



HAL
open science

Robust optimization for network design

Yacine Al Najjar

► **To cite this version:**

Yacine Al Najjar. Robust optimization for network design. Computer Science [cs]. Institut Polytechnique de Paris, 2022. English. NNT : 2022IPPAS014 . tel-04704058

HAL Id: tel-04704058

<https://theses.hal.science/tel-04704058v1>

Submitted on 20 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT
POLYTECHNIQUE
DE PARIS

NNT : 2022IPPAS014

Thèse de doctorat



Robust Optimization for Network Design

(Optimisation Robuste pour la Conception de Réseaux)

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom SudParis

École doctorale n°626 Ecole doctorale de l'Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat : Mathématiques et informatique

Thèse présentée et soutenue à Palaiseau, le 9 novembre 2022, par

YACINE AL-NAJJAR

Composition du Jury :

Arie Koster Professor at RWTH Aachen University, Germany	Rapporteur
Joanna Tomasik Professor at CentraleSupélec	Rapporteur
Neil Olver Associate Professor at London School of Economics and Political Science, England	Examineur
Adam Ouorou Research director at Orange Labs	Examineur
Michael Poss Research director at CNRS Montpellier	Examineur
Jérémie Leguay Research director at Huawei Technologies	Co-encadrant
Walid Ben-Ameur Professor at Telecom SudParis	Directeur de thèse

RESUME ETENDU EN FRANCAIS

Étant donnée la nature dynamique du trafic, nous étudions la variante du problème de dimensionnement robuste de réseaux qui consiste à déterminer la capacité à réserver sur chaque lien d'un réseau de telle sorte que chaque demande appartenant à un polytope donné puisse être routée. L'objectif est soit de minimiser la congestion soit un coût linéaire.

Dans le chapitre 2, nous étudierons l'approximabilité de la variante avec un routage fractionnaire et dynamique (i.e., le routage peut dépendre du scénario actuel de la demande) dans des graphes non dirigés. En utilisant le théorème PCP et une réduction bien choisie, nous prouverons tout d'abord que le problème de minimisation de la congestion ne peut être approché en dessous d'aucun facteur constant. Ensuite, en utilisant la conjecture ETH, nous prouverons une borne inférieure de $\Omega(\log n / \log \log n)$ sur l'approximabilité de ce problème. Cela implique que le célèbre taux d'approximation $O(\log n)$ de Räcke est presque optimal. En utilisant la méthode de la relaxation Lagrangienne, nous obtiendrons une nouvelle preuve de cette borne. Une conséquence importante de cette réduction basée sur la relaxation Lagrangienne est que les résultats d'inapproximabilité prouvés pour le problème de minimisation de la congestion peuvent être transférés au problème de réservation de capacité avec un coût linéaire. En particulier, cela nous permet de répondre à une question ouverte de Chekuri (2007). Cela nous permet également de d'obtenir une nouvelle preuve d'un résultat de Goyal et al (2009) montrant que la solution optimale pour le problème de coût linéaire avec un routage statique peut être $\Omega(\log n)$ plus chère qu'une solution avec un routage dynamique. Nous étudierons ensuite la complexité du problème lorsque les chemins sont donnés en entrée du problème. Nous montrerons en particulier que si seulement deux chemins sont donnés en entrée, alors le problème reste tout de même difficile à approcher en dessous d'une certaine (petite) constante.

Dans le chapitre 3, nous porterons notre attention sur la variante avec un graphe dirigé. Nous montrerons qu'une solution avec un routage statique optimal donne une $O(\sqrt{k}) = O(n)$ -approximation du routage dynamique optimal, où k est le nombre de commodités et n le nombre de noeuds. Cela améliore une borne prouvée précédemment par Hajiaghayi et al. (2005) et coïncide, à un facteur constant près, avec la borne inférieure de $\Omega(n)$ prouvée par Ene et al. (2016) et la borne inférieure de $\Omega(\sqrt{k})$ prouvée par Azar et al. (2003). Enfin nous étudierons une généralisation du problème où la demande de chaque commodité ne peut être routée que sur un sous ensemble d'arcs donnés en entrée. En particulier, nous montrerons que ce problème ne peut être approché en dessous d'un facteur de $k^{\frac{c}{\log \log k}}$ pour une certaine constante c (resp. $2^{\log^{1-\epsilon} k}$

pour tout $\epsilon > 0$) sauf si $NP \subseteq SUBEXP$ (resp. $NP \subseteq QP$).

Dans le chapitre 4 nous étudierons plusieurs reformulations du problème de dimensionnement robuste de réseaux permettant d'améliorer la méthode de routage affine. Nous montrerons tout d'abord que la formulation par noeud-arc peut être moins restrictive que la formulation par arc-chemin. Nous fournirons également une formulation naturelle équivalente à la formulation par noeud-arc. Nous étudierons ensuite plusieurs formulations basées sur des relaxations des contraintes de conservation de flot. Enfin, nous étudierons des formulations basées sur des agrégations de commodité par source ou par destination. Comme attendu, cela permet de réduire la taille de la formulation et donc d'améliorer les temps de résolution. Un résultat plus surprenant encore est que cela peut aussi permettre d'améliorer la qualité de la solution. Pour chacune de ces formulations, nous présenterons des exemples simples montrant que ces formulations peuvent améliorer la qualité de la solution. Nous présenterons également des résultats expérimentaux montrant que chacune de ces formulations permet d'améliorer la qualité de la solution sur des instances réalistes.

Enfin dans le chapitre 5 nous proposerons une stratégie intermédiaire entre l'approche statique et l'approche dynamique pour s'approcher encore plus dynamique tout en contrôlant la complexité. Il s'agira d'une approche qu'on pourrait qualifier de multi-statique. L'idée sera de proposer de choisir un ensemble de faces du polytope représentant l'ensemble d'incertitude de telle manière que l'union des ces faces contienne tous les points extrêmes non-dominés de cet ensemble. Ensuite, un routage statique serait envisagé pour chacune de ces faces. Nous verrons que cette approche est modulaire puisque, d'une part, en prenant comme faces les points extrêmes, on obtient l'approche dynamique, et d'autre part, si on prend comme face l'intégralité du polytope, on obtient l'approche statique. Nous proposerons quelques mécanismes pour générer les faces d'une manière itérative afin de baisser les coûts en évitant l'explosion combinatoire correspondant au cas dynamique. Deux polytopes seront plus particulièrement étudiés, le polytope du type budget et le polytope du type hose.

Mots Clés: *Optimisation robuste, Dimensionnement de réseaux, Approximation, Inapproximabilité.*

ABSTRACT

Given the dynamic nature of traffic, we investigate the robust network design problem where we have to determine the capacity to reserve on each link so that each demand vector belonging to a polyhedral set can be routed. The objective is either to minimize congestion or a linear cost. When routing is assumed to be fractional and dynamic (i.e., dependent on the current traffic vector), we first prove that the robust network design problem with minimum congestion cannot be approximated within any constant factor, settling an open question by Chekuri (2007). Then, using the ETH conjecture, we get a $\Omega(\frac{\log n}{\log \log n})$ lower bound for the approximability of this problem. We next focus on the variant in which the underlying graph is directed. We prove that an $O(\sqrt{k}) = O(n)$ -approximation can be obtained by solving the problem under static routing, where k is the number of commodities and n is the number of nodes. We show that a natural generalization of the problem cannot be approximated within a ratio of $k^{\frac{c}{\log \log k}}$ for some constant c (resp. $2^{\log^{1-\epsilon} k}$ for any $\epsilon > 0$) unless $NP \subseteq SUBEXP$ (resp. $NP \subseteq QP$).

