UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Ingegneria Civile, Edile e Ambientale

Master Degree in Mathematical Engineering

Final Dissertation

Developing new methods for routing and optimal transport on networks

Thesis supervisor

Prof. Mario Putti

Thesis co-supervisor

Dr. Caterina De Bacco

Candidate

Alessandro Lonardi

Academic Year 2019/2020

Alla mia famiglia e alla Valpolicella.

Contents

Introduction **1**

1 Optimal Transport Model **4**
1.1 Building blocks . . . 4
1.1.1 Commodities . . . 5
1.1.2 Pressure gradients . . . 6
1.1.3 Additional constraints . . . 7
1.1.4 Transport density . . . 7
1.2 Adaptation equations . . . 8
1.2.1 Kirchoff's law . . . 9
1.2.2 Ansatz . . . 10
1.2.3 Dynamics . . . 10
1.3 Numerical routines . . . 11
1.3.1 MCOT algorithm . . . 11
1.3.2 MSG algorithm . . . 12
1.4 Numerical analysis: MCOT vs. MSG . . . 14
1.4.1 MCOT: Transport density fraction . . . 15
1.4.2 MCOT: Pressure gradients . . . 16
1.4.3 MSG: traffic and efficiency . . . 19
1.4.4 Idle edges . . . 22
1.4.5 Total length . . . 23

2 Message Passing Model **26**
2.1 Theoretical introduction . . . 27
2.1.1 The EDP problem . . . 27
2.1.2 The Message Passing routine . . . 29
2.1.3 Weighted matching problem mapping . . . 29
2.2 Numerical routines . . . 31
2.2.1 SMP-EDP algorithm . . . 31
2.2.2 Reinforcement . . . 33
2.2.3 SMSG algorithm . . . 34
2.3 Numerical analysis: SMP-EDP vs. SMSG . . . 35
2.3.1 Performance . . . 35
2.3.2 Accommodation speed . . . 38
2.3.3 Traffic . . . 39
2.3.4 Cost . . . 40
2.3.5 Idle edges . . . 41

2.3.6 Total length 42

3 Comparison of the models **44**
3.1 Two different settings 44
3.2 Length and traffic 46
3.3 Practical applications 47

Conclusions **48**

A Additional simulations: MCOT **54**

B Additional simulations: SMP-EDP **55**

Notation

Most of the notations used throughout this work are standard and defined on the spot. However, for the sake of completeness, in Table (1) we provide a summary of some of the most recurrent symbols together with their meaning.

symbol	*meaning*
$\mathbb{R}$ ($\mathbb{R}_+$)	(non-negative) real numbers
$\mathbb{R}^d$	space of d-dimensional real vectors
$\boldsymbol{A} \equiv (A_i^j)_i^j \in \mathbb{R}^{n,m}$	$n \times m$ matrix with entries in $\mathbb{R}$
$\boldsymbol{A}_i \in \mathbb{R}^m$	i-th row of matrix $\boldsymbol{A}$
$\boldsymbol{A}^j \in \mathbb{R}^n$	j-th column of matrix $\boldsymbol{A}$
$\|\|\cdot\|\|_2$	ℓ^2 norm (e.g. $\boldsymbol{x} \in \mathbb{R}^n$ column vector, $\|\|\boldsymbol{x}\|\|_2 := \sqrt{\boldsymbol{x}^\intercal \boldsymbol{x}}$)
δ_{ij}	Kronecker delta
$U(a,b)$	uniform distribution in the interval (a,b) with $a < b \in \mathbb{R}$
$U\{\mathcal{S}\}$	discrete uniform distribution in the set $\mathcal{S}$
diag$[\boldsymbol{x}]$	$n \times n$ diagonal matrix with $\boldsymbol{x} \in \mathbb{R}^n$ as diagonal
$\mathcal{G}$	undirected graph
$\mathcal{V}$ ($\|\mathcal{V}\| = N$)	set of nodes/vertices of $\mathcal{G}$ (its cardinality)
$\mathcal{E}$ ($\|\mathcal{E}\| = E$)	set of links/edges of $\mathcal{G}$ (its cardinality)
$\mathcal{E}_v$	set of edges of connected to a node $v \in \mathcal{V}$
$\mathcal{C}$ ($\|\mathcal{C}\| = M$)	set of commodities (its cardinality)
	Optimal Transport Model
$\boldsymbol{B} \in \mathbb{R}^{N,E}$	oriented incidence matrix of $\mathcal{G}$
$\boldsymbol{G} \in \mathbb{R}^{N,M}$	In-flowing mass matrix
$\boldsymbol{F} \in \mathbb{R}^{N,M}$	Out-flowing mass matrix
$\boldsymbol{\ell} \in \mathbb{R}_+^E$	vector of edge lengths
$\hat{\boldsymbol{\mu}} \in \mathbb{R}_+^E$	conductivity/transport density vector
$\boldsymbol{P} \in \mathbb{R}^{N,M}$	pressure/potential matrix
$\Delta\cdot := \boldsymbol{L}^{-1}\boldsymbol{B}^\intercal\cdot$	"discrete gradient" operator, here $\boldsymbol{L} := \text{diag}[\boldsymbol{\ell}]$
	Message Passing Model
$\boldsymbol{I}_{vw} \in \mathbb{R}^M$	current vector passing through $(v,w) \in \mathcal{E}$
$\Lambda_v^i = \{0,+1,-1\}$	{transit node/source/sink} for commodity $i \in \mathcal{C}$
E_{vw}	message associated to $(v,w) \in \mathcal{E}$
	Acronyms
EDP	Edge-Disjoint Path
MP	Message Passing
MCOT	Multi-Commodity Optima Transport algorithm
MSG	Multi-Start Greedy algorithm
SMP-EDP	Sequential Message Passing Edge-Disjoint Path algorithm
SMSG	Sequential Multi-Start Greedy algorithm

Table 1: Summary of notation.

Introduction

Optimal path selection and network routing play a crucial role in the constantly expanding domain of network science. Indeed, these concepts stand at the core of several practical applications and innovations from which we benefit on a daily basis. These range from the Internet itself, in which data transmission among users and servers is regulated via routing protocols [35], to telecommunication networks, where the decision making process controlling package transfer is pivotal to guarantee Quality of Service [13]. Other fields where efficient traffic management routines are extensively studied are urban transportation (see [21] for an overview), and biological networks [38, 47–49]. The former deals with compelling tasks both for human activities and civil logistics, where the importance of high performing road networks could contribute, for example, to partially resolving environment-related issues. The second, although apparently not closely related to our everyday routine, has indeed a significant connection with research on optimization of railway and highway networks. In fact, the study of biological networks applies to many natural systems where complex transportation structures spontaneously emerge seemingly as consequence of optimality principles. Such are metabolic networks, as for example blood circulation systems in animals, and areal branches and underground roots in plants, which try to explore as much as possible the surrounding space for delivering or absorbing vital nutrients [7, 8]. However, recent work concentrated on the study of "Pysarum Machines" as model of biological optimizers that nowadays serve as paradigm for network modeling [2–4, 37, 43, 49]. Moreover, biologically-inspired models constitute the foundation of what we call the Multi-Commodity Optimal Transport (MCOT) algorithm, one of the two building blocks of this thesis.

While the potential range of applications for routing and optimal transport on networks is wide, a proper theoretical *and* computational efficient modeling framework needed to address this problem is still lacking. In particular, while several progresses have been recently made theoretically in optimal transport theory, they remain mainly abstract efforts, not applicable in practice. On the other side, the main computationally efficient algorithmic implementations currently being used by practitioners are based on simple heuristics like greedy optimizations, which do not make use of relevant theoretical insights provided by optimal transport theory or statistical physics.

We consider Multiple Sources-Multiple Sinks (MS-MS) routing problems, where we have a set of sources (or senders) and a set of sinks (or receivers) communicating in an underlying network structure. The former are nodes in the network who send some quantity (e.g. information), the latter are those receiving it. Among MS-MS, we distinguish two different settings, which we will consider in the remain of the thesis:

1. *One Source-Multiple Sinks.* In this case information is sent from a certain source but is not directed towards another specific sink. Instead, each source's information

can be collected by any among the sink nodes. In this context each source is a commodity.

2. *One Source-One Sink*. In this case there is a one-to-one pairing between a given pair of source and sink. In other words, a source needs to transfer information only to one particular sink. We refer to these source-sink (or sender-receiver) pairings as communications or commodities.

These two classes of problems are commonly studied in computer networking. In this setting OS-MS and OS-OS models are respectively addressed as multicast and unicast routing schemes. In the first case, a simultaneous distribution of information towards a group of destinations is performed; in the second case a package sent from a machine is directed to only one other device.

In this thesis we develop theoretical models and the corresponding algorithmic implementation for these two problem settings. Specifically, we propose Multi-Commodity Optimal Transport (MCOT) for the One Source-Multiple Sinks setting and Sequential Message Passing with Edge Disjoint Path constraint (SMP-EDP) for the One Source-One Sink setting.

The MCOT model is a principled generalization of the slime mold Physarum dynamics recently developed by Facca et al. [15] and originally proposed by Tero et al. [48], which has been further studied in subsequent works [47, 49]. In this thesis we extend the so-called *adaptation equations* characterizing the dynamics of the problem to the MS-MS framework. The goal of this model is twofold: i) adapt theoretical principles of optimal transport theory to a concrete routing application; ii) highlight the connection between a biological system like the slime mold dynamics and the abstract optimal transport formalism studied in [44, 53–55]. Moreover, this has an intuitive physical interpretation and outperforms a heuristic multi-start greedy (MSG) algorithm. These two models are extensively compared in term of metrics as network traffic and total path length.

The starting point to build the SMP-EDP algorithm is the formal definition of edge disjoint path problem. This is a constrained optimization problem where the goal is finding, given a network topology and a set source-sink pairings, the maximum number of paths joining each source with its sink that can be accommodated at the same time given no edge is allowed to have more than one communication passing through it. This is called edge-disjoint constraint, and it results in having paths associated to different communications not overlapping on the edges. This problem is largely applied in transmission services requiring high quality data transmission and full bandwidth exploitation. Some examples are: Routing and Wavelength Assignment (RWA) problems in optical networks [26, 45, 57], where each light-path is reserved for a single wavelength. Or accommodation problems in telecommunications, from local wireless networks to global coverage architectures [46, 59]. During the last decades, several studies have been devoted to solve this problem. Some of the most common approaches are greedy algorithms [12, 23, 25, 50], linear programming [9, 23, 27], evolutionary algorithms [20, 40] and ant-colony routines [10, 41].

In this thesis we consider the approach of message-passing (MP) algorithms, a method that uses principles of statistical physics to effectively incorporate global information and process long-range interactions between network users as they compete for using the same infrastructure. Specifically, we extend the work of Altarelli et al. [5] by relaxing the hard constraint of edge-disjointedness to allow for more than one active network route to pass

through an edge. Loosening this constraint by means of a sequential method, we are able to achieve the accommodation of all the communications of the networks, maintaining all the advantages of the MP formulation. Moreover, our algorithm largely outperforms a sequential multi-start greedy (SMSG) method, used as benchmark in absence of any theoretically supported ground truth. The two methods are compared in terms of several metrics, from total path length, to traffic management and number of accommodated source-sink pairs.

Outline of the thesis.

- Chapter 1. In the first chapter we present the MCOT model. Starting from its theoretical derivation, we present the algorithm and its properties. An extensive comparison with a multi-start greedy algorithm is performed.
- Chapter 2. Here, first we discuss the key points regarding the theoretical derivation of the SMP-EDP model. Then we show the results of exhaustive numerical experiments in which the algorithm is tested against a multi-start greedy benchmark routine.
- Chapter 3. This chapter is dedicated to a comparison of the two models previously presented. The goal is to highlight their differences and similarities, in order to be able to improve both methods in the future.
- Conclusions. In this final chapter we close the thesis by providing a summary of the key points of our analysis and of possible future works.

Chapter 1

Optimal Transport Model

Optimal transport (OT) models, other than a purely theoretical interest, displayed a great applicability to a large variety of contexts. Starting from the investigation of the path-finding ability of slime mold in mazes [33, 38], these models have been employed to replicate railroad networks [49, 52] and road networks [1, 43, 51]. Other applications are modeling path formation in wireless sensor architectures [24], and designing supply chain structures [60]. Moreover, optimal transport theory has also been exploited in medicine, for example to replicate the vascular structures of human placentas [56].

In recent years OT has been investigated with several techniques, from Physarum (slime mold) solvers used by most of the works just cited, to more complex biologically-inspired models as in [17], where Physarum Machines are used to solve MS-MS transshipment (minimum cost flow) problems. A large contribution in the study of OT is given by linear optimization methods, indeed the dual formulation of the original Monge OT problem (1781) [34] proposed by Kantorovic [22] is inspired by a discrete linear program [14]; for a brief presentation on the state-of-the-art we can refer to [42]. In most of the current literature routing problems are tackled considering constrained settings. Precisely, to overcome their complexity different simplifications are made to the problem formulation, e.g. imposing heuristic traffic constraints to manage congestion, or limiting the capacity of the amount of mass the network can transport.

In this first chapter of the thesis is devoted to introducing the MCOT model, first presenting its mathematical formulation and then describing the details of its numerical implementation. Before moving to the model construction it is necessary to highlight the purpose of MCOT. What we want to do is, given a graph $\mathcal{G}$ and a set of sources/sinks (nodes sending/receiving mass), to build an algorithm capable to optimally distribute the mass flowing in the network. The meaning of "optimality" in this context is anything but simple, in particular we want a method capable of minimizing the network total length while managing traffic and rerouting at a moderate computational cost. One of the purposes of the numerical experiments performed in this thesis is that of allowing a quantitative characterization of these properties, showing how the MCOT routine outperforms the state-of-the-art.

1.1 Building blocks

The first step to construct the MCOT model is to introduce its main building blocks. All the elements we present are combined together in what we refer as the adaptation

equations (cfr. [48]), which entirely characterizes the dynamics of the problem. Many of the objects in our formulation have the same physical interpretation of those of the Physarum Solvers. Despite that, many crucial differences between the two methods arise. Indeed, in this chapter we explain all these similarities and distinctions in detail.

1.1.1 Commodities

The topology of the network on which the accommodation of mass is performed is given by an unweighted and undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. Here $\mathcal{V}$ is the set of vertices, and $\mathcal{E}$ the set of edges. Nodes and links have respectively cardinality $|\mathcal{V}| = N$ and $|\mathcal{E}| = E$. Some of the nodes $\mathcal{G}$ form the set of commodities, these are vertices that play the role of either sources or/and sinks of mass, we label them with $\mathcal{C} \subseteq \mathcal{V}$.

In the context of railway networks we can think of $\mathcal{C}$ as stations, where people can both go on or off a train. Instead the set $\mathcal{V} \setminus \mathcal{C}$ represents the set of junctions, in these passengers can only move from one train to another, but neither exit nor enter from and in the network. Another practical and intuitive visualization may be the one of river networks, here commodities are sources of water and wells while the remaining vertices represent confluence points of rivers where different channels merge together.

Formally we can define the set of commodities as

$$\mathcal{C} := \{v \in \mathcal{V} \,:\, g^v \text{ or } f^v \neq 0\},$$

where the two terms g^v and f^v measure respectively the amount of in-flowing and out-flowing mass of a node v. Collecting the couples $(g^v, f^v)\,\forall v \in \mathcal{V}$ we can build matrices $\boldsymbol{G} \in \mathbb{R}^{N,M}$ and $\boldsymbol{F} \in \mathbb{R}^{N,M}$. These can be constructed following the element-wise definition $\forall v \in \mathcal{V}$, $\forall i \in \mathcal{C}$

$$G_v^i := \delta_{vi}\, g^i \tag{1.1a}$$

$$F_v^i := \mathbb{I}_{\{v\in\mathcal{C}\}}(1 - \delta_{vi})\widetilde{F}_v^i \tag{1.1b}$$

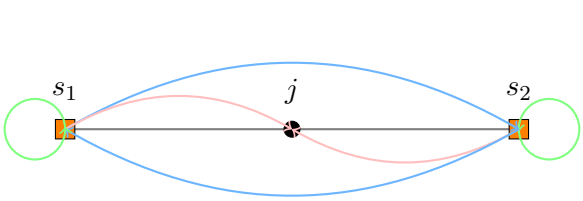
where

$$\sum_{v\in\mathcal{V}} \widetilde{F}_v^i = f^i \quad \forall i \in \mathcal{C} \tag{1.2}$$

and $\mathbb{I}_{\{v\in\mathcal{C}\}}$ denotes the indicator function on set $\mathcal{C}$.

It is immediate to understand that in matrix $\boldsymbol{G}$ we are simply collecting the in-flowing mass contributions given by all the g^v terms. In particular, every row $\boldsymbol{G}_v$ is associated to a node $v \in \mathcal{V}$ and i) it is entirely equal to 0 if $v \notin \mathcal{C}$, i.e. vertex v is a "junction"; or ii) the in-flowing mass contribution to the vertex is given solely by $G_i^i = g^i$ if $v = i \in \mathcal{C}$. The definition of matrix $\boldsymbol{F}$ is less straightforward, it should should be intended as follows: every N-dimensional column $\boldsymbol{F}^i$ represents the out-flowing mass produced by a node $i \in \mathcal{C}$. This mass is distributed among all the nodes $v \in \mathcal{C} \setminus \{i\}$ which contributes $\widetilde{F}_v^i$ each, meanwhile the entries of the vertices in $\mathcal{V} \setminus (\mathcal{C} \cup \{i\})$ are equal to 0. This seemingly articulated definition is crucial to guarantee the well-posedness of the problem, it is in fact important to build the model in such a way that mass entering a source i cannot outflows from a "junction" or from i itself. This latter constraint has to be imposed so that we avoid trivial dynamics where no mass is actually moving trough the network.

To further clarify the rationale behind definition (1.1b), that still may be not evident, it is useful to resort again to the railroad network framework. Suppose we have a simple railway made with two stations $\{s_1, s_2\}$ and a junction $\{j\}$. We connect the nodes as in figure 1.1, in such a way that every passenger in order to move from s_1 to s_2 has to travel through j. In this setting the only non-zero terms of the 3×2 matrix $\boldsymbol{F}$ are: $F^{s_1}_{s_2}$, measuring the amount of passengers entering station s_1 and exiting station s_2, and $F^{s_2}_{s_1}$, measuring the number of people traveling from s_2 to s_1. The entries $F^{s_1}_{j}$ and $F^{s_2}_{j}$ are equal to 0 because no people can exit the network from the junction. Instead, $F^{s_2}_{s_2}$ and $F^{s_1}_{s_1}$ are equal to 0 because to be that every person (unit of mass) entering the network is actually moving along a travel route joining two different stations (commodities).



Mass contributions:

- $F^{s_1}_{s_1} = F^{s_2}_{s_2} = 0$
- $F^{s_1}_{j} = F^{s_2}_{j} = 0$
- $F^{s_2}_{s_1}, F^{s_1}_{s_2} \neq 0$

Figure 1.1: Railway network toy model. The orange squared nodes are the sources/sinks, the black circle is the junction. The physical edges of the netrork are drawn in gray, the colored curved lines serve to give an idea of the which routes are associated with the entries of $\boldsymbol{F}$.

Lastly, equality (1.2) expresses conservation of the out-flowing mass generated by every commodity $i \in \mathcal{C}$. In particular, each f^i term has to be exactly distributed between all the other nodes of the graph and no mass has to "get stuck" in the network.

1.1.2 Pressure gradients

The force exerted by the mass entering in source $i \in \mathcal{C}$ on a vertex $v \in \mathcal{V} \setminus \{i\}$ is encoded by a pressure variables p^i_v, collecting together all these $N \times M$ terms we can build the matrix $\boldsymbol{P} \in \mathbb{R}^{N,M}$. Every N-dimensional column $\boldsymbol{P}^i$ of this matrix contains the contributions to the total pressure p^i that $i \in \mathcal{C}$ exercises on the entire network, formally this reads

$$p^i = \sum_{v \in \mathcal{V}} P^i_v \quad \forall i \in \mathcal{C}.$$

In constructing the adaptation equations it is in our interests to measure differences of pressure between neighboring vertices, so that we are able to replicate the typical mechanism of Physarum Solvers where the width of a link $e \in \mathcal{E}$ is regulated by the force gradient between neighboring nodes linked by e. To be precise, we start by the defining for all $i \in \mathcal{C}$

$$\delta p^i_e := p^i_v - p^i_w, \tag{1.3}$$

where with $e = (v, w) \in \mathcal{E}$ we label the edge joining nodes w and v. All pressure difference terms are be collected in a $E \times M$ matrix, that can be easily written in compact form with the matrix multiplication $\boldsymbol{B}^\intercal \boldsymbol{P}$. Here $\boldsymbol{B} \in \mathbb{R}^{N,E}$ is the signed incidence matrix of

the undirected graph $\mathcal{G}$, this is constructed by assigning a fictitious orientation to the edges of the network and by setting $B_v^e = \pm 1$ if an edge e has node v as start/end point, 0 otherwise.

