) \geq \varepsilon$ *and* $\mathbb{P}[\mathcal{B} \leq \varepsilon/4] \geq \delta$*, for some constant* $\delta > 0$*, there exists a constant* $\gamma = \frac{24A_{\max}}{\delta A_{\min}}$ *such that with probability at least* $1 - \mathrm{e}^{-\delta n_-/8}$*, for all edges* e*,* $f_e \leq \gamma r/n_+$*.*

Proof. We let $L \equiv L_{eq}(G_0)$ denote the equilibrium latency and g denote a Nash flow of the original instance (G_0, r, l). Since G_0 is a 0-latency simplified network and g is a Nash flow of (G_0, r, l), there are $L_1, L_2 > 0$, with $L_1 + L_2 = L$, such that: (i) for any edge e incident to s, if $b_e < L_1$, $g_e > 0$ and $a_e g_e + b_e = L_1$, while $g_e = 0$, otherwise, and (ii) for any edge e incident to t, if $b_e < L_2$, $g_e > 0$ and $a_e g_e + b_e = L_2$, while $g_e = 0$, otherwise. Namely, all used edges incident to s (resp. to t) have latency L_1 (resp. L_2) in the Nash flow g. Wlog we assume that $L_1 \geq L_2$, and thus, $L_1 \geq L/2 \geq \varepsilon/2$.

We next show that (i) if $L \geq \varepsilon$ and $\mathbb{P}[\mathcal{B} \leq \varepsilon/4] \geq \delta$, then with probability at least $1 - \mathrm{e}^{-\delta n_-/8}$, $L \leq \frac{24A_{\max}r}{\delta n_+}$, and (ii)that for any e, $f_e \leq L/A_{\min}$. The lemma follows by combining (i) and (ii).

We start with the proof of (i). Let e be any edge incident to s with $b_e \leq \varepsilon/4$. By the discussion above, in the Nash flow g of (G_0, r, l), $g_e > 0$ and $a_e g_e + b_e = L_1$. Using that $L_1 \geq L/2 \geq \varepsilon/2$, we obtain that:

$$L_1 = a_e g_e + b_e \leq a_e g_e + \varepsilon/4 \Rightarrow g_e \geq \frac{L_1 - \varepsilon/4}{a_e} \geq \frac{L_1}{2a_e} \geq \frac{L}{4A_{\max}} \qquad (5.1)$$

Moreover, since $\mathbb{P}[\mathcal{B} \leq \varepsilon/4] \geq \delta$, we use Chernoff bounds[2], and obtain that:

$$\mathbb{P}[|\{e \in E_s(G_0) \text{ with } b_e \leq \varepsilon/4\}| \geq \delta n_s/2] \geq 1 - \mathrm{e}^{-\delta n_s/8} \qquad (5.2)$$

Combining (5.1) and (5.2), we obtain that if $L \geq \varepsilon$ and $\mathbb{P}[\mathcal{B} \leq \varepsilon/4] \geq \delta$, with probability at least $1 - \mathrm{e}^{-\delta n_-/8}$, the flow rate r is at least $\frac{L\delta n_s}{8A_{\max}}$, or equivalently, that:

$$L \leq \frac{8A_{\max}r}{\delta n_s} \leq \frac{24A_{\max}r}{\delta n_+} \qquad (5.3)$$

[2]We use the Chernoff bound form of relation 3.8, where $X = \{e \in E_s(G_0) \text{ with } b_e \leq \varepsilon/4\}$, $(1-\epsilon)\mathbb{E}[X] = \delta n_s/2$ and $\mathbb{E}[X] = \delta n_s$.

The last inequality holds because $|n_s - n_t| \leq 2n_-$ (G_0 is balanced)[3]. This concludes the proof of (i).

To prove (ii), we observe that in the best subnetwork equilibrium flow f, no used edge e has latency greater than L. Therefore, for any used edge e incident to either s or t, we have that:

$$a_e f_e + b_e \leq L \Rightarrow f_e \leq \frac{L}{a_e} \leq \frac{L}{A_{\min}} \tag{5.4}$$

Moreover, any edge e in the intermediate subnetwork of G has $f_e \leq L/A_{\min}$ due to the flow conservation constraints. This concludes the proof of (ii). □

Having proved lemma 5.5, lemma 5.4 is straight forward.

We recall that we always assume that $L_{eq}(G) \geq \varepsilon$, since otherwise the problem of approximating BestSubEL is trivial. Moreover, by the definition of reasonable latency functions, we have that for any constant $\varepsilon > 0$, there is a constant $\delta > 0$, such that $\mathbb{P}[\mathcal{B} \leq \varepsilon/4] \geq \delta$.

So from now on, we can assume, with high probability and wlog, that the Nash flow in the best subnetwork of any simplified instance (G_0, r, l) with $r \leq \frac{B_{\max} n_+}{A_{\min} \varepsilon}$, routes $\mathcal{O}(1)$ units of flow on any used edge and on any used path. □

Approximating the Best Subnetwork of Simplified Networks. We proceed to derive an approximation scheme for the best subnetwork of any simplified instance (G_0, r, l). Again, we will use Barman's work on [31] and produce a proof analogous to theorem 4.10's first part of the proof.

Theorem 5.6. *Let G_0 be a balanced 0-latency simplified network with reasonable latencies, let $r > 0$, and let H_0^* be the best subnetwork of (G_0, r, l). Then, for any $\varepsilon > 0$, we can compute, in time $n_+^{\mathcal{O}(A_{\max}^2 r^2/\varepsilon^2)}$, a flow f and a subnetwork H_0 consisting of the edges used by f, such that (i) f is an ε-Nash flow of (H_0, r, l) and (ii) $l_{P_i}(f) \leq L_{eq}(H_0^*) + \varepsilon/2$ for every $P_i \in \mathcal{P}$ with $f_{eqP_i} > 0$. Moreover, if $r \leq \frac{B_{\max} n_+}{A_{\min} \varepsilon}$ and $\mathbb{P}[\mathcal{B} \leq \varepsilon/4] \geq \delta$ for some constant $\delta > 0$, then (iii) there exists a constant $\rho > 0$, such that $f_e \leq \rho + \varepsilon$, for all e.*

Proof. Let the Nash Equilibrium flow of the Best Subnetwork H_0^* be f_{eq}.

Then, there will be an $S \subseteq [\mu]$, where $\mu = n_+ \cdot n_-$ is the number of paths, such that $\forall i, j \in S : f_{eqP_i}, f_{eqP_j} > 0$ and $l_{P_i}(f_{eq}) = l_{P_j}(f_{eq}) = L_{eq}(H^B)$. If $S \subset [\mu]$, then $T = [\mu] - S$ and $\forall i \in T : f_{eqP_i} = 0$ and $l_{P_i}(f_{eq}) \neq L_{eq}(H^B)$.

Suppose that we have the following set of vectors $X = \{x_1, x_2, \ldots, x_\mu\}$. Every $x_i, i \in [\mu]$, is a vector of $m = |E|$ elements, and corresponds to the

[3] It is relatively easy to show that $n_+ \leq 3n_s$. If $n_s = n_+$ then this is trivial. If on the other hand $n_s = n_-$, then $n_t = n_+$. Now, since $|n_s - n_t| \leq 2n_-$ we have that $n_+ - n_- \leq 2n_-$ or $n_+ \leq 3n_- = 3n_s$.

path P_i. The vector x_i's component x_{ij}, represents the presence of edge e_j at the corresponding path P_i. That is, if edge e_j belongs to the path P_i, then $x_{ij} = 1$, else $x_{ij} = 0$.

Obviously, for every path only 3 x_{ij}'s should have value equal to 1, while all the rest should have value equal to 0.