Affine routing can be used to obtain better solutions also with polynomial-time algorithms. It consists in restricting the routing to depend on the demands in an affine way. We first show that a node-arc formulation is less conservative than an arc-path formulation. We also provide a natural cycle-based formulation that is shown to be equivalent to the node-arc formulation. To further reduce the solution's cost, several new formulations are proposed. They are based on the relaxation of flow conservation constraints. The obtained formulations have been further improved through aggregation. As might be expected, aggregation allows us to reduce the size of formulations. A more striking result is that aggregation reduces the solution's cost. We finally propose an intermediate strategy between static and dynamic routing that can be seen as a new variant of multi-static routing. We consider some faces of the uncertainty set whose union contains all non-dominated extreme points of the polytope. Then a static routing is considered for each of these faces.

Keywords: *Robust Optimization, Network Design, Approximation, Inapproximability.*

TABLE OF CONTENTS

RESUME ETENDU EN FRANCAIS	i
ABSTRACT	iii
1 Introduction	1
1.1 Optimizing under uncertainty	2
1.2 Robust network design problems	4
1.3 Variants with flow restrictions	7
1.3.1 Routing over a single path	7
1.3.2 Routing over a given set of paths	8
1.3.3 Edge restricted routing	9
1.3.4 K-splittable routing	9
1.3.5 Tree routing	9
1.4 Review of traffic uncertainty sets	10
1.4.1 Budget model	10
1.4.2 Hose model	10
1.4.3 Set of all routable demands	10
1.5 Known relationships between problem variants	11
1.5.1 From undirected graphs to directed graphs	11
1.5.2 Node-capacited and edge-capacited instances in directed graphs	11
1.5.3 From congestion to linear cost	12
1.6 Intermediate strategies between static and dynamic routing	13
1.6.1 Affine routing	13
1.6.2 Multipolar routing	14
1.6.3 Multi-static routing	15
1.7 Known theoretical results on complexity and approximability	16
1.7.1 Fractional routing	16

1.7.2	Other variants with flow restrictions	17
1.7.3	Results for the hose model	18
1.8	Related problems	18
1.8.1	Other variants on robust network design	18
1.8.2	Two-stage adaptive optimization	19
1.9	Thesis contributions	20
1.9.1	Inapproximability results in the undirected case	20
1.9.2	Approximability and inapproximability results in the directed case	21
1.9.3	Affine routing	22
1.9.4	Multi-static routing	23
1.10	List of publications	23
2	Robust network design problem in undirected graphs	24
2.1	Inapproximability under some constant factor $c > 1$	25
2.2	Inapproximability in undirected graphs	26
2.2.1	A $\Omega(\frac{\log n}{\log \log n})$ approximability lower bound	29
2.3	From minimum congestion to linear cost	30
2.4	Inapproximability with restrictions on a set of given paths	33
3	Robust network design problem in directed graphs	36
3.1	Preliminaries	37
3.2	Gap between static and dynamic routing strategies	39
3.3	Inapproximability for the problem with edge restrictions	44
4	Affine routing for the robust network design problem	50
4.1	Possible affine formulations	51
4.1.1	Initial node-arc formulation	51
4.1.2	Arc-path formulation	52
4.1.3	Elementary cycle-based formulation	54
4.2	Relaxing the flow conservation constraints	56
4.3	A cut-based formulation	58
4.4	Extended graph formulation	62
4.5	Aggregation	64
4.5.1	Aggregation for $\mathcal{F}_=$	65

4.5.2	Sink aggregation for \mathcal{F}_+	67
4.5.3	Source aggregation for \mathcal{F}_-	68
4.6	Solution Methods and Numerical Evaluation	69
4.6.1	Uncertainty sets	69
4.6.2	Problem solving methods	70
4.6.3	Network instances	72
4.6.4	Numerical results	73
4.7	Conclusion	77
5	A new multi-static routing approach	79
5.1	Decomposition of the uncertainty set	79
5.2	Some preliminary numerical experiments	82
6	Conclusion and perspectives	84
6.1	Summary of contributions	84
6.2	Thesis outlook	85
	REFERENCES	86

Appendices

Appendix A	A modular approach for solving robust design problems	95
A.1	Abstract representation of a robust optimization problem	96
A.2	Demand polytope representation	97
A.2.1	Cutting plane procedure	97
A.3	Automatic compact reformulation procedure	97

CHAPTER 1

Introduction

Network optimization [Wang et al. (2008), Luna (2006)] plays a crucial role for telecommunication operators allowing them to carefully invest in their infrastructure, i.e. reducing capital expenditures. As Internet traffic is ever increasing, the network capacity needs to be expanded through careful investments every year or even half-year.

Beyond traditional carrier networks that build and maintain their own physical infrastructure, Over-The-Top (OTT) operators are also building wide area overlay networks to obtain worldwide, long-haul and cost effective services. These companies are renting and reselling bulk capacity from transit operators and Internet eXchange Points (IXP). For instance, SD-WAN (Software-Defined Wide Area Networks) operators build high-performance and low cost WANs (Wide Area Networks) for enterprises based on Software-Defined Networking [Michel and Keller (2017)] technologies by efficiently mixing expensive private lines with low cost Internet access. Other OTT such as cloud service providers or online platforms (e.g., video streaming, social networks) are leasing a mixture of connectivity services to interconnect central and regional data centers. In all cases, network connectivity, whether leased or owned, has a cost and needs to be carefully designed to optimize profits.

In this thesis, we consider the robust network design problem to efficiently size telecommunication networks and optimize the use of resources. This problem consists in routing traffic demands between different pairs of sources and destinations of a network while taking into account the uncertainty on the traffic. We consider two variants of the problem. The first kind consists in choosing the capacities on the edges so that all envisioned traffic matrices can be routed. This can model the problem of buying and installing physical equipment or of renting virtual capacity to a network service provider. Several capacity models can be considered. For example, we can choose between a discrete set of capacities with different costs. In this case, it can be modeled using a piece-wise constant objective function. It has been for instance used by Facebook [Ahuja et al. (2021)] for the planning of their backbone network between data centers. In this thesis, we focus on a simpler variant where we minimize a linear cost of the chosen capacities. Algorithms for this simple problem can be used as subroutine to

solve more complex problems.

The other variant of the robust network design problem consists in minimizing the congestion in the network. This is motivated by the fact that highly congested networks have poorer performance in terms of latency. Several objective functions can be considered. For example, queuing theory can be used to give an approximation of the latency/serving time as a function of the load of the network equipment [Ben-Ameur and Ouorou (2006), Hijazi et al. (2013)]. Here we focus on the simple variant of minimizing the maximum congestion over all edges. A similar approach has been for instance used, also by Facebook [Kumar et al. (2018)], for the engineering of traffic.