To properly measure the pressure "gradient" associated to every link $e \in \mathcal{E}$ we divide every contribution introduced in (1.3) by its edge length. Precisely we define

$$(\Delta \boldsymbol{P})_e^i := \frac{\delta p_e^i}{\ell_e} \quad \forall e \in \mathcal{E}, \forall i \in \mathcal{C}, \tag{1.4}$$

where ℓ_e is the length of edge $e \in \mathcal{E}$. In order to rewrite the previous definition in a more elegant fashion we can collect all the ℓ_e terms in a vector $\boldsymbol{\ell} \in \mathbb{R}_+^E$, containing the lengths of the links in the graph. Using this object we construct the diagonal matrix $\boldsymbol{L} := \mathrm{diag}[\boldsymbol{\ell}] \in \mathbb{R}^{E,E}$ that allows to write all the contributions of equation (1.4) in a $E \times M$-dimensional matrix reading as

$$\Delta \boldsymbol{P} := \boldsymbol{L}^{-1} \boldsymbol{B}^\intercal \boldsymbol{P}.$$

This last quantity contains the pressure gradients acting on every edge, and generated by every in-flowing and out-flowing mass term associated to each commodity. We may refer to

$$\Delta \cdot := \boldsymbol{L}^{-1} \boldsymbol{B}^\intercal \cdot$$

as discrete gradient operator because of its straightforward similarity with its more common continuous counterpart.

1.1.3 Additional constraints

In addition to constraint (1.2), matrices $\boldsymbol{G}, \boldsymbol{F} \in \mathbb{R}^{N,M}$ need to satisfy the following requirements to guarantee the well-posedness of the problem. First, we must have that

$$g^i \leqslant \sum_{\substack{j \in \mathcal{C} \\ j \neq i}} f^j \quad \forall i \in \mathcal{C}, \tag{1.5}$$

that is: for every commodity $i \in \mathcal{C}$ the quantity of in-flowing mass cannot be greater than its total out-flowing contribution, obtained summing on all the vertices $\mathcal{C} \setminus \{i\}$. Moreover, to maintain mass balance we need the following chain of equalities to hold for every commodity $i \in \mathcal{C}$

$$\sum_{v \in \mathcal{V}} G_v^i - F_v^i = g^i - \sum_{v \in \mathcal{V}} F_v^i = 0. \tag{1.6}$$

This last expression says that the no mass can remain in the network. Precisely, every unit of mass entering a source i must flow out from a sink different from i itself, in such a way that the difference in (1.6) is identical to 0 for every $i \in \mathcal{C}$.

1.1.4 Transport density

In order to present all the necessary quantities needed to construct the adaptation equations we need to introduce one last object, the transport density (or conductivity).

Correspondingly to what is done for Physarum Solvers we associate to every edge $e \in \mathcal{E}$ a variable $\hat{\mu}_e > 0$, that intuitively can be thought as the area of the (tubular) section of a link. Its physical interpretation is provided by studies on slime molds [18, 36, 39, 48], where the idea of "current reinforcement" is extensively discussed. This consists in the typical behavior of molds based on feedback mechanisms regulating the thickness of each link, in which high rates of protoplasmic materials stream stimulate an increase in tube diameters. This positive feedback mechanism between flux and edge thickness is indeed encoded by adaptation equations, in which the evolution in time of each $\hat{\mu}_e$ follows the principle: the greater the flux, the thicker the channel.

It is important to notice that since we modeled pressure gradients separating the contributions provided by each commodity in $\mathcal{C}$, in principle we could define M communication-dependent transport densities μ_e^i. However, this would have the effect of losing all information on traffic occupancy, resulting in a selfish accommodation of the commodities. This fundamental step of making transport densities communication independent is indeed is carried out by the adaptation equations, and will be largely discussed along the thesis.

For convenience of notation, analogously to what has been done with the other quantities on the model we collect all the transport densities into a vector $\hat{\boldsymbol{\mu}} \in \mathbb{R}_+^E$ and we define a diagonal matrix $\boldsymbol{M} := \mathrm{diag}[\hat{\boldsymbol{\mu}}] \in \mathbb{R}_+^{E,E}$.

1.2 Adaptation equations

Before moving to the analysis and formulation of the equations, we point out again that in the majority of the current literature routing problems are rephrased considering simplified settings. Here, to avoid intractable problems heuristic constraints are often imposed to deal with traffic, or limitations on the capacity of the edges are built ad-hoc for each case of study. Furthermore, most of the state-of-the-art approaches studying routing on networks are incapable of returning a solution while taking in account the occupancy state of all the edges of the graph at the same time, limiting the dynamics to reach sub-optimal results. Our formulation of the multi-commodity flow problem consists in an improved original approach to the ones already present in the current literature. In fact the method we are presenting is naturally able to take in account global information on the traffic of each edge while performing accommodation.

Using all the elements introduced in §1.1 we can write the adaptation equations that we propose in this thesis. These are presented with a similar fashion to the one of [15], where the authors study, in a continuous framework, the connections between the adaptive Physarum dynamics and Monge Kantorovich Optimal Transport algorithms for the solution of Basis Pursuit problem for the case of one commodity. Here, however, we consider the existence of more than one commodity and adapt the equations accordingly to take into account how this impacts the problem's setting and solution. In particular, it is crucial to distinguish the pressure p^i exercised by a commodity $i \in \mathcal{C}$. This implies that at each edge we should distinguish a transport density per commodity μ_e^i, which is due to δp_e^i. Having said that, the straightforward generalization of the model proposed in [15] would be to consider one adaptation equation per commodity. However, this would make the $|\mathcal{C}|$ problems independent, and this generalization trivial. Instead, we need a quantity to couple the different commodities together, as in a realistic scenario where commodities compete for the network infrastructure. This quantity should be a function

of the various transport densities $f(\{\mu_e^i\}^i)$, as this is the main variable controlling how the mass of different commodities travels through edges. There could be several choices for this function, here we consider the simple scenario

$$f(\{\mu_e^i\}^i) = \mu_e^i =: \hat{\mu}_e, \quad \forall i \in \mathcal{C}. \tag{1.7}$$

This implies that all the transport densities μ_e^i on edges should be equal, and equal to an overall transport density $\hat{\mu}_e$. In this way, we can now consider only this quantity and propose an adaptation equation only for it, which implicitly couples all the individual commodities together allowing for a straightforward mathematical non-trivial generalization of the one-commodity case. While this assumption might be valid in certain cases but too strict in others, depending on the application considered, we leave for future work more complex functions $f(\{\mu_e^i\}^i)$ which might lead to intractable adaptation equations. Taking this into account, we propose the following adaptation equations

$$\boldsymbol{B}\boldsymbol{M}\boldsymbol{L}^{-1}\boldsymbol{B}^\intercal\ \boldsymbol{P} = \boldsymbol{G} - \boldsymbol{F} \tag{1.8a}$$

$$\forall e \in \mathcal{E}, \forall i \in \mathcal{C} : \mu_e^i =: \hat{\mu}_e \tag{1.8b}$$

$$\forall e \in \mathcal{E} : \dot{\hat{\mu}}_e = \hat{\mu}_e^\beta ||(\boldsymbol{\Delta P})_e||_2^2 - \hat{\mu}_e. \tag{1.8c}$$

In the system (1.8a–1.8c), matrices $\boldsymbol{G}, \boldsymbol{F} \in \mathbb{R}^{N,M}$ and $\boldsymbol{B} \in \mathbb{R}^{N,E}$ are data input of the problem. These quantities may vary dependently on the context we are working on, and are directly related to the nature of the problem itself and to the underlying network topology. Instead, the pressure $\boldsymbol{P} \in \mathbb{R}^{N,M}$ and the transport density vector $\hat{\boldsymbol{\mu}} \in \mathbb{R}_+^E$ are dependent variables, which values are updated by solving the adaptation equations.

Moreover, in (1.8c) the additional scalar parameter $\beta > 0$ has been introduced, which relevance is largely discussed and justified in several works on OT [44, 54, 55]. This, intuitively, serves as tuning for the importance given to traffic congestion. That is, regulating its value we can control tendency of mass to travel on common routes or to spread among different paths. To be more precise, for $\beta > 1$ the solution returned by the adaptation equations is a graph with a smaller number of thick edges (high values of $\hat{\mu}_e$), meaning flow is consolidated through fewer routes. Instead, if $\beta < 1$ the result is a network with many thin (low values of $\hat{\mu}_e$) links, meaning the paths distribute more widely through the network. In the former case a lower "importance" is given to traffic congestion, reason for which a solution with thicker edges is preferred to one where mass is rerouted on a high number of links, meanwhile in the second situation we strongly penalize traffic favoring rerouting. Formally, this tendency of the model of preferring one behavior with respect to the other is given by subadditivity (resp. superaddivity) of the function $\hat{\mu}_e \mapsto \hat{\mu}_e^\beta$ for $\beta < 1$ (resp. $\beta > 1$). The limit case of $\beta = 1$ corresponds to the case in which traffic neither favored nor disadvantaged, so all the mass gets distributed along the shortest paths joining sources and sinks.

We now proceed in a brief discussion of each one of the equations building the system (1.8a–1.8c).

1.2.1 Kirchoff's law

Equation (1.8a) is Kirchoff's law for the network, expressing conservation of mass. Indeed referring to the description of the quantities done in §1.1 we notice that the r.h.s of this equation is exactly the difference of in-flowing and out-flowing mass entering and exiting

the graph. Instead, the left hand side describes the mass flux moving through the edges. We can exhibit clearly the role played by each term separating the contribution of every commodity $i \in \mathcal{C}$. In particular taking

$$\boldsymbol{B}\boldsymbol{M}\boldsymbol{L}^{-1}\boldsymbol{B}^{\intercal}\ \boldsymbol{P}^i = \boldsymbol{G}^i - \boldsymbol{F}^i \quad \forall i \in \mathcal{C}$$

we equate two N-dimensional vectors; the entries of the one on the r.h.s are of the form

$$G_v^i - F_v^i = \begin{cases} g^i & v = i \\ -\widetilde{F}_v^i & v \in \mathcal{C} \setminus \{i\} \\ 0 & v \in \mathcal{V} \setminus \mathcal{C} \end{cases}$$

and express either the mass entering i, or exiting from a vertex $v \in \mathcal{C} \setminus \{i\}$ because of commodity i. The terms on the l.h.s. are of the form

$$\sum_{e \in \mathcal{E}_v} \frac{\hat{\mu}_e}{\ell_e} \delta p_e^i \quad \forall v \in \mathcal{V},$$

where with $\mathcal{E}_v$ we label the set of links attached to node $v \in \mathcal{V}$. Each one is proportional to the ratio "area/length" of an edge e, measuring the ease with which a mass can move through the link, multiplied by the pressure difference, encoding the force acting on e.

1.2.2 Ansatz

As said, the ansatz (1.8b) plays a crucial role in merging the traffic information given by every commodity in $\mathcal{C}$, so that we can build a model that is capable of adapting the values of the conductivities taking in account *all* the contribution given by the sources in the network *at once*. Formally we construct the model in such a way that $\hat{\mu}_e$, $\forall e \in \mathcal{E}$ is independent with respect to i, i.e. $\hat{\mu}_e = f(\{\mu_e^i\}^i)$. This passage is critical in the formulation of the problem because by jointly measuring the amount of mass provided by each node in $\mathcal{C}$ we obtain a system of equations that naturally regulates the ease with which mass travels though links based on the force exerted by every commodity.

Again, equation (1.8b) is a particular case of the more general (1.7); in this thesis we consider only the simplest case with $f \equiv \mathrm{Id}$, that is $\mu_e^i = \hat{\mu}_e$ $\forall e \in \mathcal{E}$, $\forall i \in \mathcal{C}$.

1.2.3 Dynamics

In equations (1.8c) we write the actual dynamics of the transport densities. Here we see that the E time derivatives on the l.h.s. are balance by two contributions. One is simply equal to $-\hat{\mu}_e$, and causes an exponential decay of the conductivities when no force is acting on the link e. This first factor gets balanced by the positive terms $\hat{\mu}_e^{\beta} ||(\boldsymbol{\Delta P})_e||_2^2$ with which we measure the force acting on the edge e produced by all the commodities together.

These terms replicate exactly the typical behavior of slime molds, where a feedback mechanism regulates the thickness of each link; enforcing transportation in edges through which a large amount of mass moves, and shrinking tubes with low rates of protoplasmic materials stream.

1.3 Numerical routines

1.3.1 MCOT algorithm

The numerical implementation of the system of equations (1.8a−1.8c) has done been via a numerical scheme summarized in the following pseudocode. The algorithm is discussed extensively in this section.

Algorithm 1 MCOT algorithm

input: graph topology $\mathcal{G}(\mathcal{V}, \mathcal{E})$
input: set of communications $\mathcal{C}$
input: in-flowing and out-flowing mass $g_i, f_i \, \forall i \in \mathcal{C}$
input: $\beta > 0$
construct $\boldsymbol{G}, \boldsymbol{F} \in \mathbb{R}^{N,M}$
initialize $\hat{\boldsymbol{\mu}}(0) \in \mathbb{R}^{E}$
while `convergence == False` **do**
    solve eq. (1.8a) for the pressure matrix $\overset{\text{return}}{\rightarrow} \boldsymbol{P} \in \mathbb{R}^{N,M}$
    set ansatz given by eq. (1.8b)
    solve eq. (1.8c) with a finite difference scheme: $\hat{\boldsymbol{\mu}}(t) \overset{\text{update}}{\rightarrow} \hat{\boldsymbol{\mu}}(t+1)$
return $\{\hat{\boldsymbol{\mu}}^*, \boldsymbol{P}^*\}$, conductivity and pressure at convergence

The scheme above provides a general outline of the method designed to solve the problem; however, in the experiments presented in thesis we made some specific choices regarding the implementation of the code. It is essential to highlight that the core of the scheme remains always the one of algorithm (1) independently on the setting, minor changes has been made, and can be done, with the purpose of improving performance in different frameworks, e.g. when working with synthetic data rather than with real networks. In particular, in our implementation we modified four main aspects that are worth describing: i) the artificial construction of $\boldsymbol{G}, \boldsymbol{F} \in \mathbb{R}^{N,M}$ in absence of a real mass assignation, ii) the initialization of the transport density vector $\hat{\boldsymbol{\mu}} \in \mathbb{R}^{E}_{+}$, iii) which convergence criteria is set by the boolean variable `convergence` and, lastly, iv) the choice of the finite difference scheme used to solve equation (1.8c).

In detail, there are some cases in which $\boldsymbol{G}$ and $\boldsymbol{F}$ are not given as data of the problem. For example, when working with synthetic networks it may be necessary to both artificially sample $\mathcal{C}$ and construct these matrices. Different techniques are equivalently valid, the only requirement that they have to fulfill are indeed set by expressions (1.1a) and (1.1b), coupled with constraints (1.2), (1.5) and (1.6) which ensures mass conservation. Our technique, to which we refer as Auxiliary Receiver method, is summarized in pseudocode (2). This briefly consists in randomly set an in-flowing g^i contribution to each commodity in $\mathcal{C}$, and correspondingly assign the same amount of mass to a node $j_i \in \mathcal{C} \setminus \{i\}$, so that mass balance is satisfied.

The transport densities are sampled as $\hat{\mu}_e \sim U(0, 1)$. This decision has been made by means of trial and error: first we initialized the conductivities as $\hat{\mu}_e = \text{const.} \, \forall e \in \mathcal{E}$ choosing different values for const., however, sampling transport densities uniformly proved to considerably reducing the number of iterations needed to reach convergence.

Several criteria could be in principle chosen to decide whether the scheme has converged or not. For example, a common and elementary technique consists in directly

Algorithm 2 Auxiliary Receivers method

initialize the variable $M(= 50$ conventionally in our experiments)
for each $i \in \mathcal{C}$ **do**
    sample $g^i \sim U\{0, \mathrm{M}\}$
    sample index as $j_i \sim U\{\mathcal{C} \setminus \{i\}\}$
    set $F^i_{j_i} = -g^i$

exploiting mass balance, formally we could define a residual from equation (1.8a) as

$$\texttt{res} := \sum_{i \in \mathcal{C}} \left|\left| \boldsymbol{B}\boldsymbol{M}\boldsymbol{L}^{-1}\boldsymbol{B}^{\intercal}\ \boldsymbol{P}^i - (\boldsymbol{G}^i - \boldsymbol{F}^i) \right|\right|_2$$

and eventually set the convergence criterion

$$\texttt{if } (\texttt{res} < \texttt{tol}) \texttt{ then } \{\texttt{convergence == True}\}$$

where `tol` is a scalar parameter set a priori.

An equivalently valid choice is the one made in [15], that we adopt in our code. Precisely, what we do is using the discretized time derivative of the transport densities by setting the following condition

$$\texttt{if } \left(\frac{||\hat{\boldsymbol{\mu}}(t + \Delta t) - \hat{\boldsymbol{\mu}}(t)||_2}{\Delta t ||\hat{\boldsymbol{\mu}}(t)||_2} < \texttt{tol} \right) \texttt{ then } \{\texttt{convergence == True}\}.$$

The argument in parenthesis is the time derivative of the conductivity vector divided by its norm, expressing the relative variation of $\hat{\boldsymbol{\mu}}$ between two consecutive time steps. What we do is simply imposing convergence when this relative quantity is small enough i.e. smaller than `tol`, so that the routine stops when the transport densities are almost stationary. In our simulations the tolerance is conventionally set to $\texttt{tol} = 10^{-3}$.

Lastly, to solve (1.8c) we use a Forward Euler method with fixed time step, approximating for every $e \in \mathcal{E}$ the conductivities $\hat{\mu}_e(t + \Delta t)$ as

$$\hat{\mu}_e(t + \Delta t) = \hat{\mu}_e(t) + f(\hat{\boldsymbol{\mu}}(t), \boldsymbol{P}(t), t)\Delta t + \mathcal{O}(\Delta t^2) \simeq \hat{\mu}_e(t) + f(\hat{\boldsymbol{\mu}}(t), \boldsymbol{P}(t), t)\Delta t,$$

where with f we label the r.h.s of equation (1.8c). The utilization of more advanced finite difference scheme could be part of future investigations.

1.3.2 MSG algorithm

In the numerical analysis performed in this thesis, the MCOT scheme is tested against a Multi-Start Greedy algorithm. The choice of this benchmark algorithm is backed by the fact that this second routine is widely employed as state-of-the-art, and by its moderate computational cost that allows to perform a comparison at a reasonable price. Moreover, the necessity of having a second benchmark method naturally arises being a theoretical baseline for the performance of MCOT still missing. The method is schematized in pseudocode (3).