If g_{P_i} is the path P_i's corresponding flow, then $g_{P_i} \cdot x_i$ gives g_{P_i}'s contribution to each edge of that path.

Let's normalize all f_{eqP_i} to a new f'_{eqP_i}, such that $\sum_{i=1}^{\mu} f'_{eqP_i} = 1$. Then, f'_{eqP_i} should be defined as:

$$f'_{eqP_i} = \frac{f_{eqP_i}}{r}$$

Let's also define the following vector:

$$\nu = f'_{eqP_1} x_1 + f'_{eqP_2} x_2 + \ldots + f'_{eqP_\mu} x_\mu \tag{5.5}$$

Then, the following is valid:

$$\begin{aligned}
\nu &= f'_{eqP_1} x_1 + f'_{eqP_2} x_2 + \ldots + f'_{eqP_\mu} x_\mu \\
&= \frac{1}{r}(f_{eqP_1} x_1 + f_{eqP_2} x_2 + \ldots + f_{eqP_\mu} x_\mu) \\
&= \frac{1}{r}(\sum_{P \in \mathcal{P}: e_1 \in P} f_{eqP}, \sum_{P \in \mathcal{P}: e_2 \in P} f_{eqP}, \ldots, \sum_{P \in \mathcal{P}: e_m \in P} f_{eqP}) \\
&= \frac{1}{r}(f_{eq e_1}, f_{eq e_2}, \ldots, f_{eq e_m})
\end{aligned} \tag{5.6}$$

It should be obvious that $\nu \in \mathrm{conv}(X)$.

Using McDiarmid's and Kahane's inequalities, we will find a vector $\nu' \in \mathrm{conv}(X)$ such that $\|\nu - \nu'\|_p \leq \epsilon$, where ϵ is a positive constant.

Indeed. $f'_{eq} = (f'_{eqP_1}, f'_{eqP_2}, \ldots, f'_{eqP_\mu})$ could be interpreted as a probability distribution over vectors $x_1, x_2, \ldots, x_\mu$. That is, under probability distribution f'_{eq} vector x_i is drawn with probability f'_{eqP_i}. The vector ν is the mean of this distribution. Specifically, the jth component of ν is the expected value of the random variable that takes value x_{ij} with probability f'_{eqP_i}, where x_{ij} is the jth component of vector x_i. We succinctly express these component-wise equalities as follows:

$$\mathbb{E}_{v \sim f'_{eq}}[v] = \nu$$

Now, let $v_1, v_2, \ldots, v_\kappa$ be κ i.i.d. draws from f'_{eq}. The sample mean vector is defined to be $\frac{1}{\kappa}\sum_{i=1}^{\kappa} v_i$. Below we specify function $g : X^\kappa \to \mathbb{R}$ to quantify the p-norm distance between the sample mean vector and the ν.

$$g(v_1, v_2, \dots, v_\kappa) := \left\| \frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i - \nu \right\|_p$$

where p is norm, with $2 \leq p < \infty$.

Now, for every $j \in [\mu]$ the following is valid:

$$\max_{x \in X} \|x\|_p = (1^p + 1^p + 1^p)^{\frac{1}{p}} = 3^{\frac{1}{p}}$$

We will use McDiarmid's inequality. In particular, we will establish that with positive probability the sample mean vector defined over $\kappa := \left\lceil \frac{256pA_{\max}^2 r^2 3^{\frac{2}{p}}}{\varepsilon^2} \right\rceil$ draws, is $\epsilon = \frac{\varepsilon}{4A_{\max}r}$ close to ν in p-norm. Hence, the stated claim is implied by the probabilistic method.

For any κ tuple $(v_1, v_2, \dots, v_i, \dots, v_\kappa) \in X^\kappa$ and $v'_i \in X$, we show that $|g(v_1, v_2, \dots, v_i, \dots, v_\kappa) - g(v_1, v_2, \dots, v'_i, \dots, v_\kappa)|$ is no more than $\frac{2}{\kappa} \cdot 3^{\frac{1}{p}}$. We can assume without loss of generality that $g(v_1, v_2, \dots, v_i, \dots, v_\kappa) \geq g(v_1, v_2, \dots, v'_i, \dots, v_\kappa)$, since the other case is symmetric.

Setting $u := \frac{1}{\kappa} \sum_{j \neq i} v_j - \nu$ we have:

$$\begin{aligned}
g(v_1, v_2, \dots, v_i, \dots, v_\kappa) - g(v_1, v_2, \dots, v'_i, \dots, v_\kappa) &= \left\| u + \frac{1}{\kappa} v_i \right\|_p - \left\| u + \frac{1}{\kappa} v'_i \right\|_p \\
&\leq \|u\|_p + \frac{1}{\kappa} \|v_i\|_p - \|u\|_p + \frac{1}{\kappa} \|v'_i\|_p \\
&\leq \frac{1}{\kappa} \|v_i\|_p + \frac{1}{\kappa} \|v'_i\|_p \\
&\leq \frac{1}{\kappa} \max \{\|x\|_p\} + \frac{1}{\kappa} \max \{\|x\|_p\} \\
&= \frac{2}{\kappa} \max \{\|x\|_p\} \\
&= \frac{2}{\kappa} \cdot 3^{\frac{1}{p}}
\end{aligned}$$

Given that g satisfies $|g(v_1, v_2, \dots, v_i, \dots, v_\kappa) - g(v_1, v_2, \dots, v'_i, \dots, v_\kappa)| \leq \frac{2}{\kappa} 3^{\frac{1}{p}}$, we can apply Mc-Diarmid's inequality, with $c_i = \frac{2}{\kappa} 3^{\frac{1}{p}}$ for all $i \in [\kappa]$, to obtain:

$$\mathbb{P}(|g - \mathbb{E}[g]| \geq t) \leq 2e^{\frac{-\kappa t^2}{2 \cdot 3^{\frac{2}{p}}}} \tag{5.7}$$

By using the same approach as in the proof of theorem 4.9 we can prove that:

$$\mathbb{E}[g] \leq 2\mathbb{E}_{v_i, r_i} \left\| \sum_{i=1}^{\kappa} r_i \frac{v_i}{\kappa} \right\|_p \tag{5.8}$$

where $r_1, r_2, \ldots, r_\kappa$ be a sequence of i.i.d. Rademacher ± 1 random variables.

At this point we can apply Kahane's inequality with $u_i = \frac{v_i}{\kappa}$ to obtain:

$$\begin{aligned}
\mathbb{E}_{v_i, r_i} \left\| \sum_{i=1}^{\kappa} r_i \frac{v_i}{\kappa} \right\|_p &= \mathbb{E}_{v_i} \left[\mathbb{E}_{r_i} \left[\left\| \sum_{i=1}^{\kappa} r_i \frac{v_i}{\kappa} \right\|_p \middle| v_1 \ldots v_\kappa \right] \right] \\
&\leq \mathbb{E}_{v_i} \left[\sqrt{p} \left(\sum_{i=1}^{\kappa} \left\| \frac{v_i}{\kappa} \right\|_p^2 \right)^{\frac{1}{2}} \right] \\
&\leq \mathbb{E}_{v_i} \left[\sqrt{p} \left(\sum_{i=1}^{\kappa} \frac{\left(\max_{x \in X} \|x\|_p \right)^2}{\kappa^2} \right)^{\frac{1}{2}} \right] \\
&= \sqrt{p} \frac{3^{\frac{1}{p}}}{\sqrt{\kappa}} \qquad (5.9)
\end{aligned}$$

By using relations (5.8) and (5.9) we have that:

$$\mathbb{E}[g] \leq 2\sqrt{p} \frac{3^{\frac{1}{p}}}{\sqrt{\kappa}}$$

Thus, for sample size:

$$\kappa \geq 16 \cdot \frac{p 3^{\frac{2}{p}}}{\epsilon^2} \qquad (5.10)$$

we have $\mathbb{E}[g] \leq \frac{\epsilon}{2}$.

