Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Bachelor's Thesis

Network Security Group, Department of Computer Science, ETH Zurich

Critical Networking: A Theoretical Analysis of GMA

by Lucas Dodgson

Spring 2021

ETH student ID: 18-922-260
E-mail address: lucas.dodgson@inf.ethz.ch

Supervisor: Prof. Dr. Adrian Perrig
Advisor: Giacomo Giuliari

Date of submission: August 23, 2021

Abstract

Critical networking applications require strong packet delivery guarantees, which the current Internet's best-effort paradigm cannot provide. One method for providing stronger guarantees is to create bandwidth allocations for network paths, effectively protecting communication from external congestion. However, many challenges arise in the implementation and enforcement of these allocations. In a recent publication, the Network Security Group at ETH Zürich formalized the constraints of this allocation process in the distributed path allocation ($\mathrm{PA}^{\mathrm{dist}}$) problem and introduced the global myopic allocation (GMA) algorithm which solves $\mathrm{PA}^{\mathrm{dist}}$.

In this thesis we answer many of the open questions regarding the performance of GMA while analyzing various theoretical aspects of it. First, we derive a new fairness notion, NBP-fairness, from the GMA algorithm which is based on the local policies of nodes and is applicable to the setting of $\mathrm{PA}^{\mathrm{dist}}$. Secondly, we prove that GMA is not a globally optimal algorithm, although we do find GMA to be globally optimal in the important subset of graphs in which the limiting factor is inter-node link capacity. Finally, we find an optimal method for determining how to best initialize a node's policy. Unfortunately, this approach suffers from the tragedy of the commons. Therefore, we discuss some possible solutions to this problem as well as their tradeoffs.

Acknowledgements

I would like to thank Giacomo Giuliari for all of his input, advice, guidance, and feedback. Additionally, I would like to thank Prof. Dr. Adrian Perrig for supervising this thesis, Marc Wyss for the constructive discussions on a variety of the algorithms properties, Simon Scherrer for his helpful input on the game theoretic tradeoffs, and finally, my parents and Leonardo Galli for provided their feedback along the way.

Contents

Abstract iii

Acknowledgements v

1 Introduction 1
- 1.1 Contributions . . . 2
- 1.2 Organization . . . 2

2 Background 5
- 2.1 Distributed path allocation problem . . . 5
  - 2.1.1 Allocation graphs . . . 5
  - 2.1.2 (Distributed) path allocation problem . . . 6
- 2.2 Global myopic allocation algorithm . . . 6
- 2.3 Fairness . . . 7
  - 2.3.1 Max-min fairness . . . 8
  - 2.3.2 Proportional fairness . . . 8
  - 2.3.3 Other fairness notions . . . 9
- 2.4 Game theory . . . 9

3 Analysis of GMA's Fairness 11
- 3.1 Problem statement . . . 11
- 3.2 NBP-fairness . . . 11
- 3.3 Reasoning behind NBP-fairness . . . 12
  - 3.3.1 Bounds implied by NBP-fairness . . . 13
- 3.4 Comparison to other fairness notions . . . 15
- 3.5 GMA is NBP-fair . . . 16
  - 3.5.1 Properties **(F1)** and **(F2)** . . . 16
  - 3.5.2 Property **(F3)** . . . 16

4 Global Optimality of GMA 23
- 4.1 Problem statement . . . 23
- 4.2 Definitions of global optimality . . . 23
- 4.3 GMA is not globally optimal . . . 25
  - 4.3.1 GMA and Opt 1 . . . 25
  - 4.3.2 GMA and Opt 2 . . . 28
- 4.4 Subset of graphs for which GMA is globally optimal . . . 30
  - 4.4.1 Case 1: i is a local interface . . . 32

4.4.2 Case 2: j is a local interface 33
4.4.3 Case 3: Neither i nor j is a local interface 34

5 Optimal Initialization of Allocation Matrices **37**
5.1 Problem statement 37
5.2 Setup 37
5.3 Maximize the sum of allocations 37
5.4 Maximizing the product of allocations 39
5.4.1 Two-neighbor-minimum case 39
5.4.2 General case 41
5.5 Simulation results 42
5.5.1 Single attachment 42
5.5.2 Multi attachment 44
5.6 Game theoretical trade-off 45

6 Discussion **49**

7 Related Work **51**

8 Conclusion **53**

Bibliography **55**

A Appendix **59**
A.1 Varying allocation matrices without changing convergents and divergents 59
A.2 Varying allocation matrices as in fairness proof – when does GMA have no improvement 60

Declaration of Originality **63**

1 Introduction

Historically, allocating resources in a distributed network has been a difficult problem. As a consequence of this, in the context of bandwidth over the internet, most modern-day applications are based on the best-effort delivery paradigm, which relies on congestion control to maintain connectivity. While this paradigm has worked well for many applications, it has been problematic for others. These applications lie in areas which range from blockchain transactions, where one might occur a financial loss due to delays or interruptions to connectivity [1], inter-bank transactions, for which reliability is essential [13], all the way to medical applications, where a surgeon performing remote surgery requires a reliable connection [20]. All these applications rely on quick and reliable data delivery.

In the literature, there are many proposed solutions to this problem which try to allocate resources. However, they either have to be highly centralized and thus are not widely applicable [13], or have a too high associated complexity to be feasible, for example, because they require too much state to be stored [7, 14].

The GMA algorithm and open questions. In a recent publication Giuliari et al. [12] introduced and formalized the distributed path allocation ($\mathrm{PA}^{\mathrm{dist}}$) problem. This problem consists of allocating resources to paths given only local on-path information without causing overuse of any links. Additionally, the authors introduced the global myopic allocation (GMA) algorithm, a solution to this problem. The GMA algorithm provides unconditional allocations to paths while ensuring that no congestion occurs and has many desirable properties such as being pareto optimal.

One interesting consequence of the GMA algorithm is that it completely bypasses many of the problems of the current paradigm in regard to fairness. In the world of best-effort deliveries, where congestion control algorithms are used to determine the sending rates, there is a lot of active research on how to measure the fairness of algorithms and how to decide if they are deployable or not [10, 25]. None of these fairness notions are applicable to the GMA algorithm, because it completely bypasses the need for congestion control due to the unconditional allocations it provides. It remains important though that these allocations are done in a fair way. Therefore, we study the fairness properties of the GMA algorithm and analyze how these differ from the status quo.

Additionally, while many of the properties of the GMA algorithm have already been researched, various open questions regarding the theoretical aspects of GMA remain. In this thesis we will answer the question of whether GMA is globally optimal in the class of algorithms which solve the PA^{dist} problem. This will show whether it is possible to improve the allocations determined by the GMA algorithm or not.

Finally, open questions remain in the prerequisites for the GMA algorithm. The algorithm assumes that every node has a local policy, expressed through its allocation matrix, to decide how to split up its resources. The quality of the allocations to paths varies widely depending on how these are initialized. We study the tradeoff to global path allocations in regard to how nodes split up their resources between their neighbors and themselves.

In this thesis we answer these questions and further analyze various properties of the GMA algorithm. By doing this we not only answer these open questions but we also provide further features with which future solutions to the distributed path allocation problem can be evaluated. In this thesis we (i) derive a new fairness notion which is based on local policies and applicable to the distributed setting, (ii) answer the question on whether GMA is globally optimal, and (iii) find an optimal way of initializing allocation matrices.

1.1 Contributions

The main contributions of this thesis are the following:

- We design a neighbor-based policy fairness notion (NBP-fairness). We then apply this notion to GMA, thus proving that GMA is fair.
- We take a deeper dive into some of GMA's properties. We find an exact equation to model the influence of a single node's policy to paths; we show that GMA is not a globally optimal algorithm, and that GMA fully utilizes all links in all graphs where inter-node links are the constraining factor and thus is globally optimal for these.
- We find a way to optimally initialize all allocation matrices for GMA which depends on the average path length. Due to this initialization suffering from the tragedy of the commons we analyze the game theoretic tradeoffs in this.

1.2 Organization

Chapter 2 contains the necessary background information on the formalization of the distributed path allocation problem; the GMA algorithm; some important fairness notions; and some basics in game theory.
In Chapter 3 we present a new fairness notion; prove that GMA is fair according to this notion; and provide a comparison to existing fairness notions. We also motivate why this fairness notion is general enough to apply to other problems as well.

Chapter 4 will be used to answer the question of GMA's global optimality including finding a sensible definition of global optimality for the distributed path allocation problem.
In Chapter 5 we find the initialization policy that leads to optimal allocations. We also explore the game theoretical consequences of this policy.
Then in Chapter 6 we discuss our findings; in Chapter 7 we highlight some related work, and in Chapter 8 we draw our closing thoughts.
Finally, in Appendix A we provide some extended proofs of various properties.

2 Background

2.1 Distributed path allocation problem

The problem was formally introduced by Giuliari et al. together with the GMA algorithm [12]. We will first define allocation graphs and then the path allocation problem itself.

2.1.1 Allocation graphs

An *allocation graph* is defined as a standard directed graph, augmented with a set of interfaces at every node. An interface can either be an *external interface*, in which case it is attached to the end of one of the edges of this node, or a *local interface* which is not associated with any edge.

Every node k in this graph expresses its local policy through a non-negative *allocation matrix* $M^{(k)}$. The *pair allocation* between the interface i and the interface j represents the maximum amount of resource that can be used by all paths which start at or enter the node at interface i and end at or exit the node at interface j. This pair allocation is encoded in the allocation matrix in the entry $M^{(k)}_{i,j}$.

Finally, the *convergent/divergent* of some interface is defined as the maximum amount of resource that can be allocated to all paths entering/leaving that interface. That is, the convergent is defined as: $CON^{(k)}_j = \sum_i M^{(k)}_{i,j}$ and the divergent is defined as: $DIV^{(k)}_i = \sum_j M^{(k)}_{i,j}$.

One final requirement is that if i is an external interface, the divergent of i is smaller or equal to the capacity of the incoming edge and the convergent of i is smaller or equal to the capacity of the outgoing edge at i.

Paths. In an allocation graph, we define some *path* π which visits ℓ nodes $N^1, ..., N^\ell$ as $\pi = \left[\left(N^1, i^1, j^1\right), ..., \left(N^\ell, i^\ell, j^\ell\right)\right]$.[1] For simplicity we write $M^{(k)}_{i,j}$ instead of $M^{(N^k)}_{i^k,j^k}$ and $\pi = \left[\left(i^1, j^1\right), ..., \left(i^\ell, j^\ell\right)\right]$.

These paths can either be terminated or preliminary. A path is called *terminated* if it starts and ends in a local interface and *preliminary* otherwise.

[1] These paths are allowed to contain loops. We also implicitly assume that there is an edge between $j^{(k-1)}$ and $i^{(k)}$

Finally, we call a path *simple* if it contains any node at most once and the path *valid* if all pair interfaces (i, j) used by the path have $M_{i,j}^{(k)} > 0$. We observe that if some $M_{i,j}^{(k)}$ is 0, the path cannot have a positive allocation under the above constraints.

2.1.2 (Distributed) path allocation problem

The *path allocation* (PA) problem is defined as the problem of allocating resources to paths in some allocation graph.[2] An allocation to a path is an amount allocated to that path which is then reserved for this path on every edge and interface pair used by it.

One obvious constraint on these allocations is *no-over-allocation*. This property states that no edges or interface pair is ever over-allocated, which means that if we consider all possible paths that use this edge or interface pair, the sum of their allocations is smaller or equal to the capacity of the edge or the pair allocation of the interface pair.

For the *distributed path allocation problem* (PA^{dist}), we then restrict our algorithm's input to only contain local information about the path for which it is computing the allocation. That is, in addition to no-over-allocation we now have the *locality* property as well. This states that the path allocation depends only on the on-path allocation matrices $M^{(1)}, \ldots., M^{(\ell)}$. That is, we do not allow the algorithm to have a global view of the graph to compute the allocation given to some particular path. This is desirable, because in many networks it may be costly to find the entire topology and keep it up to date.

Additionally, there are three supplementary properties which are desirable for any solution to either the PA or the PA^{dist} problem. The first is *usability* which requires that every valid path gets a positive allocation. The second property *efficiency* holds, if the algorithms complexity is at most polynomial as a function of its input size. Finally, the *monotonicity* property expresses that if for some simple path π the pair allocation in one node is increased and all other ones remain the same, the resulting allocation must not decrease.

2.2 Global myopic allocation algorithm

The global myopic allocation (GMA) algorithm is a solution to the PA^{dist} problem. For a path π it calculates the allocation given to π as follows:

$$\mathcal{G}(\pi) = \min_x \left(\prod_{k=1}^{x-1} \frac{M_{i,j}^{(k)}}{CON_j^{(k)}} \cdot M_{i,j}^{(x)} \cdot \prod_{k=x+1}^{\ell} \frac{M_{i,j}^{(k)}}{DIV_i^{(k)}} \right) \tag{2.1}$$

This equation can also be used to calculate allocations on preliminary paths.

[2]We will treat the resource as bandwidth here, but other interpretations are also possible.

This algorithm fulfills both conditions of no-over-allocation and locality, thus it is a valid solution to the PA$^{\text{dist}}$problem. Additionally, it has the three supplementary properties listed above and is pareto optimal according to the following definition, as was defined by Giuliari et al. [12].

P-Opt Pareto Optimality: Consider the class of all algorithms fulfilling the requirement of either PA or PA$^{\text{dist}}$. Algorithm $\mathcal{A}$ from this class is Pareto optimal if there is no other algorithm $\mathcal{B}$ from the same class that can provide at least the same path allocation for every path of every allocation graph, and a strictly better allocation for at least one path. Formally, if there exists a graph with a path π for which $\mathcal{B}(\pi) = \mathcal{A}(\pi) + \delta$ with $\delta > 0$, then there exists at least one other path π', possibly in a different graph, where $\mathcal{B}(\pi') = \mathcal{A}(\pi') - \delta'$ with $\delta' > 0$.

This means that there is no other algorithm in the class of PA$^{\text{dist}}$ that can provide at least as large allocations for every path in any allocation graph and a strictly better allocation for at least one path. That means that there is no trivial way of improving the algorithm.

Proofs of all these properties were provided by Giuliari et al. [12].

2.3 Fairness

The topic of fairness is one that appears in many disciplines, all the way from philosophy to social sciences and computer networks [22]. Therefore, it is not surprising that the problem of defining a fairness notion is a complicated one where there are many tradeoffs to be made. Whenever limited means must be assigned to various agents, the question arises as to how to do this in a fair way. In the following, we will look at fairness in computer networks where we have limited link capacities, various flows (respectively paths) that use them, and we want to divide up this bandwidth "fairly" between the flows. But these ideas extend to other areas as well.

Intuitively, one might expect that something as simple as "there is an equal allocation of resources between participants" would suffice as a fairness notion because that ultimately is what corresponds to one's intuition. But sadly, it is not always that simple as there are three major issues with the above notion. To start off, it is often unclear what the resource we are dividing up is. For example, should the resource be the capacity of a certain link or rather the total network utilization? Secondly, it is unclear who we are equally allocating the resource to. In computer networks is this per flow or per end host pair? A good example, where the first definition would struggle is that a user could just start ten flows instead of one and would then combined get an "unfair" allocation in comparison to some other user who only uses one flow [26]. Finally, the intuition lacks some form of expressing demand. Giving a flow more allocation than it wants would be fair in the above definition but would be wasteful.

While there is no definition that solves all these problems, we will nevertheless look at the two most popular fairness notions in the networking literature.

In the following, we say an allocation is *feasible* if all flows have a non-negative share of the bandwidth allocated to them and the sum of all allocations of flows that use any given link does not exceed the capacity of the link.

2.3.1 Max-min fairness

Max-min fairness was first introduced to networks by Bertsekas and Gallager [5] and is amongst the most popular notions used. It is defined on the allocation given to some set of flows. An allocation achieves max-min fairness if and only if (i) it is feasible and (ii) the only way to increase the allocation of any flow is to decrease the allocation of some other flow with an equal or smaller allocation. If we are in the case where flows have certain demands, we need to modify the second condition to only be a problem if the flow being decreased is not above its demand. The setting with no demands can be seen equivalent to the setting where all demands are infinitely large.