Due to the diversity of applications generating traffic and the inherent difficulty in predicting user behaviours and traffic variations over time, it is difficult in practice to obtain an accurate estimation of traffic in advance.

Another source of uncertainty can come from possible failures of network equipments. If the traffic is underestimated when planning capacity, the performance of the network from high congestion induces losses and latency. Another use case is private networks [Duffield et al. (1999)] for enterprise services where a company buy connectivity between several offices and the headquarter. The capacity is bought using an access bandwidth (i.e. we only know upper bounds on the traffic that can be sent and received for each node). In this case, the goal for the network operator is to ensure that most or all traffic realization can be sent through a VPN (Virtual Private Network). The exact traffic scenarios are difficult to anticipate.

A naive solution to cope with uncertainty on traffic is to overestimate it. However, this can lead to high costs by using more network equipments. Finer models that directly take into account the uncertainty can provide the best of both world: having a low cost and, at the same time, an efficient network.

In the rest of this chapter we first review some general techniques to model uncertainty. Then we formally present the main robust network design problem on which the thesis focuses. We provide further details on problem variants with flow restrictions and different types of polyhedral traffic demands. We explain the relationships between some variants of robust network design and we present some of the main routing strategies that can be seen as an intermediate between fully-dynamic routing and static routing. We review the main theoretical results related to robust network design and we briefly present other robust network optimization problems of interest. Finally, the main contributions of the thesis are summarized.

1.1 Optimizing under uncertainty

We will now describe some possible solutions to directly integrate demand uncertainty into our models. One classical method to model uncertainties is *stochastic optimization*

introduced by [Dantzig (1955)], see [Infanger (1992)] for a comprehensive overview. This could be applied to our problem by considering several demand scenarios with an assigned probability and then by minimizing the average cost over those scenarios, for example the congestion. This type of approach has two main drawbacks, the first one is that obtaining precise estimation on the distribution of the demand can be challenging, the second one is that the size of the model increases with the number of scenarios considered and it can quickly lead to intractable models. Another alternative is to use *chance constrained optimization* introduced in [Charnes et al. (1958)] see [Prékopa (2013)] for an extensive review. Constraints (for example the edge capacity constraint in our context) are here required to be satisfied only with some specific probability. Solving a chance constrained optimization problem directly poses many challenges: the problem is not easy to solve, and the feasibility region of such problem is usually non-convex. To overcome those computational difficulties, approximations are often used to solve chance constrained problems. One of the methods to approximately solve chance constrained problem is *robust optimization* introduced in [Soyster (1973)]. In robust optimization the coefficients of the constraints can take any value inside a given uncertainty set and the goal is to find a solution that is feasible for any possible realisation of the coefficients. [Ben-Tal and Nemirovski (2000)] study the probabilistic guarantees on constraint violations that can be derived from robust optimization. They show that under some conditions related to the distribution of the coefficients, using an ellipsoidal uncertainty provides a robust solution satisfying the constraints with some prescribed high probability. In a similar vein, [Bertsimas and Sim (2004)] introduce the *budget* uncertainty set. They show that, under some assumptions on the distribution of coefficients of a constraint, the *budget* uncertainty set leads to solution that is feasible with high probability.

A more general way to model uncertainty is *distributional robust optimization* where instead of considering a single probability distribution on the coefficients of the objective function and constraints an *ambiguity* set is considered. [Scarf (1958)] first introduced this approach for the news vendor problem. See [Rahimian and Mehrotra (2022)] for a survey on distributional robust optimization. Distributional robust optimization can be seen as generalized stochastic optimization and chance constrained optimization by taking the ambiguity set to contain only a single probability measure. It can also be seen as a generalization of robust optimization by taking the ambiguity set to be all distributions with support in a given set.

In this thesis we will model demand uncertainty using robust optimization. We will consider that the set of possible demand scenarios is a polytope. The polyhedral set appears naturally, for example a known upper bound on the sum of some demands leads straightforwardly to a linear constraint. Other families can also be considered, for

example, ellipsoids or more general convex sets. We then impose that the set of edge capacities is feasible for all possible demand scenarios for the linear cost problem. In one of the problem variants, we minimise the worst case congestion over all demand scenarios.

1.2 Robust network design problems

We will now introduce some notations and precisely define several variants of the network design problem, see [Ben-Ameur et al. (2012), Chekuri (2007)] for surveys on related problems and the lecture notes of [Olver (2018)]. All the problems that we will consider have as input a graph $G = (V, E)$ (either directed or undirected). A set of commodities \mathcal{H} , where each commodity has a source $s(h)$ and destination node $t(h)$ in the graph G . The decision we must take is how to route the demand. This is specified by giving for each commodity $h \in \mathcal{H}$ and demand scenario $d \in \mathcal{D}$ a flow vector $x_h(d) \in \mathbb{R}^E$ of value 1 between the source $s(h)$ and the destination of the commodity h , i.e., a vector satisfying the following flow conservation constraints:

$$\sum_{e \in \delta_+(v)} x_{h,e}(d) - \sum_{e \in \delta_-(v)} x_{h,e}(d) = \begin{cases} 1, & \text{if } v = s(h) \\ -1, & \text{if } v = t(h) \\ 0 & \text{otherwise} \end{cases} \quad \forall h \in \mathcal{H}, v \in V, d \in \mathcal{D} \quad (1.1a)$$

The flow sent on edge e for commodity h in a demand scenario $d \in \mathcal{D}$ is then $f_h(d) := d_h x_h(d)$.

If the routing can be adapted freely to each demand scenario we call it *dynamic* routing. On the opposite side we have *static* routing where we must choose, independently of the demand scenario $d \in \mathcal{D}$ a flow x_h of value 1 for each commodity. Some intermediate solutions can also be considered between those two extremes. We will present some of them in Section 1.6. The routing is called *fractional* if the flow can use several paths without particular restrictions. In Section 1.3 we will present several variants of the problem with some restrictions on the flow.

Given a routing f , let $u \in \mathbb{R}^E$ be the vector whose components u_e are the maximum flow that is sent (i.e. capacity used) on edge e , i.e., $u_e = \max_{d \in \mathcal{D}} \sum_{h \in \mathcal{H}} f_{h,e}(d)$. We will consider two objective functions depending on u . The first one is the *linear cost* objective where we have as input a cost $\lambda_e \geq 0$ for each edge $e \in E$ and the goal is to minimize $\text{obj}(u) = \sum_{e \in E} \lambda_e u_e$. The other objective function that we will consider is the *congestion* where we have as input the capacity $c_e \geq 0$ of each edge $e \in E$ and the goal is to minimize $\text{obj}(u) = \max_{e \in E} \frac{u_e}{c_e}$. Depending on if the routing is static or dynamic

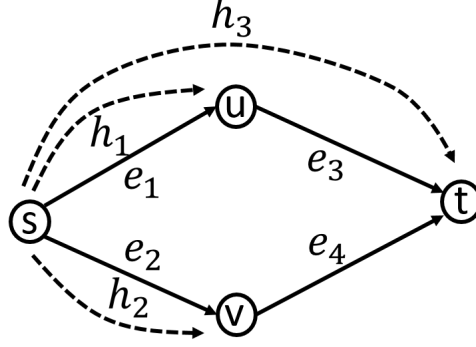


Fig. 1.1 Example of a robust network design instance

and if the objective function is the linear cost or the congestion cost, we have 4 different variants of the problem. We will note those problems $\text{obj}_{\text{adaptivity-frac}}$ where obj can be lin for the linear cost objective or cong for the congestion objective. The keyword adaptivity can be sta for a static routing or dyn for a dynamic routing. Other variants of the robust routing problem can be obtained by considering some restrictions on the flow used to route the demands of the commodities (some of them are presented in Section 1.3). For brevity and because we will mostly consider unrestricted fractional routing, we will also use $\text{obj}_{\text{adaptivity}}$ to denote $\text{obj}_{\text{adaptivity-frac}}$ if it clear from the context that we are considering (unrestricted) fractional routing. We note $\mathcal{U}(\mathcal{D})$ the set of capacities u such that all demand scenarios in \mathcal{D} can be routed without exceeding the capacities.