Setting $t = \frac{\epsilon}{2}$ in relation (5.7), and by following the same guidelines of theorem 4.9, we may prove that $\mathbb{P}(g \geq \epsilon) \leq 2e^{-2}$ or equivalently that $\mathbb{P}\Big(\left\| \frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i - \nu \right\|_p \geq \epsilon \Big) \leq 2e^{-2}$.

Therefore, with positive probability:

$$\left\| \frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i - \nu \right\|_p \leq \epsilon \qquad (5.11)$$

The latter means that we have found a vector $\nu' := \frac{1}{\kappa} \sum_{i=1}^{\kappa} v_i$, such that $\nu' \in \text{conv}(X)$ and $\|\nu - \nu'\|_p \leq \epsilon$.

Since $v_1, v_2, \ldots, v_\kappa$ are κ i.i.d. draws from f'_{eq}, vector ν' could be expressed as:

$$\begin{aligned}
\nu' &= x_1 \cdot f'_{P_1} + x_2 \cdot f'_{P_2} + \ldots + x_\mu \cdot f'_{P_\mu} \qquad (5.12) \\
&= \frac{\kappa_1}{\kappa} \cdot x_{\kappa_1} + \frac{\kappa_2}{\kappa} \cdot x_{\kappa_2} + \ldots + \frac{\kappa_k}{\kappa} \cdot x_{\kappa_k} \qquad (5.13)
\end{aligned}$$

where $\kappa_1 + \kappa_2 + \ldots + \kappa_k = \kappa$ and $1 \le \kappa_1, \kappa_2, \ldots, \kappa_k \le \kappa$.

Also, it should be clear that all f'_{P_i} are the normalized values. So $f'_{P_i} = \frac{f_{P_i}}{r}$, where $\sum_{i=1}^{\mu} f_{P_i} = r$.

Vector ν' could be expressed in a form similar to (5.6) as follows:

$$\nu' = \frac{1}{r}(f_{e_1}, f_{e_2}, \ldots, f_{e_m}) \tag{5.14}$$

Taking into account relations (5.6) and (5.14), relation (5.11) could be analyzed as follows:

$$\begin{aligned}
&\left\|\nu - \nu'\right\|_p \le \epsilon \Rightarrow \\
&\left\|\frac{1}{r}\left(f_{eq_{e_1}} - f_{e_1}, f_{eq_{e_2}} - f_{e_2}, \ldots, f_{eq_{e_m}} - f_{e_m}\right)\right\|_p \le \epsilon \Rightarrow \\
&\sum_{i=1}^{m} \left|f_{eq_{e_i}} - f_{e_i}\right|^p \le r^p \cdot \epsilon^p \Rightarrow \\
&\left|f_{eq_{e_i}} - f_{e_i}\right| \le r \cdot \epsilon, \forall i \in [m]
\end{aligned} \tag{5.15}$$

Now, let's consider a path P_i with $f_{eq_{P_i}} > 0$. Suppose now that P_i's edges are $e_{z_1}, e_{z_2}, e_{z_3}$, with $l_{e_{z_2}}(x) = 0 \cdot x + 0 = 0$. Then by relation (5.15) we have:

$$\begin{aligned}
|L_{eq}(H_0^*) - l_{P_i}(f)| &= \left|a_{e_{z_1}}(f_{eq_{e_{z_1}}} - f_{e_{z_1}}) + 0 + a_{e_{z_3}}(f_{eq_{e_{z_3}}} - f_{e_{z_3}})\right| \\
&\le A_{\max}\left(\left|f_{eq_{e_{z_1}}} - f_{e_{z_1}}\right| + \left|f_{eq_{e_{z_3}}} - f_{e_{z_3}}\right|\right) \le 2A_{\max} r\epsilon
\end{aligned}$$

Thus, $\forall i \in S, |L_{eq}(H_0^*) - l_{P_i}(f)| \le 2A_{\max} r\epsilon$. So, if we choose $\epsilon = \frac{\varepsilon}{4A_{\max} r}$ then:

$$|L_{eq}(H_0^*) - l_{P_i}(f)| \le \frac{\varepsilon}{2}$$

or better:

$$l_{P_i}(f) \le L_{eq}(H_0^*) + \frac{\varepsilon}{2} \tag{5.16}$$

Now, since $\epsilon = \frac{\varepsilon}{4A_{\max} r}$, relation (5.10) should be:

$$\begin{aligned}
\kappa &\ge 16 \cdot \frac{p 3^{\frac{2}{p}}}{\epsilon^2} \\
&\ge 256 \cdot \frac{p A_{\max}^2 r^2 3^{\frac{2}{p}}}{\varepsilon^2}
\end{aligned} \tag{5.17}$$

Thus, it suffices to consider only a $\kappa := \left\lceil \frac{256 p A_{\max}^2 r^2 3^{\frac{2}{p}}}{\varepsilon^2} \right\rceil$ number of paths.

Also, relation (5.15) becomes:

$$\left|f_{eq_{e_i}} - f_{e_i}\right| \leq \frac{\varepsilon}{4A_{\max}} \tag{5.18}$$

Now, for $p = 2 \cdot \ln 3$ (which is the case that κ gets its minimum value) we have that $256 \cdot 2 \cdot \ln 3 \cdot 3^{\frac{1}{\ln 3}} \simeq 1529.005$, which means that κ suffices to be $\kappa := \left\lceil \frac{1530 A_{\max}^2 r^2}{\varepsilon^2} \right\rceil$.

Therefore, since $\mu = n_+ n_-$, then by exhaustive search in time $T = n_+^{\mathcal{O}(\kappa)}$ we can find a subnetwork (the one that minimizes the latency of the maximum latency path) such that:

$$T = n_+^{\mathcal{O}(\frac{A_{\max}^2 r^2}{\varepsilon^2})}$$

Moreover, from lemma 5.4 we have that for all edges e, $f_{eq_e} \leq \rho$. So, having found the suitable flow f, and by the use of relation (5.18), property (iii) is easy to be proved.

□

5.5 Extending the Solution to the Good Network

Given a good instance (G, r, l), we create the 0-latency simplification G_0 of G, and using theorem 5.6, we compute a subnetwork H_0 and an $\varepsilon/6$-Nash flow f, that comprise an approximate solution to BestSubEL for (G_0, r, l). Next, we show how to extend f to an approximate solution to BestSubEL for the original instance (G, r, l). The intuition is that the 0-latency edges of H_0 used by f to route flow from V_s to V_t can be "simulated" by low-latency paths of G_m. We first formalize this intuition for the subnetwork of G induced by the neighbors of s with (almost) the same latency B_s and the neighbors of t with (almost) the same latency B_t, for some B_s, B_t with $B_s + B_t \approx L_f$. We may think of the networks G and H_0 in the lemma below as some small parts of the original network G and of the actual subnetwork H_0 of G_0. Thus, we obtain the following lemma, which serves as a building block in the proof of lemma 5.8.