There exists a simple procedure to find the max-min fair solution which is the water-filling procedure. We start off by giving all flows an allocation of 0. We then increase the allocation of all flows together, until some flows are constrained by the capacity of some link they use. We freeze the allocation of those constrained flows and repeat the previous step for all other flows until there are none left to increase [6].

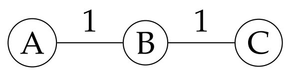

Figure 2.1: A simple network.

As an example, let us consider the network in Figure 2.1 and the three paths: [A,B], [B,C], and [A,B,C]. The max-min fair allocation on these paths would give each path an allocation of 0.5.

From this, we also observe one of the tradeoffs made by max-min fairness. If we look at any given link, we can say the resource of that link is equally divided between the two paths that use it. But if we take a global view, we notice that the path of length two ([A,B,C]) uses a global share of 1, whereas the two shorter paths each only use a global share of 0.5. This problem leads us directly to an alternative approach, proportional fairness.

2.3.2 Proportional fairness

Introduced to computer networks by Kelly et al. [17], *proportional fairness* is defined as follows: Let us assume we have n flows and let $x_1, \ldots, x_n$ be the allocation given to these flows. We say an allocation is proportionally fair if and only if it is feasible and the allocation maximizes the following sum amongst all feasible allocations.

$$\sum_{i=1}^{n} \log(x_i)$$

An approximation of this has been found to be implemented by various real world congestion control algorithms [18].

If we now take another look at the example given in Figure 2.1, the proportionally fair allocation would give our paths of length one ([A,B] and [B,C]) an allocation of $\frac{2}{3}$ and the path of length two an allocation of $\frac{1}{3}$. Thus, we can see that proportional fairness favors shorter paths.

2.3.3 Other fairness notions

Both of the above are fairness notions on the allocations given to paths themselves. There are a large variety of other notions, for example, based around ″TCP fairness“. These try to quantify how ″fair“ some congestion control algorithm is. One notable example of this would be Jain's Fairness Index [16] which, given two algorithms, returns a measure of how fair they are to each other.

But there has been a large amount of criticism [8, 10, 25] of this paradigm and no perfect solution for this has been found yet. We will not cover these in further depth here.

2.4 Game theory

Since subsequent sections draw on an understanding of game theory, we aim to give a quick primer on it in this section. We will do this by considering a simple example to illustrate some of the concepts and tradeoffs that one finds.

As an example, we consider the *prisoner's dilemma* [23]. We have two players where each has the option to either remain silent (cooperate with the other player) or act selfishly (betray the other player). If they both remain silent, they both need to spend one year in prison. If one of them acts selfishly and the other remains silent, the one who acted selfishly will go free and the one who stayed silent will spend three years in prison. But if they both act selfishly, they both end up spending two years in prison.

The natural question that arises is what the best action for any player is. The *Nash equilibrium* of this game will be for both players to act selfishly [21]. The Nash equilibrium is a point at which neither player wants to change their strategy. Assume I am one of the players in the prisoner's dilemma, if the other person remains silent my best outcome is to act selfishly (otherwise I would get a year in prison). Similarly, if the other person acts selfishly my best outcome again would be to act selfishly (like this I get 2 years compared to 3 years in prison). Therefore, if everyone in this game acts rationally, the outcome will be that both players end up in prison for two years.

But clearly the outcome of both players staying silent (cooperating with each other) would be better for both of them. If they both did that, they would end up spending half as long in prison as they would in the Nash equilibrium.

The *iterated prisoner's dilemma* game is a version of the prisoner's dilemma where there are multiple rounds and in each round the players either cooperate with each other or act selfishly. One strategy for this is *Tit for Tat*. The idea is to cooperate by default and if in the most recent round the other person acted selfishly you act selfishly in this round as a form of punishment.

A tournament was held in 1979 where various strategies were submitted and tested against each other in playing the iterated prisoner's dilemma. The Tit-for-Tat strategy [2] was found to perform the best. In a second tournament the Tit-for-Tat strategy again came out on top [3].

3 Analysis of GMA's Fairness

3.1 Problem statement

In this section we present our neighbor-based policy (NBP) fairness notion. This notion is applicable to distributed path allocation algorithms.

The aim for this notion is that it is intuitive and simple to understand. While we want to present it formally, it is important that all aspects of the notion correspond to some intuition of fairness. Additionally, the notion should depend and respect a node's local policy.

Let us consider a notion along the lines of "If the algorithm calculates the exact same allocation as GMA it is fair, otherwise it is not.". While GMA does have many properties we want in a fairness notion, we see that such a notion has no practical value. Thus, we seek to come up with a notion which is not too closely tied to the GMA algorithm.

Finally, we also prove that GMA is fair according to NBP-fairness.

3.2 NBP-fairness

NBP-fairness applies to algorithms which solve the $\mathrm{PA}^{\mathrm{dist}}$ problem. That means, we assume we have allocation matrices, convergents, and divergents as described in Section 2.1. Additionally, due to the definition of $\mathrm{PA}^{\mathrm{dist}}$ we have given that the algorithm has the locality and no-over allocation properties. Other than that, there are two main parts to the fairness notion.

First off, any NBP-fair algorithm should be pareto optimal according to the definition P-Opt. This is wanted as an algorithm which is inefficient and could be trivially improved is clearly undesirable.

The second part involves making sure the algorithm respects the node policies and adjusts the path allocations according to them. So if we consider some arbitrary simple path π of length ℓ, $\pi = \left[\left(i^1, j^1\right), ..., \left(i^\ell, j^\ell\right)\right]$, we want to enforce a stronger version of the monotonicity property.

Assume some node $k \in \{1, .., \ell\}$ increases its pair allocation between i^k and j^k by some factor $\delta \geq 0$; that is, we now have

$$M'^{(k)}_{i,j} = M^{(k)}_{i,j} + \delta$$

$$CON'^{(k)}_{j} = CON^{(k)}_{j} + \delta$$

$$DIV'^{(k)}_{i} = DIV^{(k)}_{i} + \delta$$

and all other allocation matrices and entries remain the same. We then require that the allocation to the path π is increased by a factor of at least:

$$\left(1 + \frac{\delta}{M_{i,j}^{(k)}}\right) \cdot \min\left\{\frac{DIV_i^{(k)}}{DIV_i^{(k)} + \delta}, \frac{CON_j^{(k)}}{CON_j^{(k)} + \delta}\right\}$$

That is, we enforce that the increase is greater than 0 unless $M_{i,j}^{(k)} = CON_i^{(k)}$ or $M_{i,j}^{(k)} = DIV_i^{(k)}$.[1] Additionally, we want the increase to depend on the ratio of the relative increase in the pair allocation to the relative increase in the convergent/ divergents. In the following we will use $\mathcal{A}(\pi)$ to denote the allocation before the change and $\mathcal{A}'(\pi)$ the allocation after the change. To summarize our new fairness notion:

NBP-fairness An algorithm $\mathcal{A}$ is neighbor-based policy fair, if the following three properties hold.

(F1) The algorithm $\mathcal{A}$ is a valid solution to the $\mathrm{PA}^{\mathrm{dist}}$ problem.

(F2) The algorithm $\mathcal{A}$ is pareto optimal in the sense of P-Opt.

(F3) Let π be some arbitrary simple path and $\mathcal{A}'$ as described above. Then we require:

$$\frac{\mathcal{A}'(\pi)}{\mathcal{A}(\pi)} \geq \left(1 + \frac{\delta}{M_{i,j}^{(k)}}\right) \cdot \min\left\{\frac{DIV_i^{(k)}}{DIV_i^{(k)} + \delta}, \frac{CON_j^{(k)}}{CON_j^{(k)} + \delta}\right\}$$

3.3 Reasoning behind NBP-fairness

Here we aim to motivate the individual components of NBP-fairness and the fact that this notion is neither too broad nor too narrow.

In the above, we already shortly motivated the three components of the fairness notion. To reiterate, the first property **(F1)** follows directly from the class of algorithms we are analyzing. The second property **(F2)** follows from the fact that any other algorithm could be trivially improved which is undesirable. For the third property **(F3)** let us quickly reiterate the motivation. Assume some node changes its local policy such that it now favors this pair allocation more; We want the increase to be dependent on the ratio of the increase in the pair allocation and the increase in the convergent/divergent. So, if currently the pair allocation is tiny compared to the convergent and divergent, we want the increase to be almost linear whereas if the pair allocation already almost fully saturates the

[1] So the increase is greater than 0, unless a node is already giving this interface pair a fraction of 1.

convergents and divergents we expect a smaller relative increase as that node already is contributing a large fraction to this path.

In what follows, we will first motivate how this fairness notion implies certain lower and upper bounds on the allocations of paths and then we will also show that this fairness notion does more than just accept the GMA algorithm, and thus is more widely applicable.

3.3.1 Bounds implied by NBP-fairness

NBP-fairness is a change-based notion. What that means is that through **(F3)** a certain change in the allocations is enforced when some node varies its policy. So, while some other notion might say that the allocation to a certain path has some lower bound, this is not directly the case with NBP-fairness which instead enforces that if an input is changed in a certain way the output should change in a corresponding fashion.

One issue with this is that even if the change is done in a fair way, we still have a problem if we start in an extremely "unfair" state. For example, if a path initially has an allocation of 0 or very close to 0, the above fairness notion would not make a lot of sense because even if we increase an allocation by some large factor it might still be very close to 0.

To motivate why this cannot occur we will look at a simple example shown in Figure 3.1. Note that we assume all edges here to be bidirectional. We will show that due to the increase formula and no over allocation there is an upper bound on the allocation for any given path. Then we observe that given the local view of a path one can similarly derive an upper bound on all other paths that this path shares interfaces with. Thus, in combination with Pareto optimality this then implies some lower bound on the allocation to this path.

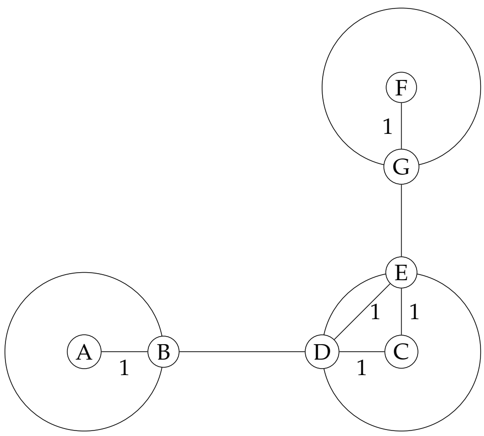

Figure 3.1: Simple example to motivate NBP-fairness.

Let us start by looking at the path $\pi_1 = [(F, G), (E, C)]$. Assume we have some NBP-fair algorithm $\mathcal{A}$. What we will show is that there is an upper bound on the allocation of π_1. In combination with pareto optimality we can then conclude that the path $\pi_2 = [(F, G), (E, D), (B, A)]$ will have a certain lower bound.

To do this, assume we are increasing the pair allocation (E,C) by δ. Due to **(F3)** we know that

$$\mathcal{A}'(\pi_1) \geq \mathcal{A}(\pi_1) \cdot (1+\delta) \cdot \min\left\{\frac{2}{2+\delta}, \frac{2}{2+\delta}\right\}$$

for any $\delta > 0$. If we now look at what happens when delta approaches infinity, we observe that we get $\mathcal{A}'(\pi_1) \geq \mathcal{A}(\pi_1) \cdot 2$. Now we add in the consideration that the pair allocation (F,G) still only is 1. Thus, to have no-over allocation the algorithm needs $\mathcal{A}'(\pi_1) \leq 1$, which directly gives us that we have $\mathcal{A}(\pi_1) \leq 0.5$.

The same bound can be shown for π_2 as well. Finally, we observe that just given the local view of either π_1 or π_2 both of these bounds can be derived. Thus, through pareto optimality we get $\mathcal{A}(\pi_1) = 0.5 = \mathcal{A}(\pi_2)$. We observe that in this graph NBP-fairness implies a bound which corresponds to the allocation GMA gives to these paths ($\mathcal{G}(\pi_1) = 0.5$ and $\mathcal{G}(\pi_2) = 0.5$).

More than just GMA. In the above, we motivated that NBP-fairness is not too broad in the sense that it doesn't accept unfair algorithms. But while doing so, we just showed that in certain situations this notion enforces the algorithm to give the same allocations as GMA. We will use an example here to illustrate that this in general is not the case and therefore motivate the fact that NBP-fairness is more widely applicable.

To consider this, we look at a slight modification of the previous graph. The new graph is in Figure 3.2. The only difference to Figure 3.1 is that now the pair allocation (C,D) is 0.5 instead of 1.

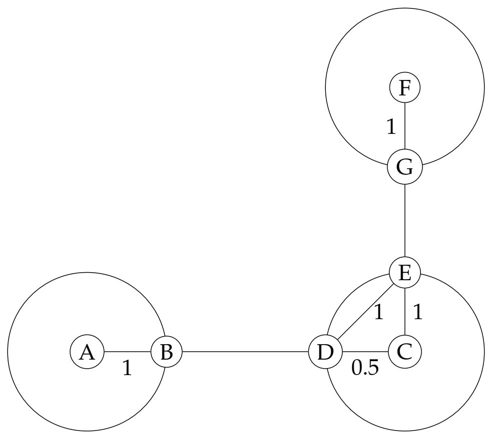

Figure 3.2: A second simple example to motivate NBP-fairness.

Again, we consider the same two paths π_1 and π_2. We observe that the GMA allocation for π_1 and π_2 are still the same. We proceed similarly to before to find some bounds. We know that if we increase the pair allocation (E,C) by some $\delta > 0$ for any NBP-fair algorithm $\mathcal{A}$ we must have

$$\mathcal{A}'(\pi_1) \geq \mathcal{A}(\pi_1) \cdot (1 + \delta) \cdot \min \left\{ \frac{2}{2 + \delta}, \frac{1.5}{1.5 + \delta} \right\}$$

Similarly to before, if we let δ grow arbitarily large we need $\mathcal{A}'(\pi_1) \geq 1.5 \cdot \mathcal{A}(\pi_1)$. Due to no over allocation, we then get that we have $\mathcal{A}(\pi_1) \leq \frac{2}{3}$.

Again, the same can be shown for π_2 if we increase the pair interface (E,D) instead. Together with pareto optimality we then get that $\mathcal{A}(\pi_1) \in \left[\frac{1}{3}, \frac{2}{3}\right]$. Thus, we can see that the fairness notion allows for various algorithms. For example, one algorithm in this graph could favor shorter paths whereas a different one could favor the longer path.

3.4 Comparison to other fairness notions

As discussed in Section 2.3 most existing fairness notions either are widely known notions like max-min or proportional fairness, or they are fairness notions in highly specific areas like congestion control algorithms.

Let us consider both max-min and proportional fairness. The allocations determined by these vary depending on the paths being considered. So, if we remove some path from the set of paths we are considering, the max-min or proportional fair allocation on some of the other paths may change. This is fundamentally incompatible with the PA^{dist} problem though, because we do not know which other paths are being used and which are not.[2] From this, it follows that most of the existing fairness notions are not directly applicable to the PA^{dist} problem.

Additionally, these fairness notions do not give a node any way to express its preference. If a node really dislikes one of its neighbors these fairness notions would force it to take a certain amount of traffic from that neighbor. But giving a node some control over how its own bandwidth is divided up is a requirement for our idea of neighbor-based fairness.

Our fairness notion takes these issues into consideration. Firstly, it is a notion on algorithms in PA^{dist}. That is, given some algorithm we can determine if it is fair or not without needing to consider various graphs with various sets of paths which as outlined above would be incompatible with the very nature of the PA^{dist} problem. Our notion also is based on nodes local policies. By varying its local policy, a node can have a direct influence on which paths get how much of an allocation. So, this notion is policy based which is a major difference to most of the fairness notions found in the literature.

[2] In fact, we do not even know which other paths exist.

3.5 GMA is NBP-fair

In this section we will prove that GMA is fair according to NBP-fairness.

3.5.1 Properties (F1) and (F2)

GMA has these properties as proven in the paper which introduced the GMA algorithm [12]. As such, we will not reiterate the proofs here.