The motivation behind the method is the following: first we select one random commodity from the set $\mathcal{C}$ and we perform a Multi-Sink Dijkstra Weighted routine by running the function `WeightedDijkstra`. This computes all the cheapest paths π joining

Algorithm 3 MSG algorithm

input: graph topology $\mathcal{G}(\mathcal{V}, \mathcal{E})$
input: set of communications $\mathcal{C}$
input: in-flowing and out-flowing mass $g_i, f_i \, \forall i \in \mathcal{C}$ (used alg. (2))
input: $\gamma > 0$
for $n = 1, n \leqslant N_{\text{tot}}, n \leftarrow n + 1$ **do**
    initialize sinks list $\mathcal{S} = \mathcal{C}$
    assign traffic to every edge $e \in \mathcal{E}$, $t_e(0) = 0$
    assign to every edge $e \in \mathcal{E}$ a cost $c_e(0) = \ell_e(t_e + 1)^\gamma$
    while $\mathcal{C}$ `!=` $\varnothing$ **do**
        sample $i_* \sim U\{\mathcal{C}\}$
        $\mathcal{S} \leftarrow \mathcal{S} \setminus i_*$
        while g_{i_*} `!=` 0 **do**
            do `WeightedDijkstra`(source $= i_*$, sinks $= \mathcal{S}$, weight $= \boldsymbol{c}$) $\overset{\text{return}}{\rightarrow}$ {`BestPath`, $s_* \in \mathcal{S}$} (see [19] for documentation)
            for $e \in$ `BestPath` **do**
                $t_e \leftarrow t_e + 1$, $c_e \leftarrow \ell_e(t_e + 1)^\gamma$
            if $g_{i_*} \geqslant f_{s_*}$ **then**
                $g_{i_*} \leftarrow g_{i_*} - f_{s_*}, f_{s_*} \leftarrow 0$
                $\mathcal{S} \leftarrow \mathcal{S} \setminus s_*$
            if $g_{i_*} < f_{s_*}$ **then**
                $f_{s_*} \leftarrow f_{s_*} - g_{i_*}, g_{i_*} \leftarrow 0$
                $\mathcal{C} \leftarrow \mathcal{C} \setminus i_*$
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{i_*\}$
    for $e \in \mathcal{E}$ **do**
        if $t_e \geqslant 1$ **then** C^n_{tot} += c_e
choose outcome with lower C^n_{tot}

the selected source and every other sink of the graph using a Dijkstra's (SPF) algorithm; the total cost of each path π is calculated by summing the cost c_e of every edge $e \in \pi$, and the cheapest `BestPath` is chosen among these routes. Formally this is

$$\texttt{BestPath} = \underset{\pi \in \Pi_{(i_*, \mathcal{S})}}{\arg\min} \left\{ C(\pi) := \sum_{e \in \pi} c_e \right\}$$

where $\Pi_{(i_*, \mathcal{S})}$ denotes the set of all the cheapest paths joining i_* and the sinks in $\mathcal{S}$.

After having found the `BestPath` associated with the endpoint s_* we run an "emptying procedure" that consists in transferring the mass g_{i_*} to sink s_*. Here it is important to notice that if the in-flowing mass of node i_* is greater than the capacity f_{s_*} the SPF scheme is repeated keeping i_* fixed until g_{i_*} is equal to 0. Moreover, when $g_{i_*} = 0$, the node i_* gets removed from the set of commodities, to avoid re-sampling it when iterating the cycle. Analogously, when $f_{s_*} = 0$ the vertex s_* is eliminated from $\mathcal{S}$, being its capacity to accommodate the g_{i_*} equal to zero. One last important detail is the following, when a node i_* is sampled it gets removed by the sets of sinks, and re-added at the end of the accommodation. This operation has the same purpose of construction

(1.1b). Namely, avoiding trivial dynamics where the mass in-flowing from a commodity exits the graph from the exact same node.

In algorithm (3) we also introduced the new scalar parameter $\gamma > 0$, which has a direct analogy with the exponent $\beta > 0$ of MCOT. The purpose of γ is indeed controlling traffic congestion, and the intuition behind its functioning is straightforward. For values of $\gamma < 1$ the cost contribution $(t_e + 1)^\gamma$ is subadditive, while if $\gamma > 1$ this is superadditive, meaning that in the former case traffic congestion is preferred over rerouting, while in the second taking different routes is favored over concentrating mass on less edges. Following this reasoning it comes natural to compare experiments performed with MCOT and MSG where $\beta = 1/\gamma$.

Lastly, we notice that the accommodation scheme is executed N_{tot} times for every instance, this is what makes the algorithm Multi-Start. We use this expedient to remedy the intrinsically poor performances of greedy routines where i) the index i_* is chosen at random at every iteration, ii) global traffic occupancy is not taken in account and every source choose selfishly its sink.

1.4 Numerical analysis: MCOT vs. MSG

In this section we show some of the results of our numerical experiments. The idea behind the choice of the quantities we decide to present is twofold: first, we try to analyze separately the different settings upon which MCOT and MSG are built, looking for connections and similarities in the problem formulations. In MCOT we have in fact a mechanism of accommodation similar to the one of Physarum Solvers, where the idea of traffic is not encoded by a discrete variable expressing the occupancy state of a link (as for the MSG), but instead it is given by the transport densities. Secondly, we try to directly compare the two methods using metrics that are shared by the models, examples are the total length of the graph, or the number of non occupied edges.

The analysis has been entirely performed on a benchmark internet-like topology obtained using the BRITE graph generator [28]. Performances of MP on these kind of graphs have been studied in [5]. These experiments, as we will see later on in the thesis, have been used as baseline for the implementation of SMP-EDP. The network comes from a setting in which spatial position of nodes is not relevant, because of that we assigned conventional lengths $\ell_e = 1 + 10^{-3}\varepsilon, \varepsilon \sim U(0,1)$ to every edge $e \in \mathcal{E}$. The purpose of the bias ε is to ease the selection of `BestPath` when running `WeightedDijkstra`, so that we avoid obtaining routes with the same cost $C(\pi)$. This choice is easily modifiable when using the schemes, for example to model railroad or road networks in which spatial coordinates of stations and junctions play a crucial role in the experimental setting. Independently on the framework, however, the qualitative reasoning explaining all the results we are presenting does not change.

The structure of this section is the following. We divide the text in different subsections, each one associated with a quantity studied in the analysis. These metrics are first briefly presented, and a motivation for its relevance its given; after this first introduction we show and discuss our numerical results.

1.4.1 MCOT: Transport density fraction

The first quantity that we inspect is what we refer as transport density fraction. This is the portion of total conductivity occupied by each edge in the graph. Precisely, for every link $e \in \mathcal{E}$ we compute the ratio

$$R_e = \frac{\hat{\mu}^*_e}{\sum_{e' \in \mathcal{E}} \hat{\mu}^*_{e'}}$$

and we collect these objects in a E-dimensional vector $\boldsymbol{R}$. The entries of this array are consecutively sorted is ascending order and plotted against the edge indices, the purpose of this procedure is highlighting how different behaviors in traffic congestion arise when running the model changing β. The expected outcome, consistently with the discussion in §1.2, is the following. For $\beta < 1$ the mass traveling through the graph is redirected to many thin the edges to avoid congestion, this will reflect in having a larger fraction of $\mathcal{E}$ a occupying a consistent portion of $\sum_{e' \in \mathcal{E}} \hat{\mu}^*_{e'}$, resulting in bigger tails when plotting $\boldsymbol{R}$ vs. $\mathcal{E}$. If $\beta > 1$ the model naturally reroutes mass on fewer more trafficked links, in this case the algorithm produces a network where the total transport density is distributed among less edges, correspondingly we obtain smaller tails in the plot $\boldsymbol{R}$ vs. $\mathcal{E}$.

We can clearly see this behavior in figure 1.2, obtained running the MCOT scheme on the mesh `blrand_10_1` used in [5], choosing $M = 50$ random commodities among the $N = 500$ total nodes of the graph, and averaging 5 different realizations of the algorithm. Precisely, in this plot four different stem plots are displayed, realized setting $\beta \in \{0.5, 1.0, 1.5, 2.0\}$. For the one obtained fixing $\beta = 0.5$ we see that the entries of $\boldsymbol{R}$ are distributed in such a way that most of the links in the graph occupy a large amount of the total area of the plot, producing thick tails. Instead in the two cases where $\beta > 1$ the total transport density $\sum_{e' \in \mathcal{E}} \hat{\mu}^*_{e'}$ is mainly concentrated on approximately 100 edges, while the remaining links have negligible values of R_e.

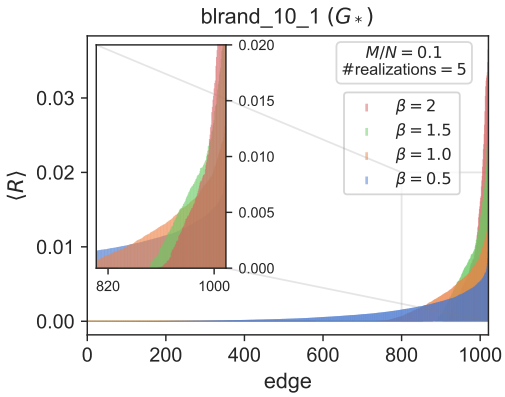


Figure 1.2: Sorted entries of $\boldsymbol{R}$ vs. $\mathcal{E}$. MCOT, mesh: `blrand_10_1`, $N = 500$, $M = 50$, $E = 1020$, $\beta \in \{0.5, 1.0, 1.5, 2.0\}$. Results averaged over 5 realizations. Similar plots for $M = 30, 40$ are in figs. A.1a and A.1b in appendix A.

From the analysis of this first quantity it is clear that not all the edges in a network have the same "importance" in terms traffic, meaning that the mass may be concentrated on a small fraction of the total number of links. Supported by this motivation we perform

a "trimming procedure" of the graph at convergence, that consists in cutting those edges that are "not important" for the dynamics. Here, we must formally define what we mean with the term "importance" of a link.

A first technique could be deleting all the edges which transport density is below a certain fixed threshold $\tau > 0$, deciding that not relevant links are those with μ_e^* small enough. This corresponds to constructing the set

$$\mathcal{E}_\tau := \{e \in \mathcal{E} \, : \hat{\mu}_e^* > \tau, \text{ with } \tau > 0\},$$

with which we build the network

$$\mathcal{G}_\tau := \mathcal{G}(\mathcal{V}, \mathcal{E}_\tau).$$

However, in the context we are studying this elementary approach has the disadvantage of cutting links not taking care to which nodes these are attached. That is, after the trimming process we could end up with a graph where one or more commodities are disconnected from the rest of the graph. This is something that clearly we want to avoid, since on a network where sources and sinks are disconnected accommodation cannot performed. To overcome this problem we use a different cutting method: we iteratively delete edges in increasing order of $\hat{\mu}_e^*$, paying attention that at least a path between any pairing of commodities in $\mathcal{C}$ exists. We label the network obtained in this way with $\mathcal{G}_* := \mathcal{G}(\mathcal{V}, \mathcal{E}_*)$. The advantage of this refined procedure is that it can erase edges occupying a small fraction of $\sum_{e'} \hat{\mu}_{e'}^*$, while not cutting out sources from the giant component of the graph. Throughout the following numerical analysis we study the MCOT model using the trimmed network $\mathcal{G}_*$.

1.4.2 MCOT: Pressure gradients

The second part of this section is devoted to an intensive study of the discrete pressure gradient matrix $\boldsymbol{\Delta P}^* \in \mathbb{R}^{E,M}$, quantity that can provide useful insights on the MCOT algorithm.

Initially, we are interested in seeing how the pressure gradients are distributed on the edges of the network. This examination is done with the purpose of understanding how the graph distributes the force acting on its edges, when changing β from the subadditive to the superadditive regime. Also in this case a prediction of the result can be done beforehand, and the intuition behind that comes once again from Physarum Solvers. It is known [47, 48] that in their construction it is assumed that the flow of mass in the links is modeled as a Poiseuille flow; because of that, recurring to the Hagen-Poiseuille equation we expect to obtain $\delta p_e \sim 1/A_e^2$, where A_e is the section of the tube e. Moreover, since in our model $\hat{\mu}_e^* \sim A_e$, to bigger values of the transport densities (when $\beta > 1$) correspond smaller pressure gradients. While for $\beta < 1$, when the graph has thinner edges, δp_e attains larger values.

This qualitative explanation is confirmed by the following plot. Precisely, we show four histograms containing the entries of the E-dimensional array $\sum_{i \in \mathcal{C}} |\Delta P_e^{*i}|$ obtained for $\beta \in \{0.5, 1.0, 1.5, 2.0\}$. The simulation has been performed on `blrand_10_1`, with $M = 50$.

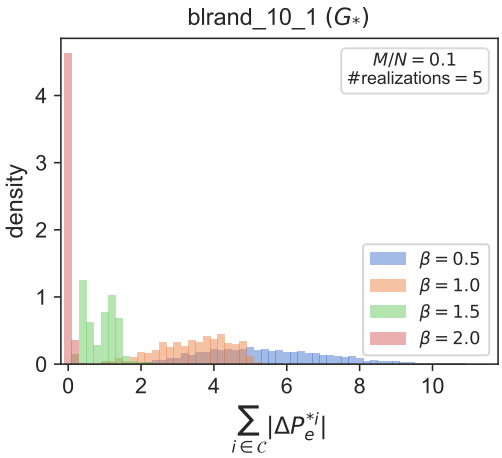


Figure 1.3: Histograms showing pressure gradients distribution on $\mathcal{E}$. MCOT, mesh: `blrand_10_1`, $N = 500$, $M = 50$, $E = 1020$, $\beta \in \{0.5, 1.0, 1.5, 2.0\}$. Results computed over 5 realizations. Similar plots for $M = 30, 40$ are in figs. A.1c and A.1d in appendix A.

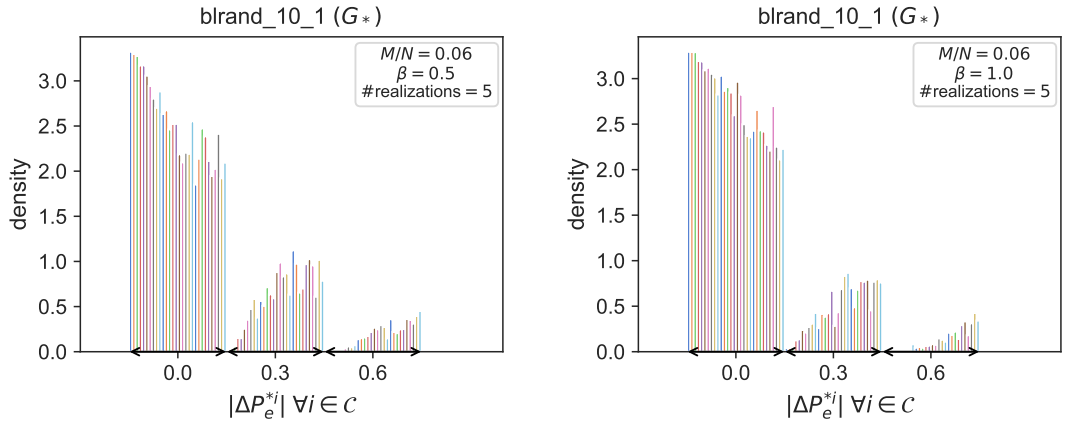
In figure 1.3 we can see that for $\beta = 0.5$ the bins tend to have larger values and bigger tails, as a confirmation of the fact that $\mathcal{G}_*$ has many thin edges and that $\delta p_e \sim 1/A_e$. Meanwhile as β grows the entries of the vector entries concentrate around small values and the histogram has smaller tails. This behavior is strongly evident when $\beta = 2$, in this case in fact all the entries of $\sum_{i \in \mathcal{C}} |\Delta P_e^{*i}|$ are highly clustered around 0.

Now, we consider again the discrete pressure gradient matrix $\boldsymbol{\Delta P}^* \in \mathbb{R}^{E,M}$. What we want to look at is again the distribution of its entries over the edges of the graph, but i) we want to separate the pressure distribution produced by each commodity in $\mathcal{C}$, ii) we are interested in understanding which is the relation between the in-flowing mass terms g^i their correspondent E-dimensional arrays $|\boldsymbol{\Delta P}^{*i}| \, \forall i \in \mathcal{C}$, where the absolute values is applied element-wise to each entry of the vector.

In order to perform this analysis we build M histograms associated to every commodity in $\mathcal{C}$ that we order from left to right for increasing values of g^i. This construction, that we repeated using four different values of $\beta \in \{0.5, 1.0, 1.5, 2.0\}$, is displayed in figure 1.4. The simulations were performed on the mesh `blrand_10_1` fixing $M = 30$.

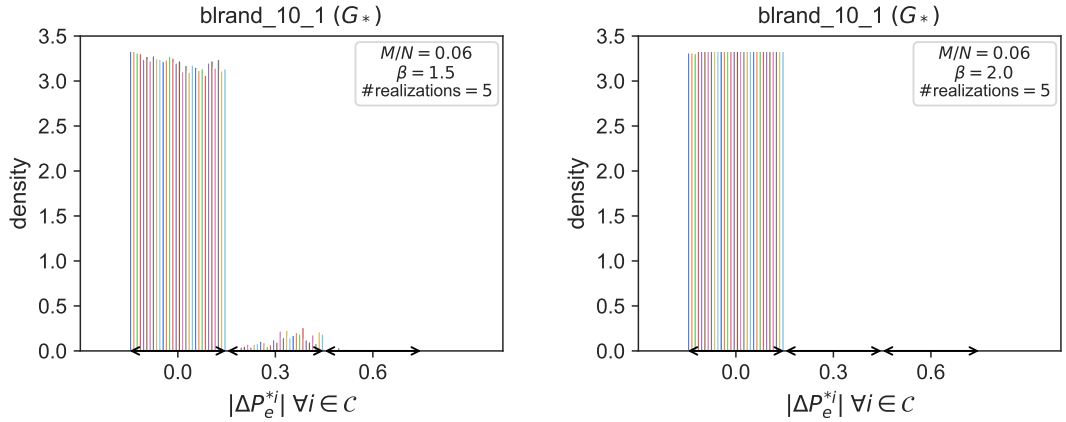
Looking at the four panels of figure 1.4 we can notice two different characteristic traits of our model. The first is fundamentally the one that we displayed in figure 1.3, namely that when for large values of the exponent β all the pressure gradients tend to concentrate around small values, while decreasing β pressure differences are higher and more widely distributed. This feature can be seen in these plots noticing that for $\beta = 0.5$ a large contribution to the total area of the histograms is given by the bins concentrated around higher values of $|\Delta P_e^{*i}|$. This portion of the area progressively decreases when increasing β, for $\beta = 1$ we see that both the second and the third cluster of bins are smaller; this trend is even more evident when $\beta = 1.5$, here in fact the third group of bins has almost completely disappeared. Ultimately if $\beta = 2.0$ most of the pressure histogram entries are concentrated around 0.

Furthermore, we may see how the bins are either almost monotonic increasing or decreasing. This is symptomatic of another intuitive behavior of our model, that is:



(a) $\beta = 0.5$

(b) $\beta = 1.0$



(c) $\beta = 1.5$

(d) $\beta = 2.0$

Figure 1.4: Histograms showing the pressure gradients distribution for each $i \in \mathcal{C}$ separately. MCOT, mesh: `blrand_10_1`, $N = 500$, $M = 30$, $E = 1020$, $\beta \in \{0.5, 1.0, 1.5, 2.0\}$. Results computed over 5 realizations. Similar plots for $M = 40$ are in figs. A.1e–A.1h in appendix A.

to higher values of g_i correspond higher values of pressure gradients. This, other than being supported by physical intuition, i.e. it is reasonable to expect that inserting a larger amount of mass in a node the force exerted by that vertex will be higher, is also formally verifiable with equation (1.8a) where pressure terms are directly proportional to in-flowing mass. In particular, the first cluster of bins is approximately monotonically decreasing while the other clusters are approximately monotonically increasing, meaning that a larger portion of the total pressure gradient contribution tend to be occupied by those commodities with higher values of g^i.

To complete this first part of the numerical analysis we inspect the relation between the entries of matrix $\boldsymbol{\Delta P}^* \in \mathbb{R}^{E,M}$ and the commodities in $\mathcal{C}$.

Formally what we have done in the numerical implementation is the following. We built a stem plot where in the x-axis we inserted the commodity indices $i \in \mathcal{C}$, and on the y-axis the total absolute pressure difference each commodity exercises on all the edges in $\mathcal{E}$: $\sum_{e \in \mathcal{E}} |\Delta P_e^{*i}|$, $\forall i \in \mathcal{C}$. The commodities, also in this case, are ordered from the left to

the right for increasing in-flowing mass. Moreover we kept fixed for every β the in-flowing and the out-flowing mass assigned to each node, together with the assignations of sources and sinks in $\mathcal{V}$. By doing that we hoped we might be able to compare qualitatively the effect that the exponent has on different experiments with the same setting, meaning that we wanted to find a trend where relative variation between each $\sum_{e\in\mathcal{E}} |\Delta P_e^{*i}|$ remains qualitative the same $\forall i \in \mathcal{C}$, while the gradient (in absolute value) are suppressed as β grows. However this seems not to be the case, since the complex topology of the network doesn't allow the comparison of instances of the graph with different exponents. This, indeed, could be objective of future investigations. Still, to provide a better graphical understanding of how pressure gradients are distributed among edges we put beside the stem plot four instances of the mesh at convergence with $\beta \in \{0.5, 1.0, 1.5, 2.0\}$.