For illustration we represented in Figure 1.1 a simple example of an instance for the robust network design with 4 nodes s, u, v, t , 4 edges e_1, e_2, e_3, e_4 and \mathcal{H} composed of 3 commodities h_1, h_2, h_3 . The sources of the 3 commodities are all equal to s and the destination of commodity h_1 (resp. h_2, h_3) is u (resp. v, t). We consider the following demand polytope \mathcal{D} :

$$d_{h_1} + d_{h_2} = 1 \tag{1.2a}$$

$$d_{h_3} = 1. \tag{1.2b}$$

We consider the congestion problem on this instance with capacities c_e of 1 for all edges. The demand of commodities h_1 (resp. h_2) can only be routed using the edge e_1 (resp. e_2) because it's the only path from note s to node u (resp. v) in this graph. The congestion cannot be less than 1 because the edge e_1 of capacity 1 must at least carry the demand of commodity h_1 in the scenario where $d_{h_1} = 1$. Under dynamic routing, a solution with congestion of 1 can be obtained by routing $1 - d_{h_1}$ of the demand of commodity h_3 on the path (e_1, e_3) and $1 - d_{h_2}$ unit of flow on the path (e_2, e_4) . On the other hand, under a static routing, the congestion must be at least 1.5. This can be seen by observing that this contains some symmetry: if we exchange u (resp. e_1, e_3, h_1) with v (resp. e_2, e_4, h_2) we obtain an equivalent instance. This implies that there is an

optimal solution that also contains some symmetries: the flow of commodity h_3 must be split equally between path (e_1, e_3) , and path (e_2, e_4) (if there is an optimal solution that sends strictly more than $x > 0.5$ unit of flow of commodity h_3 on the path (e_1, e_3) then by symmetry there is also an optimal solution that sends x unite of flow of commodity h_3 on path (e_2, e_4) , we can then take the average of those solutions).

In summary, we present below a linear programming model for the dynamic routing variant. We will have one variable $x_{e,h}(d)$ for each $e \in E, h \in \mathcal{H}, d \in \mathcal{D}$. Equivalently, we can also see $x_{e,h}$ as a function from the demand uncertainty set \mathcal{D} to \mathbb{R}_+ .

$$\begin{aligned} \min \text{obj}(u) \\ \sum_{e \in \delta_+(v)} x_{h,e}(d) - \sum_{e \in \delta_-(v)} x_{h,e}(d) &= \begin{cases} 1, & \text{if } v = s(h) \\ -1, & \text{if } v = t(h) \\ 0 & \text{otherwise} \end{cases} \\ \forall h \in \mathcal{H}, v \in V & \end{aligned} \quad (1.3a)$$

$$\sum_{h \in \mathcal{H}} d_h x_{h,e}(d) \leq u_e, \quad \forall e \in E, d \in \mathcal{D} \quad (1.3b)$$

$$x_{h,e}(d) \geq 0, \quad \forall e \in E, \forall h \in \mathcal{H}, d \in \mathcal{D} \quad (1.3c)$$

We can also model the problem in terms of variables $f_{h,e}(d)$ (this will be more appropriate to use in the affine routing and multipolar routing cases). However, when we consider a static routing, it will be with respect to formulation (1.3).

The static routing problem can then be formulated as follows:

$$\begin{aligned} \min \text{obj}(u) \\ \sum_{e \in \delta_+(v)} x_{h,e} - \sum_{e \in \delta_-(v)} x_{h,e} &= \begin{cases} 1, & \text{if } v = s(h) \\ -1, & \text{if } v = t(h) \\ 0 & \text{otherwise} \end{cases} \\ \forall h \in \mathcal{H}, v \in V, d \in \mathcal{D} & \end{aligned} \quad (1.4a)$$

$$\sum_{h \in \mathcal{H}} d_h x_{h,e} \leq u_e, \quad \forall e \in E, d \in \mathcal{D} \quad (1.4b)$$

$$x_{h,e} \geq 0, \quad \forall e \in E, \forall h \in \mathcal{H}, d \in \mathcal{D}. \quad (1.4c)$$

Observe that $x_{h,e}$ does not depend on d .

As summary, we will give below the abbreviations for the different problems:

Routing \rightarrow Objective \downarrow	Static	Dynamic
Linear cost	lin_{sta}	lin_{dyn}
Congestion	cong_{sta}	cong_{dyn}

Both the linear cost and congestion problems under static routing variants can be solved in polynomial-time, see [Ben-Ameur and Kerivin (2005)] and Appendix A. On the other hand, as we will see in Section 1.7, those two variants in general become theoretically difficult to solve under dynamic routing. Nonetheless, some methods have been devised to obtain exact solutions [Mattia (2013), Zeng and Zhao (2013), Ayoub and Poss (2016)]. If the capacities are required to be integer the problem becomes more complicated to solve, in this case finding some valide inequalities can help alleviate those difficulties, see [Koster et al. (2013), Raack et al. (2011)].

In the remaining of the manuscript, we refer to the *static/dynamic gap* (or *ratio*) for the congestion (resp. linear cost) problem as the value $\text{cong}_{\text{sta}}/\text{cong}_{\text{dyn}}$ (resp. $\text{lin}_{\text{sta}}/\text{lin}_{\text{dyn}}$).

1.3 Variants with flow restrictions

We now present some additional variants of the problems. Where we have some constraints on the flow of each commodity. Let \mathcal{X}_h be the polytope of flow vectors from $s(h)$ to $t(h)$ of value 1. Suppose that we have as input a subset $\mathcal{X}_h^{\text{allowed}} \subseteq \mathcal{X}_h$ of allowed flow. And we consider the routing problems with the additional constraints $x_h(d) \in \mathcal{X}_h^{\text{allowed}}$. We consider this quite abstract model because it allows us to capture several interesting routing problems and some of the results we have proved on the static/dynamic gap also apply for this more general problem.

This generalised model allows us to model, among others, the *single path routing*, *path restricted routing*, *K-splittable routing*. We will present below those problems.

1.3.1 Routing over a single path

In the single routing problem, each commodity must be routed over a single path i.e. we must have $x_h(d) = \mathbb{1}_p$ for some $s(h) - t(h)$ path, where $\mathbb{1}_p$ is the vector of \mathbb{R}^E whose components are 1 for the edges that are in the path and 0 otherwise. We denote

this problem as $\text{obj}_{\text{adaptivity-sing}}$. Although single path routing usually leads to higher cost solution, it is preferred in practice as all the traffic of an origin-destination flow can be sent in a single tunnel, with MPLS protocol for instance [Paolucci et al. (2013)].