3.5.2 Property (F3)

To start off, we define:

$$\mathcal{G}_x(\pi) = \left(\prod_{k=1}^{x-1} \frac{M_{i,j}^{(k)}}{CON_j^{(k)}} \cdot M_{i,j}^{(x)} \cdot \prod_{k=x+1}^{\ell} \frac{M_{i,j}^{(k)}}{DIV_i^{(k)}} \right)$$

If we then let $x^* = \arg\min_x \left(\mathcal{G}_x(\pi)\right)$, we observe that $\mathcal{G}(\pi) = \mathcal{G}_{x^*}(\pi)$.

Let $\pi = \left[\left(i^1, j^1\right), ..., \left(i^\ell, j^\ell\right)\right]$ be an arbitrary simple path. And $k \in \{1, .., \ell\}$ be the node which is increasing its pair allocation between i^k and j^k by some factor $\delta \geq 0$. That is, in our modified graph we have:

$$\begin{aligned} M_{i,j}'^{(k)} &= M_{i,j}^{(k)} + \delta \\ CON_j'^{(k)} &= CON_j^{(k)} + \delta \\ DIV_i'^{(k)} &= DIV_i^{(k)} + \delta \end{aligned}$$

We will use $\mathcal{G}'(\pi)$ to denote the allocation after the modification.

We observe that all terms in the formula used to calculate respectively all the inputs to the algorithm $\mathcal{G}'(\pi)$ will be the same as for $\mathcal{G}(\pi)$ except for the above three. To prove the desired property, we will first derive an exact formula for the increase and then prove that the exact formula has the desired properties.

To find the exact formula for the increase we perform a case distinction on whether we are increasing in the node x^* (the minimum pre modification) or whether we are increasing in some other node.

Case 1: $k \neq x^*$

Let us first consider the case where x^* remains the minimizer. That is, we have $\mathcal{G}'_{x^*}(\pi) = \mathcal{G}'(\pi)$.
We observe that if $k < x^*$ we have

$$\begin{aligned}
\mathcal{G}'(\pi) &= \mathcal{G}(\pi) \cdot \frac{CON_j^{(k)}}{M_{i,j}^{(k)}} \cdot \frac{M'^{(k)}_{i,j}}{CON'^{(k)}_j} \\
&= \mathcal{G}(\pi) \cdot \frac{CON_j^{(k)}}{M_{i,j}^{(k)}} \cdot \frac{M_{i,j}^{(k)} + \delta}{CON_j^{(k)} + \delta} \\
&= \mathcal{G}(\pi) \cdot \left(1 + \frac{\delta}{M_{i,j}^{(k)}}\right) \cdot \frac{CON_j^{(k)}}{CON_j^{(k)} + \delta}
\end{aligned}$$

Similarly, if $k > x^*$ we have

$$\begin{aligned}
\mathcal{G}'(\pi) &= \mathcal{G}(\pi) \cdot \frac{DIV_i^{(k)}}{M_{i,j}^{(k)}} \cdot \frac{M'^{(k)}_{i,j}}{DIV'^{(k)}_i} \\
&= \mathcal{G}(\pi) \cdot \frac{DIV_i^{(k)}}{M_{i,j}^{(k)}} \cdot \frac{M_{i,j}^{(k)} + \delta}{DIV_i^{(k)} + \delta} \\
&= \mathcal{G}(\pi) \cdot \left(1 + \frac{\delta}{M_{i,j}^{(k)}}\right) \cdot \frac{DIV_i^{(k)}}{DIV_i^{(k)} + \delta}
\end{aligned}$$

It is clear the property holds in this case.

Therefore, we are now interested in seeing under what conditions a node $y \neq x^*$ will become the new minimizer of Equation (2.1). To do so, we assume $y \neq x^*$ becomes the new minimizer. We allow that y and x^* both were minimizers before our modification but we require that x^* no longer is a minimizer after our modification.[3] That is, we assume that:

$$\begin{aligned}
\mathcal{G}_{x^*}(\pi) &\leq \mathcal{G}_y(\pi) \\
\mathcal{G}'_{x^*}(\pi) &> \mathcal{G}'_y(\pi)
\end{aligned}$$

We now perform a case distinction on the possible values for y. And either prove that it is impossible for y to replace x^* as the minimizer or derive the conditions for y to become the new minimizer. Once done, we will verify that the bound holds in all of them.

[3]Otherwise the above formulas would still apply.

Case 1.1.a: $k = y$ **and** $y < x^*$

Claim: This case will never occur.

Proof: We perform a proof by contradiction. Assume $y = k$ is the new minimizer. We then have:

$$\mathcal{G}'_{x^*}(\pi) = \mathcal{G}_{x^*}(\pi) \cdot \frac{CON_j^{(y)}}{M_{i,j}^{(y)}} \cdot \frac{M'^{(y)}_{i,j}}{CON'^{(y)}_j} = \mathcal{G}_{x^*}(\pi) \cdot \frac{CON_j^{(y)}}{M_{i,j}^{(y)}} \cdot \frac{M_{i,j}^{(y)} + \delta}{CON_j^{(y)} + \delta}$$

$$\mathcal{G}'_y(\pi) = \mathcal{G}_y(\pi) \cdot \frac{M'^{(y)}_{i,j}}{M_{i,j}^{(y)}} = \mathcal{G}_y(\pi) \cdot \left(1 + \frac{\delta}{M_{i,j}^{(y)}}\right)$$

Since we need $\mathcal{G}_{x^*}(\pi) \leq \mathcal{G}_y(\pi)$ and $\mathcal{G}'_{x^*}(\pi) > \mathcal{G}'_y(\pi)$ the following must hold:

$$\begin{aligned} 1 + \frac{\delta}{M_{i,j}^{(y)}} &< \frac{CON_j^{(y)}}{M_{i,j}^{(y)}} \cdot \frac{M_{i,j}^{(y)} + \delta}{CON_j^{(y)} + \delta} \\ &< \left(1 + \frac{\delta}{M_{i,j}^{(y)}}\right) \cdot \left(\frac{CON_j^{(y)}}{CON_j^{(y)} + \delta}\right) \end{aligned}$$

Which clearly does not hold because $\delta > 0$ and $CON_j^{(y)} > 0$.

Case 1.1.b: $k = y$ **and** $y > x^*$

Similarly to Case 1.1.a, this will never occur. The proof works completely analogously, one just needs to replace all the mentions of convergents with divergents.

Case 1.2.a: $x^* > k$ **and** $y > k$

Claim: This case will never occur.

Proof: We perform a proof by contradiction. Assume $y > k$ is the new minimizer. We then have:

$$\mathcal{G}'_{x^*}(\pi) = \mathcal{G}_{x^*}(\pi) \cdot \frac{CON_j^{(k)}}{M_{i,j}^{(k)}} \cdot \frac{M'^{(k)}_{i,j}}{CON'^{(k)}_j}$$

$$\mathcal{G}'_y(\pi) = \mathcal{G}_y(\pi) \cdot \frac{CON_j^{(k)}}{M_{i,j}^{(k)}} \cdot \frac{M'^{(k)}_{i,j}}{CON'^{(k)}_j}$$

Since

$$\frac{CON_j^{(k)}}{M_{i,j}^{(k)}} \cdot \frac{M'^{(k)}_{i,j}}{CON'^{(k)}_j} \geq 0$$

and $\mathcal{G}_{x^*}(\pi) \leq \mathcal{G}_y(\pi)$ it must hold that $\mathcal{G}'_{x^*}(\pi) \leq \mathcal{G}'_y(\pi)$ which yields the desired contradiction.

Case 1.2.b: $x^* < k$ **and** $y < k$

Similarly to Case 1.2.a, this will never occur. Once again, we observe that the proof works completely analogously, one just needs to replace all mentions of convergents with divergents.

Case 1.2.c: $x^* > k$**, and** $y < k$

Unlike the above cases this case can occur. In the following, we will analyze under which conditions this occurs.
We observe:

$$\mathcal{G}'_{x^*}(\pi) = \mathcal{G}_{x^*}(\pi) \cdot \frac{CON_j^{(k)}}{M_{i,j}^{(k)}} \cdot \frac{M'^{(k)}_{i,j}}{CON'^{(k)}_j} = \mathcal{G}_{x^*}(\pi) \cdot \left(1 + \frac{\delta}{M_{i,j}^{(k)}}\right) \frac{CON_j^{(k)}}{CON_j^{(k)} + \delta}$$

$$\mathcal{G}'_{y}(\pi) = \mathcal{G}_{y}(\pi) \cdot \frac{DIV_i^{(k)}}{M_{i,j}^{(k)}} \cdot \frac{M'^{(k)}_{i,j}}{DIV'^{(k)}_i} = \mathcal{G}_{y}(\pi) \cdot \left(1 + \frac{\delta}{M_{i,j}^{(k)}}\right) \frac{DIV_i^{(k)}}{DIV_i^{(k)} + \delta}$$

From this, it becomes clear that in this case it is indeed possible for the minimum to change. For example, if $\mathcal{G}_y(\pi) = \mathcal{G}_{x^*}(\pi)$, y would become the new minimizer if $\frac{CON_j^{(k)}}{CON_k^{(j)}+\delta} > \frac{DIV_i^{(k)}}{DIV_i^{(k)}+\delta}$ i.e., if $CON_j^{(k)} > DIV_i^{(k)}$.

In this case, we can also directly observe the desired bound for $\mathcal{G}'(\pi)$. From the above it directly follows that:

$$\mathcal{G}'(\pi) \geq \mathcal{G}(\pi) \cdot \left(1 + \frac{\delta}{M_{i,j}^{(k)}}\right) \cdot \min\left\{\frac{DIV_i^{(k)}}{DIV_i^{(k)} + \delta}, \frac{CON_j^{(k)}}{CON_j^{(k)} + \delta}\right\}$$

To find a more exact bound we observe that

$$\mathcal{G}_y(\pi) = \left(\prod_{r=y+1}^{x^*-1} \frac{CON_j^{(r)}}{DIV_i^{(r)}}\right) \cdot \frac{CON_j^{(y)}}{DIV_i^{(x^*)}} \cdot \mathcal{G}_{x^*}(\pi)$$

Since y becomes the new minimizer we get:

$$\begin{aligned}\mathcal{G}'(\pi) &= \mathcal{G}'_y(\pi)\\ &= \mathcal{G}(\pi)\cdot\left(1+\frac{\delta}{M^{(k)}_{i,j}}\right)\cdot\frac{DIV^{(k)}_i}{DIV^{(k)}_i+\delta}\left(\prod_{r=y+1}^{x^*-1}\frac{CON^{(r)}_j}{DIV^{(r)}_i}\right)\cdot\frac{CON^{(y)}_j}{DIV^{(x^*)}_i}\end{aligned}$$

Case 1.2.d: $a \neq y$**,** $x^* < a$**, and** $y > a$

This case works analogously to Case 1.2.c, once again we replace convergents by divergents and vice versa and find:

$$\begin{aligned}\mathcal{G}'(\pi) &= \mathcal{G}'_y(\pi)\\ &= \mathcal{G}(\pi)\cdot\left(1+\frac{\delta}{M^{(k)}_{i,j}}\right)\cdot\frac{CON^{(k)}_j}{CON^{(k)}_j+\delta}\left(\prod_{r=x^*+1}^{y-1}\frac{DIV^{(r)}_i}{CON^{(r)}_j}\right)\cdot\frac{DIV^{(y)}_i}{CON^{(x^*)}_j}\end{aligned}$$

Deriving the bound

If we now combine the above, we find that the new allocation after the increase is the following:
Case $x^* > k$:

$$\mathcal{G}'(\pi) = \mathcal{G}(\pi)\cdot\left(1+\frac{\delta}{M^{(k)}_{i,j}}\right)\cdot\min\left\{\min_{y<k}\left(\left(\prod_{r=y+1}^{x^*-1}\frac{CON^{(r)}_j}{DIV^{(k)}_i}\right)\cdot\frac{CON^{(y)}_j}{DIV^{(x^*)}_i}\cdot\frac{DIV^{(k)}_i}{DIV^{(k)}_i+\delta}\right),\right.\\ \left.\frac{CON^{(k)}_j}{CON^{(k)}_j+\delta}\right\}$$

Case $x^* < k$:

$$\mathcal{G}'(\pi) = \mathcal{G}(\pi)\cdot\left(1+\frac{\delta}{M^{(k)}_{i,j}}\right)\cdot\min\left\{\min_{y>k}\left(\left(\prod_{r=x^*+1}^{y-1}\frac{DIV^{(r)}_i}{CON^{(r)}_j}\right)\cdot\frac{DIV^{(y)}_i}{CON^{(x^*)}_j}\cdot\frac{CON^{(k)}_j}{CON^{(k)}_j+\delta}\right),\right.\\ \left.\frac{DIV^{(k)}_i}{DIV^{(k)}_i+\delta}\right\}$$

The final observation still required is that because x^* was the minimizer before the change, we have that for any $y < x$ that $\left(\prod_{r=y+1}^{x^*-1}\frac{CON^{(r)}_j}{DIV^{(k)}_i}\right)\cdot\frac{CON^{(y)}_j}{DIV^{(x^*)}_i} > 1$ and for any $y > x$ that $\left(\prod_{r=x^*+1}^{y-1}\frac{DIV^{(r)}_i}{CON^{(r)}_j}\right)\cdot\frac{DIV^{(y)}_i}{CON^{(x^*)}_j}\cdot\frac{CON^{(k)}_j}{CON^{(k)}_j+\delta} < 1$.

With this, we conclude that in the case where we are increasing along some node which is not our minimizer that GMA fulfills the desired bound.

Case 2: $k = x^*$

In this section we will show that the formula derived above is also applicable to this case. Again, let us first consider under what conditions a node $y \neq x^*$ will become the minimizer.

To do this, we assume $y \neq x^*$ is the new minimizer. That is, we again assume:

$$\begin{aligned} \mathcal{G}_{x^*}(\pi) &\leq \mathcal{G}_y(\pi) \\ \mathcal{G}'_{x^*}(\pi) &> \mathcal{G}'_y(\pi) \end{aligned}$$

To treat this case, we will use the following which directly follows from Equation (2.1).

$$\mathcal{G}(\pi) = \prod_{r=1}^{\ell} M_{i,j}^{(r)} \cdot \min_x \left(\prod_{r=1}^{x-1} \frac{1}{CON_j^{(r)}} \cdot \prod_{r=x+1}^{\ell} \frac{1}{DIV_i^{(r)}} \right)$$

We then observe that

$$\begin{aligned} \mathcal{G}'_{x^*}(\pi) &= \prod_{r=1}^{\ell} M_{i,j}^{(r)} \cdot \left(1 + \frac{\delta}{M_{i,j}^{(k)}} \right) \cdot \left(\prod_{r=1}^{x^*-1} \frac{1}{CON_j^{(r)}} \cdot \prod_{r=x^*+1}^{\ell} \frac{1}{DIV_i^{(r)}} \right) \\ &= \left(1 + \frac{\delta}{M_{i,j}^{(k)}} \right) \mathcal{G}(\pi) \end{aligned}$$

We once again perform a case distinction on the possible values for our new minimizer y.