In figure 1.5 we show the results of our experiments. The simulations have been performed on the mesh `blrand_10_1`, setting $M = 30$.

Looking at these figures some observations on our model can be done. First of all we have, again, that to higher values of β correspond smaller values of the pressure gradients: in this case we notice that the stem plots gets largely suppressed when increasing β. In fact we pass from having values ranging approximately in the interval $(0, 80)$ for $\beta = 0.5$, to having all the stems close to 0 if $\beta = 2.0$. Apart from that, we see that the stem plots are not exactly monotonically increasing, meaning that to higher g^i not always correspond higher values of the pressure gradients on $\mathcal{G}_*$. This is a clear signal that traffic, namely the interaction with other commodities, plays a crucial role in rerouting the mass in the graph.

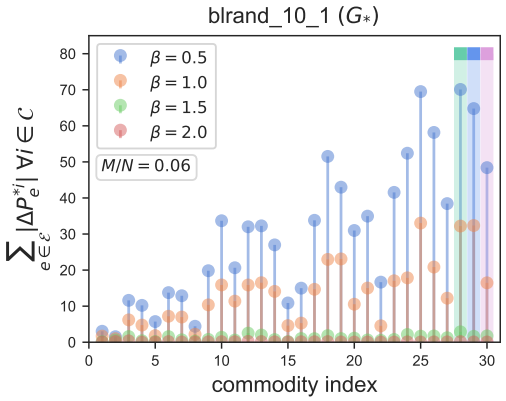
As said, figures 1.5b−1.5e serve to provide an intuition on how the mass is distributed in the network. Here the size of the orange squared nodes (the commodities) are proportional to their in-flowing mass and the 3 nodes with the biggest g^i have been colored in green, blue and purple. Moreover, we superimposed to the faded gray links of width $\propto \hat{\mu}_e$ other edges colored in the same way as commodities $28, 29, 30$. Their width is

$$\text{width}_e^i \propto \frac{|\Delta P_e^{*i}|}{\sum_{i\in\{28,29,30\}} \sum_{e\in\mathcal{E}} |\Delta P_e^{*i}|},$$

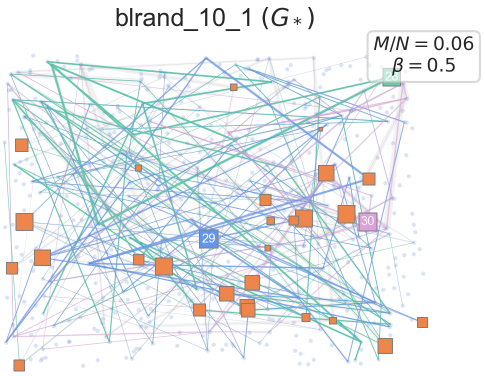
so they are proportional to the pressure gradient caused separately by each commodity on every link, which is a first naive and intuitive way to try to split the contribution of the conductivities $\hat{\mu}_e$ that are merged by equation (1.8b). From the plots we have a qualitative impression of the effect that β has on penalizing or favoring traffic congestion, furthermore we can see how the topology of the mesh affects mass rerouting. For example, focusing on figure 1.5e we clearly notice how, even if the purple source has the biggest in-flowing mass, its pressure contribution is consistently smaller than the green one. Indeed mass entering $i = 28$, before exiting the network gets rerouted between many "junction" nodes from which it cannot exit.

1.4.3 MSG: traffic and efficiency

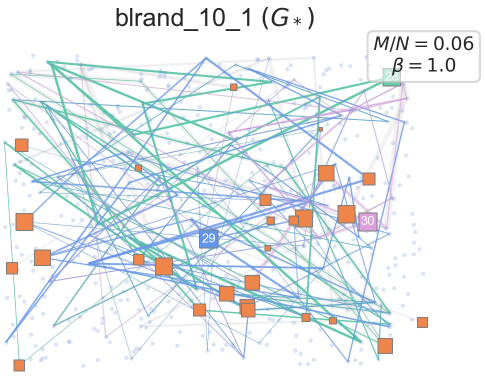
This subsection of thesis is devoted to studying the results of the numerical experiments performed with the MSG scheme. In particular, we want to find a way to quantitatively measure network traffic, focusing on trying to understand which is its relation with the



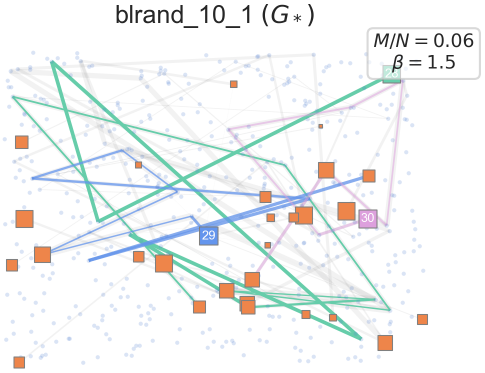
(a) $\sum_{e\in\mathcal{E}} \Delta P_e^{*i}$ vs. $\mathcal{C}$.



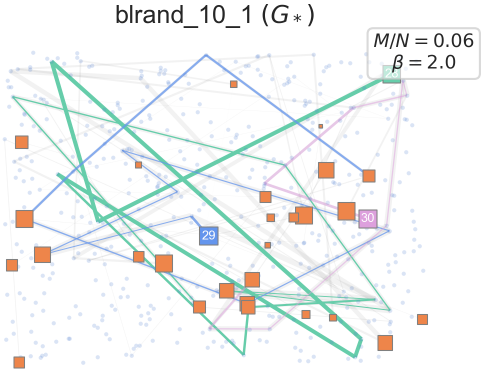
(b) $\beta = 0.5$.



(c) $\beta = 1.0$.



(d) $\beta = 1.5$.



(e) $\beta = 2.0$.

Figure 1.5: Panels **(a)**: $\sum_{e\in\mathcal{E}} |\Delta P_e^{*i}|$ vs. $\mathcal{C}$. Panel **(b-e)**: Qualitative explanation of non-trivial interaction among commodities. $\ell_e = 1 + \varepsilon 10^{-3}, \varepsilon \sim U(0,1)$, the position of each node is sampled at random in the square $[0,1] \times [0,1]$ for visualization purpose. MCOT, mesh: `blrand_10_1`, $N = 500$, $M = 30$, $E = 1020$, $\beta \in \{0.5, 1.0, 1.5, 2.0\}$. Results computed on one single instance, keeping $\{g^i\}^i$ and $\{f^i\}^i$ fixed for every β. Similar plots for $M = 40, 50$ are in figs. A.1i and A.1j in appendix A.

exponent $\gamma > 0$. This analysis has been carried out by constructing the following two plots:

- a histogram of the traffic distribution on the edges of the network. This measure is performed for different values of γ, ranging in the set $\{0.0, 0.5, 1.0, 1.5, 2.0\}$, so that we can have a clear visualization of the effect that γ has on mass rerouting.

- a plot of the efficiency vs. γ. This quantity has been defined with the purpose of evaluating the fraction of edges e for which the `BestPath`s of two or more sources overlap. A formal expression is provided by

$$\text{efficiency} := 1 - \frac{|\{e \in \mathcal{E} \,|\, t_e > 1\}|}{|\mathcal{E}|}. \tag{1.9}$$

As we have done for most of the previous metrics, by analyzing algorithm (3), we can give a qualitative explanation of the behavior of the plots. Precisely, we recall how the cost of an edge changes as a power law w.r.t γ, namely for every $e \in \mathcal{E}$ we have $c_e = (t_e + 1)^\gamma$. Practically, this means that when we perform the accommodation routine, as we increase the value of γ the `WeightedDijkstra` scheme tend to choose longer path with less overlaps against shorter routes which edges have been already traversed. The effect on the traffic distribution is the following, for $\gamma = 0$ the final network has short routes with many overlaps, since the cost of each edge remains equal to ℓ_e at every iteration of the scheme. Instead, increasing γ, it is reasonable to expect a graph where routes taken by commodities are progressively longer and the overlaps monotonically decrease. The same behavior reflects on the plot of the efficiency, here we have a smaller fraction of overlaps when γ grows, so to get a monotonically increasing curve.

Traffic distributions and efficiency are displayed in the following plots. As usually, the simulations have been performed on the mesh `blrand_10_1`.

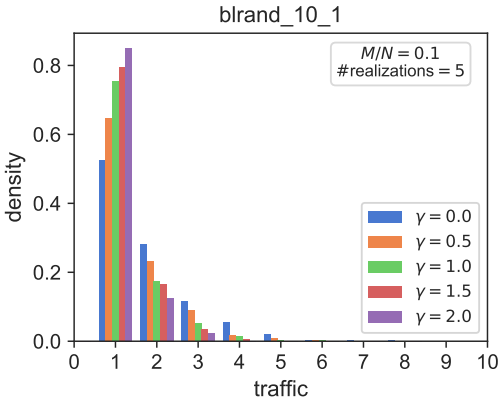


Figure 1.6: Histograms showing the traffic distribution on $\mathcal{E}$ for different values of γ. MSG, mesh: `blrand_10_1`, $N = 500$, $M = 50$, $E = 1020$, $\gamma \in \{0.0, 0.5, 1.0, 1.5, 2.0\}$. Results averaged over 5 realizations. Similar plots for $M = 30, 40$ are in figs. A.1k and A.1l in appendix A.

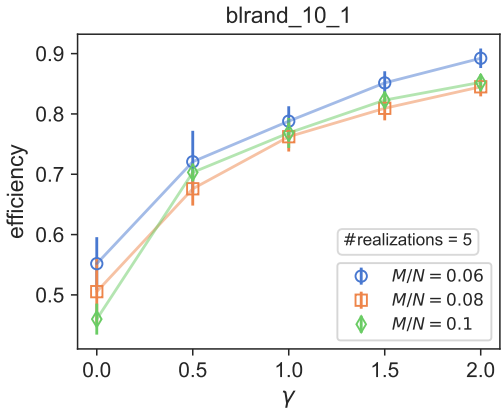


Figure 1.7: efficiency vs. γ. MSG, mesh: `blrand_10_1`, $N = 500$, $M = 30, 40, 50$, $E = 1020$. Results averaged over 5 realizations.

Looking at figure 1.6 we see the behavior just described. When increasing γ the bin cluster centered around 1 increases, highlighting that the more γ grows, the stronger is the tendency of the scheme to return routes with less overlaps. On the contrary the remaining bin clusters have the opposite tendency, with a decay in the y-axis for increasing values of γ. In figure 1.7 we see that all the curves of the efficiency are monotonically increasing, in particular we pass from having a fraction of trafficked edges approximately equal to 0.5, to getting values around 0.9 for $\gamma = 2.0$.

1.4.4 Idle edges

In this last part of the numerical analysis we focus on the comparison of our two methods, studying metrics useful to find their differences and similarities, other than highlighting the better performances of the MCOT algorithm respect to the MSG. Here, following the intuition given in §1.3.2, to do a meaningful comparison between the results we evaluate experiments obtained fixing the exponents so that $\beta = 1/\gamma$.

The first quantity that we analyze is the idle edge fraction. This has two different formal definitions for the contexts of MCOT and MSG. These are, respectively, the portion of edges that gets trimmed in the cutting procedure performed to obtain $\mathcal{G}_*$, formally

$$\text{edge idle fraction} := 1 - \frac{|\mathcal{E}_*|}{|\mathcal{E}|}.$$

And the amount of edges that does not get occupied when running the MSG routine, that is

$$\text{edge idle fraction} := \frac{|\{e \in \mathcal{E} \,|\, t_e = 0\}|}{|\mathcal{E}|}. \tag{1.10}$$

The two definitions above have the same meaning, since both the quantities are measuring how many link the in-flowing mass occupies in the accommodation process. In this regard, we may notice that when running MCOT scheme in principle all the edges gets occupied i.e. $\hat{\mu}_e^* > 0 \, \forall e \in \mathcal{E}$, in this case it is the cutting scheme discussed above that plays a key role in allowing a meaningful comparison between the methods.

The plot idle edge fraction vs. β is shown in the figure 1.8. The experiments have been performed, also in this case, on the mesh `blrand_10_1` taking different values of the ration M/N.

Looking at the plot we can see that for both algorithms the fraction of idle edges is monotonically increasing with β. This trend is consistent with all the theoretical discussion we have done up to this points. Indeed, in the MCOT model when increasing β we tend to have a smaller fraction of edges occupying the larger portion of the total transport density (see fig. 1.2), so a bigger number of links will be trimmed off to obtain $\mathcal{E}_*$. Instead in the MSG model, as we have just seen in the previous subsection, a big value of γ (small value of β) forces longer routes with no overlap, meanwhile if γ is small (resp. β is high) more traffic congestion is allowed, resulting in shorter paths and less occupied edges.

Despite having the same qualitative behavior, it is import to highlight the differences between the results obtained with the two routines. What we can clearly notice is that the MCOT algorithm enhances the tendency of the model to favor or penalize traffic congestion. In fact in the subadditive regime, when $\beta = 0.5$, the fraction of idle edges of the MCOT model is lower than the MSG one. Instead, if β increases and we pass to the superadditive regime, the idle edges fraction is higher for the MCOT. In this sense then the Optimal Transport approach performs better than the Greedy one, accentuating the inclination to ease or impede congestion. Already from this first preliminary result we can notice how the MCOT algorithm is a simple and efficient tool to easily perform routing optimization while managing traffic on the network.

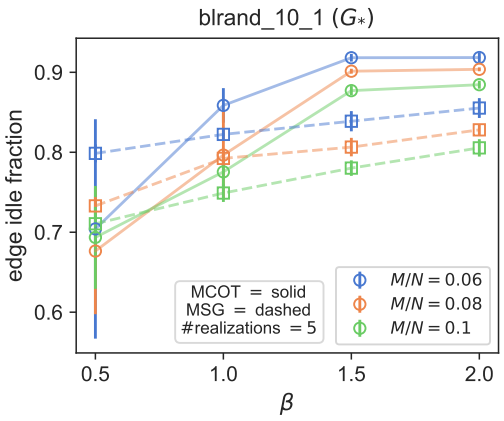


Figure 1.8: idle edge fraction vs. β. Mesh: `blrand_10_1`, $N = 500$, $M = 30, 40, 50$, $E = 1020$. Results averaged over 5 realizations.

1.4.5 Total length

A second part of our analysis is devoted to the study of the total length of the graph. In particular, we want to study how this quantity changes with the exponent β, and with complexity ratio M/N. Precisely, we study the normalized length L/L_0 obtained dividing the total length of the graph at convergence

$$L := \sum_{e \in \mathcal{E}} \ell_e t_e$$

where the length of the mesh `blrand_10_1` is denoted by L_0. Analogously to what happened studying the fraction of idle edges, the trimming process returning $\mathcal{G}_*$ is crucial. In fact in the MCOT model all the edges remain occupied, however trimming those edges that are "not important" for the dynamics we are able to construct a graph $\mathcal{G}_*$ that can be meaningfully compared with the network using the MSG routine.

The results obtained with our numerical experiments are displayed in figures 1.9 and 1.10.

Looking at figure 1.9 we notice that the behavior of the total length is as we could have expected. Increasing β both the models displays a monotonic decrease of L/L_0, consistent with the fact that more traffic is preferred over rerouting for large exponents. Moreover, for bigger values of the ratio M/N, when the complexity i.e. the number of sources to accommodate grows, the length of the graph gets bigger (under the same value of β). Furthermore, even if the qualitative trend of the results is the same for both algorithms there are also important differences. First, we notice that for $\beta = 0.5$ and $M/N = 0.04$ the models return two networks with approximately the same length. What happens here is that, being the number of sources in $\mathcal{C}$ small the rerouting process does not stand out since most of the mass can be distributed without the need of changing paths. However, progressively increasing the ratio M/N the total length of $\mathcal{G}_*$ is also increasing more and more w.r.t. the one of the MSG model. When passing to high values of β, to the superadditive regime, to high values of the complexity M/N are associated remarkably smaller values of L/L_0 for the solid line (MCOT) w.r.t the dashed ones (MSG). This tendency of enhancing the rerouting process is the same effect that we observes when studying figure 1.8.

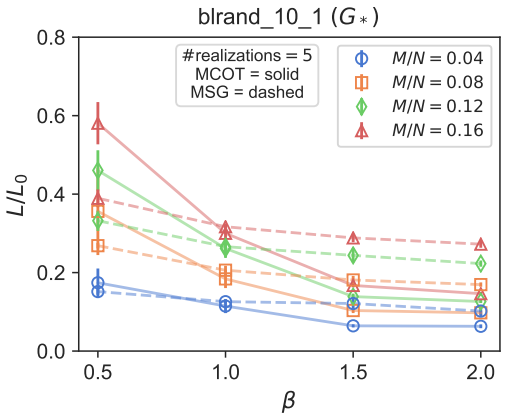


Figure 1.9: L/L_0 vs. β. Mesh: `blrand_10_1`, $N = 500$, $M = 30, 40, 50$, $E = 1020$. Results averaged over 5 realizations.

When analyzing figure 1.10 we can appreciate the same typical features of the models from a different perspective. Here we have, as expected, that the total length of the graph is monotonically increasing with the ratio M/N. Apart from that, the fundamental thing to notice is how β serves as a regulatory parameter for traffic congestion. This mechanism is particularly visible in the MCOT algorithm; in fact this method compared to the MSG returns graphs with larger total length when setting $\beta = 0.5$, and notably smaller values of L/L_0 for β in the superadditive regime, i.e. $\beta = \{1.5, 2.0\}$.

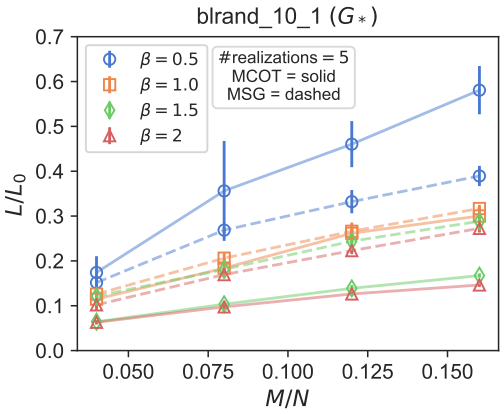


Figure 1.10: L/L_0 vs. M/N. Mesh: `blrand_10_1`, $N = 500$, $\beta \in \{0.5, 1.0, 1.5, 2.0\}$, $E = 1020$. Results averaged over 5 realizations.

Chapter 2

Message Passing Model

In the last decades, Message Passing has been largely employed in several fields of science as spin-glass theory [30, 31] and combinatorial optimization [32]. Only more recently this technique has been used to tackle routing problems, a seminal paper in this setting is [5].

Here, the authors propose a distributed algorithm to solve the EDP problem built upon MP. The method, other than being theoretically justified, is largely tested against several benchmark algorithms. The results of this extensive analysis show how MP outperforms most of the state-of-the-art methods, in this sense the capability of MP routine to take in account the global traffic occupancy of the network proves to be fundamental for performance improvement. Lastly, it is highlighted how the MP technique provides a viable polynomial time implementation opening interesting perspectives in the solution of routing problems.

Because of these key features, the MP routine of [5] is used as a baseline to construct the SMP-EDP model, which, in turn, provides a novel approach able to overcome most of the limitations of the ones present in the current literature.

Another interesting model, with better performance than most of the state-of-the-art is developed in [58]. Here the authors use an approach derived by physics of interacting polymers to build a distributed routing algorithm capable of considering all individual path choices simultaneously. The model, however, presents two primary issues: the algorithm shows issues in converging when taking large values of the complexity ratio M/N, and the analytical derivation of the key equations may result impractical to the reader due to its complexity. The remedy proposed to solve the first problem is to use what is referred to as decimation procedure, which consists in fixing communication paths already biased towards a particular choice of routes. For what concern the second issue, the model remains intractable with respect to the rather self-contained analytical derivation of the MP routine provided in [5].