This problem can be modeled as a restricted flow problem by taking \mathcal{X}_h to be the discrete set $\{\mathbb{1}_p \mid p \text{ a path from } s(h) \text{ to } t(h)\}$.

1.3.2 Routing over a given set of paths

In this section we present the path restricted routing problem where demands can only be routed over a set of paths and some of its applications in practice. Although this problem has not been much studied from a theoretical perspective it can have several practical applications, see for example the B4 [Jain et al. (2013)] controller from Google that routes inter-data center traffic over a set of private network connections. Later in this thesis, we will prove several results concerning this problem. In the path restricted routing problem we suppose that we are given as input for each commodity $h \in \mathcal{H}$ a set of allowed path \mathcal{P}_h . And the routing is constrained to be on those paths. Suppose that a commodity has some delay constraint l_h and each edge $e \in E$ has a delay l_e . Then we can define \mathcal{P}_h as the set of paths with end-to-end delay less than l_h i.e.

$$\mathcal{P}_h = \{p \text{ an } (s(h), t(h)) \text{ - path} \mid \sum_{e \in p} l_e \leq l_h\}$$

This problem can also be used to model other constraints on the paths such as precedence constraints where a node v_1 must be visited before a node v_2 . This can be useful for network function chaining [Bhamare et al. (2016)].

These problems can be solved either by preselecting heuristically a small set of paths satisfying those constraints or by generating the paths on the fly in a column generation procedure using a constrained shortest path sub-problem. This can be done quite straightforwardly in the static routing but it can be more tricky in other variants.

We present below a model for this problem.

$$\begin{aligned} & \min \text{obj}(u) \\ & \sum_{h \in \mathcal{H}} \sum_{p \in \mathcal{P}_h \mid p \ni e} d_h x_{h,p}(d) \leq u_e, \quad \forall e \in E, d \in \mathcal{D} \end{aligned} \tag{1.5a}$$

$$\sum_{p \in \mathcal{P}_h} x_{h,p}(d) \geq 1, \quad \forall h \in \mathcal{H}, d \in \mathcal{D} \tag{1.5b}$$

$$x_{h,p}(d) \geq 0, \quad \forall h \in \mathcal{H}, \forall p \in \mathcal{P}_h, d \in \mathcal{D} \tag{1.5c}$$

This problem can be seen as a problem with flow restrictions by taking

$$\mathcal{X}_h^{allowed}(d_h) = \text{conv}_{p \in \mathcal{P}_h} \mathbb{1}_p$$

where $\mathbb{1}_p$ is the vector in \mathbb{R}^H equals to 1 on the edge in the path p and 0 otherwise.

1.3.3 Edge restricted routing

The *edge restricted* routing variant consists in adding the constraint stating that each commodity can only be routed over a given subset of allowed edges $E_h \subseteq E$. These restrictions seem to be quite natural to ensure quality of service requirements such as delay constraints. We will investigate the approximability of this problem in Chapter 3.

1.3.4 K-splittable routing

In the K-splittable routing the flow of each commodity must be on at most K paths, useful for load balancing for example. This problem can be modeled as a network design problem with flow restrictions by taking \mathcal{X}_h to be the non-convex set

$$\left\{ \sum_{i=1}^K \alpha_i \mathbb{1}_{p_i} \mid \forall i, K \geq i \geq 1 : p_i \text{ a path from } s(h) \text{ to } t(h), \sum_{i=1}^K \alpha_i = 1 \right\}$$

Notice that the set of path $\{p_i \mid K \geq i \geq 1\}$ here is not given but must be chosen. When the number of paths $K \geq 2$, an intermediate strategy between the static routing and dynamic routing, called *semi-oblivious routing*, has been considered in the literature [Hajiaghayi, Kleinberg and Leighton (2007)]. In this approach, the set of paths must be valid for all possible demand scenarios d . The split ratio is then adapted dynamically to each realisation of the demand scenario d . This problem can have applications in telecommunication networks to adjust load balancing over a stable set of paths.

1.3.5 Tree routing

Another restriction that has been considered in the literature is that the routing of commodities must be on a single tree. This can be useful for multicast routing [Lim and Kim (2000)], generally used to stream multimedia content or in conferencing applications. Formally, this corresponds to adding the constraint $\sup(\sum_{h \in \mathcal{H}} x_h(d)) \in \text{trees}(G)$, where $\sup(x)$ for a vector $x \in \mathbb{R}_+^E$ denotes the set of edges $e \in E$ such that $x_e > 0$ and $\text{trees}(G)$ denotes the set of subset of edges in G that forms a tree. Unlike other flow restrictions previously introduced in this section, the tree routing restriction involves coupling constraints between the routing of the different commodities. This distinction will be important in Chapter 2. We will denote this problem by $\text{obj}_{\text{adaptivity-tree}}$.

1.4 Review of traffic uncertainty sets

Several uncertainty sets for traffic demands have been considered for this problem in the literature. The robust network design problem under general polyhedral uncertainty has first been considered in [Ben-Ameur and Kerivin (2003)].

1.4.1 Budget model

A classical uncertainty set in the robust optimization community is the *budget polytope* introduced in [Bertsimas and Sim (2004)] for the general robust optimization problem. If for each commodity $h \in \mathcal{H}$ we have a minimum and a maximum demand value d_h^{min} , d_h^{max} and if not all the demands can (or are likely to) be close to their maximum value d_h^{max} at the same time, the traffic demand can be modeled using the budget polytope. This polytope is defined using a parameter Γ that considers the set of demands $d \in \mathbb{R}^{\mathcal{H}}$ such that $d_h = d_h^{min} + z_h(d_h^{max} - d_h^{min})$ and $\sum_{h \in \mathcal{H}} z_h \leq \Gamma$.

1.4.2 Hose model

The other uncertainty sets presented in the remaining of Section 1.4 have been introduced in the telecommunication community, independently from the robust optimization community.

The *hose model* polytope has been introduced in [Duffield et al. (1999), Gupta et al. (2001)]. This polytope is defined as follows: for each node v of the graph we are given a maximum traffic input m_v^{in} and a maximum traffic output m_v^{out} . The demand polytope \mathcal{D} is defined as the set of vectors $d \in \mathbb{R}^{\mathcal{H}}$ satisfying the constraints $\sum_{h|s(h)=v} d_h \leq m_v^{out}$ and $\sum_{h|t(h)=v} d_h \leq m_v^{in}$ for all nodes $v \in V$.

The *capped hose model*, introduced by [Fr chet te et al. (2014)], is a natural extension of the hose model. Additionally to the maximum traffic input m_v^{in} and traffic output m_v^{out} of each node, there is also a maximum demand value d_h^{max} for each individual commodity h . A special case of this more general polytope, called the *masked hose model* [Bosman and Olver (2017)], considers that d_h^{max} are either 0 or ∞ .