Case 2.a: $y < x^*$

We observe

$$\begin{aligned} \mathcal{G}'_y(\pi) &= \prod_{r=1}^{\ell} M_{i,j}^{(r)} \cdot \left(1 + \frac{\delta}{M_{i,j}^{(k)}} \right) \cdot \left(\prod_{r=1}^{y-1} \frac{1}{CON_j^{(r)}} \cdot \prod_{\substack{r=y+1 \\ r \neq k}}^{\ell} \frac{1}{DIV_i^{(r)}} \cdot \frac{1}{DIV_i^{(k)} + \delta} \right) \\ &= \left(1 + \frac{\delta}{M_{i,j}^{(k)}} \right) \left(\prod_{r=y+1}^{k-1} \frac{CON_k^{(r)}}{DIV_i^{(r)}} \right) \left(\frac{CON_j^{(y)}}{DIV_i^{(k)} + \delta} \right) \cdot \mathcal{G}(\pi) \end{aligned}$$

One can see that for y to become the new minimum the following needs to hold:

$$\left(\prod_{r=y+1}^{k-1} \frac{CON_k^{(r)}}{DIV_i^{(r)}} \right) \left(\frac{CON_j^{(y)}}{DIV_i^{(k)} + \delta} \right) < 1$$

Case 2.b: $y > x^*$

In this case analogously to Case 2.a one can derive that:

$$\mathcal{G}'_y(\pi) = \left(1 + \frac{\delta}{M_{i,j}^{(k)}}\right)\left(\prod_{r=a+1}^{y-1} \frac{DIV_i^{(r)}}{CON_j^{(r)}}\right)\left(\frac{DIV_i^{(y)}}{CON_j^{(k)} + \delta}\right) \cdot \mathcal{G}(\pi)$$

We again observe that y becomes the new minimum if the following holds:

$$\left(\prod_{r=a+1}^{y-1} \frac{DIV_i^{(r)}}{CON_j^{(r)}}\right)\left(\frac{DIV_i^{(y)}}{CON_j^{(k)} + \delta}\right) < 1$$

Having completed this, we now derive the following formula for the increased allocation:

$$\begin{aligned}\mathcal{G}'(\pi) = \mathcal{G}(\pi) \cdot \left(1 + \frac{\delta}{M_{i,j}^{(k)}}\right) \cdot \min \Bigg\{ &\min_{y<x} \left(\prod_{r=y+1}^{k-1} \frac{CON_k^{(r)}}{DIV_i^{(r)}}\right)\left(\frac{CON_j^{(y)}}{DIV_i^{(k)} + \delta}\right), \\ &\min_{y>x} \left(\prod_{r=a+1}^{y-1} \frac{DIV_i^{(r)}}{CON_j^{(r)}}\right)\left(\frac{DIV_i^{(y)}}{CON_j^{(k)} + \delta}\right), \\ &1 \Bigg\}\end{aligned}$$

Again, we make the crucial observation that because x^* was the minimizer at the beginning we have that

$$\left(\prod_{r=y+1}^{k-1} \frac{CON_k^{(r)}}{DIV_i^{(r)}}\right)\left(\frac{CON_j^{(y)}}{DIV_i^{(k)}}\right) > 1$$

and

$$\left(\prod_{r=y+1}^{k-1} \frac{CON_k^{(r)}}{DIV_i^{(r)}}\right)\left(\frac{CON_j^{(y)}}{DIV_i^{(k)} + \delta}\right) > 1$$

from which the desired bound then directly follows.

As such, we have shown that GMA is fair according to NBP-fairness.

In Appendix A we will also derive the bounds for GMA in case we only change the pair allocation and leave the convergent and divergent as constant. We then will also analyze for which conditions the above increase will be 0.

4 Global Optimality of GMA

4.1 Problem statement

In this part of the thesis, we aim to analyze whether GMA is a globally optimal algorithm in the class of PA^{dist}.

To do this, we will on the one hand need to find a definition for global optimality that makes sense. Once we have done that, we will prove that GMA is not globally optimal.

At this point it also is important to note that if we draw an undirected link in the following figures, we implicitly take this to mean there are two directed links. The value we assign to this link is the pair allocation provided in each direction on this link.

4.2 Definitions of global optimality

As a first definition one might consider the following:

1. **Global Optimality:** Consider the class of all algorithms fulfilling the requirement of either PA or PA^{dist}. Algorithm $\mathcal{A}$ from this class is globally optimal if there is no other algorithm $\mathcal{B}$ from the same class, such that for *every* interface pair, the product of the allocations of all valid paths using this pair calculated with $\mathcal{B}$ is strictly larger than the ones calculated with $\mathcal{A}$.

We will now illustrate with an example why this is a bad definition. Let us consider the graph in Figure 4.1.

One possible algorithm which would be optimal in the above definition would give the paths [(A,B), (E,D)] and [(D,E), (B,A)] each an allocation of 0.5. Since

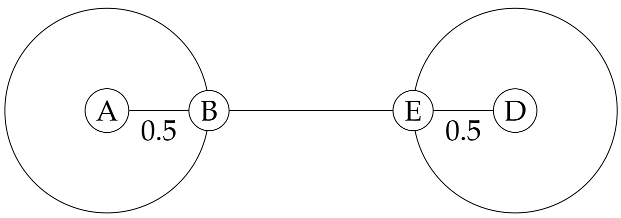

Figure 4.1: Graph which shows why Definition 1 is not a good optimality definition.

these are the only two valid paths in this graph, it is clear we have maximized the product. Thus, by the above definition this algorithm would be globally optimal regardless of what we do with any other paths. Therefore, we could say the algorithm for any other path which is not of the above form gives them an allocation of 0. That would mean, in the majority of topologies most paths would have an allocation of 0 which is obviously not optimal. The above notion doesn't even enforce pareto optimality. We also observe that if we replace the product with a sum in the definition the exact same counter example works. Thus, this clearly is a too weak property to be of real interest.

This leads us to the following two definitions of global optimality both of which do not suffer from the above problem. These are effectively pareto optimality on the product or sum of all paths using a certain interface pair.

Opt 1 **Global Optimality (·):** Consider the class of all algorithms fulfilling the requirements of either PA or PA^{dist}, we say algorithm $\mathcal{A}$ from this class is globally optimal if there is no other algorithm $\mathcal{B}$ from the same class, such that for any interface pair the *product* of the allocations of all valid paths using this pair calculated with $\mathcal{B}$ is at least as large as the *product* when the allocations are calculated with $\mathcal{A}$ and strictly larger for at least one interface pair.

Opt 2 **Global Optimality (+):** Consider the class of all algorithms fulfilling the requirements of either PA or PA^{dist}, we say algorithm $\mathcal{A}$ from this class is globally optimal if there is no other algorithm $\mathcal{B}$ from the same class, such that for any interface pair the *sum* of the allocations of all valid paths using this pair calculated with $\mathcal{B}$ is at least as large as the *sum* when the allocations are calculated with $\mathcal{A}$ and strictly larger for at least one interface pair.

If we define the utilization of a pair interface as the sum of the allocations of all paths that use it, an alternative version of Opt 2 would be to say that there is no other algorithm that achieves an equal or larger utilization of all pair interfaces and a strictly larger utilization of at least one pair interface.

In the following, we will first show that GMA is not optimal in the sense of either of the above definitions in the class of algorithms of PA^{dist}. However, we will then show that for an important subset of graphs the GMA algorithm is globally optimal in the sense of Opt 2.

4.3 GMA is not globally optimal

4.3.1 GMA and Opt 1

Let us first consider the product definition. We formulate the following claim:

Claim: GMA is not a globally optimal algorithm in the class of PA^{dist} according to Opt 1.

Proof: To show this, we will propose an alternative algorithm $\mathcal{B}$ that solves PA^{dist}, performs strictly better than GMA in at least one graph, and is just as good as GMA in all other graphs. From that, it then directly follows that GMA is not globally optimal according to this metric.

To start off, let us consider the graph in Figure 4.2 which is the graph in which our new algorithm will outperform GMA in the sense of this version of global optimality.

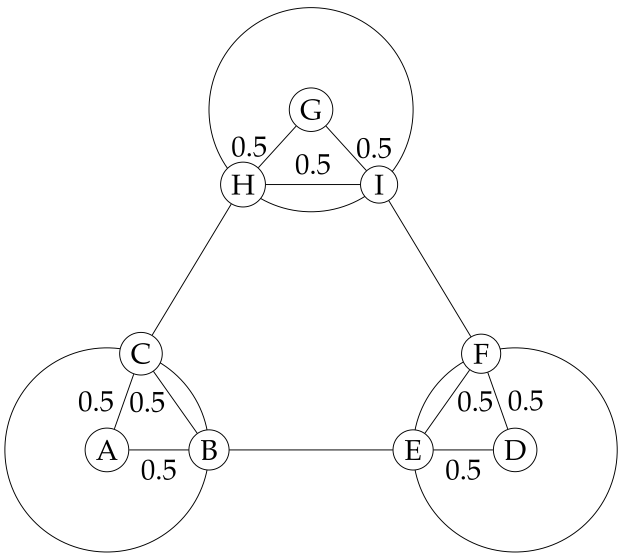

Figure 4.2: Graph for which GMA is not globally optimal according to Opt 1.

In this graph we have three nodes. Each with one local interface[1] and two exterior interfaces. For each node we set all the pair allocations between two interfaces to 0.5 (where the pair allocation from an interface to itself will remain 0). Thus, each interface has a convergent and a divergent of 1.

In this graph we will now consider the following two paths:

$$\pi_1 = [(D, F), (I, H), (C, B), (E, F), (I, G)]$$

[1] here called A, D, and G although the naming is arbitrary

and

$$\pi_2 = [(\mathrm{D}, \mathrm{F}), (\mathrm{I}, \mathrm{H}), (\mathrm{C}, \mathrm{B}), (\mathrm{E}, \mathrm{F}), (\mathrm{I}, \mathrm{H}), (\mathrm{C}, \mathrm{B}), (\mathrm{E}, \mathrm{F}), (\mathrm{I}, \mathrm{G})]$$

We observe the following about these paths:

- The paths are both $D - G$ paths and use the same set of pair interfaces.
- π_2 uses any interface pair at most twice and π_1 at most once.

Note that when we talk about the "graph" in this we implicitly also allow any larger (and disconnected) graphs which have this graph as a subgraph.
In Algorithm $\mathcal{B}$ we present the alternative algorithm to GMA.

Algorithm $\mathcal{B}$: ALTERNATIVE PA$^{\text{DIST}}$ ALGORITHM

Require: All on path allocation matrices; some path π
  if π corresponds exactly to π_1 in the graph in Fig. 4.2 **then**
    Return $\frac{1}{64}$
  else
    if π corresponds exactly to π_2 in the graph in Fig. 4.2 **then**
      Return $\frac{3}{256}$
    else
      Return the GMA allocation $\mathcal{G}(\pi)$.
    end if
  end if

At this point, it is also important to note that due to the symmetry of the graph there is no difference to the algorithm between π_1 and the following path: $\pi_3 = [(\mathrm{G}, \mathrm{H}), (\mathrm{C}, \mathrm{B}), (\mathrm{E}, \mathrm{F}), (\mathrm{I}, \mathrm{H}), (\mathrm{C}, \mathrm{A})]$. So, for example, we find $\mathcal{B}(\pi_3) = \frac{1}{64}$. We will now prove the following three claims about Algorithm $\mathcal{B}$.

1. For certain interface pairs, in certain graphs this algorithm has a larger product than GMA.
2. For all other interface pairs this new algorithm doesn't have a smaller product than GMA.
3. This algorithm is in the class of PA$^{\text{dist}}$.

Proof of property 1

The graph in Figure 4.2 will be a graph where this algorithm performs better than GMA in all interface pairs. From this, property 1 directly follows.
Proof: For any interface pair (x, y) in the graph, we can find at least one path of the form π_1 and at least one path of the form π_2 such that both of these paths use this particular interface pair.

We also observe that there will be an equal number of paths of the form π_2 as of the form π_1 that use this interface pair.[2]
So, we let a be the number of paths of the form π_1 respectively π_2 that use the interface pair (x, y).[3] For an algorithm $\mathcal{A}$ the product of all paths using this interface pair is then of the form

$$\mathcal{A}(\pi_1)^a \cdot \mathcal{A}(\pi_2)^a \cdot c(\mathcal{A})$$

where $c(\mathcal{A})$ represents the product of the allocations of all the (infinitely many) other paths that use this interface pair.

If we now calculate the GMA allocation on these two paths, we find $\mathcal{G}(\pi_1) = \frac{1}{32}$ and $\mathcal{G}(\pi_2) = \frac{1}{256}$.
We get the product for GMA to be: $\left(\frac{1}{32} \cdot \frac{1}{256}\right)^a \cdot c(\mathcal{G})$.

Whereas Algorithm $\mathcal{B}$ has $\mathcal{B}(\pi_1) = \frac{1}{64}$ and $\mathcal{B}(\pi_2) = \frac{3}{256}$. We also note that $c(\mathcal{B}) = c(\mathcal{G})$, as for all paths that are not of the form π_1 or π_2 the new algorithm calculates the same allocation as GMA. So, the product using our new algorithm is: $\left(\frac{1}{64} \cdot \frac{3}{256}\right)^a \cdot c(\mathcal{G})$.

As all valid paths have a nonzero and positive allocation and $a \geq 1$ we conclude:

$$\begin{aligned}
&\left(\frac{1}{32} \cdot \frac{1}{256}\right)^a \cdot c(\mathcal{G}) < \left(\frac{1}{64} \cdot \frac{3}{256}\right)^a \cdot c(\mathcal{G}) \\
\iff &\left(\frac{1}{32} \cdot \frac{1}{256}\right)^a < \left(\frac{1}{64} \cdot \frac{3}{256}\right)^a \\
\iff &\frac{1}{8192} < \frac{3}{16384}
\end{aligned}$$

From which it directly follows that property 1 holds.

Proof of property 2

This property follows directly from the definition of the algorithm.

If we are looking at an interface pair which is not in this specific graph all paths using this interface pair will have their allocation calculated by the GMA formula. Therefore, the products will be equal.

Since we know that for all interface pairs in our specific graph Algorithm $\mathcal{B}$ is strictly better, we conclude that this property holds.

[2] For each pair (u, v) of local interfaces there is exactly one $u - v$ path of form π_1 and one $u - v$ path of the form π_2. These use the same sets of pair interfaces.

[3] In the definition of global optimality it is not specified how we count a path that uses an interface multiple times. But regardless of if we count it only once or multiple times this proof still works. Here we formulate it as if the path only counts once.

Proof of property 3

We need to show that Algorithm $\mathcal{B}$ has two properties: No over allocation and locality.

No over allocation: If we are not in the specific graph this follows directly from the definition of the algorithm and the fact that GMA has this property.

If we are in this specific graph, let for any interface pair a be the number of paths of the form π_1 that use it and b be the number of paths of the form π_2 that use it. Here, if a path uses an interface pair multiple times, we count it multiple times. So we now have $a \neq b$.

We have $a \leq b \leq 2 \cdot a$ because π_1 uses any pair interface at most once and π_2 at most twice. So if for any interface pair we decrease the allocation of a paths by ϵ (here we would have $\epsilon = \frac{1}{64}$) and increase the allocations of b paths by $\frac{\epsilon}{2}$, the no over allocation property of GMA is preserved. As this is exactly what Algorithm $\mathcal{B}$ does, we conclude that it has this property.

Locality: Apart from the if-statements this again directly follows from the fact that the GMA algorithm has this property.

For the if-statements what is important to note is that the if condition can be decided using only the on-path allocation matrices. This holds, because there is a unique set of on path allocation matrices that determine π_1 and π_2. And once one has those allocation matrices, the graph is uniquely determined by them [4] as the path visits all nodes. Hence, with our local policy we can determine if the path is either π_1 or π_2.

With that the proof is concluded.

4.3.2 GMA and Opt 2

Let us now consider the sum definition. We formulate the following claim:

Claim: GMA is not a globally optimal algorithm in the class of $\mathrm{PA}^{\mathrm{dist}}$ according to Opt 2.

Proof: This time, we consider the graph in Figure 4.3 which has 4 nodes. Our pair allocations now are a bit more complicated and we no longer have constant convergents and divergents. Why this change is needed will be addressed in Section 4.4. In the graph in Figure 4.3 we will consider the following paths:

$$\pi_1 = [(\mathrm{J},\mathrm{K}),(\mathrm{H},\mathrm{I}),(\mathrm{F},\mathrm{E}),(\mathrm{B},\mathrm{C}),(\mathrm{L},\mathrm{K}),(\mathrm{H},\mathrm{G})]$$
$$\pi_2 = [(\mathrm{J},\mathrm{K}),(\mathrm{H},\mathrm{I}),(\mathrm{F},\mathrm{E}),(\mathrm{B},\mathrm{A})]$$
$$\pi_3 = [(\mathrm{D},\mathrm{E}),(\mathrm{B},\mathrm{C}),(\mathrm{L},\mathrm{K}),(\mathrm{H},\mathrm{G})]$$

[4] At least the connected component these paths lie in.