The structure of this chapter is the following. First, we provide a short theoretical introduction on the MP method. This presentation is made without any purpose of completeness, it rather serves to briefly show which are the key elements composing the setting on which SMP-EDP is based. Secondly, we want to formally construct the SMP-EDP routine and the SMSG benchmark algorithm. These two methods are tested in terms of both computational efficiency and performances, i.e. computing meaningful metrics to study their capability of efficiently managing traffic congestion.

2.1 Theoretical introduction

In this section we present the building blocks needed to construct the SMP-EDP model; first, we formally define the EDP problem, then we introduce the Message Passing equations. The last subsection is devoted to the explanation of the mapping of the EDP problem into a weighted matching one. This step, as will be clear later in the thesis, is essential to overcome the computational bottleneck introduced by the MP equations.

The derivations are done following for the most part [5] and [29].

2.1.1 The EDP problem

The EDP problem can be formally stated as follows. Given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and a set of M communications $\mathcal{C}$, i.e. pairings between sources and sinks, we want to find the maximum number of edge-disjointed paths joining them that can be accommodated on the network.

To solve this problem we map the EDP problem into a Minimum Weight EDP one (MWEDP), formulation that takes in account both edge disjointness and path length optimization. Formally, this consists in finding M communication paths $\pi_i, i \in \mathcal{C}$, that i) satisfy the edge disjoint constraint, and ii) minimize the total cost, that we define as

$$C := \sum_{i \in \mathcal{C}} c(\pi_i) = \sum_{i \in \mathcal{C}} \sum_{e \in \pi_i} c_e,$$

with c_e cost of a link $e \in \mathcal{E}$. Practically, the EDP problem can be recovered from the second formulation using an expedient: we connect each source-sink couple with an extra edge and we assign to these M extra edges a cost $c_e > |\mathcal{E}|$. This step is essential to guarantee that when performing the MWEDP accommodation each commodity gets connected by a path, and the algorithm is forced to choose an extra edge only if no other route can be taken while respecting the EDP constraint. Ultimately, after the accommodation of every source-sink pair is done the extra links are removed, so that we achieve both cost minimization and maximization in terms of the number of source-sink couples.

The mathematical formulation of the problem can be done first introducing the current vectors $\boldsymbol{I}_{vw} \in \mathbb{R}^M \, \forall (v, w) \equiv e \in \mathcal{E}$, which entries are defined as

$$I^i_{vw} := \begin{cases} +1 & \text{if communication } i \text{ passes from } v \text{ to } w \\ -1 & \text{if communication } i \text{ passes from } w \text{ to } v \\ 0 & \text{otherwise.} \end{cases}$$

Each current has to satisfy Kirchoff's law, analogously to equation (1.8a) in §1.2 we need $\forall v \in \mathcal{V}$ the constraint

$$\sum_{w \in \partial v} I^i_{vw} - \Lambda^i_v = 0 \quad \forall i \in \mathcal{C} \tag{2.1}$$

to be satisfied. Here we defined for each node $v \in \mathcal{V}$ and for each communication $i \in \mathcal{C}$

the variable Λ_v^i as

$$\Lambda_v^i := \begin{cases} +1 & v \text{ is a source for } i \in \mathcal{C} \\ -1 & v \text{ is a sink for } i \in \mathcal{C} \\ 0 & \text{otherwise.} \end{cases}$$

This quantity, to some extent, encodes the same information provided by the two $N \times M$-dimensional matrices $\boldsymbol{G}, \boldsymbol{F}$ constructed for the MCOT model, in fact it allows to express with one single object the distinction between sources and sinks. Moreover, before continuing it is important to notice that each array contains M entries, allowing us to register the edge occupancy caused by each communication separately. In this sense the EDP problem setting is different from the MCOT, where to regulate traffic congestion it is crucial to merge traffic given by different commodities.

The formulation of the problem can be further refined by noticing that because of the EDP constraint each vector $\boldsymbol{I}_{vw} \in \mathbb{R}^M$ can be parametrized by a variable taking $2M + 1$ values, associated to cases where one entry of $\boldsymbol{I}$ is equal to ± 1, and the remaining are equal to 0, or when the vector is identically equal to 0. It is clear how the set of currents $\{\boldsymbol{I}_{vw}\}_{(vw)\equiv e \in \mathcal{E}}$ completely characterizes the occupancy of the network and using them we can formulate the MWEDP problem in a compact fashion. Formally the problem reads as

$$\{\boldsymbol{I}_e^*\} = \underset{\{\boldsymbol{I}_e\} \in \mathcal{I}}{\arg\min} \left\{ C(\{\boldsymbol{I}_e\}) := \sum_{e \in \mathcal{E}} c_e f(||\boldsymbol{I}_e||) \right\}, \tag{2.2}$$

where

$$||\boldsymbol{I}_e|| := \sum_{i \in \mathcal{C}} |I_e^i|,$$

the EDP constraint is ensured defining f as

$$f(||\boldsymbol{I}_e||) := \begin{cases} 0 & \text{if } ||\boldsymbol{I}_e|| = 0 \\ 1 & \text{if } ||\boldsymbol{I}_e|| = 1 \\ \infty & \text{if } ||\boldsymbol{I}_e|| > 1 \end{cases}$$

and we label with $\mathcal{I}$ the space of all the currents for which equation (2.1) holds, namely

$$\mathcal{I} := \left\{ \{\boldsymbol{I}_e\}_e \,\middle|\, \sum_{w \in \partial v} I_{vw}^i - \Lambda_v^i = 0 \quad \forall i \in \mathcal{C}, \forall v \in \mathcal{V} \right\}.$$

In [5] the authors set $c_e = \ell_e = 1 \, \forall e \in \mathcal{E}$, so that the final cost of the network is equal to the total path length. In the SMP-EDP model, instead, the cost of each edge does not remain fixed. Indeed the underlying idea of our algorithm is to construct an iterative routine in which, at every step, each link get sequentially updated dependently on its occupancy state in the past. It is by exploiting this expedient that we relax the original formulation of the problem and we are able to accommodate all the source-sink couples.

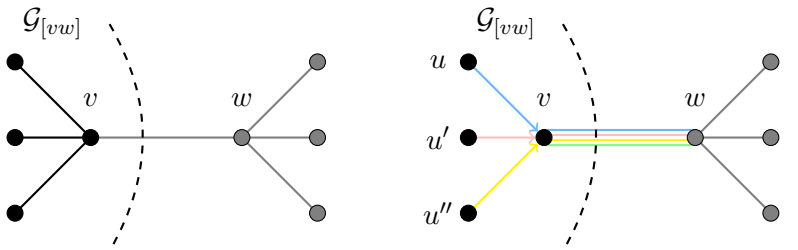
2.1.2 The Message Passing routine

The optimization problem (2.2) can be solved using a Message Passing routine. More precisely, MP is able to provide a solution that is exact on trees and approximately correct for locally tree-like cyclic graphs (Theorem 14.4, [29]).

The MP algorithm can be derived with the following construction. First, given a tree $\mathcal{G}$ we build the modified cavity graph $\mathcal{G}_{[vw]}$, made by the connected component of $\mathcal{G} \setminus (v, w)$ (figure 2.1a). Secondly, we introduce the message $E_{vw}(\boldsymbol{I}_{vw})$, function of the M-dimensional current passing trough the link $(v, w) \in \mathcal{E}$; this is the minimum cost $C(\{\boldsymbol{I}_{vw}\})$ among all the current configurations on the cavity graph $\mathcal{G}_{[vw]}$ satisfying equation (2.1), given that the vector $\boldsymbol{I}_{vw}$ has been fixed a priori. We label the set of these currents with $\mathcal{I}_{[vw]}$. Using these objects, and exploiting the absence of loops in the subtree $\mathcal{G}_{[vw]}$ (necessary condition for MP to converge exactly to $\{\boldsymbol{I}^*_e\}$), we can write the min-sum update

$$E_{vw}(\boldsymbol{I}_{vw}) = \min_{\{\boldsymbol{I}_{uv}\} \in \mathcal{I}_{[vw]}} \left\{ \sum_{u \in \partial v \setminus w} E_{uv}(\boldsymbol{I}_{uv}) \right\} + f(||\boldsymbol{I}_{vw}||). \tag{2.3}$$

The underlying idea of this method is that we can write an exact recursive equation for E_{vw} summing the cost contributions coming from the nodes $u \in \partial v \setminus w$ and from the additional fixed current $\boldsymbol{I}_{vw}$. See figure 2.1b for an intuitive representation of the min-sum recursion (2.3).



(a) Modified cavity graph $\mathcal{G}_{[vw]}$.

(b) min-sum recursion, E_{vw} gets updated using the information in $\{(u, v), (u', v), (u'', v)\} \in \partial v \setminus w$

Figure 2.1: Figurative representation of MP.

Practically, when running the model, equations of the form (2.3) are iterated from an arbitrary initialization of the currents. Once they converge, we can retrieve the optimal configuration $\{\boldsymbol{I}^*_{vw}\}_{(vw) \in \mathcal{E}}$ given by

$$\boldsymbol{I}^*_{vw} = \arg\min_{\boldsymbol{I}_{vw}} \{E_{vw}(\boldsymbol{I}_{vw}) + E_{wv}(-\boldsymbol{I}_{vw}) - f(||\boldsymbol{I}_{vw}||)\}$$

where the last term is subtracted to avoid counting counting twice the edges.

2.1.3 Weighted matching problem mapping

Even if equation (2.3) in theory provides a way of finding the solution the EDP problem, it also introduces a computational bottleneck that we need to overcome. Precisely, to solve

the min-sum recursion one needs to take in account all the possible currents consistent with Kirchoff's law (2.1) and the EDP constraint; the number of such objects grows exponentially with the degree of node v.

In order to reduce the computational complexity of the method we map it to a maximum weight matching problem on an auxiliary weighted fully connected graph $\mathcal{G}'_v$. Precisely, taken a node $v \in \mathcal{V}$ we define $\mathcal{G}'_v$, a complete network which nodes are the neighbors of $v \in \mathcal{V}$ (see figure 2.2). Its weight matrix is a symmetric matrix $\boldsymbol{Q}$ which entries are

$$Q_{wu} := \min_{1 \leqslant |j| \leqslant M} \{E_{wv}(j) + E_{uv}(-j)\} + E_{wv}(0) + E_{uv}(0).$$

Here j parametrizes the $2M+1$ possible values the current vector $\boldsymbol{I}_{wu}$ can take, namely $I^i_{wu} = \delta_{ij}$ for $j > 0$, $I^i_{wu} = -\delta_{ij}$ for $j < 0$ and $I^i_{wu} = 0$ if $j = 0$. In order to initialize the matrix $\boldsymbol{Q}$ we need to do $\mathcal{O}(MK^2)$ operations, with K cardinality of the set of nodes in $\mathcal{G}'_v$.

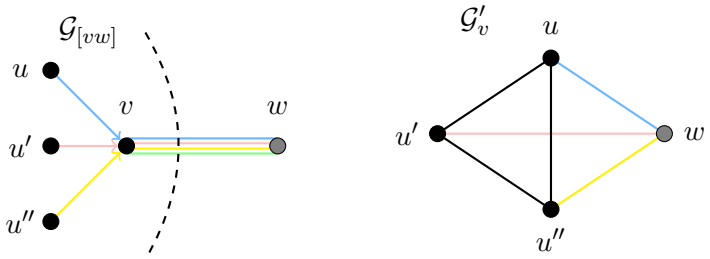


Figure 2.2: Building the auxiliary complete graph $\mathcal{G}'_v$.

This formulation can be further refined by noticing the following. We take a vertex $w \in \partial v$, and we suppose that the communication passing through the edge (v, w) is indexed by i. If we know also the other node where i is passing through, conventionally labeled by u, then the least costly configuration in the remaining part of ∂v is

$$q^{\min}_{uw} = -m_{uw} + \sum_{x \in \partial v \setminus \{u,w\}} E_{xv}(0)$$

where m_{uw} is the maximum weight of a second matching on a complete graph $\mathcal{G}''_{vwu}$ with $k-2$ edges, that we obtain removing w and u together with their incident edges from $\mathcal{G}'_v$. A representation of this construction is given in figure 2.3. The crucial improvement provided by this procedure is that the matching condition, that is: no colored edges in the in $\mathcal{G}''_{vwu}$ can share a node, translates in forbidding edge overlaps. Therefore, thanks to this mapping we are able to reduce the cost of the recursion (2.3) from exponential to polynomial. We may also notice that the minimum weight is $q^{\min}_{uw}$ is independent of $i \in \mathcal{C}$, in fact a configuration of $\mathcal{G}''_{vwu}$ where one of the currents j is equal to i has higher cost and it is naturally eliminated when performing the minimization to compute $q^{\min}_{uw}$. Notice that exploiting this observation we can reduce the computational complexity of the algorithm by a further factor M.

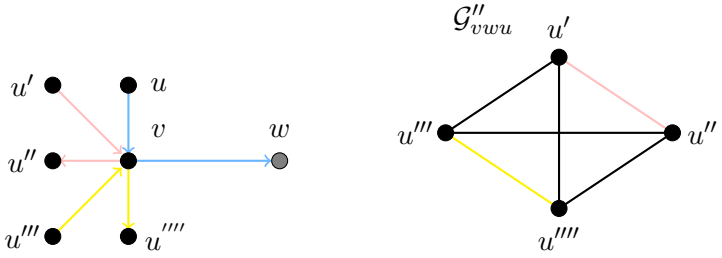


Figure 2.3: Building the auxiliary complete graph $\mathcal{G}''_{vwu}$.

Ultimately, to complete the mapping of E_{vw} we need to minimize over the node u, namely for iteration t of the min-sum recursion (2.3) we have

$$E_{vw}^{t+1}(i) = \min_{u \in \partial v \setminus w} \left\{ E_{uv}^{t}(i) + q_{uw}^{\min} \right\} + c_{vw}(i). \tag{2.4}$$

This last expression has an intuitive interpretation. The update of the message E_{vw} is done by taking in account three contributions: i) the entering (resp. exiting if E_{vw} is entering in v) message over the neighboring nodes $\partial v \setminus w$, in accordance with the underlying rationale of MP for which we update the information of a link using the one entering in v (resp. w), ii) the optimal cost of the remaining neighbor $q_{uw}^{\min}$ and iii) the infrastructure cost of the link (v, w): $c_{vw}(\mu)$.

Finally, we notice that in order to evaluate each term inside the brackets in equation (2.4) we need to perform a matching optimization on every complete graph $\mathcal{G}'''_{vwu}$, $\forall u \in \partial v \setminus w$. Each problem has a complexity of the order $\mathcal{O}(K^3 \log K)$ [16]. Taking in account that we have $\mathcal{O}(K^2)$ possible combinations of v and w, and summing this last contribution to the one given by the construction of $\boldsymbol{Q}$, we get the final complexity of the problem

$$\mathcal{O}(K^5 \log K + MK^2).$$

that is polynomial in both M and K.

A last remark is needed. The mapping procedure can be similarly performed when $i = 0$, i.e. no current is passing through (v, w). In this case a matching on the $(k-1)$-node fully connected graph composed by the nodes $u \in v \setminus w$ is done. Instead, if v is either a source or a sink, i.e. $\Lambda_v^i = \pm 1$ for a certain $i \in \mathcal{C}$, the same computation can be performed using the following expedient. We add an extra auxiliary node to the network, indexed by the communication label i, and connected to v. Then, we initialize at $t = 0$ the message E_{iv} and we never update it. In particular, we set $E_{iv}^0(j) = -\infty$ if $0 < \pm j = i$ (i is resp. a sender or a receiver), and $E_{iv}^0(j) = +\infty$ otherwise.

2.2 Numerical routines

2.2.1 SMP-EDP algorithm

One of the purposes of this thesis is relaxing the MP routine, developing a novel algorithm able to accommodate all the sources in $\mathcal{C}$ at a moderate computational cost. Precisely, in this regard we may notice how the hard constraint set by edge-disjointness limit the MP routine to be able to allocate only a fraction M_{acc} of the M commodities in $\mathcal{C}$,

since when having more and more routes on the network it may be impossible to join a source/sink pairing without overlapping a previously occupied link. Even if w.r.t. many of the state-of-the-art method MP improves performances in term of M_{acc} [5] we want to proceed further, building a scheme able to reach the goal $M_{\text{acc}} = M$ softening the EDP constraint.

The core structure of SMP-EDP is contained in the following pseudocode[1], which explanation this section is devoted to.

Algorithm 4 SMP-EDP algorithm

input: graph topology $\mathcal{G}(\mathcal{V}, \mathcal{E})$
input: number of communications M
input: $\gamma > 0$
do `AssignCouples`$(M, \mathcal{V}) \overset{\text{return}}{\rightarrow} \mathcal{C}$ (M source-sink pairings)
assign traffic to every edge $e \in \mathcal{E}$, $t_e(0) = 0$
assign to every edge $e \in \mathcal{E}$ a cost $c_e(0) = \ell_e(t_e + 1)^\gamma$
associate to each link a binary occupancy $\text{occ}_e = 0$ ($\{0, 1\} = \{$"free", "occupied"$\}$)
while $\mathcal{C}$ `!=` $\varnothing$ **do**
    do `MP`$(\mathcal{G}, \mathcal{C}, \boldsymbol{c}) \overset{\text{return}}{\rightarrow} \{\mathcal{C}', M_{\text{acc}}, \texttt{BestPath}\}$ (in `MP` occ_e is employed to ensure edge-disjointness)
    for $e \in$ `BestPath` **do**
        $t_e \leftarrow t_e + 1$, $c_e \leftarrow \ell_e(t_e + 1)^\gamma$
        $\text{occ}_e \leftarrow 0$
    $\mathcal{C} \leftarrow \mathcal{C} \setminus \mathcal{C}'$

As we may see from the structure of the algorithm, the EDP constraint gets relaxed using this strategy. After initially giving as inputs the graph topology and creating the pairings between source-sink couples, we associate to every edge $e \in \mathcal{E}$ a cost c_e (collected in the array $\boldsymbol{c} \in \mathbb{R}^E$) and a traffic variable t_e. The first quantity differs from the one defined in traditional MP, in this context it has the purpose of controlling traffic congestion. Precisely, in the original MP routine of [5], the cost of each edge is $c_e = \ell_e$. Instead in our model it is essential to update the cost as for MCOT exploiting the traffic occupancy of the network, so that we can, for example, force rerouting if some edges are already occupied.

Even more in detail, we can follow the structure of algorithm (4) step by step. First, the S-S coupling is made with the function `AssignCouples`, then the MP routine is sequentially performed. At each iteration, the MP scheme returns three objects: i) the set of pairs $\mathcal{C}' \subseteq \mathcal{C}$ it has been able to accommodate, ii) the cardinality of this set $|M_{\text{acc}}| = \mathcal{C}'$, and iii) the set of links `BestPath`. This last quantity contains the edges in $\mathcal{E}$ that are used during a single MP execution, their occupancy state is encoded by the binary variable occ_e which allows to perform the accommodation guaranteeing edge-disjointness. After each run of the `MP` function, the traffic t_e and the cost c_e of these links gets updated as in algorithm (3), namely $t_e \leftarrow t_e + 1$ and $c_e \leftarrow \ell_e(t_e + 1)^\gamma$. The occupancy state, in turn, is set again to 0 for every edge $e \in$ `BestPath`. It is by employing two different traffic variables that we can construct the sequential update routine: the "local" (in the sense

[1] For an open-source version of the MP routine used as baseline to build algorithm (4) see: https://github.com/cdebacco/MP_EDP.

that it gets re-initialized at every iteration) quantity occ_e has the only purpose of ensuring no edge overlapping, the "global" traffic occupancy, instead, plays a crucial role in traffic congestion ensuring that changing the exponent γ we can regulate rerouting. The last operation in the cycle is removing those S-S couples that have been accommodated, so that when `MP` is launched again the allocation is performed solely on the remaining commodities.

It is important to notice how after each iteration of MP both the cost and the traffic occupancy of the links gets updated as in the MSG scheme. Precisely, for $\gamma > 1$ the cost of the edges is superlinear w.r.t. the traffic, while for $\gamma < 1$ it is sublinear. Qualitatively this means that for "small" values of the exponent taking the same routes multiple times is favored, up to the limit case of $\gamma = 0$ where path allocation does not influence consecutive iterations. Instead, enlarging γ rerouting is preferred over traffic congestion.