Lemma 5.7. *We assume that $G(V, E)$ is an $(n, p, 1)$-good network, with a possible violation of the maximum degree bound by s and t, but with $|V_s|, |V_t| \leq 3knp/2$, for some constant $k > 0$. Also, the latencies of the edges in $E_s \cup E_t$ are not random, but there exist constants $B_s, B_t \geq 0$, such that for all $e \in E_s$ we have that $l_e(x) = B_s$, and for all $e \in E_t$ we have that $l_e(x) = B_t$. We let $r > 0$ be any traffic rate, let H_0 be any subnetwork of the 0-latency simplification G_0 of G, and let f be any flow of (H_0, r). We assume that there exists a constant $\rho' > 0$, such that for all $e \in E(H_0)$, $0 < f_e \leq \rho'$. Then, for any $\epsilon_1 > 0$, with high probability, wrt the random* *choice of the latency functions of G, we can compute in* $\text{poly}(|V|)$ *time a subnetwork G' of G, with $E_s(G') = E_s(H_0)$ and $E_t(G') = E_t(H_0)$, and a flow g of (G', r, l) such that (i) $g_e = f_e$ for all $e \in E_s(G') \cup E_t(G')$, (ii) g is a $7\epsilon_1$-Nash flow in G', and (iii) $L_g(G') \leq B_s + B_t + 7\epsilon_1$.*

Proof. For convenience and wlog, we assume that $E_s(G) = E_s(H_0)$ and that $E_t(G) = E_t(H_0)$, so that we simply write V_s, V_t, E_s, and E_t from now on. For each $e \in E_s \cup E_t$, we let $g_e = f_e$. So, the flow g satisfies (i), by construction.

We compute the extension of g through G_m as an "almost" Nash flow in a modified version of G, where each edge $e \in E_s \cup E_t$ has a capacity $g_e = f_e$ and a constant latency $l_e(x) = B_s$ if $e \in E_s$, and $l_e(x) = B_t$ if $e \in E_t$. All other edges e of G have an infinite capacity and a (randomly chosen) reasonable latency function $l_e(x)$.

We let g be the flow of rate r that respects the capacities of the edges in $E_s \cup E_t$ and minimizes $\text{Pot}(g) = \sum_{e \in E} \int_0^{g_e} l_e(x)dx$. Such a flow g can be computed in strongly polynomial time (see e.g., [35]). The subnetwork G' of G is simply G_g, namely, the subnetwork that includes only the edges used by g. It could have been that g is not a Nash flow of (G, r, l), due to the capacity constraints on the edges of $E_s \cup E_t$. However, since g is the expression's $\text{Pot}(g)$ minimizer, for any $u \in V_s$ and $v \in V_t$, and any pair of $s - t$ paths P, P' going through u and v, if $g_P > 0$, then $l_P(g) \leq l_{P'}(g)$.

We next adjust the proof of [21]' lemma 5 (or lemma 3.16 above), and show that for any $s - t$ path P used by g, $l_P(g) \leq B_s + B_t + 7\epsilon_1$. To prove this, we let $P = (s, u, \ldots, v, t)$ be the $s - t$ path used by g that maximizes $l_P(g)$. We show the existence of a path $P' = (s, u, \ldots, v, t)$ in G of latency $l_{P'}(g) \leq B_s + B_t + 7\epsilon_1$. Therefore, since g is a minimizer of $\text{Pot}(g)$, the latency of the maximum latency g-used path P, and thus the latency of any other g-used $s - t$ path, is at most $B_s + B_t + 7\epsilon_1$, i.e., g satisfies (iii). Moreover, since for any $s - t$ path P, $l_P(g) \geq B_s + B_t$, g is an $7\epsilon_1$-Nash flow in G'.

Let $P = (s, u, \ldots, v, t)$ be the $s - t$ path used by g that maximizes $l_P(g)$. To show the existence of a path $P' = (s, u, \ldots, v, t)$ in G of latency $l_{P'}(g) \leq B_s + B_t + 7\epsilon_1$, we start from $S_0 = \{u\}$ and grow a sequence of vertex sets $S_0 \subseteq S_1 \subseteq \cdots \subseteq S_{i^*}$, stopping when $|\Gamma(S_{i^*})| \geq 3n/5$ for the first time.

We use the expansion properties of G, and condition (3), on the distribution of $\mathcal{B}$, in the definition of good networks, and show that these sets grow exponentially fast, and thus, $i^* \leq \ln n$, with high probability. Moreover, we show[4] that there are edges of latency $\epsilon_1 + o(1)$ from $S_0 = \{u\}$ to each vertex

[4]The intuition is that if among the edges e incident to $V_s \cup V_t$, we keep only those with $b_e \leq \epsilon_1$, and among all the remaining edges e, we keep only those with $b_e \leq \epsilon_1 / \ln n$, then due to condition (3) on the distribution of $\mathcal{B}$ $\big(\forall \eta > 0,\ \mathbb{P}[\mathcal{B} \leq \eta/\ln n]np = \omega(1)\big)$, a good network G remains an expander.

of S_1, and edges of latency $\epsilon_1/\ln n+o(1/\ln n)$ from S_i to each vertex of S_{i+1}, for all $i=1,\dots,i^*-1$. Thus, there is a path of latency at most $2\epsilon_1+o(1)$ from u to each vertex of S_{i^*}. Similarly, we start from $T_0=\{v\}$ and grow a sequence of vertex sets $T_0\subseteq T_1\subseteq\cdots\subseteq T_{j^*}$, stopping when $|\Gamma(T_{j^*})|\geq 3n/5$ for the first time. By exactly the same reasoning, we establish the existence of a path of latency at most $2\epsilon_1+o(1)$ from each vertex of T_{j^*} to v.

Finally, since $|\Gamma(S_{i^*})|\geq 3n/5$ and $|\Gamma(T_{j^*})|\geq 3n/5$, the neighborhoods of S_{i^*} and T_{j^*} contain at least $n/10$ vertices in common. With high probability, most of these vertices can be reached from S_{i^*} and from T_{j^*} using edges of latency $\epsilon_1+o(1)$. Putting everything together, we find a $u-v$ path (in fact, many of them) of length $\mathcal{O}(\ln n)$ and latency at most $6\epsilon_1+o(1)\leq 7\epsilon_1$.

For completeness, we next give a detailed proof, by adjusting the arguments in the proof of lemma 3.16. For convenience, for each vertex x, we let $d_s(x)$ (resp. $d_t(x)$) be the latency (wrt g) of the shortest latency path from s to x (resp. from x to t). Also, for any $\delta>0$, we let $P_b(\delta)\equiv\mathbb{P}[\mathcal{B}\leq\delta]$ denote the probability that the additive term of a reasonable latency is at most δ. Recall also that by hypothesis, there exists a constant $\rho'>0$, such that for all $e\in E(H_0)$, $f_e\leq\rho'$. Hence, the total flow through G (and through H_0) is $r\leq\rho' n_+$.

At the conceptual level, the proof proceeds as explained above. We start with $S_0=\{u\}$. By hypothesis, the flow entering u is at most ρ'. By the expansion property of good networks and by Chernoff bounds[5]. Now, since , with high probability, there are at least $P_b(\epsilon_1)np/4$ edges e adjacent to u with $b_e\leq\epsilon_1$. At most half of these edges have flow greater than $\frac{8\rho'}{P_b(\epsilon_1)np}$, thus there are at least $P_b(\epsilon_1)np/8$ edges adjacent to u with latency, wrt g, less than $\frac{8A_{\max}\rho'}{P_b(\epsilon_1)np}+\epsilon_1$. We now let $d_1=B_s+\frac{8A_{\max}\rho'}{P_b(\epsilon_1)np}+\epsilon_1$ and $S_1=\{x\in V: d_s(x)\leq d_1\}$. By the discussion above, $|S_1|\geq P_b(\epsilon_1)np/8$.