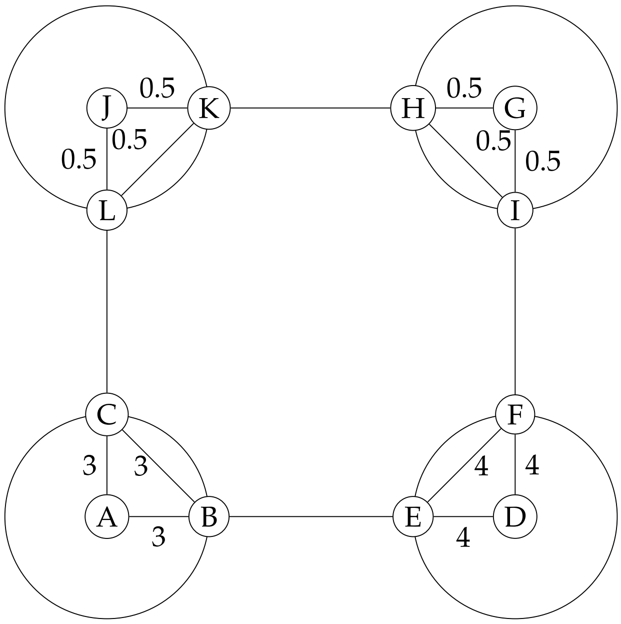

Figure 4.3: Graph for which GMA is not globally optimal according to Opt 2.

We observe, that when calculating the allocations for these paths we again have access to the entire graphs topology because they visit all the nodes in the graph and thus get access to all the nodes' allocation matrices. We can again modify the allocation of these paths in only this graph without violating the locality property of the $\mathrm{PA}^{\mathrm{dist}}$ problem.

If we were to decrease the allocation of π_1 by ϵ (where $\epsilon < \mathcal{G}(\pi_1)$) and increase the allocation of π_2 and π_3 by ϵ that the allocation on all pairs of interfaces with the exception of (B,A) and (D,E) remain the same. The allocation on (B,A) and (D,E) would be increased by ϵ.

In the following, we will show that the sum of the allocation of all paths using the pair interface (B,A) and (D,E) is smaller than the pair allocation (3 respectively 4). From this, it will follow that one can construct an algorithm similarly to Algorithm $\mathcal{B}$ which proves that GMA is not globally optimal in the sense of Opt 2. We will not explicitly present this algorithm, nor will we prove all of its properties as these work completely analogous to how they did for Algorithm $\mathcal{B}$.
Let us start with the pair interface (D,E).

All paths that use (D,E) will either be the path $\pi = [(D, E), (B, A)]$, $\pi' = [(D, E), (B, C), (L, J)]$, or will have the prefix $\pi'' = [(D, E), (B, C), (L, K)]$. We calculate $\mathcal{G}(\pi) = \frac{3}{2}$, $\mathcal{G}(\pi') = \frac{1}{8}$ and $\mathcal{G}(\pi'') = \frac{1}{8}$. From which we conclude that the sum of the allocations of all paths using the pair interface (D,E) is at most $\frac{14}{8}$.

The same argument can be made for (A,B). Observe that since $CON_{\mathrm{I}} = 1$ all paths that end in $[(\ldots, I), (F, E), (B, A)]$ get an allocation of at most 1.

To conclude, we have now shown how we can increase the utilization on these two pair interfaces while keeping the sum on all other pair interfaces the same.

From this we conclude that GMA is not globally optimal according to this definition either.

4.4 Subset of graphs for which GMA is globally optimal

What we will show here, is that in a certain subset of graphs the GMA algorithm is globally optimal according to the definition Opt 2. What we aim to show is that if we consider all finite and infinite paths in this subset of graphs, GMA will fully utilise every link; therefore, it will be impossible to increase the sum on any link without causing over-allocation.

Henceforth, we make the following assumption:

Let i be some interface in a node a and j an interface in a node b. If there is a link between i and j we assume the following conditions hold:

$$\begin{aligned} DIV_i^{(a)} &= CON_j^{(b)} \\ CON_i^{(a)} &= DIV_j^{(b)} \end{aligned}$$

That is, whenever we have a direct link between two nodes, we assume that the allocation matrices are set such that the sum over all the allocation entries entering the interface by this link is equal to the sum over all the allocation entries leaving the interface by this link.

This is a realistic assumption for many real-world graphs, as usually the limiting factor is link capacity between nodes. Convergents and divergents are defined such that they do not exceed these capacities, but nodes would want to set their allocation matrices such that they fully use the inter-node capacity.

The useful thing about this assumption is that it allows us to simplify Equation (2.1). The key observation is that every node becomes a minimizer. We define $\mathcal{G}_x(\pi)$ as before in Section 3.5. That is, it is the GMA equation where we plug in x as the minimizer.

Let π be some arbitrary path of length ℓ and x some arbitrary node in this path. We first assume that $x \neq 1$. Using the above assumption, we then find:

$$\mathcal{G}_x(\pi) = \left(\prod_{k=1}^{x-1} \frac{M_{i,j}^{(k)}}{CON_j^{(k)}} \cdot M_{i,j}^{(x)} \cdot \prod_{k=x+1}^{\ell} \frac{M_{i,j}^{(k)}}{DIV_i^{(k)}}\right) \tag{4.1}$$

$$= \left(\prod_{k=1}^{x-2} \frac{M_{i,j}^{(k)}}{CON_j^{(k)}} \cdot \frac{M_{i,j}^{(x-1)}}{CON_j^{(x-1)}} M_{i,j}^{(x)} \cdot \prod_{k=x+1}^{\ell} \frac{M_{i,j}^{(k)}}{DIV_i^{(k)}}\right) \tag{4.2}$$

$$= \left(\prod_{k=1}^{x-2} \frac{M_{i,j}^{(k)}}{CON_j^{(k)}} \cdot M_{i,j}^{(x-1)} \frac{M_{i,j}^{(x)}}{DIV_i^{(x)}} \cdot \prod_{k=x+1}^{\ell} \frac{M_{i,j}^{(k)}}{DIV_i^{(k)}}\right) \tag{4.3}$$

$$= \left(\prod_{k=1}^{x-2} \frac{M_{i,j}^{(k)}}{CON_j^{(k)}} \cdot M_{i,j}^{(x-1)} \cdot \prod_{k=x}^{\ell} \frac{M_{i,j}^{(k)}}{DIV_i^{(k)}}\right) = \mathcal{G}_{x-1}(\pi) \tag{4.4}$$

Where the above assumptions were used in Equation (4.2).

Similarly, if we assume $x \neq \ell$ we find:

$$\mathcal{G}_x(\pi) = \left(\prod_{k=1}^{x-1} \frac{M_{i,j}^{(k)}}{CON_j^{(k)}} \cdot M_{i,j}^{(x)} \cdot \prod_{k=x+1}^{\ell} \frac{M_{i,j}^{(k)}}{DIV_i^{(k)}}\right)$$

$$= \left(\prod_{k=1}^{x-1} \frac{M_{i,j}^{(k)}}{CON_j^{(k)}} \cdot M_{i,j}^{(x)} \cdot \frac{M_{i,j}^{(x+1)}}{DIV_i^{(x+1)}} \cdot \prod_{k=x+2}^{\ell} \frac{M_{i,j}^{(k)}}{DIV_i^{(k)}}\right)$$

$$= \left(\prod_{k=1}^{x-1} \frac{M_{i,j}^{(k)}}{CON_j^{(k)}} \cdot \frac{M_{i,j}^{(x)}}{CON_j^{(x)}} \cdot M_{i,j}^{(x+1)} \cdot \prod_{k=x+2}^{\ell} \frac{M_{i,j}^{(k)}}{DIV_i^{(k)}}\right) = \mathcal{G}_{x+1}(\pi)$$

Using this, we conclude the following:

$$\mathcal{G}(\pi) = \prod_{k=1}^{\ell} M_{i,j}^{(k)} \cdot \prod_{k=2}^{\ell} \frac{1}{DIV_i^{(k)}} = \prod_{k=1}^{\ell} M_{i,j}^{(k)} \cdot \prod_{k=1}^{\ell-1} \frac{1}{CON_j^{(k)}} \tag{4.5}$$

So when calculating the GMA equation we can choose an arbitrary node as the minimizer.

At this point, it is also important to note that this formula, just like the GMA formula, can be used to calculate the allocation on preliminary (non-terminated) paths.

In the following, we analyze the maximal utilization of any pair interface (i, j) in some arbitrary node a. To do this, we will assume we are in a case where we are allocating resources to all possible paths.

We will prove that for any node a and pair of interfaces i, j the pair allocation $M_{i,j}^{(a)}$ is equal to the sum of the resource allocation of all paths using that interface pair under the above assumptions.

The proof will be structured similarly to the proof of no-over allocation in the GMA paper [12]. So we let i, j be arbitrary interfaces in some arbitrary node a. We aim to show that the pair allocation $M_{i,j}^{(a)}$ is equal to the sum of the resource allocations of all paths going through that interface pair. To show this, we perform a case distinction on the pair i, j:

4.4.1 Case 1: i is a local interface

We define S_t^x as the set of terminated paths of length at most x that start in (i, j) and S_p^x be the set of preliminary paths of length exactly x that start in (i, j). We will show that:

$$\forall x \geq 1 : \sum_{\pi \in S_p^x} \mathcal{G}(\pi) + \sum_{\pi \in S_t^x} \mathcal{G}(\pi) = M_{i,j}^{(a)}$$

from which the desired property directly follows.

Proof: We perform a proof by induction over x.
Base case x=1: We have $S_p^1 = \{[i, j]\}$ and $S_t^1 = \{\}$
Since $\mathcal{G}([i, j]) = M_{i,j}^{(a)}$ we conclude:

$$\sum_{\pi \in S_p^x} \mathcal{G}(\pi) + \sum_{\pi \in S_t^x} \mathcal{G}(\pi) = M_{i,j}^{(a)}$$

Induction step:
Induction hypothesis: For some x we assume: $\sum_{\pi \in S_p^x} \mathcal{G}(\pi) + \sum_{\pi \in S_t^x} \mathcal{G}(\pi) = M_{i,j}^{(a)}$; and we then show that: $\sum_{\pi \in S_p^{x+1}} \mathcal{G}(\pi) + \sum_{\pi \in S_t^{x+1}} \mathcal{G}(\pi) = M_{i,j}^{(a)}$

Similarly to the proof of no-over allocation, for some preliminary path π of length ℓ,[5] we let Z be the node which is connected to j^ℓ and we let i^Z be corresponding interface in Z.
We then, using "+" to denote list concatenation, define the local extension of the path π as $E_{loc}(\pi) = \{\pi + [(i^Z, j^Z)]\}$ where j^Z is a local interface and the non local extension as $E_{ext}(\pi) = \bigcup_{j^Z \in I_{ext}^{(Z)}} \{\pi + [(i^Z, j^Z)]\}$ where $I_{ext}^{(Z)}$ is the set of external interfaces of node Z. Their union we then define as $E(\pi)$.

[5]Which we denote here as $[(i, j), (i^2, j^2), ..., (i^\ell, j^\ell)]$

Now onto the actual proof:

$$\begin{aligned}
&\sum_{\pi\in S_p^{x+1}} \mathcal{G}(\pi) + \sum_{\pi\in S_t^{x+1}} \mathcal{G}(\pi) = \left(\sum_{\pi\in S_p^x} \sum_{\phi\in E_{ext}(\pi)} \mathcal{G}(\pi)\right) + \left(\sum_{\pi\in S_t^x} \mathcal{G}(\pi) + \sum_{\pi\in S_p^x} \sum_{\phi\in E_{loc}(\pi)} \mathcal{G}(\pi)\right) \\
&= \sum_{\pi\in S_t^x} \mathcal{G}(\pi) + \sum_{\pi\in S_p^x} \sum_{\phi\in E(\pi)} \mathcal{G}(\pi) \\
&= \sum_{\pi\in S_t^x} \mathcal{G}(\pi) + \sum_{\pi\in S_p^x} \sum_{\phi\in E(\pi)} \min\left(\mathcal{G}(\pi)\cdot \frac{M_{i,j}^{(Z)}}{DIV_i^{(Z)}}, \prod_{k=1}^{\ell} \frac{M_{i,j}^{(k)}}{CON_j^{(k)}} \cdot M_{i,j}^{(Z)}\right)
\end{aligned}$$

In the last step we just distinguish between the two cases where in Equation (2.1) the minimizer stays unchanged or the case where the newly added node becomes the minimizer.

Up until here the proof has been structured as in the proof of no-over allocation. It is at this point where we exploit the property that under our assumptions we can always take the first node as the minimizer. That is, we use Equation (4.5).

$$\begin{aligned}
&\sum_{\pi\in S_t^x} \mathcal{G}(\pi) + \sum_{\pi\in S_p^x} \sum_{\phi\in E(\pi)} \min\left(\mathcal{G}(\pi)\cdot \frac{M_{i,j}^{(Z)}}{DIV_i^{(Z)}}, \prod_{k=1}^{\ell} \frac{M_{i,j}^{(k)}}{CON_j^{(k)}} \cdot M_{i,j}^{(Z)}\right) \\
&= \sum_{\pi\in S_t^x} \mathcal{G}(\pi) + \sum_{\pi\in S_p^x} \sum_{\phi\in E(\pi)} \mathcal{G}(\pi)\cdot \frac{M_{i,j}^{(z)}}{DIV_i^{(Z)}} = \sum_{\pi\in S_t^x} \mathcal{G}(\pi) + \sum_{\pi\in S_p^x} \mathcal{G}(\pi) = M_{i,j}^{(a)}
\end{aligned}$$

The second last step follows from the definition of the divergent and the last step follows from our induction hypothesis.

4.4.2 Case 2: j is a local interface

This case works similarly to Case 1. The only difference is that we extend our paths backwards instead of forwards and instead of the divergent we use the convergent. We define S_t^x as the set of paths of length at most x that start in a local interface and end in (i, j) and S_p^x as the set of paths that start at an external interface, have length exactly x, and end in (i, j).
We will again show that:

$$\forall x \geq 1 : \sum_{\pi\in S_p^x} \mathcal{G}(\pi) + \sum_{\pi\in S_t^x} \mathcal{G}(\pi) = M_{i,j}^{(a)}$$

from which the desired property follows.

Proof: We perform a proof by induction over x.
Base case x=1: We have $S_p^1 = \{[i,j]\}$ and $S_t^1 = \{\}$
Since $\mathcal{G}([i,j]) = M_{i,j}^{(a)}$, we again conclude:

$$\sum_{\pi \in S_p^x} \mathcal{G}(\pi) + \sum_{\pi \in S_t^x} \mathcal{G}(\pi) = M_{i,j}^{(a)}$$

Induction step:
Induction hypothesis: For some x we assume: $\sum_{\pi \in S_p^x} \mathcal{G}(\pi) + \sum_{\pi \in S_t^x} \mathcal{G}(\pi) = M_{i,j}^{(a)}$; and we then show that: $\sum_{\pi \in S_p^{x+1}} \mathcal{G}(\pi) + \sum_{\pi \in S_t^{x+1}} \mathcal{G}(\pi) = M_{i,j}^{(a)}$

Assume we have a path $\pi \in S_p^x$ of the form $\left[\left(i^x, j^x\right), \ldots, \left(i^2, j^2\right), \left(i, j\right)\right]$, we let Z be the node which is connected to i^x and we let j^Z be the corresponding interface in Z.

We then define the local extension of the path π as $E_{loc}(\pi) = \left\{\left[\left(i^Z, j^Z\right)\right] + \pi\right\}$, where i^Z is a local interface and the non-local extension as
$E_{ext}(\pi) = \bigcup_{i^Z \in I_{ext}^{(Z)}} \left\{\left[\left(i^Z, j^Z\right)\right] + \pi\right\}$. Their union we again define as $E(\pi)$.