1.4.3 Set of all routable demands

The polytope containing all possible demand vectors has been introduced by [Azar et al. (2003)] in the context of oblivious routing which consists in finding a fixed routing with a good worst case competitive ratio between the congestion obtained from this fixed routing and the optimal routing for any demand vector (without specific constraints other than positivity). It has been proved by [R ck e (2002)], that there always exists a routing having a polylogarithmic competitive ratio. However it was still an open

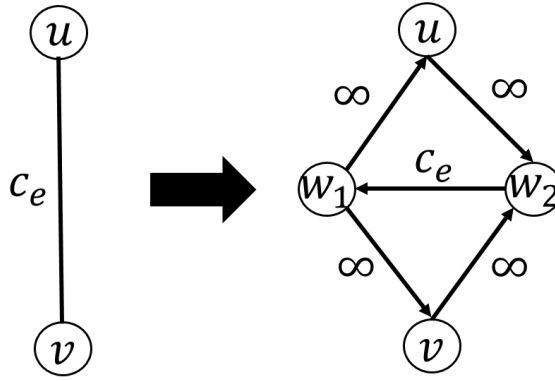


Fig. 1.2 From undirected instance to directed instances.

question if such routing could be found in polynomial-time. In [Azar et al. (2003)] it has been proved that finding an optimal oblivious routing is equivalent to finding a static routing minimizing the congestion over the polytope containing all possible demand vectors that can be routed through the network (routing is here dynamic). This in turn leads to a proof that such routing could be found in polynomial-time.

1.5 Known relationships between problem variants

Let us now present some established relationships between different problems: routing in directed vs undirected graphs, minimizing either a linear cost or the congestion.

1.5.1 From undirected graphs to directed graphs

An instance with an undirected graph can be modeled using a directed graph instance. This can be done by removing each edge e between two nodes u, v , adding two new nodes w_1, w_2 and adding 5 directed edges: for instance in the example of Figure 1.2 we add edges from u to w_2 , from w_2 to w_1 , from w_1 to u and v , from u and v to w_2 and we put the same capacity (or cost) for the edge from w_2 to w_1 that was on the edge e . As we will see, results about the static/dynamic gap provide evidences that an instance with a directed graph cannot be modeled by an undirected graph instance, at least without an exponential increase of the size of the instance.

1.5.2 Node-capacitated and edge-capacitated instances in directed graphs

While only edge-capacitated networks are considered in this thesis, other variants such as node-capacitated networks can also be studied. However, a node-capacitated instance can be transformed into an edge-capacitated instance and vice-versa.

We will first describe how an instance with capacities on the nodes can be represented by an instance with capacities on the edges. This can be done by replacing each node v by two nodes v_1 and v_2 and adding an edge (v_1, v_2) whose capacity is the same

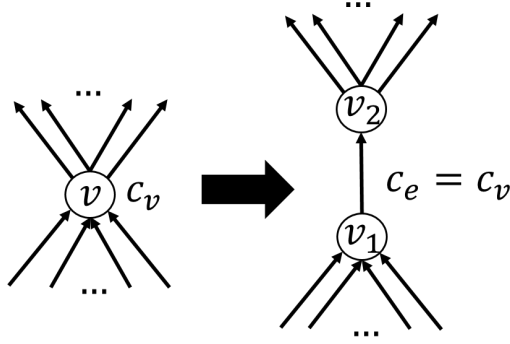


Fig. 1.3 From node-capacitated to edge-capacitated instances.

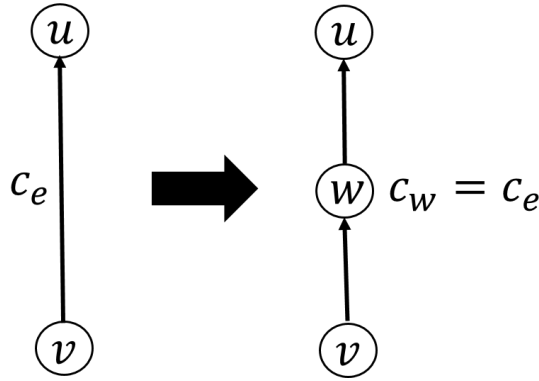


Fig. 1.4 From node-capacitated to edge-capacitated instances.

as the capacity of node v . Each edge that had v as destination (resp. as source) will have v_1 (resp. v_2) as destination (resp. as source) in the new graph (see Figure 1.3).

Conversely, an edge-capacitated instance can be simulated by a node-capacitated instance. This can be done by replacing each edge e with source v and destination u by one node w with the same capacity than the removed edge and by adding two edges, one from v to w and one from w to u (see Figure 1.3).

1.5.3 From congestion to linear cost

We will now introduce a result from [Goyal et al. (2009)] which states that if the static/dynamic gap is less than or equal to α for the congestion problem, then the same applies to the linear cost problem.

Proposition 1.5.1. [Goyal et al. (2009)] : Assume that for a given graph G , a set of commodities \mathcal{H} , a demand polytope \mathcal{D} and some $\alpha > 1$ we have $\text{cong}_{\text{sta}} \leq \alpha \text{cong}_{\text{dyn}}$ for **any** vector $c \in \mathbb{R}_+^E$. Then $\text{lin}_{\text{sta}} \leq \alpha \text{lin}_{\text{dyn}}$ for **any** cost vector $\lambda \in \mathbb{R}_+^E$ (for the instance given by the data $G, \mathcal{H}, \mathcal{D}, \lambda$).

Proof. Given a cost vector λ , let $c_{\text{dyn}}^* \in \mathbb{R}_+^E$ be the reservation vector (i.e., u) obtained when the linear cost is minimized under dynamic routing. For the capacity vector c_{dyn}^*

we clearly have $\text{cong}_{\text{dyn}} \leq 1$ and $\text{cong}_{\text{sta}} \leq \alpha \text{cong}_{\text{dyn}} \leq \alpha$. Therefore, αc_{dyn}^* is a feasible reservation vector for the lin_{sta} problem and its cost is α times the cost of c_{dyn}^* . \square

A consequence of Proposition 1.5.1 is that if we know an upper bound on the static/dynamic gap for the congestion problem on a certain class of instances then this result can be transferred to the linear cost problem. In the other direction, if a lower bound can be found on the static/dynamic gap for the linear cost problem from some classes of instances (i.e., some instances achieving a certain bound can be found), then this implies that there is also a lower bound on the static/dynamic gap for the linear cost problem.

1.6 Intermediate strategies between static and dynamic routing

There are at least two reasons to consider routing strategies that are intermediate between the static and dynamic routing. The first one is to obtain a routing that gives a better solution than the static one (in terms of cost) while being easier to solve than the dynamic routing problem. In this class of methods that achieve such trade off, one can at least consider the following strategies: *affine routing*, *multi-static routing*, *multipolar routing*. Second, static routing is easier to implement in practice while dynamic routing requires a continuous adaptation of routing.