We now inductively define a sequence of vertex sets S_i and upper bounds d_i on the latency of the vertices in S_i from s, such that $S_i\subseteq S_{i+1}$ and $d_i<d_{i+1}$. This sequence stops the first time that $|\Gamma(S_i)|\geq 3n/5$. We inductively assume that the vertex set S_i and the upper bound d_i on the latency of the vertices in S_i are defined, and that $|\Gamma(S_i)|<3n/5$. By the expansion property of good networks $|\Gamma(S_i)\setminus S_i|\geq np|S_i|/3$, for sufficiently large n. Thus, with probability at least $1-\mathrm{e}^{-P_b(\epsilon_1/\ln n)np|S_i|/24}$, there are at least $P_b(\frac{\epsilon_1}{\ln n})np|S_i|/6$ vertices outside S_i that are connected to a vertex in S_i by an edge e with $b_e\leq\epsilon_1/\ln n$ (we use of Chernoff bounds again[6]). Let

[5]We repeatedly use the following form of the Chernoff bound (Let $X_1,\dots,X_k$ be random variables independently distributed in $\{0,1\}$, and let $X=\sum_{i=1}^k X_i$. Then, for all $\epsilon\in(0,1)$, $\mathbb{P}[X<(1-\epsilon)\mathbb{E}[X]]\leq\mathrm{e}^{-\epsilon^2\,\mathbb{E}[X]/2}$, where e is the basis of natural logarithms. The latter means that $\mathbb{P}[X\geq(1-\epsilon)\mathbb{E}[X]]\geq 1-\mathrm{e}^{-\epsilon^2\,\mathbb{E}[X]/2}$. Here we have that $X=\{e\in\Gamma(u)\text{ with }b_e\leq\epsilon_1\}$, $(1-\epsilon)\mathbb{E}[X]=P_b(\epsilon_1)np/4$ and from relation 3.4 we have that $\mathbb{E}[X]=P_b(\epsilon_1)np$. We should also have in mind that $np\geq\ln n$.

[6]Here we have that $X=\{e\in\Gamma(S_i)\setminus S_i\text{ with }b_e\leq\epsilon_1/\ln n\}$, $(1-\epsilon)\mathbb{E}[X]=\frac{1}{2}P_b(\epsilon_1/\ln n)np|S_i|/3$ and $\mathbb{E}[X]=P_b(\epsilon_1/\ln n)np|S_i|/3$.

S'_i be the set of such vertices, and let E_i be the set of edges that for each vertex $v \in S'_i$, includes a unique edge $e \in E_i$ with $b_e \leq \epsilon_1/\ln n$ connecting v to a vertex in S_i. Since the flow g may be assumed to be acyclic, a volume $r \leq \rho' n_+$ of flow is routed through the cut $(S_i, V \setminus S_i)$. Then, at most half of the edges in E_i have flow greater than $2\rho' n_+/|S'_i|$. Consequently, at least half of the vertices $v \in S'_i$ have latency from s:

$$\begin{aligned} d_s(x) &\leq d_i + \frac{\epsilon_1}{\ln n} + A_{\max}\frac{2\rho' n_+}{|S'_i|} \\ &\leq d_i + \frac{\epsilon_1}{\ln n} + \frac{12 A_{\max}\rho' n_+}{P_b(\frac{\epsilon_1}{\ln n})np|S_i|} \end{aligned}$$

Thus, we define the next latency upper bound d_{i+1} in the sequence as:

$$d_{i+1} = d_i + \frac{\epsilon_1}{\ln n} + \frac{12 A_{\max}\rho' n_+}{P_b(\frac{\epsilon_1}{\ln n})np|S_i|}\,,$$

and we let $S_{i+1} = \{x \in V(G) | d_s(x) \leq d_{i+1}\}$. By the discussion above, and using the inductive definition of S_i's, we obtain that:

$$\begin{aligned} |S_{i+1}| &\geq \Big(\tfrac{1}{12}P_b(\epsilon_1/\ln n)np + 1\Big)|S_i| \\ &\geq \Big(\tfrac{1}{12}P_b(\epsilon_1/\ln n)np + 1\Big)^i |S_1| \end{aligned}$$

We recall that i^* is the first index i such that $|\Gamma(S_i)| \geq 3n/5$. Then, the inequality above implies that:

$$i^* \leq \frac{\ln\left(3n/(5|S_1|)\right)}{\ln\left(\frac{1}{12}P_b(\epsilon_1/\ln n)np + 1\right)} \leq \frac{\ln\left(24n/(5P_b(\epsilon_1)np)\right)}{\ln\left(\frac{1}{12}P_b(\epsilon_1/\ln n)np + 1\right)}$$

Using that $pn \geq \ln n$ and that $P_b(\epsilon_1/\ln n)np = \omega(1)$, the inequality above implies that $i^* \leq \ln n$, for sufficiently large n.

Therefore, we obtain an upper bound on the latency from s of any vertex in S_{i^*}:

$$
\begin{aligned}
d_{i^*} &\leq d_0 + i^* \frac{\epsilon_1}{\ln n} + \sum_{i=1}^{i^*} \frac{12 A_{\max} \rho' n_+}{P_b(\frac{\epsilon_1}{\ln n}) np |S_i|} \\
&\leq d_1 + \frac{\epsilon_1}{\ln n} \ln n + \sum_{i=1}^{\ln n} \frac{12 A_{\max} \rho' n_+}{P_b(\frac{\epsilon_1}{\ln n}) np \left(\frac{1}{12} P_b(\frac{\epsilon_1}{\ln n}) np + 1\right)^i |S_1|} \\
&= d_1 + \epsilon_1 + \frac{12 A_{\max} \rho' n_+}{P_b(\frac{\epsilon_1}{\ln n}) np |S_1|} \sum_{i=1}^{\ln n} \left(\frac{1}{12} P_b(\frac{\epsilon_1}{\ln n}) np + 1\right)^{-i} \\
&\leq \left(B_s + \frac{8 A_{\max} \rho'}{P_b(\epsilon_1) np} + \epsilon_1\right) + \epsilon_1 + \frac{96 A_{\max} \rho' n_+}{P_b(\frac{\epsilon_1}{\ln n}) P_b(\epsilon_1) (np)^2} \sum_{i=1}^{\infty} 2^{-i} \\
&\leq B_s + 2\epsilon_1 + \frac{8 A_{\max} \rho'}{P_b(\epsilon_1) np} + \frac{144 A_{\max} \rho' k}{P_b(\frac{\epsilon_1}{\ln n}) P_b(\epsilon_1) np}
\end{aligned}
$$

For the penultimate inequality, we use that $P_b(\epsilon_1/\ln n)np = \omega(1)$, which implies that $1 + P_b(\epsilon_1/\ln n)np/12 \geq 2$, for n sufficiently large. For the last inequality, we use that $n_+ \leq 3knp/2$, for some constant $k > 0$, by hypothesis.

Moreover, we observe that the probability that the above construction fails is at most:

$$
\begin{aligned}
\sum_{i=1}^{i^*} \mathrm{e}^{-P_b(\epsilon_1/\ln n)np|S_i|/24} &\leq \sum_{i=1}^{i^*} \mathrm{e}^{-\left(\frac{1}{12} P_b(\epsilon_1/\ln n)np + 1\right)^i |S_1|/24} \\
&\leq \ln n \, \mathrm{e}^{-\left(\frac{1}{12} P_b(\epsilon_1/\ln n)np + 1\right) P_b(\epsilon_1)np/192}
\end{aligned}
$$

Therefore, the construction above succeeds with high probability.