Now onto the actual proof:

$$\begin{aligned}
&\sum_{\pi \in S_p^{x+1}} \mathcal{G}(\pi) + \sum_{\pi \in S_t^{x+1}} \mathcal{G}(\pi) = \left(\sum_{\pi \in S_p^x} \sum_{\phi \in E_{ext}(\pi)} \mathcal{G}(\pi)\right) + \left(\sum_{\pi \in S_t^x} \mathcal{G}(\pi) + \sum_{\pi \in S_p^x} \sum_{\phi \in E_{loc}(\pi)} \mathcal{G}(\pi)\right) \\
&= \sum_{\pi \in S_t^x} \mathcal{G}(\pi) + \sum_{\pi \in S_p^x} \sum_{\phi \in E(\pi)} \mathcal{G}(\pi) \\
&= \sum_{\pi \in S_t^x} \mathcal{G}(\pi) + \sum_{\pi \in S_p^x} \sum_{\phi \in E(\pi)} \min\left(\frac{M_{i,j}^{(Z)}}{CON_j^{(Z)}} \cdot \mathcal{G}(\pi), \prod_{k=1}^{\ell} M_{i,j}^{(Z)} \cdot \frac{M_{i,j}^{(k)}}{DIV_i^{(k)}}\right)
\end{aligned}$$

Again, we use Equation (4.5) here and observe:

$$\begin{aligned}
&\sum_{\pi \in S_t^x} \mathcal{G}(\pi) + \sum_{\pi \in S_p^x} \sum_{\phi \in E(\pi)} \min\left(\frac{M_{i,j}^{(Z)}}{CON_j^{(Z)}} \cdot \mathcal{G}(\pi), \prod_{k=1}^{\ell} M_{i,j}^{(Z)} \cdot \frac{M_{i,j}^{(k)}}{DIV_i^{(k)}}\right) \\
&= \sum_{\pi \in S_t^x} \mathcal{G}(\pi) + \sum_{\pi \in S_p^x} \sum_{\phi \in E(\pi)} \frac{M_{i,j}^{(Z)}}{CON_j^{(Z)}} \cdot \mathcal{G}(\pi) = \sum_{\pi \in S_t^x} \mathcal{G}(\pi) + \sum_{\pi \in S_p^x} \mathcal{G}(\pi) = M_{i,j}^{(a)}
\end{aligned}$$

The second last step follows from the definition of the convergent and the last step follows from our induction hypothesis.

4.4.3 Case 3: Neither i nor j is a local interface

The proof of this case will be structured in two phases. The first phase is done analogously to Case 2 and the second phase is done analogously to Case 1.

First, we define S_t^x as the set of preliminary paths of length at most x that start in a local interface and end in (i,j) and S_p^x as the set of preliminary paths that start at an external interface, have length x and end in (i,j).

Completely analogously to Case 2 one can then show that

$$\forall x \geq 1 : \sum_{\pi \in S_p^x} \mathcal{G}(\pi) + \sum_{\pi \in S_t^x} \mathcal{G}(\pi) = M_{i,j}^{(a)}$$

For the second phase we take an arbitrary path $\pi \in S_p^x$ or $\pi \in S_t^x$ of length ℓ. We then define $S_{\pi,p}^x$ as the set of all preliminary paths that start with π and have length of exactly $\ell + x$. Similarly, we define $S_{\pi,t}^x$ as the set of all paths that end in a local interface that start with π and have length smaller or equal to $\ell + x$. Completely analogously to Case 1, one can then show that:

$$\forall x \geq 1 : \sum_{\pi' \in S_{\pi,p}^x} \mathcal{G}(\pi') + \sum_{\pi' \in S_{\pi,t}^x} \mathcal{G}(\pi') = \mathcal{G}(\pi)$$

From the combination of the above two statements the desired property then directly follows, which concludes the proof.

5 Optimal Initialization of Allocation Matrices

5.1 Problem statement

In this section we analyze the tradeoffs made by the nodes when choosing how to initialize their allocation matrices. Under the assumption that all nodes follow the same method we find an optimal method.

We allow the method to depend on some aspects of the graph and the paths over which we are trying to optimize, although it should not depend on the entire topology as that would defeat the purpose of $\mathrm{PA}^{\mathrm{dist}}$.

We will additionally use the well known Barabási–Albert random graph model [4] to generate scale-free graphs on which we will test our findings. Scale-free graphs have been found to be a good approximation of real-world technological networks [9].

Finally, there is a game theoretical tradeoff in these initializations. We will discuss the issue and some methods of dealing with it.

5.2 Setup

In GMA we assume all nodes in the network have some policy by which they determine how they want to split up their capacity. They express this through their allocation matrices M. The problem of how to best initialize these matrices has yet to be answered. Here we aim to analyse which policies lead to a globally optimal state. That is, what are some ways of initializing the allocation matrices such that we optimise some criteria.

The goal is to find some policy that if followed by all nodes is the best possible such policy. One limitation of this is that the solution we will find will not apply to the general case in which we allow all nodes to have completely different policies and arbitrary allocation matrices.

5.3 Maximize the sum of allocations

A first intuition might be that given some graph and a set of paths on that graph, a good goal would be to maximize the sum of all the allocations to these paths.

To show that this is not a good metric we look at the example given in Figure 5.1. We have $n > 2$ nodes that form a cycle. Each node has one interior and two external interfaces.

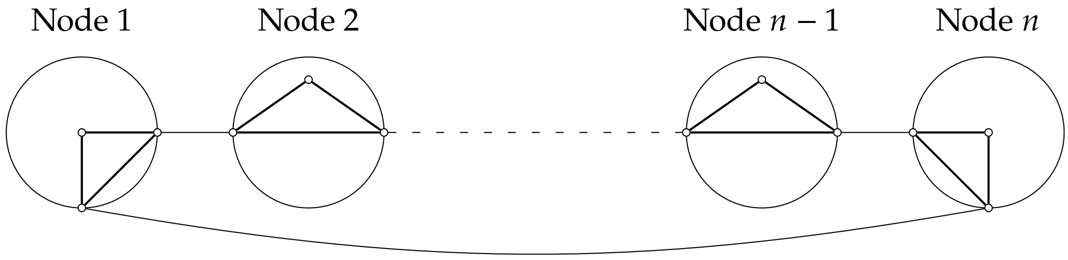

Figure 5.1: Example where maximizing the sum does not lead to good results.

Assume we now want to find some policy for the nodes such that we maximize the sum over all simple paths in this network. That is the set of paths we are considering is the set of all simple paths in this graph.

Let us assume all nodes split up their capacity as follows: They provide a fraction of $\alpha \in [0,1]$ to the internal interface and split up the remaining capacity evenly between their two neighbors. Additionally, we assume that all external interfaces have a convergent and divergent of 1.

Under these assumptions the allocation matrices would look as follows:

$$M_{i,j} = \begin{cases} 0 & \text{if } i = j \\ \alpha & \text{if } i \text{ or } j \text{ is a local interface} \\ \beta := 1 - \alpha & \text{otherwise} \end{cases}$$

We will now show the following:

Claim: In the above case the sum of the allocations to all simple paths has a unique maximum for $\alpha = 1$.

Proof: Let us consider some path π that visits k nodes in the above graph. So this path has length $k - 1$.

It is clear that $\mathcal{G}(\pi) = \alpha^2 \cdot \beta^{k-2}$. So as an example, a path that visits two nodes (has a length of one) has an allocation of α^2 and if we set α to 1 the allocation would be 1.

Now in the above graph it is clear that for every node a we have two neighbors. So, two simple paths of length one originate from any node. If we set $\alpha = 1$ those paths will get an allocation of 1. While all other paths would get an allocation of 0. Therefore, the sum of our allocations would be $2 \cdot n$.
To see that this is a maximum consider some other value of $\alpha \in [0, 1)$, such that the sum of paths is at least $2 \cdot n$.

Each path must originate from the interior interface of some node. We know that for every pair interface the sum of the paths using that pair interface is smaller than the allocation given to that. So for every node at most $2 \cdot \alpha$ will be allocated to all the paths originating from its interior interface. This then implies that in total at most $2 \cdot n \cdot \alpha$ can be allocated to all our paths. It is clear that for $\alpha \in [0, 1)$ it does not hold that $2 \cdot n \leq 2 \cdot n \cdot \alpha$, from which the correctness of the claim follows.

Because of this we conclude that the goal of maximizing the sum of allocations is not a sensible one to focus on as this leads to all paths that visit more than two nodes receiving an allocation of 0, which causes connectivity to collapse.

5.4 Maximizing the product of allocations

After finding that the sum is not a good metric, the next natural choice would be the product. We therefore want to find which initialization for some set of paths leads to a maximum in the product of the allocations. The reason for this is that the product will not run into the problem of it giving most paths an allocation of 0 as giving any path an allocation of 0 (or even close to 0) will make the product zero or tiny.

Let us consider an arbitrary graph. In the following, we make a simplifying assumption that all interfaces have the same convergents and divergents (:= 1). This assumption is required to make the problem tractable.

Again, we assume our policy for a node is that it gives its interior interface a fraction of α and evenly splits the remaining capacity between all of its neighbors.

First in Section 5.4.1 we will maximize the product under the constraint that all nodes have at least two neighbors. Once we have analyzed the simpler case, we will discuss the general case in Section 5.4.2.

Each node having at least two neighbors simplifies our argument as providing an α fraction to the internal interface of a node with only one neighbor would lead to the remaining $1 - \alpha$ fraction being unused which would be wasteful.

5.4.1 Two-neighbor-minimum case

As discussed above, we make the simplifying assumption here that all nodes have at least two neighbors.

Thus, for some node a with $d^{(a)}$ neighbors its allocation matrix is constructed as follows:

$$M^{(a)}_{i,j} = \begin{cases} 0 & \text{if } i = j \\ \alpha & \text{if } i \text{ or } j \text{ is a local interface} \\ \frac{1-\alpha}{d^{(a)}-1} & \text{otherwise} \end{cases}$$

Again, given some path π that visits ℓ nodes we can calculate the allocation of the path using GMA (Eq. (2.1)):

$$\mathcal{G}(\pi) = \prod_{a=1}^{\ell} M_{i,j}^{(a)} = \alpha^2 \cdot (1-\alpha)^{\ell-2} \cdot \prod_{a=2}^{\ell-1} \frac{1}{d^{(a)} - 1}$$

Let us first consider the case where we want to choose $\alpha \in [0,1]$ to maximize the allocation of only this specific path π. If $\ell = 2$ it is clear that this is maximal for $\alpha = 1$.

Now let us look at the case where $\ell > 2$. We define $c := \prod_{a=2}^{\ell-1} \frac{1}{d^{(a)}-1}$

$$\begin{aligned}\frac{\partial}{\partial a}\mathcal{G}(\pi) &= c \cdot \left(2\alpha \cdot (1-\alpha)^{\ell-2} - (\ell-2) \cdot (1-\alpha)^{\ell-3} \cdot \alpha^2\right) \\ &= c \cdot (1-\alpha)^{\ell-3} \cdot \alpha \cdot (2 - \alpha \cdot \ell)\end{aligned}$$

We observe that the only $\alpha \in (0,1)$ for which the derivative is 0 is $\alpha = \frac{2}{\ell}$.

Additionally, since $\mathcal{G}(\pi) = 0$ if $\alpha = 0$ or $\alpha = 1$ and $\mathcal{G}(\pi) > 0$ for $\alpha \in (0,1)$, we can conclude that $\alpha = \frac{2}{\ell}$ is the maximum $\alpha \in [0,1]$.

Multiple paths. Now let us consider the case where we have an arbitrary set of paths $\pi_1, \ldots, \pi_n$. We define c_i and ℓ_i for π_i analogously to c and ℓ above for π. We let ℓ be the average number of nodes in these paths so we have $\ell = \frac{1}{n}\sum_{i=1}^{n} \ell_i$.

Our goal is to maximize $\prod_{i=1}^{n} \mathcal{G}(\pi_i)$.

$$\begin{aligned}\prod_{i=1}^{n} \mathcal{G}(\pi_i) &= \prod_{i=1}^{n} c_i \cdot \alpha^2 \cdot (1-\alpha)^{\ell_i-2} \\ &= \alpha^{2n} \cdot (1-\alpha)^{\sum_{i=1}^{n} \ell_i - 2} \cdot \prod_{i=1}^{n} c_i \\ &= \alpha^{2n} \cdot (1-\alpha)^{n(\ell-2)} \cdot \prod_{i=1}^{n} c_i\end{aligned}$$

If we then define $c := \prod_{i=1}^{n} c_i$, we once again observe that if $\ell = 2$ the optimal α would be 1.

Now let us consider the case where $\ell > 2$. We observe that if $\alpha = 0$ or $\alpha = 1$ our product is 0. Additionally, for all other $\alpha \in (0,1)$ this product is greater than 0. So we again use the derivative to find the maximal α.

$$\begin{aligned}\frac{\partial}{\partial a}\prod_{i=1}^{n} \mathcal{G}(\pi_i) &= c \cdot \left(2n \cdot \alpha^{2n-1} \cdot (1-\alpha)^{n(l-2)} - \alpha^{2n} \cdot n \cdot (\ell-2) \cdot (1-\alpha)^{n(l-2)-1}\right) \\ &= c \cdot n \cdot (\alpha\ell - 2) \cdot \alpha^{2n-1} \cdot (1-\alpha)^{n(\ell-2)-1}\end{aligned}$$

Once again, we observe that the only root of the derivative in the range $(0, 1)$ is for $\alpha = \frac{2}{\ell}$. From which we again conclude that the product is maximized for that value of α.

So we can conclude that, if we know the average path length, we can determine the optimal initialization for the case where each node has at least two neighbors.

5.4.2 General case

As mentioned before, the previous result depends on the assumption that each node has at least two neighbors. To generalize, we will initialize our matrices as follows:

$$M_{i,j}^{(a)} = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{if } d^{(a)} = 1 \\ \alpha & \text{if } i \text{ or } j \text{ is a local interface} \\ \frac{1-\alpha}{d^{(a)}-1} & \text{otherwise} \end{cases} \tag{5.1}$$

The only change here is that if some node only has one neighbor, we now allocate all our bandwidth to that neighbor. Once again, we will find the optimal value for α. Let us proceed similarly to above:

We observe four possible cases for some path π of length $\ell + 1$. Where c is defined as before.

$$\mathcal{G}(\pi) = \begin{cases} \alpha^2 \cdot (1-\alpha)^{\ell-2} \cdot c & \text{if } \pi \text{ starts and ends in nodes with more than 1} \\ & \text{neighbors} \\ 1 \cdot \alpha \cdot (1-\alpha)^{\ell-2} \cdot c & \text{if } \pi \text{ starts at a node with more than 1 and ends} \\ & \text{at a node with 1 neighbor} \\ \alpha \cdot 1 \cdot (1-\alpha)^{\ell-2} \cdot c & \text{if } \pi \text{ starts at a node with 1 and ends at a node} \\ & \text{with more than 1 neighbor} \\ 1 \cdot 1 \cdot (1-\alpha)^{\ell-2} \cdot c & \text{if } \pi \text{ starts and ends in nodes with 1 neighbor} \end{cases}$$

We then define x as the sum of the number of paths that start at a node with only one interface and the number of paths that end at a node with only one interface. So x is the number of paths in the second or third cases above plus two times the number of paths in the fourth case. We then find:

$$\prod_{i=1}^{n} \mathcal{G}(\pi_i) = \alpha^{2n-x} \cdot (1-\alpha)^{n(\ell-2)} \cdot \prod_{i=1}^{n} c_i$$

Once again, we first observe that if $\ell = 2$ the optimal $\alpha = 1$. Additionally, if $x = 2n$ (i.e., all paths start and end in nodes with only 1 neighbor) we have the optimal $\alpha = 0$.

For the remaining cases we again take the derivative with regards to α:

$$\frac{\partial}{\partial a}\prod_{i=1}^{n}\mathcal{G}(\pi_i) = -c\cdot(n(\alpha l-2)-\alpha x+x)\cdot\alpha^{2n-x-1}\cdot(1-\alpha)^{n(l-2)-1}$$

The only root for $\alpha \in (0,1)$ of the derivative is $\alpha = \frac{2n-x}{\ell\cdot n-x}$. We observe that if we are not in any of the above cases the product is 0 if $\alpha = 0$ or $\alpha = 1$. Therefore, if we are not in one of the above two cases, we find the optimal α to be $\frac{2n-x}{\ell\cdot n-x}$.

We observe that this point is harder to compute than the previous one. While in many networks one might know the average path length, it is harder to know what fraction of paths start or end in a node with only one neighbor.

5.5 Simulation results

In this section we will use simulations to validate the previous findings. We will generate random graphs according to the Barabási–Albert model. Then for every pair of nodes we take the shortest path between those two nodes. For various values of $\alpha \in [0,1]$ we will then initialize the matrices as described in Equation (5.1).