1.6.1 Affine routing

The affine routing method has been presented in [Ouorou and Vial (2007)] to approximately compute lin_{dyn} for the path restricted problem variant. It consists in restraining the dynamic path flow $f_{h,p}^{h'}(d)$ to depend affinely on the demand scenario $d \in \mathcal{D}$. The decision variables are now the coefficients $x_{h,p}^{h'}$ for $h' \in \mathcal{H} \cup \{0\}$, and the dynamic path flow is $f_{h,p}^{h'}(d) = \sum_{h' \in \mathcal{H}} d_{h'} x_{h,p}^{h'}$. This problem can be modeled as the following single stage robust optimization problem:

$$\begin{aligned} & \min \text{obj}(u) \\ & \sum_{h \in \mathcal{H}} \sum_{p \in \mathcal{P}^h: p \ni e} \left(x_{p,h}^0 + \sum_{h' \in \mathcal{H}} x_{p,h}^{h'} d_{h'} \right) \leq u_e, \quad \forall e \in E, d \in \mathcal{D} \end{aligned} \quad (1.6a)$$

$$\sum_{p \in \mathcal{P}^h} \left(x_{p,h}^0 + \sum_{h' \in \mathcal{H}} x_{p,h}^{h'} d_{h'} \right) = d_h, \quad \forall h \in \mathcal{H}, d \in \mathcal{D} \quad (1.6b)$$

$$x_{p,h}^0 + \sum_{h' \in \mathcal{H}} x_{p,h}^{h'} d_{h'} \geq 0, \quad \forall h \in \mathcal{H}, p \in \mathcal{P}^h, d \in \mathcal{D} \quad (1.6c)$$

Affine routing has also been used in [Poss and Raack (2013)] to solve the (unrestricted flow) lin_{dyn} problem. It consists in restraining the dynamic edge flow $f_{h,e}^{h'}(d)$ to depend affinely on the demand scenario $d \in \mathcal{D}$. The decision variables are now the coefficients $x_{h,e}^{h'}$ for $h' \in \mathcal{H} \cup \{0\}$, and the dynamic edge flow is $f_{h,e}^{h'}(d) = \sum_{h' \in \mathcal{H}} d_{h'} x_{h,e}^{h'}$. This problem can then be modeled as the following single stage robust optimization problem:

$\min \text{obj}(u)$

$$\sum_{e \in \delta_+(v)} \left(x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'} \right) - \sum_{e \in \delta_-(v)} \left(x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'} \right) = \begin{cases} d_h, & \text{if } v = s(h) \\ -d_h, & \text{if } v = t(h) \\ 0 & \text{otherwise} \end{cases}$$

$$\forall h \in \mathcal{H}, v \in V, d \in \mathcal{D} \quad (1.7a)$$

$$\sum_{h \in \mathcal{H}} \left(x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'} \right) \leq u_e, \quad \forall e \in E, d \in \mathcal{D} \quad (1.7b)$$

$$x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'} \geq 0, \quad \forall e \in E, \forall h \in \mathcal{H}, d \in \mathcal{D} \quad (1.7c)$$

The obtained problems are of polynomial size. We will see in Appendix A how this kind of problems can be solved given a demand polytope \mathcal{D} described by inequalities, either by reformulating it as a compact linear program or by using a cutting plane procedure. Affine routing can then be seen as an approximation of dynamic routing that can be computed in polynomial-time.

1.6.2 Multipolar routing

In this subsection we present the multipolar robust routing approach, introduced in [Ben-Ameur and Żotkiewicz (2013a)] for the robust routing problem and later generalized for more general robust optimization problems in [Ben-Ameur et al. (2018a)]. Suppose that we have a finite set of points $\mathcal{W} = \{w^1, w^2, \dots, w^{|\mathcal{W}|}\}$ called *poles* such that $\mathcal{D} \subseteq \text{conv}(\mathcal{W})$. Consider the following set:

$$\Lambda := \left\{ \lambda \in \mathbb{R}_+^{|\mathcal{W}|} \mid \sum \lambda^w = 1, \text{ s.t. } \sum \lambda^w w \in \mathcal{D} \right\}$$

Note that a demand vector $d \in \mathcal{D}$ can potentially be written in several ways. For each $\lambda \in \Lambda$ there is a corresponding demand vector $d(\lambda) = \sum_{w \in \mathcal{W}} \lambda^w w$. It consist in having a variable $x_{h,e}^w$ for each $h \in \mathcal{H}, e \in E$ and $w \in \mathcal{W}$. We impose that $f_h(\lambda) = \sum_{h \in \mathcal{H}} \lambda^w x_h^w$ is a flow of value $d_h(\lambda)$ from $s(h)$ to $t(h)$. Note that the vectors x_h^w them-self are not required to be flow vectors, are not necessarily non-negative and they do not need to respect flow conservation constraints. This can be modeled as a single stage robust

optimization problem (where Λ is the uncertainty set) as follows. It can be shown that this approach is equivalent to affine routing by choosing \mathcal{W} such that $\text{conv}(\mathcal{W})$ is a simplex containing \mathcal{D} . It can also be shown that this approach is equivalent to dynamic routing by taking \mathcal{W} to be the set of extreme points of \mathcal{D} . This problem can be modeled as a single stage robust optimization problem.

$$\begin{aligned} & \min \text{obj}(u) \\ & \sum_{h \in \mathcal{H}} \left(\sum_{w \in \mathcal{W}} \lambda^w x_{h,e}^w \right) \leq u_e, \quad \forall e \in E, \lambda \in \Lambda \end{aligned} \quad (1.8a)$$

$$\sum_{e \in \delta_+(v)} \left(\sum_{w \in \mathcal{W}} \lambda^w x_{h,e}^w \right) - \sum_{e \in \delta_-(v)} \left(\sum_{w \in \mathcal{W}} \lambda^w x_{h,e}^w \right) = \begin{cases} \sum_{w \in \mathcal{W}} \lambda^w w_h, & \text{if } v = s(h) \\ - \sum_{w \in \mathcal{W}} \lambda^w w_h, & \text{if } v = t(h) \\ 0 & \text{otherwise} \end{cases} \quad (1.8b)$$

$$\forall h \in \mathcal{H}, v \in V, \lambda \in \Lambda \quad (1.8b)$$

$$\sum_{w \in \mathcal{W}} \lambda^w x_{h,e}^w \geq 0, \quad \forall e \in E, \forall h \in \mathcal{H}, \lambda \in \Lambda \quad (1.8c)$$

1.6.3 Multi-static routing

The multi-static routing method has been introduced in [Ben-Ameur (2007)]. It has also been independently studied in [Bertsimas and Caramanis (2010)]. The multi-static approach has been further studied in [Ben-Ameur and Żotkiewicz (2011)]. It consists in covering the polytope \mathcal{D} by polyhedral subsets \mathcal{D}_i for $i \in I$, i.e., such that $\mathcal{D} \subseteq \bigcup_{i \in I} \mathcal{D}_i$. This is typically done by cutting the polytope \mathcal{D} with an hyperplane $\{d \in \mathbb{R}_+^{\mathcal{H}} \mid \sum_{h \in \mathcal{H}} a_h d_h = b\}$ in two pieces $\mathcal{D}_1, \mathcal{D}_2$ defined by:

$$\mathcal{D}_1 = \{d \in \mathbb{R}_+^{\mathcal{H}} \mid \sum_{h \in \mathcal{H}} a_h d_h \geq b\}$$

$$\mathcal{D}_2 = \{d \in \mathbb{R}_+^{\mathcal{H}} \mid \sum_{h \in \mathcal{H}} a_h d_h \leq b\}$$

The covering polytopes can be recursively decomposed to obtain smaller polyhedra subsets \mathcal{D}_i . An approach to iteratively partition the covering polytopes has been independently proposed in [Bertsimas and Dunning (2016)] and in [Postek and Hertog (2016)] for general multistage robust optimization problem, where at each iteration the multi-static model is solved for the current partition and then the partition is refined using information about the current multi-static solution. [Silva et al. (2018)] applied this method for the single path routing problem and for the fractional routing problem, both with given paths. They also introduced several improvements of the method.