Similarly, we start from $T_0 = \{v\}$, and inductively define a sequence of vertex sets $T_0 \subseteq T_1 \subseteq \cdots \subseteq T_{j^*}$, and a sequence of upper bounds $d'_0 < d'_1 < \cdots < d'_{j^*}$ on the latency from t of the vertices in each T_j. We let $T_j = \{x \in V(G) | d_t(x) \leq d'_j\}$. The sequence stops as soon as $|\Gamma(T_j)| \geq 3n/5$ for the first time. Namely, j^* is the first index with $|\Gamma(T_{j^*})| \geq 3n/5$. Using exactly the same arguments, we can show that with high probability, we have that $j^* \leq \ln n$, and that:

$$
d'_{j^*} \leq B_t + 2\epsilon_1 + \frac{8 A_{\max} \rho'}{P_b(\epsilon_1) np} + \frac{144 A_{\max} \rho' k}{P_b(\frac{\epsilon_1}{\ln n}) P_b(\epsilon_1) np}
$$

Wlog we assume that $S_{i^*} \cap T_{j^*} = \emptyset$. Since $|\Gamma(S_{i^*})| + |\Gamma(T_{j^*})| \geq 6n/5$, there are at least $n/5$ edge disjoint paths of length at most 2 between S_{i^*} and T_{j^*}. Furthermore, by Chernoff bounds[7], with high probability, there

[7] Here we have that $X = \{$ Edge disjoint paths of length at most 2 between S_{i^*} and T_{j^*} with both edges e on the path having $b_e \leq \epsilon_1$ $\}$, $(1-\epsilon)\mathbb{E}[X] = \frac{5}{12} P_b(\epsilon_1)^2 n/5$ and $\mathbb{E}[X] = P_b(\epsilon_1)^2 n/5$.

are at least $P_b(\epsilon_1)^2 n/12$ such paths with both edges e on the path having $b_e \leq \epsilon_1$. At most half of these paths have flow more than $2\frac{12\rho' n_+}{P_b(\epsilon_1)^2 n}$ and thus there is a path from a vertex of S_{i^*} to a vertex of T_{j^*} that costs at most $2\epsilon_1 + 2A_{\max}\frac{24\rho' n_+}{P_b(\epsilon_1)^2 n}$.

Putting everything together, we have that there is a path P' that starts from s, moves to u, goes through vertices of the sequence $S_1, \ldots, S_{i^*}$, proceeds to a vertex of $\Gamma(S_{i^*}) \cap \Gamma(T_{j^*})$, and from there, continues through vertices of the sequence $T_{j^*}, \ldots, T_1$, until finally reaches v, and then t. The latency of this path is:

$$l_{P'}(g) \leq B_s + B_t + 6\epsilon_1 + 2\Big(\frac{8A_{\max}\rho'}{P_b(\epsilon_1)np} + \frac{48A_{\max}\rho' k}{P_b(\frac{\epsilon_1}{\ln n})P_b(\epsilon_1)np}\Big) + \frac{48A_{\max}\rho' n_+}{P_b(\epsilon_1)^2 n}$$

We recall that since the flow g is a the minimizer of $\mathrm{Pot}(g)$, for any g-used path $P = (s, u, \ldots, v, t)$, $l_P(g) \leq l_{P'}(g)$. Thus we obtain that any g-used path $P = (s, u, \ldots, v, t)$ has latency:

$$l_{P}(g) \leq B_s + B_t + 6\epsilon_1 + 2\Big(\frac{8A_{\max}\rho'}{P_b(\epsilon_1)np} + \frac{48A_{\max}\rho' k}{P_b(\frac{\epsilon_1}{\ln n})P_b(\epsilon_1)np}\Big) + \frac{48A_{\max}\rho' n_+}{P_b(\epsilon_1)^2 n}$$

Using the hypothesis that $n_+ \leq 3knp/2$, for constant $k > 0$, and that $P_b(\epsilon_1/\ln n)np = \omega(1)$, which is condition (3) in the definition of good networks, we obtain that for any constant $\epsilon_1 > 0$, $l_P(g) \leq B_s + B_t + 7\epsilon_1$, for sufficiently large n. □

Grouping the Neighbors of s and t. Let us now consider the entire network G and the entire subnetwork H_0 of G_0. Lemma 5.7 can be applied only to subsets of edges in $E_s(H_0)$ and in $E_t(H_0)$ that have (almost) the same latency under f. Hence, we partition the neighbors of s and the neighbors of t into classes V_s^i and V_t^j according to their latency. For convenience, we let $\epsilon_2 = \varepsilon/6$, i.e., f is an ϵ_2-Nash flow, and $L \equiv L_f(H_0)$. By theorem 5.6, applied with error $\epsilon_2 = \varepsilon/6$, there exists a ρ such that for all $e \in E(H_0)$, $0 < f_e \leq \rho + \epsilon_2$. Hence, $L \leq 2A_{\max}(\rho+\epsilon_2) + 2B_{\max}$ is bounded by a constant.

We partition the interval $[0, L]$ into $\kappa = \lceil L/\epsilon_2 \rceil$ subintervals, where the i-th subinterval is $I^i = (i\epsilon_2, (i+1)\epsilon_2]$, $i = 0, \ldots, \kappa - 1$. We partition the vertices of V_s (resp. of V_t) that receive positive flow by f into κ classes V_s^i (resp. V_t^i), $i = 0, \ldots, \kappa - 1$. Precisely, a vertex $x \in V_s$ (resp. $x \in V_t$), connected to s (resp. to t) by the edge $e_x = (s, x)$ (resp. $e_x = (x, t)$), is in the class V_s^i (resp. in the class V_t^i), if $l_{e_x}(f_{e_x}) \in I_i$. If a vertex $x \in V_s$ (resp. $x \in V_t$) does not receive any flow from f, x is removed from G and does not belong to any class. Hence, from now on, we assume that all neighbors of s and t receive positive flow from f, and that $V_s^0, \ldots, V_s^{\kappa-1}$ (resp. $V_t^0, \ldots, V_t^{\kappa-1}$) is a partitioning of V_s (resp. V_t). In exactly the same way, we partition the edges of E_s (resp. of E_t) used by f into k classes E_s^i (resp. E_t^i), $i = 0, \dots, \kappa - 1$.

To find out which parts of the subnetwork H_0 will be connected through the intermediate subnetwork of G, using the construction of lemma 5.7, we further classify the vertices of V_s^i and V_t^j based on the neighbors of t and on the neighbors of s, respectively, to which they are connected by f-used edges in the subnetwork H_0. In particular, a vertex $u \in V_s^i$ belongs to the classes $V_s^{(i,j)}$, for all $j \in \{0, \dots, \kappa - 1\}$ such that there is a vertex $v \in V_t^j$ with $f_{\{u,v\}} > 0$. Similarly, a vertex $v \in V_t^j$ belongs to the classes $V_t^{(i,j)}$, for all $i \in \{0, \dots, \kappa - 1\}$ such that there is a vertex $u \in V_s^i$ with $f_{\{u,v\}} > 0$. We note that a vertex $u \in V_s^i$ (resp. $v \in V_t^j$) may belong to many different classes $V_s^{(i,j)}$ (resp. to $V_t^{(i,j)}$), and that the class $V_s^{(i,j)}$ is non-empty iff the class $V_t^{(i,j)}$ is non-empty, i.e., non-empty classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$ appear in pairs. We let $k \leq \kappa^2$ be the number of pairs (i, j) for which $V_s^{(i,j)}$ and $V_t^{(i,j)}$ are non-empty. We note that k is a constant, i.e., does not depend on $|V|$ and r. We let $E_s^{(i,j)}$ be the set of edges connecting s to the vertices in $V_s^{(i,j)}$ and $E_t^{(i,j)}$ be the set of edges connecting t to the vertices in $V_t^{(i,j)}$.