A natural and crucial question that may arise now is why the relaxation of the EDP problem has not been done by simply softening the hard edge-disjoint constraint, running the MP directly updating c_e as $\ell_e(t_e+1)^\gamma$. The answer has been already given in §2.1.3, in fact a necessary requirement to build the weighted matching mapping is edge-disjointness. This condition is indeed translated in not having communications sharing the same node, that is the definition of matching, and it is fundamental to reduce the complexity of the scheme from exponential to polynomial. Therefore the EDP constraint on one side may have the drawback of worsening the results e.g. in terms total path length, but at the same time it is essential to guarantee the solvability of the problem.

2.2.2 Reinforcement

To accelerate convergence we use in each MP routine the reinforcement technique of [5, 6, 11]. This method, to some extent, can be compared to the decimation procedure of [58] where communications gets progressively fixed on links biased towards a certain route assignation when performing accommodation.

This briefly consists in the following construction. At every time iteration t of the MP routine we bias each message using a local external field of the form

$$h^t_{vw}(i) := E^t_{vw}(i) + E^t_{wv}(-i) + c_{vw}(i) \tag{2.5}$$

that tends to align the messages with themselves. The reinforcement is introduced by defining a communication-dependent cost, namely adding the fields h_{vw} to the costs c_{vw} at each time step. Formally

$$c^{t+1}_{vw}(i) = c^t_{vw}(i) + \eta^t h^t_{vw}(i) \tag{2.6}$$

with η^t time-dependent learning rate. This has the effect to lead the routine to converge faster by gradually increasing the magnitude of each external field. In practice, we set η^t as in [5], that is $\eta^t = \rho t$ fixing the growth rate to $\rho = 0.002$ in every numerical experiment.

Improvements given by reinforcement have been extensively studied in several numerical experiments performed with the MP scheme. A portion the analysis of this thesis is devoted to showing how this technique boost performance also in the SMP-EDP framework.

2.2.3 SMSG algorithm

Analogously to the MCOT case, a theoretical baseline exploitable as benchmark for the SMP-EDP scheme is still missing. To overcome this problem we build a second algorithm with which we are able to draw meaningful comparisons in terms of metrics as, for example, traffic congestion and total path length. In particular, we develop what we call a Sequential Multi-Start Greedy (SMSG) routine. This scheme has the same structure of algorithm (4), that is: we iterate accommodation steps where edge-disjointness is imposed, and we relax this hard constraint using a local occupancy occ_e and global traffic variable t_e. The crucial difference between the methods is that while in the previous case at each sequential step we used a MP scheme, here we perform the accommodation via a MSG algorithm. The SMSG scheme is briefly described by the following pseudocode.

Algorithm 5 SMSG algorithm

input: graph topology $\mathcal{G}(\mathcal{V}, \mathcal{E})$
input: number of communications M
input: $\gamma > 0$
do `AssignCouples`$(M, \mathcal{V}) \overset{\text{return}}{\to} \mathcal{C}$ (M source-sink pairings)
assign traffic to every edge $e \in \mathcal{E}$, $t_e(0) = 0$
assign to every edge $e \in \mathcal{E}$ a cost $c_e(0) = \ell_e(t_e + 1)^\gamma$
associate to each link a binary occupancy $\text{occ}_e = 0$ ($\{0, 1\} = \{$"free", "occupied"$\}$)
while $\mathcal{C}$ `!=` $\varnothing$ **do**
    for $n = 1 \leqslant N_{\text{tot}}, n \leftarrow n + 1$ **do**
        do `Greedy`$(\mathcal{G}, \mathcal{C}, \boldsymbol{c}) \overset{\text{return}}{\to} \{\mathcal{C}', M_{\text{acc}}, \texttt{BestPath}\}$ (in `MP` occ_e is employed to ensure edge-disjointness)
        for $e \in \mathcal{E}$ **do**
            if $\text{occ}_e == 1$ **then** C^n_{tot} += c_e
    choose outcome with lower C^n_{tot}
    for $e \in$ `BestPath` **do**
        $t_e \leftarrow t_e + 1$, $c_e \leftarrow \ell_e(t_e + 1)^\gamma$
        $\text{occ}_e \leftarrow 0$
    $\mathcal{C} \leftarrow \mathcal{C} \setminus \mathcal{C}'$

The underlying structure of the method, as said, is identical to the one of scheme (4). First we give as inputs the graph topology and the number of total communications, with which we create the source-sink pairings using the function `AssignCouples`. Secondly we assign to each link in $\mathcal{E}$ a cost c_e and a traffic occupancy t_e. Again, this second quantity is different from the binary variable occ_e that is introduced with the purpose of labeling edges that have been occupied when running the MSG routine, ensuring edge-disjointness. After the initialization we run iteratively a MSG routine, i.e. a Greedy algorithm repeated N_{tot} times, so that we can keep only the best realization in terms of total cost C^n_{tot}. Finally, using the best instance returned from `Greedy` we update traffic and cost as $t_e \leftarrow t_e + 1$ and $c_e \leftarrow \ell_e(t_e + 1)^\gamma$.

The SMSG routine provides and easy-to-use benchmark comparable with the SMP-EDP method. This algorithm, on one side can run at moderate computational cost. On the other hand its performances are strongly effected by the its greedy nature for which global information on edge occupancy is not exploited to maximize M_{acc}. In fact, the

accommodation of each path in `Greedy` is done as in algorithm (3), i.e. sampling at random the S-S couples. This procedure clearly affect performance, and even if we partially remedy this poor behavior by choosing the best path among multiple instances, its performances are still worse than the one of the SMP-EDP scheme.

2.3 Numerical analysis: SMP-EDP vs. SMSG

In this part of the thesis we perform a quantitative comparison between the SMP-EDP and the SMSG method. The analysis is divided in two parts: first we present some experiments that have been performed with the purpose of studying the improvements provided to SMP-EDP by the reinforcement technique described in §2.2.2. As will be clear from these preliminary results, adding reinforcement greatly boost performance and consistently reduce the script running time. For this reason in the second part of the numerical analysis the SMP-EDP model is always executed with reinforcement. The following results are devoted to draw a comparison between the schemes in terms of usual metrics as graph length or traffic occupancy.

Analogously to §1.4 the numerical analysis has been entirely performed on the mesh `blrand_10_1` used in [5]. To each link of the graph we assigned a length $\ell_e = 1 + 10^{-3}\varepsilon$, where the bias $\varepsilon \sim U(0,1)$ has been introduced to facilitate converge, avoiding degenerate cases with multiple paths having the same total cost.

2.3.1 Performance

As described in §2.2.2, we can improve the converge properties of the MCOT method using the following trick. We define a local external field h_{vw} as in equation (2.5) for every $(v,w) \in \mathcal{E}$, having the effect of gradually biasing messages to align with themselves. Inserting these objects in the communication-dependent cost (2.6) results in favoring aligned communications over the others in terms of their cost, speeding-up the accommodation routine.

To give a quantitative proof of the improvements brought by reinforcement we tested the model performing several experiments on `blrand_10_1`, while changing $\gamma \in \{0.0, 0.5, 2.0\}$ and spanning the complexity ratio M/N in the interval $[0.01, 0.1]$. For each run of the accommodation routine we measured the time $[s]$ used by our CPU to process the scripts, in figure 2.4 we display the obtained results.

Looking at the plot we notice that introducing the external fields h_{vw} greatly improves performances. More precisely, we have that for small values, i.e. $M/N = 0.01$, of the complexity ratio the elapsed time does not differ much in the two cases with and without reinforcement. However when increasing M/N, the impact of reinforcement becomes progressively more evident, up to the point where the running time get halved or reduced by 2/3. This behavior is consistent with what we may have expected, that is: for a "simpler" problem where M is low it is reasonable not to have much rerouting, thus introducing h_{vw} does not really have an effect on accelerating convergence. Instead, when we consider a "more difficult" problem, in the sense that M/N is larger, we may encounter frequently a situation in which the preferred route of more than one communication passes through a certain edge. It is in this case that the external fields h_{vw} are fundamental boost convergence.

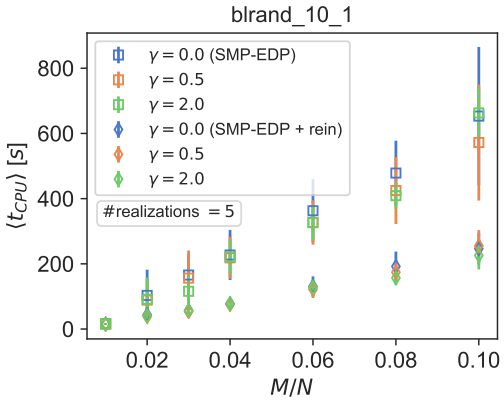


Figure 2.4: $\langle t_{\text{CPU}} \rangle$ vs. M/N. SMP-EDP, mesh: `blrand_10_1`, $N = 500$, $\gamma \in \{0.0, 0.5, 2.0\}$, $E = 1020$. Results averaged over 5 realizations.

To further study the effects of reinforcement we analyze two other quantities. Here, we not only compare different runs of the SMP-EDP model with and without reinforcement, but we test our scheme against the SMSG algorithm, with the purpose of providing a first glimpse on the advantages introduced by the SMP-EDP model over the state-of-the-art.

In particular, we build two plots in which we examine the quantities $\langle$#iterations$\rangle$ vs. M/N and M_{acc}/N vs. iteration, where an "iteration" is a step of the sequential routine in which a single MP or Greedy edge-disjoint accommodation is performed. The purpose of the first figure is to show how increasing the complexity ratio M/N influences the total number of steps needed to converge, as we may notice the ability of MP to take in account traffic occupancy is fundamental to increase the number of accommodated Source-Sink couples M_{acc} at each step, so that the number of iterations gets reduced. This characteristic trait of the SMP-EDP algorithm to maximize M_{acc} by taking in account the global traffic occupancy of the network is displayed in the second plot, indeed here we show how the fraction of allocated commodities changes for consecutive iteration.

The results of our numerical experiments are in the following figures. Each run of the experiments has been executed on `blrand_10_1`, with the ratio M/N spanning in the interval $[0.01, 0.1]$ and setting γ to values in both the sublinear and the superlinear regime, precisely $\gamma \in \{0.0, 0.5, 2.0\}$.

Analyzing figure 2.5 we notice the following. First, as we may have expected, for each of the three models the total number of iterations is monotonically increasing with the complexity M/N. This is consistent with the fact that as we add more commodities to allocate we need more iterations to reach convergence, having less free edges to occupy at every MP or Greedy step. Secondly, we see that the amount of steps used by the SMSG scheme is consistently larger than the one used by the SMP-EDP algorithm. This is a consequence of the capacity of MP of exploiting network traffic maximizing M_{acc} at each step (seen in fig. 2.6). Ultimately, we may notice that the runs of the SMP-EDP scheme in which we applied reinforcement do not differs in terms of total iterations from the ones without it. This fact, even if it may seems counter-intuitive at first sight, is in line with the effect of reinforcement. Precisely, the external fields bias messages in such a way that when having multiple communication selecting the same route a faster selection of the message is made. This, of course, have no effect in increasing M_{acc}, indeed the values of this last quantity is mainly determined by edge-disjoint constraint.

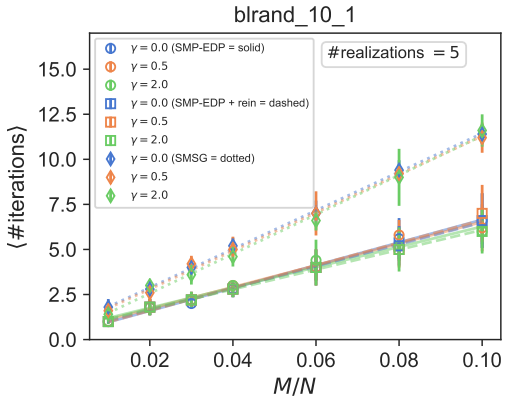
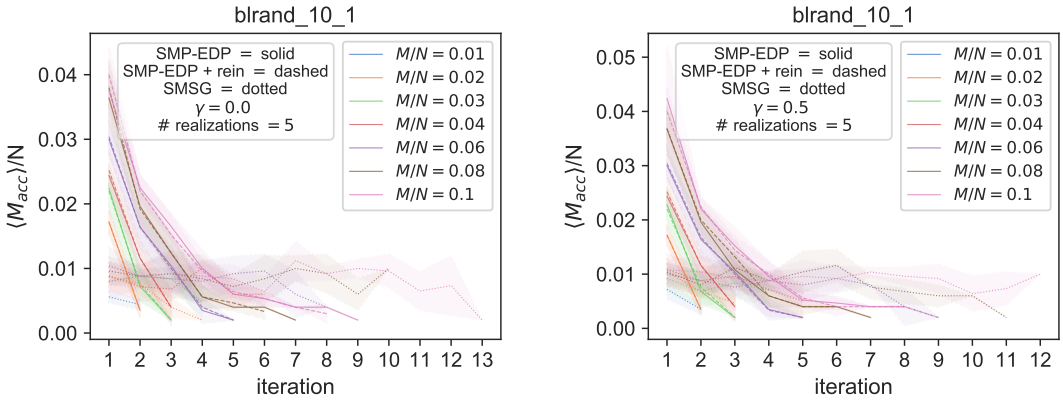
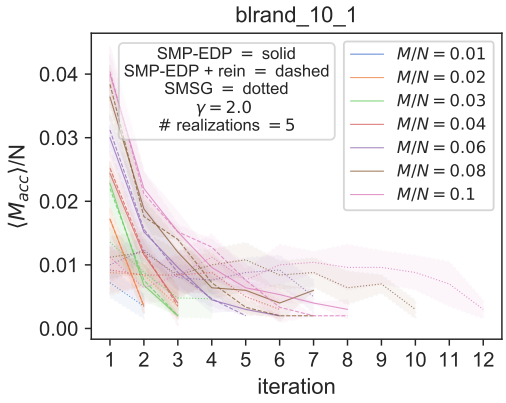


Figure 2.5: $\langle$#iterations$\rangle$ vs. M/N. Mesh: `blrand_10_1`, $N = 500$, $\gamma \in \{0.0, 0.5, 2.0\}$, $E = 1020$. Results averaged over 5 realizations.



(a) $\gamma = 0.0$

(b) $\gamma = 0.5$



(c) $\gamma = 2.0$

Figure 2.6: M_{acc}/N vs. iteration. Mesh: `blrand_10_1`, $N = 500$, $\gamma \in \{0.0, 0.5, 2.0\}$, $E = 1020$. Results averaged over 5 realizations.

In the three panels of figure 2.6 we clearly notice how performance of SMP-EDP in terms of M_{acc} overcome the one of SMSG. First, increasing M/N the number of steps needed to perform the accommodation using a certain fixed model increases. As a reference example may take fig. 2.6a, looking at the solid and dashed lines (correspondent to SMP-EDP with and without reinforcement) we notice how the total number of iterations used to accommodate all M Source-Sink pairings progressively grows from 1 (no lines is shown when $M/N = 0.01$ since one step is sufficient to allocate all the commodities) to 9 when the complexity is equal to $M/N = 0.1$. The same happens for the dotted lines, associated with the SMSG algorithm, which needs a monotonic increasing number of total iterations to perform the accommodation as M/N gets bigger. In this case, however, more steps are necessary to run the routine under the same value of M/N, w.r.t. the SMP-EDP algorithm. This behavior is consistent with what we found in figure 2.5. Moreover, we see that the solid and dashed curves rapidly decay over consecutive iterations. This clearly exhibit the capability of MP to accommodate as much Source-Sink pairings as possible. In turn, the dotted curves tend to be more flat, this trend is symptomatic of the fact Greedy routines do not perform this maximization. Instead, when running MP we sequentially sample the commodities to allocate uniformly from $\mathcal{C}$, in doing that some "bad" choices for two consecutive indexes often happen, meaning that we may select one Source-Sink pair which path blocks the second one, causing M_{acc} to decrease.

From this plots it is strongly evident that our algorithm provides a consistent improvement in performance over SMSG, which is commonly used as state-of-the-art scheme. Lastly, in force of the benefits given by reinforcement, in the rest of the analysis the SMP-EDP routine will be always executed with reinforcement to speed-up convergence.

2.3.2 Accommodation speed

A first preliminary plot that we may show is, again, M_{acc}/N vs. iteration. Its behavior and its justification remain the same we just discussed. However, in the following figure we keep M/N fixed while changing γ. For visualization purpose we fade all the curves but one, since the qualitative trend remains the same in each plot, independently of γ.

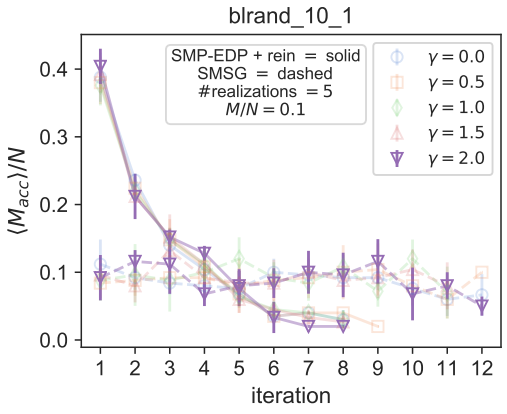
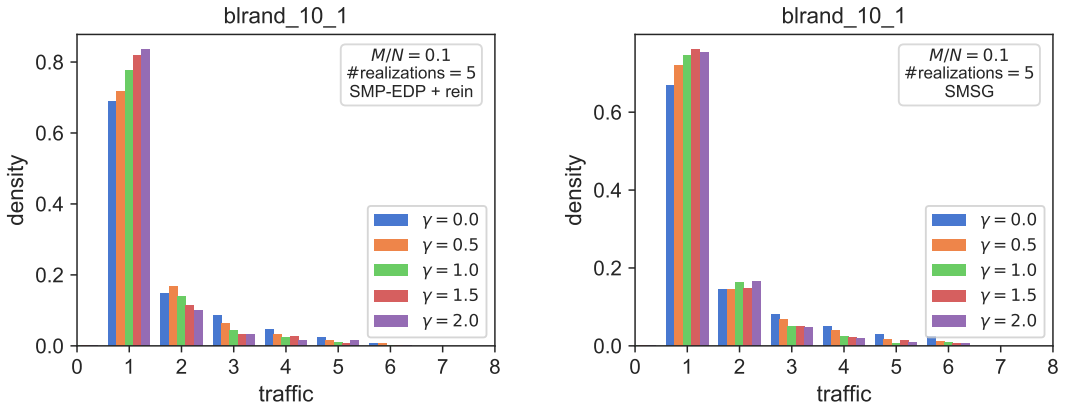


Figure 2.7: M_{acc}/N vs. iteration. Mesh: `blrand_10_1`, $N = 500$, $M = 50$, $\gamma \in \{0.0, 0.5, 1.0, 1.5, 2.0\}$, $E = 1020$. Results averaged over 5 realizations. Similar plots are in B.1a and B.1b in appendix B.

2.3.3 Traffic

In this part of the analysis we build meaningful metrics able to show how the SMP-EDP model outperforms SMSG in terms of traffic optimization over the network. First, we compare two histograms, one for each algorithm, which measure the traffic distribution over the network, i.e. they have as entries the number of commodities occupying every link in $\mathcal{E}$. Secondly we plot the efficiency, already defined in equation (1.9), vs. γ with the purpose of seeing how changing the exponent enforces or penalize traffic congestion.

Our numerical results are in the following figures. All the experiments were performed on the mesh `blrand_10_1` with γ ranging in the interval $\{0.0, 0.5, 1.0, 1.5, 2.0\}$ and $M/N \in \{0.06, 0.08, 0.1\}$.



(a) SMP-EDP **(b)** SMSG

Figure 2.8: Histograms showing the traffic distribution on $\mathcal{E}$ for different values of γ. Mesh: `blrand_10_1`, $N = 500$, $M = 50$, $E = 1020$, $\gamma \in \{0.0, 0.5, 1.0, 1.5, 2.0\}$. Results averaged over 5 realizations. Similar plots for $M = 30, 40$ are in figs. B.1c, B.1e and figs. B.1d, B.1f in appendix B.