Using these matrices, we then compute the GMA allocation on all of these paths and compute the product. The yellow bar in the plots represents the general case optimum calculated in Section 5.4.2, whereas the red bar represents the optimal value in the non general case discussed in Section 5.4.1. In these plots also observe that the y-scale is logarithmic and all values will have been divided by the minimum product. So they show the increase.

We will consider Barabási–Albert graphs. We will start off with 3 nodes and then one by one according to the rules outlined by Barabási–Albert, attach the new nodes.

5.5.1 Single attachment

In this first section we will consider Barabási–Albert graphs in which when a new node joins the network it attaches itself to a single existing one. Meaning our *attachment parameter* is one. We observe that due to this construction our graphs will be acyclic, as the number of edges will always be one smaller than the number of nodes.

To begin with, we take a look at Figure 5.2. We attach nodes until we have 100 and then for various values of α plot the product over all simple paths.

What we observe here, is that when the topology is generated like this there are a lot of nodes that end up with only one neighbor. Therefore, there is a significant difference between the two optima.

As another simulation result, we can plot the sum over all these paths. According to the results in Section 5.3 we expect this to be maximal for $\alpha = 1$. The results are plotted in Figure 5.3 and they validate the expected results.

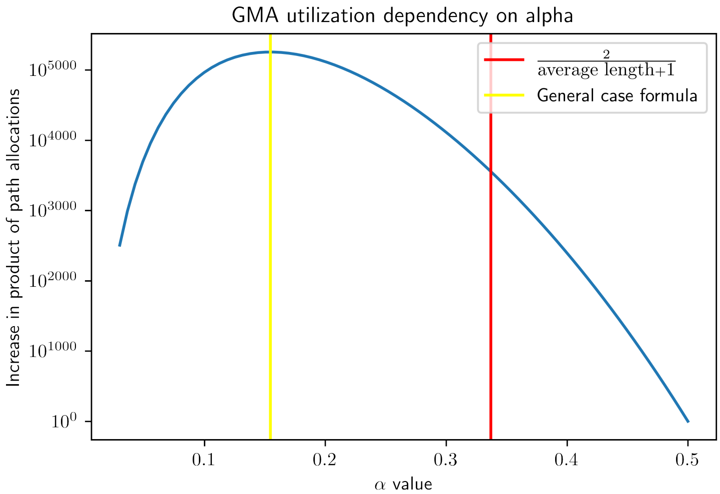

Figure 5.2: Product over all path allocations in a graph with 100 nodes generated according to the Barabási–Albert procedure with an attachment parameter of one.

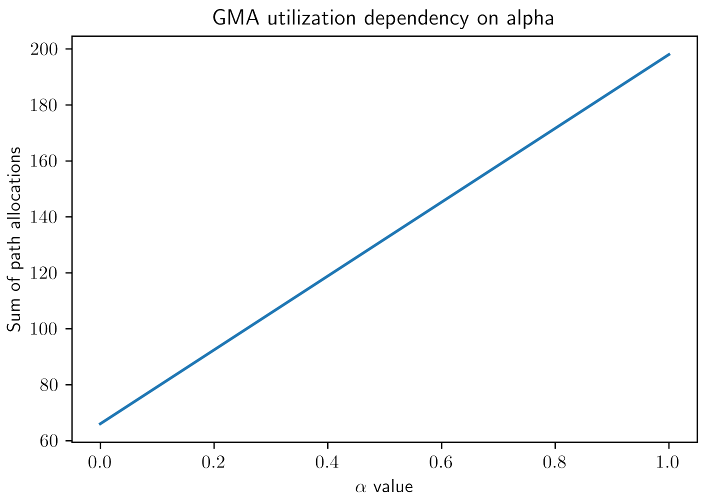

Figure 5.3: Sum over all path allocations in a graph generated according to the Barabási–Albert procedure, with an attachment parameter of one.

Additionally, we can look at the cover in these graphs. The *α-cover* for every node measures what fraction of other nodes are reachable with allocations of at least α which in many applications is a metric of interest. Note that all link capacities are set to 1. So the cover is scaled by that amount.

The setup remains the same as above. We have 100 nodes in total and an attachment parameter of one. In Figure 5.4 we can see the 10^{-4} and 10^{-6} cover and their dependency on α. For every α value we calculate the cover of the node with the highest cover, the cover of the node with the smallest cover, and the average cover of all nodes.

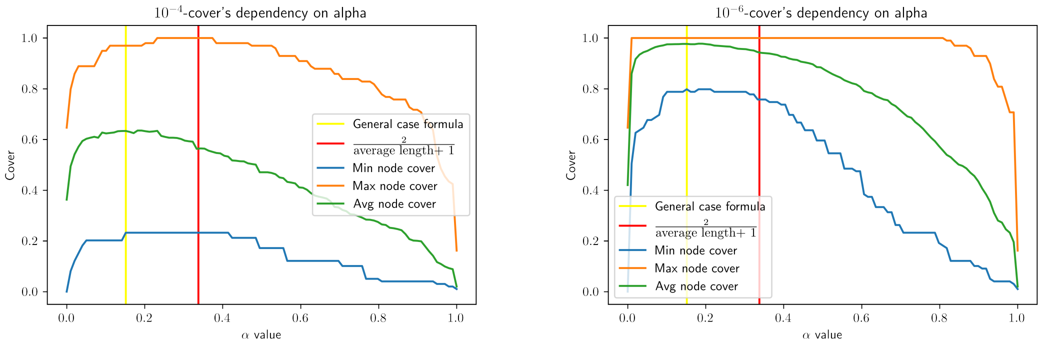

Figure 5.4: How the 10^{-4} and 10^{-6} cover in the Barabási–Albert graphs, with attachment parameter of one, change together with α.

From the plots of the cover, we make the interesting observation that the optimum for the product seems to be a good approximation of the optimal initialization for the cover.

5.5.2 Multi attachment

In this section we will do the same as in Section 5.5.1. The only difference now is that when a new node joins the network it will attach itself to two existing nodes instead of one. Thus, our attachment parameter now is two. Due to this, no node will have a degree of one and we find the two optima to be the same.

Again, to start off with let us consider 100 nodes in total. We get the results in Figure 5.5. As expected, we can see that the optima are the same. We also can see that they do correspond to the optimal initialization.

We then again take a look at the cover. We have a Barabási–Albert graph with 200 nodes and an attachment parameter of two. This time we provide the cover for both the shortest and the 5 shortest paths. The results are in Figure 5.6.

In the plots for the cover, we make two important observations. On the one hand we can see that they are more complicated functions which indicates that maximizing the cover is harder than when compared to the product. This behavior is expected as the cover is threshold based.

However, we also observe that the points which optimise the product tend to be around the optimal points for the cover and provide pretty good points for that, which would indicate that maximizing the product may be a good approximation to maximizing the cover.

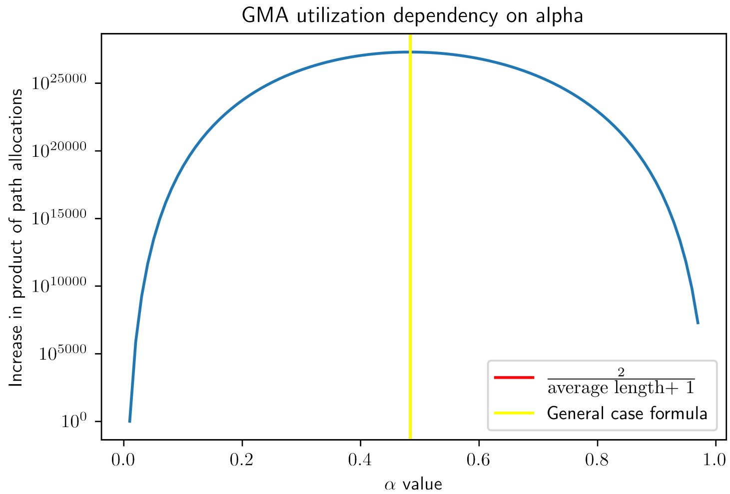

Figure 5.5: Product over all path allocations in the graphs generated according to the Barabási–Albert procedure with an attachment parameter of two.

5.6 Game theoretical trade-off

In the above we assumed that all nodes would follow the same policy with the goal of maximizing the greater good. Unfortunately, this assumption is very idealistic. The GMA algorithm is applied to the distributed setting; there is no central control to enforce that the nodes actually have these policies.

Assume we are some node a. And we want to determine some local policy in which we give our internal interface an allocation of α' and split the rest between our external interfaces as before. The paths we care about in the network will be paths that either start or end in our local interface. Paths that transit through us don't really bring much benefit to us. Then we would want to know what the optimal α' would be. For any simple path π which starts or ends in our local interface we observe that the allocation given to the path is $\mathcal{G}(\pi) = \alpha' \cdot c$. Where c is a constant dependent on all other nodes policies.

It then becomes clear that if we want to maximize the sum or the product over all such paths, we find the optimal $\alpha' = 1$. Thus, any node which acts in its self interest would just set $\alpha' = 1$.

What we then observe is that if every node acts in this fashion we end up in the same situation as we encountered in Section 5.3 when maximizing the sum over allocations. In fact, one can determine that the only Nash equilibrium in this game is if all nodes set their α to 1, as their own payoff increases with their α. In this Nash equilibrium we find that all paths of length one get an allocation of 1 and all other paths would get an allocation of 0. This would lead to a total collapse of the network as we would no longer have wider connectivity. The average cover would drop to the average number of neighbors divided by the total number of nodes

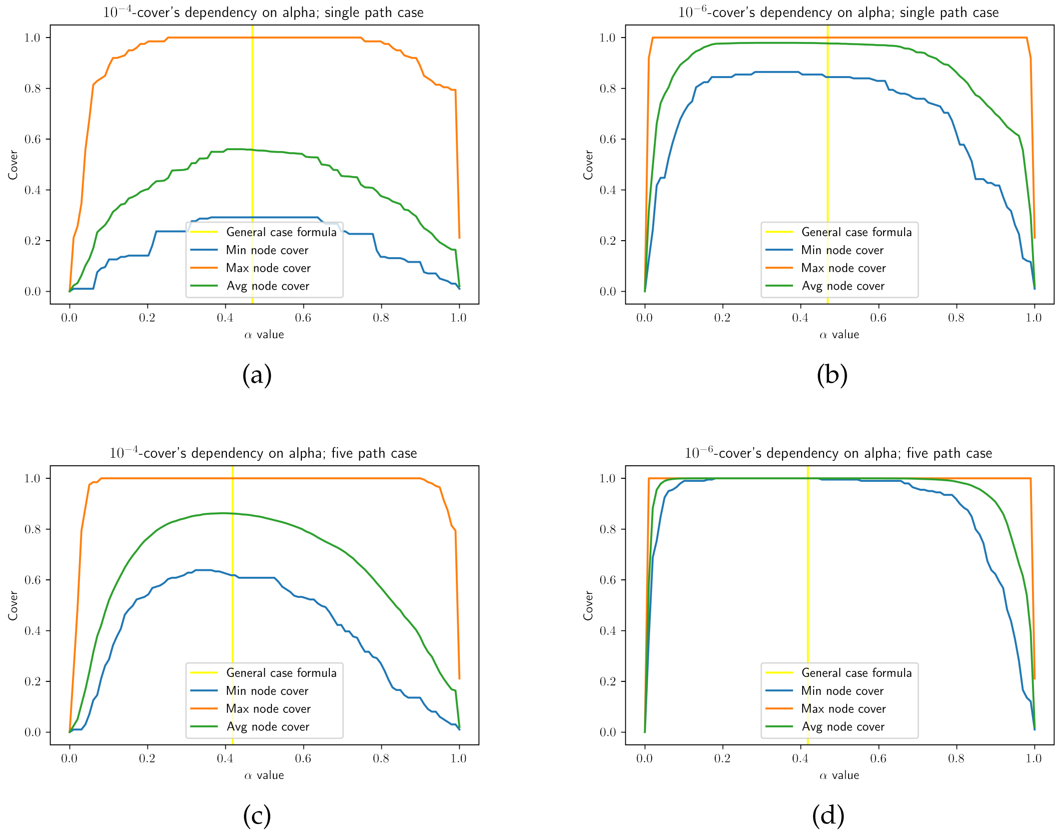

Figure 5.6: All plots concern Barabási–Albert graphs with 200 nodes and an attachment parameter of two. The first two images are the 10^{-4} (a) and 10^{-6} (b) cover when considering only the shortest paths. The last two images ((c),(d)) are the cover when we consider allocations on the (up to) 5 shortest paths between any pair of nodes.

which in the above case would be $\frac{4}{200}$. Therefore, we find ourselves in a situation analogous to the Prisoner's Dilemma [23] and the tragedy of the commons [15]. Both of these are situations in which the best outcome for all would arise if all agents cooperate, but they instead are incentivized to act selfishly.

There are two important properties of GMA and PA$^{\text{dist}}$ which will enable some solution approaches to remedy this problem. First off, assume we are some node a; By trying to calculate the allocations of some path passing through some neighbor b of ours, we get a look at b's allocation matrix and can check if b is acting selfishly or not. Secondly, any two neighboring nodes have some link between them. With these links often come some form of agreements, payments, or other contractual obligations.
We now present two possible approaches which will incentivize nodes to act for the greater good.

Tit for Tat

In Section 2.4 we discussed how in the iterated prisoner's dilemma Tit for Tat is a strategy which performs well. One idea would be to generalize this to allocation matrices. It is unspecified how this might be done in the PA^{dist} problem but in general there have to be ways to update these matrices in regular intervals, as this is needed to allow new nodes to join the network. So for each neighbor we imagine we are playing an iterated game. If in the last period they acted nicely, we divide our bandwidth for them evenly. If they behaved selfishly we could, as an example, artificially shrink the convergent and divergent at the interface connected to that node by say a factor of two which would reduce all the allocations of paths going through that interface. Of course this would also harm paths which would only use the selfishly acting node as a transit point, but it still would disincentivize the node from acting selfishly. Therefore, nodes could follow this strategy based on Tit-for-Tat to incentivize honest behavior.

One issue with this solution, is depending on how the punishment is done it may look, to the other neighbors, like you are being selfish by giving that neighbor a lower priority. But this can be remedied through clear communication on what is happening.

Contractual obligations

Another solution would be to add some contractual obligations.

Whenever two nodes agree to build a link between them this might come with a contract. If we consider a network where we know the average path has a length of four, we then could create contracts with all our neighbors that enforce that everyone optimizes their allocation matrices for an average path length of four. Breaches of this contract could, for example, be punished by some financial obligations.

We thus introduce some external cost other than bandwidth for nodes which act selfishly and rely on external entities, like governments, to help enforce these contracts in case someone violates them.

If all nodes in the network enforced such obligations on their neighbors, we could then avoid the issues arising from selfish behavior.

Of course, these contracts would also need to be expanded such that in case we find the average path length to be different that they can be updated. Still, this might be one possible solution to the problem.

6 Discussion

In this thesis we have further analyzed various properties of the GMA algorithm. We derived a new fairness notion, proved that while GMA is not globally optimal in general that it is under certain restrictions, and discussed various tradeoffs in the initialization of allocation matrices.

Fairness. In Chapter 3 we derived a new fairness notion, NBP-fairness, which is applicable to solutions of the distributed path allocation problem. To our knowledge this is the first such notion and thus will provide a metric with which future solutions to PA^{dist} can be evaluated.

Since we showed that GMA is not globally optimal, we suspect that there may be other solutions to the PA^{dist} problem out there and once those are found NBP-fairness will provide a good way to quickly evaluate and characterize these algorithms.

One current limitation with this notion is whether it is too closely tied to the ideas behind GMA or if it is more widely applicable. We motivated that NBP-fairness does not majorly constrain the algorithms, but it is hard to truly validate this without being able to try and apply NBP-fairness to other solutions.

Global optimality. In Chapter 4 we found that GMA is not globally optimal. Although this result was not what we had hoped to find, it implies that there are other algorithms for the PA^{dist} problem that might provide more optimal allocations than the GMA algorithm. Two major caveats apply to this.

Firstly, in all our examples where GMA was not globally optimal we had paths that learnt the entire topology and thus were able to perform better according to our metrics than GMA by adjusting various such paths in coordination. While it might be possible to do this for all possible topologies it would come at the cost of GMA's linear runtime. Thus, this is an area where we eagerly await new solutions to the PA^{dist} problem. We currently do not know if a globally optimal algorithm is possible without the algorithm having the optimal allocation for a large set of paths predefined. We also believe that in most real-world applications the process of "learning the entire topology" will (i) be infeasible and (ii) will provide a negligible advantage over GMA.