Building the Intermediate Subnetworks of G. The last step is to replace the 0-latency simplified parts connecting the vertices of each pair of classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$ in H_0 with a subnetwork of G_m. To this end, we randomly partition the set V_m of intermediate vertices of G into k subsets, each of cardinality (roughly) $|V_m|/k$, and associate a different such subset $V_m^{(i,j)}$ with any pair of non-empty classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$. For each pair (i, j) for which the classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$ are non-empty, we consider the induced subnetwork $G^{(i,j)} \equiv G[\{s, t\} \cup V_s^{(i,j)} \cup V_m^{(i,j)} \cup V_t^{(i,j)}]$, which is a $(n/k, p, 1)$-good network, by condition (4) in the definition of good networks, and because G is a (n, p, k)-good network. Therefore, we can apply lemma 5.7 to $G^{(i,j)}$, with $H_0^{(i,j)} \equiv H_0[\{s, t\} \cup V_s^{(i,j)} \cup V_t^{(i,j)}]$ in the role of H_0, the restriction $f^{(i,j)}$ of f to $H_0^{(i,j)}$ in the role of the flow f, and $\rho' = \rho + \epsilon_2$. Moreover, we let $B_s^{(i,j)} = \max_{e \in E_s^{(i,j)}} l_e(f_e)$ and $B_t^{(i,j)} = \max_{e \in E_t^{(i,j)}} l_e(f_e)$ correspond to B_s and B_t, and introduce constant latencies $l'_e(x) = B_s^{(i,j)}$ for all $e \in E_s^{(i,j)}$ and $l'_e(x) = B_t^{(i,j)}$ for all $e \in E_t^{(i,j)}$, as required by lemma 5.7. Thus, we obtain, with high probability, a subnetwork $H^{(i,j)}$ of $G^{(i,j)}$ and a flow $g^{(i,j)}$ that routes as much flow as $f^{(i,j)}$ on all edges of $E_s^{(i,j)} \cup E_t^{(i,j)}$, and satisfies the conclusion of lemma 5.7, if we keep in $H^{(i,j)}$ the constant latencies $l'_e(x)$ for all $e \in E_s^{(i,j)} \cup E_t^{(i,j)}$.

The final outcome is the union of the subnetworks $H^{(i,j)}$, denoted H (H has the latency functions of the original instance G), and the union of the flows $g^{(i,j)}$, denoted g, where the union is taken over all k pairs (i, j) for which the classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$ are non-empty. By construction, all edges of H are used by g. We obtain lemma 5.8 by showing that if $\epsilon_1 = \varepsilon/42$ and $\epsilon_2 = \varepsilon/6$, the flow g is an ε-Nash flow of (H, r, l), and satisfies $L_g(H) \leq L_f(H_0) + \varepsilon/2$.

Lemma 5.8. *Let any $\varepsilon > 0$, let $k = \lceil 12(A_{\max}(\rho + \varepsilon) + B_{\max})/\varepsilon \rceil^2$, let $G(V, E)$ be an (n, p, k)-good network, let $r > 0$, let H_0 be any subnetwork of the 0-latency simplification of G, and let f be an $(\varepsilon/6)$-Nash flow of (H_0, r, l) for which there exists a constant $\rho' > 0$, such that for all $e \in E(H_0)$, $0 < f_e \leq \rho'$. Then, with high probability, wrt the random choice of the latency functions of G, we can compute in $\mathrm{poly}(|V|)$ time a subnetwork H of G and an ε-Nash flow g of (H, r, l) with $L_g(H) \leq L_f(H_0) + \varepsilon/2$.*

Proof. We consider the subnetwork H (with the original latency functions of G), computed as the union of subnetworks $H^{(i,j)}$, and the flow g, computed as the union of the flows $g^{(i,j)}$, where the union is taken over all k pairs (i, j) for which the classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$ are non-empty. We recall that by construction, all edges of H are used by g. We show that if $\epsilon_1 = \varepsilon/42$ and $\epsilon_2 = \varepsilon/6$, the flow g is an ε-Nash flow of (H, r, l), and satisfies $L_g(H) \leq L_f(H_0) + \varepsilon/2$. We stress that the edge and path latencies here are calculated with respect to the original latency functions of G and under the edge congestion induced by the flow g (or the flow f).

For convenience, we let $B^{(i,j)} = B_s^{(i,j)} + B_t^{(i,j)}$ for any pair of non-empty classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$. Since the difference in the latency of any edges in the same group is at most ϵ_2, we obtain that for any edge $e \in E_s^{(i,j)}$, $B_s^{(i,j)} - \epsilon_2 \leq l_e(f_e) \leq B_s^{(i,j)}$, and similarly, that for any edge $e \in E_t^{(i,j)}$, $B_t^{(i,j)} - \epsilon_2 \leq l_e(f_e) \leq B_t^{(i,j)}$. Therefore, since H_0 is a 0-latency simplified network, and since by hypothesis, all the edges of H_0 are used by f, for any pair of non-empty classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$, and for any $s - t$ path p going through a vertex of $V_s^{(i,j)}$ and a vertex of $V_t^{(i,j)}$,

$$B^{(i,j)} - 2\epsilon_2 \leq l_p(f) \leq B^{(i,j)}$$

Moreover, since f is an ϵ_2-Nash flow of (H_0, r, l), for any $s - t$ path $P \in \mathcal{P}_{H_0}$,

$$L_f(H_0) - \epsilon_2 \leq l_P(f) \leq L_f(H_0)$$

Combining the two inequalities above, we obtain that for any pair of non-empty classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$,

$$B^{(i,j)} - 2\epsilon_2 \leq L_f(H_0) \leq B^{(i,j)} + \epsilon_2 \tag{5.19}$$

As for the flow g, by construction, we have that $g_e = f_e$ for all edges $e \in E_s \cup E_t$. Therefore, for any edge $e \in E_s^{(i,j)}$, $B_s^{(i,j)} - \epsilon_2 \leq l_e(g_e) \leq B_s^{(i,j)}$, and similarly, for any edge $e \in E_t^{(i,j)}$, $B_t^{(i,j)} - \epsilon_2 \leq l_e(g_e) \leq B_t^{(i,j)}$. Thus, by lemma 5.7, and since all the edges of any subnetwork $H^{(i,j)}$ are used by g, for any $s-t$ path P in the subnetwork $H^{(i,j)}$, $B^{(i,j)} - 2\epsilon_2 \leq l_P(g) \leq B^{(i,j)} + 7\epsilon_1$. Using relation (5.19), we obtain that for any subnetwork $H^{(i,j)}$ and any $s-t$ path P of $H^{(i,j)}$,

$$L_f(H_0) - 3\epsilon_2 \leq l_P(g) \leq L_f(H_0) + 2\epsilon_2 + 7\epsilon_1 \tag{5.20}$$

Furthermore, we recall that the subnetworks $H^{(i,j)}$ only have in common the vertices s and t, and possibly some vertices of $V_s \cup V_t$ and some edges of $E_s \cup E_t$. They have neither any other vertices in common, nor any edges connecting vertices in the intermediate parts of different subnetworks $H^{(i,j)}$ and $H^{(i',j')}$. Hence, any $s-t$ path p of H passes through a single subnetwork $H^{(i,j)}$. Therefore, and since by construction, all the edges and the paths of H are used by g, relation (5.20) holds for any $s-t$ path P of H.

Thus, we have shown that g is a $(5\epsilon_2 + 7\epsilon_1)$-Nash flow of (H, r, l), and that $L_g(H) \leq L_f(H_0) + 2\epsilon_2 + 7\epsilon_1$. Using $\epsilon_2 = \varepsilon/6$ and $\epsilon_1 = \varepsilon/42$, we obtain the performance guarantees of g as stated in lemma 5.8. $\square$

Chapter 6

Conclusions

The motivation of this research was to provide simple ways of improving network performance, by exploiting the essence of the Braess's Paradox, which as we have shown, occurs in "random" networks with high probability.