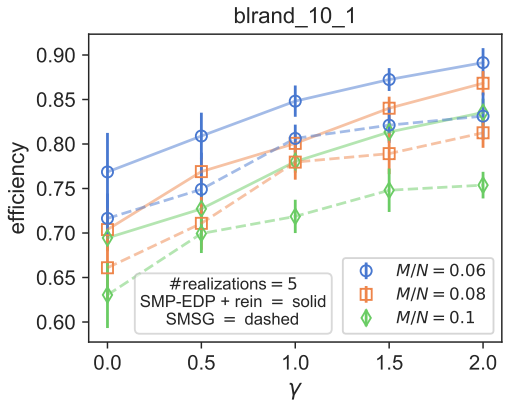


Figure 2.9: efficiency vs. γ. Mesh: `blrand_10_1`, $N = 500$, $M = 30, 40, 50$, $E = 1020$. Results averaged over 5 realizations.

Looking at figure 2.8a we see that, as the exponents grows, more edges tend to

avoid traffic congestion. This reflects on the fact that bins clustered around 1 are monotonically increasing with γ, while for values of the x-axis greater than 1 they are almost monotonically decreasing. Looking at 2.8b we notice the same qualitative behavior. However, comparing the two models we may see how the SMP-EDP model is able to manage traffic congestion outperforming the benchmark scheme. In fact, the tail of the histogram is larger in fig. 2.8b, meaning that we are more likely to obtain edges occupied by a bigger amount of commodities when running the SMP-EDP routine.

From figure 2.9 we can notice two particular features. First, all the curves shows an almost exact monotonic increasing behavior, consistent with the effect that γ has on rerouting. Also, increasing the complexity M/N we reduce the efficiency under the same value of γ, along with the principle under which adding more commodities we are more likely to observe overlaps. Other than that, we notice how the SMP-EDP scheme always returns larger values of the efficiency for every combination of γ and M/N, meaning that constructing our model we are able to outperform the benchmark routine in terms of traffic management.

2.3.4 Cost

Another metric providing useful insights on the SMP-EDP algorithm is the infrastructure cost, defined as

$$C := \sum_{e \in \mathcal{E}_t} c_e \tag{2.7}$$

with $\mathcal{E}_t := \{e \in \mathcal{E} \,|\, t_e > 0\}$. In particular, we are interested in the relation between this quantity and γ and to exhibit that we construct two plots. The first shows how the ratio $\Delta C_n / M_{\text{acc}}$ changes at each iteration of the algorithm, n; where labeling sequential steps with the subscript n, we define the cost variation

$$\Delta C_n := C_n - C_{n-1},\ C_0 \equiv 0.$$

Here C_n is the total cost given by the edges with $t_e > 0$ at time step n. The second quantity is C/M, this plot serves as a further verification to the capability of SMP-EDP of optimizing traffic.

The results of our experiments are in figures 2.10 and 2.11. The simulations have been performed on the usual benchmark mesh `blrand_10_1`.

From the first figure we notice that for increasing values of γ the infrastructure cost derivative changes from being flat to being increasing. This trend is consistent with the definition of $c_e = \ell_e(t+1)^\gamma$, that is a power law w.r.t γ. Other than that, we see that the solid curves, associated with SMP-EDP, attain lower values than the SMSG (dashed) ones under the same values of γ. This is a consequence of the capability of MP of better managing traffic congestion maximizing M_{acc}. Moreover, M_{acc} maximization leads to faster convergence i.e. fewer sequential iterations needed by the SMP-EDP scheme to accommodate all the routes, analogously to what seen in figure 2.7.

Figure 2.11 reflects definition (2.7), from which we see that the total cost is nothing but a summation of exponential functions in γ. In the plot we notice how the total infrastructure cost per communication is consistently smaller for the SMP-EDP scheme over SMSG. Once again, this is a consequence of the effectiveness of SMP-EDP in reducing traffic.

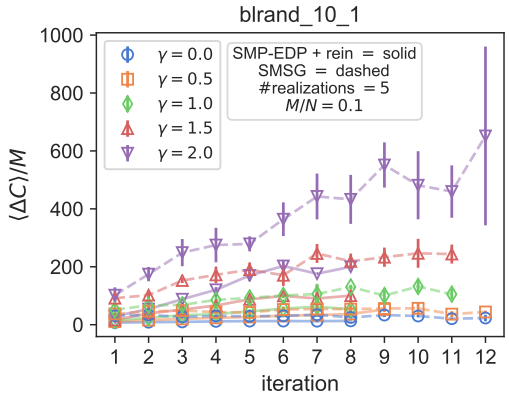


Figure 2.10: $\langle \Delta C/M_{\text{acc}} \rangle$ vs. iteration. Mesh: `blrand_10_1`, $\gamma \in \{0.0, 0.5, 1.0, 1.5, 2.0\}$, $E = 1020$. Results averaged over 5 realizations. Similar plots are in figs. B.1g and B.1h in appendix B.

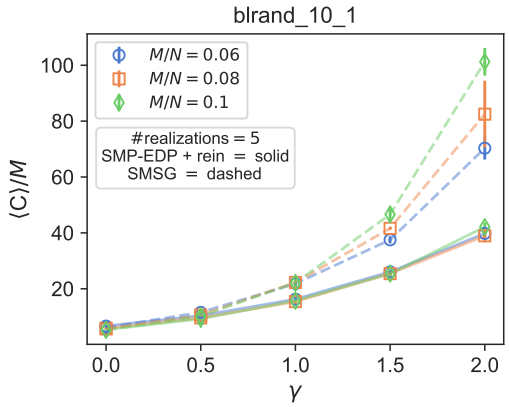


Figure 2.11: $\langle C \rangle$ vs. γ. Mesh: `blrand_10_1`, $N = 500$, $M = 30, 40, 50$, $E = 1020$. Results averaged over 5 realizations.

2.3.5 Idle edges

Analogously to what we have done in §1.4 we compute the idle edges fraction defined as in (1.10). This quantity is plotted against γ in figure 2.12. The simulations were performed on `blrand_10_1`.

Looking at the plot we may notice how all the curves are monotonically decreasing with γ, this behavior is consistent with what we should expect. In fact for larger values of the exponent rerouting is favored, reason why under the same complexity M/N more edges gets occupied. Furthermore, also in this case the SMP-EDP model outperforms SMSG. Its improvement in performance is particularly evident for bigger values of γ. In this sense it is clear that for $\gamma \in \{0, 0.5\}$, when traffic congestion is preferred over rerouting, the capacity of optimizing traffic of SMP-EDP does not take part in accommodation. Instead, when entering the linear/superlinear regime (meaning $\gamma = 1/\gamma > 1$), this characteristic feature of MP is highlighted by higher values of the free edges fraction.

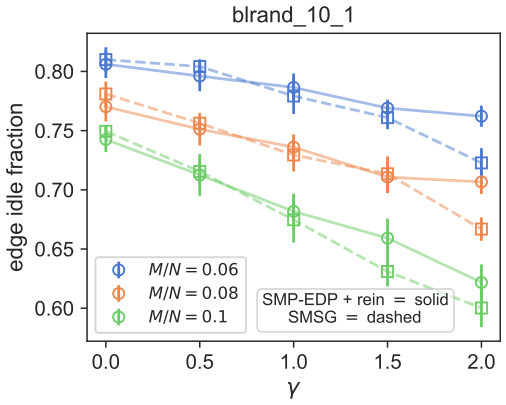


Figure 2.12: idle edge fraction vs. γ. Mesh: `blrand_10_1`, $N = 500$, $M = 30, 40, 50$, $E = 1020$. Results averaged over 5 realizations.

2.3.6 Total length

In this last part of the analysis we conclude the confrontation of the two models by studying the total length of the graph. First, we start by constructing a plot in which we compare the cumulative length, i.e. the total length of the network at each step, vs. the iteration number. From this plot, displayed in figure 2.13, we notice how for every value of the exponent γ the SMP-EDP scheme outperforms the SMSG algorithm, allocating every source-sink pairing using routes with smaller lengths.

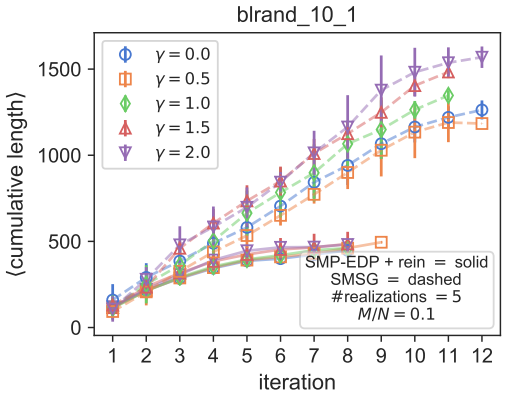


Figure 2.13: ⟨cumulative length⟩ vs. iteration. Mesh: `blrand_10_1`, $M/N = 0.1$, $E = 1020$. Results averaged over 5 realizations. Similar plots for $M/N = 0.06$ and $M/N = 0.08$ in figs. B.1i and B.1j in appendix B.

Ultimately we study the ratio L/L_0, that is the fraction between the length of the graph

$$L := \sum_{e \in \mathcal{E}} \ell_e t_e$$

and the dimension of the mesh L_0. This quantity is plotted against the exponent γ and the complexity M/N. In figure 2.14 the plot L/L_0 vs. γ is shown, looking

at the curves we can observe that the length of the network, for small values of the complexity M/N, remains almost constant. An inflection can be seen for $\gamma \geqslant 1$ and $M/N \in \{0.08, 0.10, 0.12\}$, meaning that rerouting takes place for "large enough" values of the complexity ratio. Moreover, for the same M/N vales, the SMP-EDP model returns consistently smaller length than SMSG.

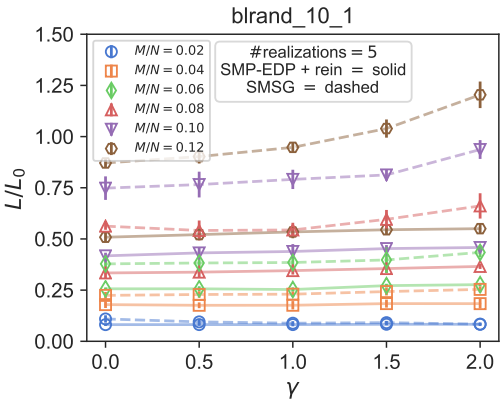


Figure 2.14: L/L_0 vs. γ. Mesh: `blrand_10_1`, $N = 500$, $E = 1020$. Results averaged over 5 realizations.

In figure 2.15 we see how, for both models, the total length of the graph increases as M/N grows. However, with the SMP-EDP scheme we are able to consistently reduce L/L_0. It is interesting to notice also how SMSG is greatly reduces performance when rerouting is forced. Indeed, the dashed lines rapidly grow when enlarging γ. On the contrary SMP-EDP is able to better manage traffic, maintaining the difference between each L/L_0 curve moderate for small and large values of γ.

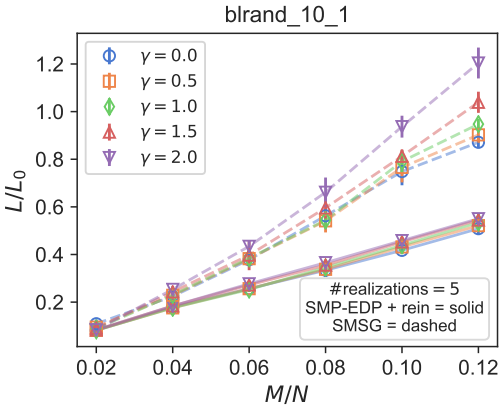


Figure 2.15: L/L_0 vs. M/N. Mesh: `blrand_10_1`, $N = 500$, $E = 1020$. Results averaged over 5 realizations.

Chapter 3

Comparison of the models

In this third chapter of the thesis we draw a qualitative comparison between MCOT and SMP-EDP. This discussion has the purpose of highlighting differences and similarities in the problems so that we can understand precisely on what terms the two algorithms match up.

It is clear that at the current state both our models are in an embryonic stage and not yet directly comparable, being in the first place the settings of the problems they are intended to solve different. In this respect, the numerical analysis done in §§1.4 and 2.3 has been designed with the twofold purpose of both highlighting typical traits of each model and to find common meaningful metrics, so that a preliminary comparison could be performed. Here we want to examine how this analysis can be refined in the future, proposing ideas constituting a possible baseline for following works.

3.1 Two different settings

A first issue that it is necessary to bring to light is the different nature of the two routing problems we are tackling. On one side we have the One Source-Multiple Sinks setting, where mass in-flowing from each commodity can outflow from any other sink of the network. On the other side, we are dealing with the One Source-One Sink problem where we make a one-to-one pairing between sources and sinks. A schematic representation of the OS-OS accommodation process is done in figure 3.1, here we see that to each commodity (orange squares) are associated a positive/in-flowing and negative/out-flowing mass contribution. Moreover, while the accommodation is performed every node can redirect its entering mass to any other sink in the graph (see for example step 1 where the pink colored in-flowing mass exits the network from two separate nodes). In figure 3.2, instead, for every source (square) of a certain color, i.e. communication index, there has to be only one other sink (diamond) colored in the same way to which we can send mass. Indeed, as we see from step 1, mass can only be rerouted in such a way that no source and sinks of two different colors can be the starting point and the endpoint of a route.

Both frameworks are interesting to study per sé and, as extensively remarked in the first part of the thesis, and are widely employed in several practical applications coming from disparate contexts. However, it is reasonable to expect that the outcome of the accommodation routine performed in one case cannot be directly matched with the other. To some extent, constructing the S-S pairs can be thought as an additional constraint

added to the One Source-Multi Sinks formulation. Indeed, this often entails producing longer routes since there is no a-priori reason for which the (random) pairings should agree with the goal of our schemes to optimize cost. A first improvement in this sense could be done adapting the MCOT model the OS-OS setting, in such a way that each source has a one-to-one matching with a sink, or equivalently by relaxing MP to the OS-MS framework.

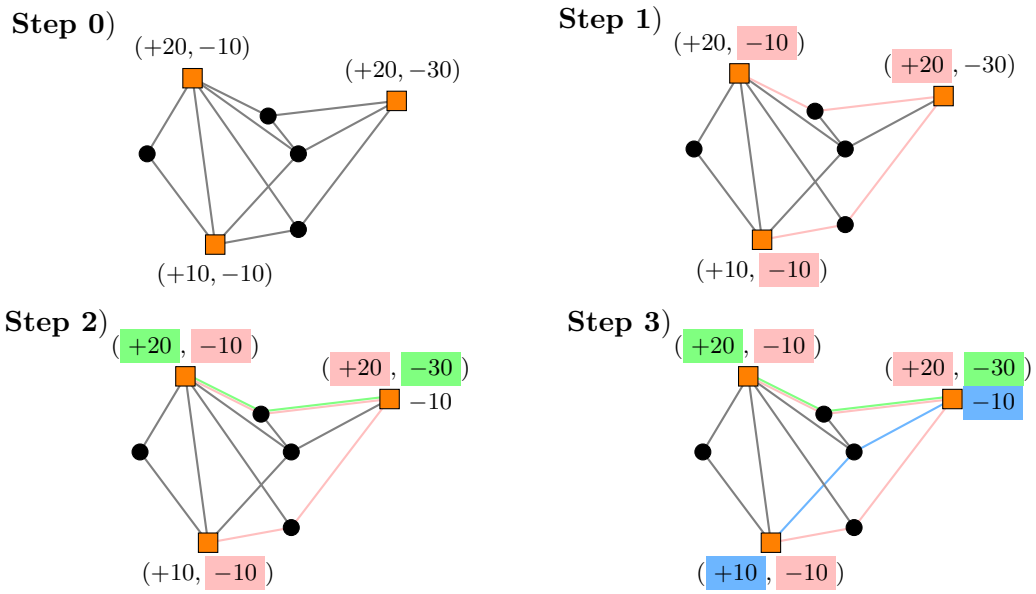


Figure 3.1: Schematic representation of the OS-MS accommodation process.

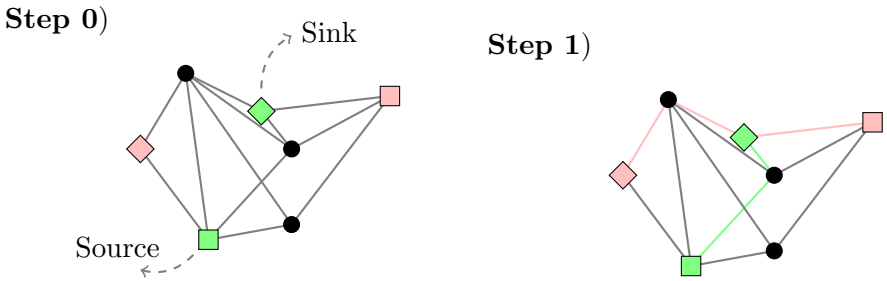


Figure 3.2: Schematic representation of the OS-OS accommodation process.

Furthermore, networks returned by the two models are not directly comparable in terms of traffic occupancy. In particular, the way traffic is encoded in the MCOT algorithm is drastically different from the modelization done in the MP context. In the first case traffic congestion is taken in account by pressure gradients, which are responsible for the regulation of transport density. Instead in the MP setting we measure traffic on a certain edge by simply counting the number of routes passing through that link.

Regarding traffic management it is interesting to see how contrasting approaches are used by the two models to perform the same task. Specifically, the MCOT algorithm controls traffic by "forgetting" commodities labeling, i.e. the evolution in time of $\hat{\mu}_e$

is carried out merging in one single variable all the separate μ_e^i contributions given by each commodity $i \in \mathcal{C}$. Formally this is done with the ansatz (1.8b) that, as already pointed out in §1.2.2, serves to group the conductivity terms coming from every source, transforming $\hat{\mu}_e$ into a global variable. On the contrary in order to manage traffic congestion, for the SMP-EDP routine it is necessary to separate each S-S pairing. In this second setting the traffic variable t_e measures exactly the number of different paths overlapping on a certain edge $e \in \mathcal{E}$ over all the consecutive iterations of algorithm (4).

To quantitatively compare the two models in terms of traffic we could directly act on equation (1.8b), modifying the choice of the function $f(\{\mu_e^i\}^i)$ used to merge the transport density contributions, with the purpose of distinguishing each commodity in $\mathcal{C}$. A meaningful choice of f seems to be everything but trivial, since the new ansatz should be able to both separate each μ_e^i term, while merging them together to regulate congestion. In this thesis we made the simplest possible ansatz choice, setting $f \equiv \mathrm{Id}$, however proposing alternatives for f could provide many useful insights on the accommodation process.

A first naive approach to measure μ_e^i separately for each $i \in \mathcal{C}$ is exploiting pressure gradients. Following the idea that the growth of the transport density $\hat{\mu}_e$ of a certain link $e \in \mathcal{E}$ is regulated by the ℓ^2-norm of the vector containing the M pressure gradient acting on link e. This result is what we represented in figures 1.5b−1.5e, where we gave a first intuitive representation of the effect that each commodity has in rerouting mass on the edges of the network, showing how the mesh topology is crucial in determining the solution of the routine.

3.2 Length and traffic

As remarked several times all along the thesis the SMP-EDP and the MCOT models have been designed with the purpose of optimizing mass rerouting while efficiently managing traffic congestion. In this sense, a quantitative analysis on how these two quantities relate to each other could be further extended by trying to measure how favoring total length optimization is affecting traffic, or vice versa.

Theoretical results backing this type of metrics are still missing, meaning that given a certain graph topology we do not know in principle which is the optimal way to perform the accommodation process while penalizing or favoring the traffic. It is for this reason that we can only study the total length/traffic trade-off using benchmark algorithms. In detail, on one side of the "total traffic vs. optimal length" balance we have selfish algorithms of Dijkstra type, where the mass in-flowing from each source gets rerouted to a sink taking the shortest path, without taking in account the remaining commodities. On the other hand, we can use as a benchmark algorithm to minimize traffic congestion the SMP-EDP routine itself, setting $\gamma \to +\infty$. Meaning that we are increasing γ as much as possible (trying to favor rerouting) while still maintaining $M_{\mathrm{acc}} = M$. Notice that in the theoretical limit $\gamma = +\infty$ we would have SMP-EDP "=" MP with the limitation of not being able to connect each source-sink pairing.

This analysis could provide a novel perspective in terms of understanding where our models are placed in the length/traffic trade-off. We could see how effectively modifying the exponents γ and β contributes on managing traffic; that is, we could exactly measure how much we are reducing traffic at the expense of forcing longer routes, or how much we are allowing more trafficked edges by minimizing the total length of the network.