Secondly, we showed that GMA is globally optimal according to the definition Opt 2 in a significant set of graphs. Considering the limiting factor for the convergents and divergents is that they must not exceed the link capacity, we suspect that in many real-world applications we will find them set to their capacity and thus GMA to be globally optimal for those problems. In certain other situations, this assumption may no longer hold though, for example, when performing punishment as discussed for the Tit-for-Tat approach in Section 5.6.

Initialization of allocation matrices. In Chapter 5 we answered various questions in regards to what some of the key tradeoffs in the initialization of the allocation matrices for GMA are. This should provide a good starting point for real world applications to determine some of the tradeoffs between acting more in one's self interest or more for the greater good. Depending on the assumptions made, we found two different optimal points (Sections 5.4.1, 5.4.2). One of these is easier to determine in general graphs than the other, but relies on the assumption that each node has at least two neighbors. Our simulation results indicate that both these points are close to the points which maximize the cover. Thus, our hope is that in most situations it will hold that (i) the two optimal points are close to each other and (ii) the cover at these optimal points is close to the maximum possible cover, as observed in our simulation results.

If (i) holds, without knowing a lot of information about the topology and the paths we are trying to maximize the product for, one can still find a point which is close to the optimum. While the optimum will be difficult to determine in the general case, the simpler formula just relies on knowing the average path length. For example, one might know that for this type of communication the average packet travels x-hops and that would be enough to then find the optimal value. If (ii) holds it would turn out that maximizing the product is a good approximation to maximizing the cover. For many applications having a high enough cover is the most important goal. But since the cover function is a lot more complicated, assumption (ii) would simplify finding this significantly.

We also discussed the game theoretical tradeoffs in the initialization and presented two possible ways to combat the issues that arise from this. While we do not know how well these solution approaches will work, we suspect they should be a helpful starting point for dealing with the problem.

7 Related Work

We discuss here relevant work that was not already discussed in Chapter 2.

A core element of our fairness notion was to come up with a notion which does not suffer from many of the current issues. There have been various other proposed solutions to the TCP fairness paradigm which have attempted instead to just move beyond using fairness. For example, in the paper "On the future of congestion control for the public internet" [10], the authors present an alternative to the current TCP-friendliness paradigm where congestion is instead enforced using economical relationships.

There is also ongoing research into various other inter-domain bandwidth allocation algorithms for path-aware internet architectures. These have different requirements though as they are based on bandwidth reservations, while GMA provides unconditional allocations.

Tradeoffs akin to the ones we touched upon in the game theoretical part of this thesis have manifested in various other contexts as well. For example, for peering agreements where various solutions have been found to bias towards a more optimal solution [11, 19, 24]. These often take a more formal and rigorous approach to finding solutions than we did here.

8 Conclusion

In this thesis we have further motivated the PA^{dist} problem and shown that there is a possibility for future research in this area.

Through NBP-fairness we have analyzed the fairness properties of GMA and seen how those vary from the status quo. From this we concluded that GMA is a fair algorithm. By showing that GMA is not globally optimal we also highlight the need for future research into the PA^{dist} problem with the goal of finding a globally optimal algorithm. Finally, in our research into the optimal initialization of allocation matrices we have shown that given the average path length we can find an initialization which provides a high cover to most nodes. This is very desirable as in most of the real-world applications the cover is the main metric of interest.

Future research. During the thesis we touched upon many questions some of which remain unanswered. We summarize them here.

The first such question arose in the study of NBP-fairness. While one possibility of research lies in applying this fairness notion to future solutions of the PA^{dist} problem, there are interesting research questions relating to what bounds for arbitrary paths can be derived from NBP-fairness similarly to how we did in Section 3.3. It may even be possible for novel solutions to the PA^{dist} problem to be derived from these bounds.

Secondly, regarding global optimality the two main open questions that remain are (i) if some simple modification of GMA is possible which would make the algorithm globally optimal and (ii) if there is some other globally optimal algorithm.

Finally, there are interesting questions relating to the game theoretical aspects and the mechanisms that need to be developed to incentivize cooperation regarding the optimal initialization of allocation matrices. A starting point could be a formal and rigorous analysis on how to disincentivize selfish behavior and motivate cooperation, as has been done for some other similar problems [24].

Bibliography

[1] M. Apostolaki, A. Zohar, and L. Vanbever. Hijacking bitcoin: Routing attacks on cryptocurrencies. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 375–392. IEEE, 2017.

[2] R. Axelrod. Effective choice in the prisoner's dilemma. *Journal of conflict resolution*, 24(1):3–25, 1980.

[3] R. Axelrod. More effective choice in the prisoner's dilemma. *Journal of conflict resolution*, 24(3):379–403, 1980.

[4] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[5] D. P. Bertsekas, R. G. Gallager, and P. Humblet. *Data networks*, volume 2. Prentice-Hall International New Jersey, 1992.

[6] T. Bonald, L. Massoulié, A. Proutiere, and J. Virtamo. A queueing analysis of max-min fairness, proportional fairness and balanced fairness. *Queueing systems*, 53(1):65–84, 2006.

[7] R. Braden, D. Clark, and S. Shenker. Rfc1633: Integrated services in the internet architecture: an overview. RFC, RFC Editor, 1994.

[8] B. Briscoe. Flow rate fairness: Dismantling a religion. *ACM SIGCOMM Computer Communication Review*, 37(2):63–74, 2007.

[9] A. D. Broido and A. Clauset. Scale-free networks are rare. *Nature communications*, 10(1):1–10, 2019.

[10] L. Brown, G. Ananthanarayanan, E. Katz-Bassett, A. Krishnamurthy, S. Ratnasamy, M. Schapira, and S. Shenker. On the future of congestion control for the public internet. In *Proceedings of the 19th ACM Workshop on Hot Topics in Networks*, pages 30–37, 2020.

[11] A. Dhamdhere, C. Dovrolis, and P. Francois. A value-based framework for internet peering agreements. In *2010 22nd International Teletraffic Congress (lTC 22)*, pages 1–8. IEEE, 2010.

[12] G. Giuliari, M. Wyss, M. Legner, and A. Perrig. GMA: A pareto optimal distributed resource-allocation algorithm. In *SIROCCO*, 2021. `https://arxiv.org/abs/2102.10314`.

[13] S. Group. Finance ipnet. `https://www.researchgate.net/profile/Thomas-Nellen/publication/228595529_2000_The_Swiss_Interbank_Clearing_System/links/5404438f0cf2bba34c1c5c40/2000-The-Swiss-Interbank-Clearing-System.pdf`, 2020.

[14] R. Guerin, H. Ahmadi, and M. Naghshineh. Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE Journal on selected areas in communications*, 9(7):968–981, 1991.

[15] G. Hardin. The tragedy of the commons. *Journal of Natural Resources Policy Research*, 1(3):243–253, 2009.

[16] R. K. Jain, D.-M. W. Chiu, W. R. Hawe, et al. A quantitative measure of fairness and discrimination. *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, 1984.

[17] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research society*, 49(3):237–252, 1998.

[18] D.-D. Luong and J. Bíró. On the proportional fairness of tcp vegas. In *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No. 01CH37270)*, volume 3, pages 1718–1722. IEEE, 2001.

[19] R. T. Ma, D. M. Chiu, J. C. Lui, V. Misra, and D. Rubenstein. Internet economics: The use of shapley value for isp settlement. *IEEE/ACM Transactions on networking*, 18(3):775–787, 2010.

[20] D. A. Meshram and D. D. Patil. 5g enabled tactile internet for tele-robotic surgery. *Procedia Computer Science*, 171:2618–2625, 2020.

[21] J. Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

[22] W. Ogryczak, H. Luss, M. Pióro, D. Nace, and A. Tomaszewski. Fair optimization and networks: A survey. *Journal of Applied Mathematics*, 2014, 2014.

[23] A. Rapoport, A. M. Chammah, and C. J. Orwant. *Prisoner's dilemma: A study in conflict and cooperation*, volume 165. University of Michigan press, 1965.

[24] S. Scherrer, M. Legner, A. Perrig, and S. Schmid. Enabling novel interconnection agreements with path-aware networking architectures. *arXiv preprint arXiv:2104.02346*, 2021.

[25] R. Ware, M. K. Mukerjee, S. Seshan, and J. Sherry. Beyond jain's fairness index: Setting the bar for the deployment of congestion control algorithms. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks*, pages 17–24, 2019.

[26] H. Zhang, D. Towsley, and W. Gong. Tcp connection game: A study on the selfish behavior of tcp users. In *13TH IEEE International Conference on Network Protocols (ICNP'05)*, pages 10–pp. IEEE, 2005.

A Appendix

A.1 Varying allocation matrices without changing convergents and divergents

Assume we have some graph and some simple path π. We then modify this graph by setting $M^{(a)}_{i,j} = M^{(a)}_{i,j} + \delta$ for some $\delta > 0$, where π uses that pair interface. We here assume the convergent and divergents don't change. (That is, the change also decreases at least two other pair allocations in the allocation matrix for node a). We will use $\mathcal{G}'(\pi)$ to denote the allocation in the modified graph.

We observe that all terms in $\mathcal{G}'$ will remain the same except for the $M^{(a)}_{i,j}$ term where we have

$$M^{(a)\prime}_{i,j} = M^{(a)}_{i,j} + \delta$$

for some $\delta \geq 0$.

We then claim that for all simple paths π which use the modified interface pair the following holds:

$$\mathcal{G}'(\pi) = \left(1 + \frac{\delta}{M^{(a)}_{i,j}}\right) \cdot \mathcal{G}(\pi)$$

Proof. We first observe that the x for which $\mathcal{G}(\pi)$ obtains its minimum depends only on the convergent and divergent of the visited nodes and not on the pair allocations along those. Thus, we can write:

$$\mathcal{G}(\pi) = \prod_{k=1}^{\ell} M^{(k)}_{i,j} \cdot \min_{x} \left(\prod_{k=1}^{x-1} \frac{1}{CON^{(k)}_{j}} \cdot \prod_{k=x+1}^{\ell} \frac{1}{DIV^{(k)}_{i}} \right)$$

Since $M^{(k)}_{i,j}$ for all nods along our path remains the same (except for a) and all convergents and divergents remain the same for all interfaces along our path we get

$$\mathcal{G}'(\pi) = \mathcal{G}(\pi) \cdot \frac{M^{(a)\prime}_{i,j}}{M^{(a)}_{i,j}} = \left(1 + \frac{\delta}{M^{(a)}_{i,j}}\right) \mathcal{G}(\pi)$$

A.2 Varying allocation matrices as in fairness proof – when does GMA have no improvement

Assume again we have some path π where we are increasing one of the pair allocations used by this path as follows:

$$\begin{aligned} M'^{(k)}_{i,j} &= M^{(k)}_{i,j} + \delta \\ CON'^{(k)}_{j} &= CON^{(k)}_{j} + \delta \\ DIV'^{(k)}_{i} &= DIV^{(k)}_{i} + \delta \end{aligned}$$

So exactly like we did for NBP-fairness.

Claim: The increasing in the allocation is 0 if and only if one of the following two conditions holds.

- $\exists y < k$ with $\mathcal{G}_y(\pi) = \mathcal{G}(\pi)$ and we have $DIV^{(k)}_{i} = M^{(k)}_{i,j}$
- $\exists y > k$ with $\mathcal{G}_y(\pi) = \mathcal{G}(\pi)$ and we have $CON^{(k)}_{j} = M^{(k)}_{i,j}$

That is, to say there is some other minimum and we already get a factor of 1 out of the node we are increasing.

Proof. We procedure similarly to how we did in Section 3.5.2. We perform a case distinction on k and x^*.

Case $k = x^*$: We have $\mathcal{G}'(\pi) \leq \mathcal{G}(\pi)$ if for the exact change we derived in Section 3.5.2 we have

$$\begin{aligned} 1 \geq \left(1 + \frac{\delta}{M^{(k)}_{i,j}}\right) \cdot \min \Bigg\{ & \min_{y<x} \left(\prod_{r=y+1}^{k-1} \frac{CON^{(r)}_{k}}{DIV^{(r)}_{i}} \right) \left(\frac{CON^{(y)}_{j}}{DIV^{(k)}_{i} + \delta} \right), \\ & \min_{y>x} \left(\prod_{r=k+1}^{y-1} \frac{DIV^{(r)}_{i}}{CON^{(r)}_{j}} \right) \left(\frac{DIV^{(y)}_{i}}{CON^{(k)}_{j} + \delta} \right), \\ & 1 \Bigg\} \end{aligned}$$

It is clear that if 1 is the minimum in the term then the increase will be greater than 0.

If we have some $y_1 < x$ such that $\left(\prod_{r=y_1+1}^{k-1} \frac{CON^{(r)}_{j}}{DIV^{(r)}_{i}}\right)\left(\frac{CON^{(y_1)}_{j}}{DIV^{(k)}_{i}+\delta}\right)$ is the minimum we know since x^* is the minimizer originally that:

$$\left(\prod_{r=y_1+1}^{k-1} \frac{CON^{(r)}_{j}}{DIV^{(r)}_{i}} \right) \left(\frac{CON^{(y_1)}_{j}}{DIV^{(k)}_{i}} \right) \geq 1$$

where equality only holds if $\mathcal{G}_{y_1} = \mathcal{G}_{x^*}$

$$\left(\prod_{r=y_1+1}^{k-1} \frac{CON_j^{(r)}}{DIV_i^{(r)}}\right)\left(\frac{CON_j^{(y_1)}}{DIV_i^{(k)}+\delta}\right) \geq \frac{DIV_i^{(k)}}{DIV_i^{(k)}+\delta}$$

Therefore:

$$\mathcal{G}'(\pi) \leq \mathcal{G}(\pi) \Rightarrow \frac{M_{i,j}^{(k)}}{M_{i,j}^{(k)}+\delta} \geq \frac{DIV_i^{(k)}}{DIV_i^{(k)}+\delta}$$

From which the desired property directly follows.

The third case follows completely analogously.

Case. $k \neq x^*$: Assume $x^* > k$: In this case we have

$$\mathcal{G}'(\pi) = \mathcal{G}(\pi) \cdot \left(1 + \frac{\delta}{M_{i,j}^{(k)}}\right) \cdot \min\left\{\min_{y<k}\left(\prod_{r=y+1}^{x^*-1} \frac{CON_j^{(r)}}{DIV_i^{(r)}}\right) \cdot \frac{CON_j^{(y)}}{DIV_i^{(x^*)}} \cdot \frac{DIV_i^{(k)}}{DIV_i^{(k)}+\delta}, \frac{CON_j^{(k)}}{CON_j^{(k)}+\delta}\right\}$$

Thus, for the increase to be 0 we need to be in either one of the following cases:

Case a: $\left(1 + \frac{\delta}{M_{i,j}^{(k)}}\right) \cdot \frac{CON_j^{(k)}}{CON_j^{(k)}+\delta} \leq 1$ which analogously to above requires that $M_{i,j}^{(k)} = CON_j^{(k)}$.

Case b:

$$\left(1 + \frac{\delta}{M_{i,j}^{(k)}}\right) \cdot \min_{y<k}\left(\prod_{r=y+1}^{x^*-1} \frac{CON_j^{(r)}}{DIV_i^{(r)}}\right) \cdot \frac{CON_j^{(y)}}{DIV_i^{(x^*)}} \cdot \frac{DIV_i^{(k)}}{DIV_i^{(k)}+\delta} \leq 1$$

Again, we observe that $\left(\prod_{r=y+1}^{x^*-1} \frac{CON_j^{(r)}}{DIV_i^{(r)}}\right) \cdot \frac{CON_j^{(y)}}{DIV_i^{(x^*)}} \geq 1$ where equality only holds if y is a minimizer from where the desired property directly follows.

The case for $x^* < k$ follows completely analogously.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Critical Networking: A Theoretical Analysis of GMA

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	First name(s):
Dodgson	Lucas Edward

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	Signature(s)
Horgen, 20.08.2021	L. Dodgson

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.