We have already seen how difficult it is to detect the so called best subnetwork, which is the one that minimizes the equilibrium latency among all subnetworks of the original network. Thus, it is of a notable importance trying to give an approximating solution to the best subnetwork equilibrium latency problem, that is an approximation of the best subnetwork and its equilibrium latency, given the difficulty of detecting it.

Chapters 1 to 3 presented the basic issues of the paradox, which were widely used in the Chapters that followed.

Chapter 4 presented approximation algorithms for the best subnetwork equilibrium latency problem, in random networks with linear latencies and polynomially many paths, each of polylogarithmic length. By these theorems, quasipolynomial running times may be achieved, for traffic rates of the size $\mathcal{O}(1)$ (or more generally $\mathcal{O}(\text{poly}(\ln\ln m))$) or even for traffic rates up to $\mathcal{O}(\text{poly}(\ln m))$, where m is the total number of the original network's edges.

Chapter 5 focused on the class of the so-called *good* selfish routing instances. These are instances that have those properties that were used exactly by [20] and [21] in order to demonstrate the occurrence of Braess's Paradox in random networks with high probability. An improvement on the best known running time for approximating the best subnetwork equilibrium latency problem in this kind of networks is given, which was originally presented in [32]. Overall, this approximation scheme runs in polynomial time if the traffic rate is $\mathcal{O}(poly(\ln\ln n))$ and in quasipolynomial time for traffic rates up to $\mathcal{O}(poly(\ln n))$, where n is the total number of the original network's vertices.

Bibliography

[1] N. Nisan, T. Roughgarden, É. Tardos and V. V. Vazirani, **Algorithmic Game Theory**. Cambridge, 2007.

[2] A.C. Kaporis and P.G. Spirakis, **Selfish splittable flows and NP-completeness**. Computer Science Review 5, 209 - 228, 2011.

[3] T. Roughgarden, **The Price of Anarchy is Independent of the Network Topology**. In Proc. of the 34th ACM Symp. on Theory of Computing (STOC '02), pages 428 - 437, 2002.

[4] T. Roughgarden, **Designing Networks for Selfish Users is Hard**. Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science, pages 472 - 481, 2001.

[5] T. Roughgarden, **Selfish Routing and the Price of Anarchy**. MIT Press, 2005.

[6] J.G. Wardrop, **Some Theoretical Aspects of Road Traffic Research**. In Proceedings of the Institute of Civil Engineers, Pt. II, volume 1, pages 325 - 378, 1952.

[7] M. Minoux, **A polynomial Algorithm for Minimum Quadratic Cost Flow Problems**. European Journal of Operational Research 18 (3), pages 377 - 387, 1984.

[8] J. Edmonds, R.M. Karp, **Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems**. Journal of the ACM 19 (2), pages 248 - 264, 1972.

[9] D.S. Hochbaum, J.G. Shanthikumar, **Convex Separable Optimization is not much Harder than Linear Optimization**. Journal of the ACM 37 (4), pages 843 - 862, 1990.

[10] E. Koutsoupias, C. Papadimitriou, **Worst-case Equilibria**. In Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science, STACS '99, Springer-Verlag, pages 404 - 413, 1999.

[11] T. Roughgarden, É. Tardos, **How Bad is Selfish Routing?** Journal of the ACM 49 (2), pages 236 - 259, Preliminary version in FOCS '00, 2002.

[12] D. Braess, **Über ein paradox aus der Verkehrsplanung**. Unternehmensforschung 12, pages 258 - 268, 1968.

[13] H. Lin, T. Roughgarden, É. Tardos, A. Walkover, **Stronger Bounds on Braess's Paradox and the Maximum Latency of Selfish Routing**. Siam J. Discrete Math, Vol. 25, No. 4. pages 1667 - 1686, 2011.

[14] T. Roughgarden, **On the Severity of Braess's Paradox, Designing Networks for Selfish Users is Hard**. Journal of the Computer and System Sciences 72 (5), pages 922 - 953, 2006.

[15] R. Tarjan, **Data Structures and Network Algorithms**. SIAM, 1983.

[16] D. Fotakis, A.C. Kaporis, P.G. Spirakis, **Efficient Methods for Selfish Network Design**. In: Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikoletseas, Wolfgang Thomas (Eds.), ICALP (2), in: Lecture Notes in Computer Science, vol. 5556, Springer, pages 459 - 471. 2009.

[17] O.L. Mangasarian, **Uniqueness of Solution on Linear Programming**. Linear Algebra and Applicat., 25: 151 - 162, 1979.

[18] I. Althöfer, **On Sparse Approximations to Randomized Strategies and Convex Combinations**. Linear Algebra and Applicat., 99: 339 - 355, 1994.

[19] M.K. Kozlov, S.P. Tarasov, L.G. Khachian, **Polynomial Solvability of Convex Quadratic Programming**. Soviet Mathematics., Doklady 20: 1108 - 1111, 1979.

[20] G. Valiant, T. Roughgarden, **Braess's Paradox in Large Random Graphs**. Random Structures and Algorithms, 37: 495 - 515, 2010.

[21] F. Chung, S.J. Young, **Braess's Paradox in Large Sparse Graphs**. 6th Workshop on Internet and Network Economics, 2010.

[22] F. Chung, S.J. Young, W. Zhao, **Braess's Paradox in Expanders**. Random Structures and Algorithms, v.41 n.4: 451 - 468, 2012.

[23] H. Lin, T. Roughgarden, É. Tardos, **A Stronger Bound on Braess's Paradox**. http://www.eecs.berkeley.edu/~henrylin/soda04.ppt

[24] M.E.J. Newman, S. H. Strogatz, D.J. Watts, **Random Graphs with Arbitrary Degree Distributions and their Applications**. Physical Review E 64, 026118, 2001.

[25] P.Erdös, A. Rényi, **On the Evolution of Random Graphs**. Publications of the Mathematical Institute of the Hungarian Academy of Sciences 5: 17-61, 1960.

[26] R. Andersen, F. Chung, L. Lu, **Drawing power law graphs**. In Proceedings of the 12th Symposium on Graph Drawing, 2004.

[27] M. Faloutsos, P. Faloutsos, C. Faloutsos, **On Power-Law Relationships of the Internet Topology**. In Proc. SigComm. ACM, 1999.

[28] C. Gkantsidis, M. Mihail, A. Saberi, **Conductance and Congestion in Power Law Graphs**. 2003.

[29] M. Makisumi, **Expanders and Related Results**. 2011.

[30] M. Mihail, C. Papadimitriou, A. Saberi, **On Certain Connectivity Properties of the Internet Topology**. 2003.

[31] S. Barman, **Approximating Caratheodory's Theorem and Nash Equilibria**. 2014.

[32] D. Fotakis, A.C. Kaporis, T. Lianeas, P.G. Spirakis, **Resolving Braess's Paradox in Random Networks**. 2014.

[33] B. Bolobás, **Random Graphs**. 2nd Ed., Cambridge Studies in Advanced Mathematics, no. 73, Cambridge University Press, 2001.

[34] R. Cole, Y. Dodis, T. Roughgarden, **Pricing Network Edges for Heterogeneous Selfish Users**. STOC, ACM, pp. 521530, 2003.

[35] L. Végh, **Strongly Polynomial Algorithm for a Class of Minimum-Cost Flow Problems with Separable Convex Objectives**. STOC, ACM, pp. 2740, 2012.

List of Figures