3.3 Practical applications

Ultimately, to complete this final discussion we mention the wide range of practical application our schemes are in principle naturally able to model. In the introduction of the thesis it has been strongly remarked how OT and MP are widely employed in the current literature as a paradigm for efficiently solve routing problems stemming from any field of science and engineering. Applications range from railway and highway networks, to blood circulation networks, or routing protocols for package transportation in telecommunication. Tackling all these problems with the SMP-EDP and the MCOT algorithms could provide on one hand improvements to the performance of state-of-the-art. On the other side it could serve as a way to deeply understand the rationale on which our routines are built upon, providing many insights on the physical intuition behind the problem formulations.

Conclusions

In this thesis we studied two novel approaches for solving routing problems on networks. The first method has been built using tools coming from optimal transport; in particular we extended the model proposed in Facca et. al. [15] to the Multiple Sources-Multiple Sinks setting, where mass can flows in and out the network from multiple nodes. In §§1.1 and 1.2 we first presented the key quantities entering in the mathematical formulation of the problem, then we coupled these objects into the adaptation equations (1.8a−1.8c) which entirely characterize the dynamics of the model. In parallel with the mathematical formulation of the problem, we constructed the MCOT algorithm (§1.3), that has been tested against a benchmark MSG routine in an extensive numerical analysis in §1.4. Our experiments displayed how our model outperforms the state-of-the-art in several respects. Precisely, we designed different metrics to display that with MCOT we are able to efficiently perform mass rerouting naturally taking in account length optimization and traffic congestion. In this regard, the MCOT showed a great adaptability in both the sublinear and the superlinear regime, being able to strongly penalize or enforce traffic by regulating the exponent β.

Secondly, we proposed a method built upon the Message Passing technique. In detail, in §2.1 we introduced the theory developed in [5], in §2.2 we extended the model to a Sequential Message Passing scheme (SMP-EDP). Our algorithm has the advantage of MP to take in account the traffic occupancy of all the edge in the network, and is able to improve the results obtained with the state-of-the-art MP routines by relaxing the EDP constraint. The scheme has been tested against a SMSG algorithm in §2.3, here the metrics used to evaluate performance showed that our model outperforms most of the methods used by practitioners nowadays. Indeed, building a sequential scheme iteratively performing MP we were able to find a route joining *all* the M source/sink couples, while optimizing total length and traffic at a moderate computational cost.

As remarked in chapter 3 the algorithms, other than providing many advancements to the current literature, hint at future developments and open many new perspectives for routing problems. In particular, our models are a first incipient modelization of two different settings: the OS-MS framework and the OS-OS one. Future works could be devoted to formulate a refined version of the models, with the MCOT algorithm working in the OS-OS framework or, vice versa, with the SMP-EDP in the OS-MS one. Furthermore, the numerical analysis presented in this thesis could be extended in many ways, a possible direction could be the study of the total length/traffic trade-off, which analysis could let us explore the effect of the exponents β and γ on traffic management. Ultimately, the algorithms could be employed in modeling real networks, giving several useful insights about problems belonging to disparate branches of science and engineering.

Bibliography

[1] Adamatzky, A. *Physarum Machines: Computers from slime mold*. World Scientific, 2010. 10.1142/7968.

[2] Adamatzky, A. Route 20, autobahn 7, and slime mold: approximating the longest roads in usa and germany with slime mold on 3-d terrains. *IEEE transactions on cybernetics 44*, 1 (2013), 126–136. arXiv:1211.0519.

[3] Adamatzky, A., and de Oliveira, P. P. B. Brazilian highways from slime mold's point of view. *Kybernetes 40*, 9/10 (2011), 1373–1394. 10.1108/03684921111169440.

[4] Adamatzky, A., and Ilachinski, A. Slime mold imitates the united states interstate system. *Complex Systems 21*, 1 (2012), 1.

[5] Altarelli, F., Braunstein, A., Dall'Asta, L., De Bacco, C., and Franz, S. The edge-disjoint path problem on random graphs by message-passing. *PloS one 10*, 12 (2015). arXiv:1503.00540.

[6] Altarelli, F., Braunstein, A., Realpe-Gomez, J., and Zecchina, R. Statistical mechanics of budget-constrained auctions. *Journal of Statistical Mechanics: Theory and Experiment 2009*, 07 (2009), P07002. arXiv:0903.2429.

[7] Banavar, J., Damuth, J., Maritan, A., and Rinaldo, A. Ontogenetic growth: Modelling universality and scaling. *Nature 420*, 6916 (2002), 626–627. 10.1038/420626a.

[8] Banavar, J., Maritan, A., and A., R. Size and form in efficient transportation networks. *Nature 399*, 6732 (1999), 130–132. 10.1038/20144.

[9] Baveja, A., and Srinivasan, A. Approximation algorithms for disjoint paths and related routing and packing problems. *Mathematics of Operations Research 25*, 2 (2000), 255–280. 10.1287/moor.25.2.255.12228.

[10] Blesa, M. J., and Blum, C. Finding edge-disjoint paths in networks: An ant colony optimization algorithm. *Journal of Mathematical Modelling and Algorithms 6*, 3 (2007), 361–391. 10.1007/s10852-007-9060-y.

[11] Braunstein, A., and Zecchina, R. Learning by message passing in networks of discrete synapses. *Physical review letters 96*, 3 (2006), 030201. arXiv:cond-mat/0511159.

[12] Carmi, P., Erlebach, T., and Okamoto, Y. Greedy edge-disjoint paths in complete graphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science* (2003), Springer, pp. 143–155. 10.1007/978-3-540-39890-5_13.

[13] Deepankar, M., and Karthikeyan, R. *Network Routing (Second Edition)*. Morgan Kaufmann, 2018. 10.1016/C2013-0-18604-7.

[14] Evans, L. C. *Partial differential equations and Monge-Kantorovich mass transfer*. CDM, 1997.

[15] Facca, E., Cardin, F., and Putti, M. Physarum dynamics and optimal transport for basis pursuit, 2018. arXiv:1812.11782.

[16] Galil, Z., Micali, S., and Gabow, H. An o(ev\logv) algorithm for finding a maximal weighted matching in general graphs. *SIAM Journal on Computing 15*, 1 (1986), 120–130. 10.1137/0215009.

[17] Gao, C., Yan, C., Wei, D., Hu, Y., Mahadevan, S., and Deng, Y. A biologically inspired model for transshipment problem. *arXiv preprint arXiv:1401.2181* (2014). arXiv:1401.2181.

[18] Guy, R. D., Nakagaki, T., and Wright, G. B. Flow-induced channel formation in the cytoplasm of motile cells. *Physical Review E 84*, 1 (2011), 016310. 10.1103/PhysRevE.84.016310.

[19] Hagberg, A. A. H., Schult, D. A., and Swart, P. J. Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th Python in Science Conference (SciPy2008), Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA)* (2008), pp. 11–15.

[20] Hsu, C.-C., and Cho, H.-J. A genetic algorithm for the maximum edge-disjoint paths problem. *Neurocomputing 148* (2015), 17–22. 10.1016/j.neucom.2012.10.046.

[21] Jia, G.-L., Ma, R.-G., and Hu, Z.-H. Review of urban transportation network design problems based on citespace. *Mathematical Problems in Engineering 2019* (2019). 10.1155/2019/5735702.

[22] Kantorovich, L. V. On the translocation of masses. *C.R. (Doklady) 321* (1942), 199–201.

[23] Kolliopoulos, S. G., and Stein, C. Approximating disjoint-path problems using greedy algorithms and packing integer programs. In *International Conference on Integer Programming and Combinatorial Optimization* (1998), Springer, pp. 153–168. 10.1007/3-540-69346-7_12.

[24] Li, K., Thomas, K., Torres, C., Rossi, L., and Shen, C.-C. Slime mold inspired path formation protocol for wireless sensor networks. In *Swarm Intelligence* (Berlin, Heidelberg, 2010), M. Dorigo, M. Birattari, G. A. Di Caro, R. Doursat, A. P. Engelbrecht, D. Floreano, L. M. Gambardella, R. Groß, E. Şahin, H. Sayama, and T. Stützle, Eds., Springer Berlin Heidelberg, pp. 299–311. 10.1007/978-3-642-15461-4_26 .

[25] Loh, R. C., Soh, S., Lazarescu, M., and Rai, S. A greedy technique for finding the most reliable edge-disjoint-path-set in a network. In *2008 14th IEEE Pacific Rim International Symposium on Dependable Computing* (2008), IEEE, pp. 216–223. 10.1109/PRDC.2008.16.

[26] Manohar, P., Manjunath, D., and Shevgaonkar, R. Routing and wavelength assignment in optical networks from edge disjoint path algorithms. *IEEE Communications Letters 6*, 5 (2002), 211–213. 10.1109/4234.1001667.

[27] Martín, B., Sánchez, Á., Beltran-Royo, C., and Duarte, A. Solving the edge-disjoint paths problem using a two-stage method. *International Transactions in Operational Research 27*, 1 (2020), 435–457. 10.1111/itor.12544.

[28] Medina, A., Lakhina, A., Matta, I., and Byers, J. Brite: An approach to universal topology generation. In *MASCOTS 2001, Proceedings Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems* (2001), IEEE, pp. 346–353. 10.1109/MASCOT.2001.948886.

[29] Mézard, M., and Montanari, A. *Information, physics, and computation*. Oxford University Press, 2009.

[30] Mézard, M., and Parisi, G. The bethe lattice spin glass revisited. *The European Physical Journal B-Condensed Matter and Complex Systems 20*, 2 (2001), 217–233. arXiv:cond-mat/0009418.

[31] Mézard, M., Parisi, G., and Virasoro, M. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, vol. 9. World Scientific Publishing Company, 1987. 10.1142/0271.

[32] Mézard, M., Parisi, G., and Zecchina, R. Analytic and algorithmic solution of random satisfiability problems. *Science 297*, 5582 (2002), 812–815. 10.1126/science.1073287.

[33] Miyaji, T., Ohnishi, I., Tero, A., and Nakagaki, T. Failure to the shortest path decision of an adaptive transport network with double edges in plasmodium system. *International Journal of Dynamical Systems and Differential Equations 1* (07 2008). 10.1504/IJDSDE.2008.019683.

[34] Monge, G. *Mémoire sur la théorie des déblais et des remblais*. Del'Imprimerie Royale, 1781.

[35] Moy, J. T. *OSPF: anatomy of an Internet routing protocol*. Addison-Wesley Professional, 1998.

[36] Nakagaki, T., and Guy, R. D. Intelligent behaviors of amoeboid movement based on complex dynamics of soft matter. *Soft Matter 4*, 1 (2008), 57–67. 10.1039/B706317M.

[37] Nakagaki, T., Tero, A., Kobayashi, R., Onishi, I., and Miyaji, T. Computational ability of cells based on cell dynamics and adaptability. *New Generation Computing 27*, 1 (2008), 57–81. 10.1007/s00354-008-0054-8.

[38] Nakagaki, T., Yamada, H., and Tóth, Á. Maze-solving by an amoeboid organism. *Nature 407*, 6803 (2000), 470–470. 10.1038/35035159.

[39] Nakagaki, T., Yamada, H., and Ueda, T. Interaction between cell shape and contraction pattern in the physarum plasmodium. *Biophysical chemistry 84*, 3 (2000), 195–204. 10.1016/S0301-4622(00)00108-3.

[40] Noronha, T. F., Resende, M. G., and Ribeiro, C. C. A biased random-key genetic algorithm for routing and wavelength assignment. *Journal of Global Optimization 50*, 3 (2011), 503–518. 10.1007/s10898-020-00877-0.

[41] Nowé, A., Verbeeck, K., and Vrancx, P. Multi-type ant colony: The edge disjoint paths problem. In *International Workshop on Ant Colony Optimization and Swarm Intelligence* (2004), Springer, pp. 202–213. 10.1007/978-3-540-28646-2_18.

[42] Quanrud, K. Approximating optimal transport with linear programs. *arXiv preprint arXiv:1810.05957* (2018). arXiv:1810.05957.

[43] Queyroi, F. Biological and shortest-path routing procedures for transportation network design, 2018. arXiv:1803.03528.

[44] Santambrogio, F. Optimal channel networks, landscape function and branched transport. *Interfaces and Free Boundaries 9*, 1 (2007), 149–169. 10.4171/IFB/160.

[45] Shekhawat, V. S., Tyagi, D. K., and Chaubey, V. Weight based edge disjoint path routing and wavelength assignment (wedp-rwa) algorithm for wdm networks. In *2008 IEEE Region 10 and the Third international Conference on Industrial and Information Systems* (2008), IEEE, pp. 1–5. 10.1109/ICIINFS.2008.4798409.

[46] Sumpter, Z., Burson, L., Tang, B., and Chen, X. Maximizing number of satisfiable routing requests in static ad hoc networks. In *2013 IEEE Global Communications Conference (GLOBECOM)* (2013), IEEE, pp. 19–24. 10.1109/GLOCOM.2013.6831041.

[47] Tero, A., Kobayashi, R., and Nakagaki, T. Physarum solver: A biologically inspired method of road-network navigation. *Physica A: Statistical Mechanics and its Applications 363*, 1 (2006), 115–119. 10.1016/j.physa.2006.01.053.

[48] Tero, A., Kobayashi, R., and Nakagaki, T. A mathematical model for adaptive transport network in path finding by true slime mold. *Journal of theoretical biology 244*, 4 (2007), 553–564. 10.1016/j.jtbi.2006.07.015.

[49] Tero, A., Takagi, S., Saigusa, T., Ito, K., Bebber, D. P., Fricker, M. D., Yumiki, K., Kobayashi, R., and Nakagaki, T. Rules for biologically inspired adaptive network design. *Science 327*, 5964 (2010), 439–442. 10.1126/science.1177894.

[50] Varadarajan, K., and Venkataraman, G. Graph decomposition and a greedy algorithm for edge-disjoint paths. In *SODA* (2004), Citeseer, pp. 379–380.

[51] Watanabe, S., and Takamatsu, A. Transportation network with fluctuating input/output designed by the bio-inspired physarum algorithm. *PloS one 9*, 2 (2014). 10.1371/journal.pone.0089231.

[52] Watanabe, S., Tero, A., Takamatsu, A., and Nakagaki, T. Traffic optimization in railroad networks using an algorithm mimicking an amoeba-like organism, physarum plasmodium. *Biosystems 105*, 3 (2011), 225–232. 10.1016/j.biosystems.2011.05.001.

[53] Xia, Q. Optimal paths related to transport problems. *Communications in Contemporary Mathematics 5*, 02 (2003), 251–279. 10.1142/S021919970300094X.

[54] Xia, Q. An application of optimal transport paths to urban transport networks. *Conference Publications 2005*, Special (2005), 904.

[55] Xia, Q. Motivations, ideas and applications of ramified optimal transportation. *ESAIM: Mathematical Modelling and Numerical Analysis 49*, 6 (2015), 1791–1832. 10.1051/m2an/2015028.

[56] Xia, Q., Croen, L. A., Fallin, M. D., Newschaffer, C. J., Walker, C., Katzman, P., Miller, R. K., Moye, J., Morgan, S., and Salafia, C. Human placentas, optimal transportation and high-risk autism pregnancies. *Journal of Coupled Systems and Multiscale Dynamics 4*, 4 (2016), 260–270. 10.1166/jcsmd.2016.1110.

[57] Xin, C., Ye, Y., Dixit, S., and Qiao, C. A joint working and protection path selection approach in wdm optical networks. In *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No. 01CH37270)* (2001), vol. 4, IEEE, pp. 2165–2168. 10.1109/GLOCOM.2001.966164.

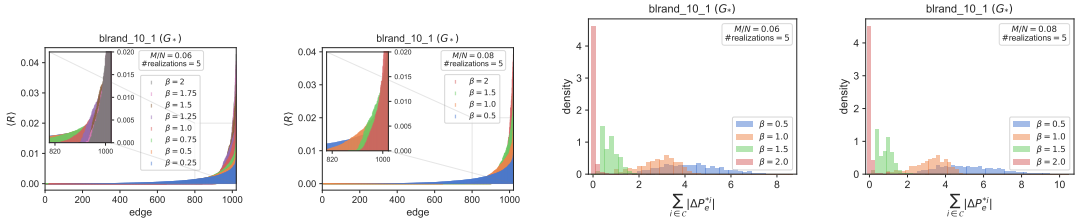
[58] Yeung, C. H., Saad, D., and Wong, K. M. From the physics of interacting polymers to optimizing routes on the london underground. *Proceedings of the National Academy of Sciences 110*, 34 (2013), 13717–13722. 10.1073/pnas.1301111110.

[59] Zeng, K., Ren, K., and Lou, W. Geographic on-demand disjoint multipath routing in wireless ad hoc networks. In *MILCOM 2005-2005 IEEE Military Communications Conference* (2005), IEEE, pp. 1–7. 10.1109/MILCOM.2005.16058430.

[60] Zhang, X., Chan, F. T., Adamatzky, A., Mahadevan, S., Yang, H., Zhang, Z., and Deng, Y. An intelligent physarum solver for supply chain network design under profit maximization and oligopolistic competition. *International Journal of Production Research 55*, 1 (2017), 244–263. 10.1080/00207543.2016.1203075.

Appendix A

Additional simulations: MCOT



(a) see fig. (1.2), $M = 30$. **(b)** see fig. (1.2), $M = 40$. **(c)** see fig. (1.3), $M = 30$. **(d)** see fig. (1.3), $M = 40$.

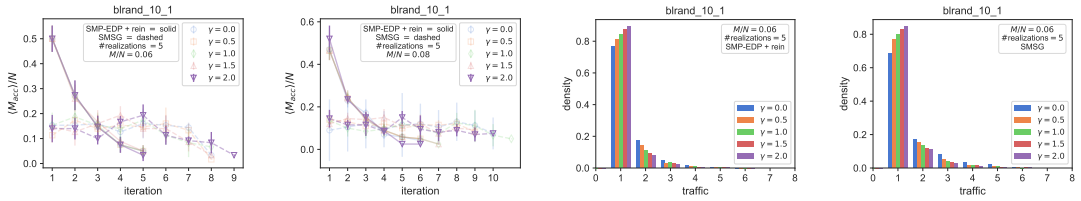
(e) see fig. (1.4a), $M = 40$. **(f)** see fig. (1.4b), $M = 40$. **(g)** see fig. (1.4c), $M = 40$. **(h)** see fig. (1.4d), $M = 40$.

(i) see fig. (1.5), $M = 30$. **(j)** see fig. (1.5), $M = 40$. **(k)** see fig. (1.6), $M = 30$. **(l)** see fig. (1.6), $M = 40$.

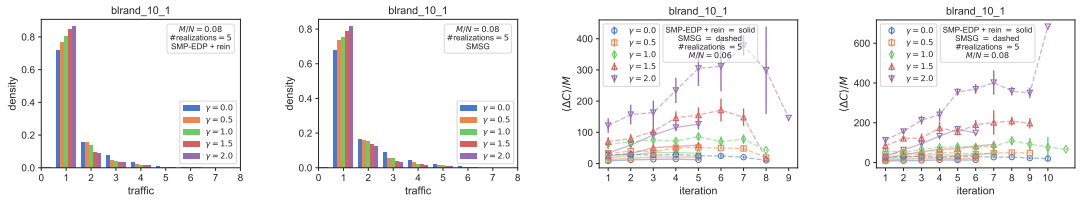
Figure A.1: Additional figures for §1.4.

Appendix B

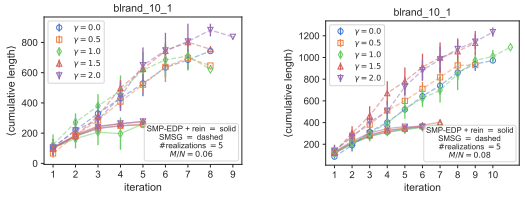
Additional simulations: SMP-EDP



(a) see fig. (2.7), $M = 30$. **(b)** see fig. (2.7), $M = 40$. **(c)** see fig. (2.8a), $M = 30$. **(d)** see fig. (2.8b), $M = 30$.



(e) see fig. (2.8a), $M = 40$. **(f)** see fig. (2.8b), $M = 40$. **(g)** see fig. (2.10), $M = 30$. **(h)** see fig. (2.10), $M = 40$.



(i) see fig. (2.10), $M = 30$. **(j)** see fig. (2.10), $M = 40$.

Figure B.1: Additional figures for §2.3.