Given such decomposition of the polytope the approach consists in choosing a potentially different routing x^i for each polyhedra subset \mathcal{D}_i . This approach always gives a solution that is at least as good as the one provided by the static approach. This can be seen by taking all routings x^i to be equal to the optimal static routing x^{opt} .

This problem can be written as a single stage robust optimization problem as follows:

$$\min \text{obj}(u)$$

$$\sum_{e \in \delta_+(v)} x_{h,e}^i - \sum_{e \in \delta_-(v)} x_{h,e}^i = \begin{cases} 1, & \text{if } v = s(h) \\ -1, & \text{if } v = t(h) \\ 0 & \text{otherwise} \end{cases}$$

$$\forall h \in \mathcal{H}, v \in V, i \in \mathcal{I}, d \in \mathcal{D}_i \quad (1.9a)$$

$$\sum_{h \in \mathcal{H}} d_h x_{h,e}^i \leq u_e, \quad \forall e \in E, i \in \mathcal{I}, d \in \mathcal{D}_i \quad (1.9b)$$

$$x_{h,e}^i \geq 0, \quad \forall e \in E, \forall h \in \mathcal{H}, i \in \mathcal{I}, d \in \mathcal{D}_i \quad (1.9c)$$

1.7 Known theoretical results on complexity and approximability

We first present results related to the variants where no specific restrictions on the flows are considered. Then, we focus on single path routing and semi-oblivious routing. Finally, we consider the case of hose polytopes. The reviewed results are mainly related to the static/dynamic gap, the complexity and the inapproximability of the dynamic routing problem.

1.7.1 Fractional routing

We first present the results for undirected graphs. It has been proved that computing lin_{dyn} is coNP-hard in undirected graphs by [Chekuri et al. (2007)]. This also implies that computing cong_{dyn} is coNP-hard. Those results have been proved for the hose model. A result attributed to A. Gupta (see [Chekuri et al. (2007)]) is that there is an upper bound of $O(\log n)$ on the static/dynamic gap where n is the number of nodes of the graph (i.e. $\text{lin}_{\text{sta}} = O(\log n)\text{lin}_{\text{dyn}}$). Since lin_{sta} can be computed in polynomial-time, this also give an $O(\log n)$ approximation for lin_{dyn} . Notice that the bound given by static routing cannot provide a better bound than $O(\log n)$ since a lower bound of $\Omega(\log n)$ is achieved by static routing for planar graphs [Maggs et al. (1997), Bartal and Leonardi (1999)] and [Goyal et al. (2009)] has shown that the gap $\frac{\text{lin}_{\text{sta-frac}}}{\text{lin}_{\text{dyn-frac}}}$ is $\Omega(\log n)$.

In the case of directed graphs, computing lin_{dyn} is also coNP-hard [Gupta et al. (2001)]. Notice that the coNP-hardness in the undirected case directly implies coNP-hardness in the directed case. When congestion is minimized, [Azar et al. (2003)] has

shown that the gap between static fractional routing and dynamic fractional routing can be $\Omega(\sqrt{k})$ while [Hajiaghayi, Kleinberg, Räcke and Leighton (2007)] proves that the gap is upper-bounded by $O(\sqrt{kn}^{\frac{1}{4}} \log n)$. Since the instance provided in [Azar et al. (2003)] contains vertices with large degree, [Hajiaghayi, Kleinberg, Räcke and Leighton (2007)] studied the version where the degree is less than some constant and all commodities have the same sink. An instance with a $\Omega(\sqrt{n})$ gap was then provided in [Hajiaghayi, Kleinberg, Räcke and Leighton (2007)], while the upper bound becomes $O(\sqrt{n} \log n)$. [Hajiaghayi, Kleinberg, Räcke and Leighton (2007)] considered also the case of symmetric demands (in that paper, symmetry means that for any two nodes u and v , the demand from u to v is equal to the demand from v to u) and shows that the upper bound of the static to dynamic ratio becomes $O(\sqrt{k} \log^{5/2} n)$. A general $\Omega(n)$ lower bound was later proposed in [Ene et al. (2016)]. They also introduced the notion of balance for directed graphs. A weighted directed graph is α -balanced if for every subset $S \subseteq V$, the total weight of edges going from S to $V \setminus S$ is within a factor α of the total weight of edges directed from $V \setminus S$ to S . Using this new parameter, they show that for single source instances an upper bound of $O(\alpha^{\frac{\log^3 n}{\log \log n}})$ holds for the competitive ratio of static routing.

In undirected node-capacitated instances, [Hajiaghayi, Kleinberg, Räcke and Leighton (2007)] have shown that the static/dynamic gap can be at most $O(\sqrt{n} \log n)$ for single source instances and $O(\sqrt{n} \log^{\frac{3}{2}} n)$ in the general case.

1.7.2 Other variants with flow restrictions

The single path robust routing problem has mainly been studied in undirected graphs. We present below those results. The $O(\log n)$ approximation, attributed to A. Gupta (see [Chekuri (2007), Goyal et al. (2009)] for a more detailed presentation), related to the linear cost under dynamic fractional routing problem is achieved by a routing on a (fixed) single tree. In particular, this shows that the ratios between the dynamic and the static solutions under fractional routing ($\frac{\text{lin}_{\text{sta-frac}}}{\text{lin}_{\text{dyn-frac}}}$) and between single path and fractional routing under the static model ($\frac{\text{lin}_{\text{sta-sing}}}{\text{lin}_{\text{sta-frac}}}$) is in $O(\log n)$ and provides an $O(\log n)$ approximation for static single path routing $\text{lin}_{\text{sta-sing}}$. On the other hand [Olver and Shepherd (2014)] show that the static single path problem cannot be approximated within a $\Omega(\log^{\frac{1}{4}-\epsilon} n)$ ratio unless $NP \not\subseteq ZPTIME(n^{\text{poly} \log(n)})$. As noticed in [Goyal et al. (2009)], this implies (assuming this complexity conjecture) that the gap $\frac{\text{lin}_{\text{sta-sing}}}{\text{lin}_{\text{sta-frac}}}$ is in $\Omega(\log^{\frac{1}{4}-\epsilon} n)$.

Focusing on the semi-oblivious routing problem in undirected graphs, it has been shown in [Hajiaghayi, Kleinberg and Leighton (2007)] that the gap between the dynamic fractional routing and a dynamic fractional routing restricted to a polynomial number of paths can be $\Omega(\frac{\log n}{\log \log n})$.

1.7.3 Results for the hose model

For the linear cost objective function and undirected graphs, an extensively studied polyhedron is the *symmetric* hose model. The demand vector is here not oriented (i.e., there is no distinction between a demand from i to j and a demand from j to i), and uncertainty is defined by considering an upper-bound limit b_i for the sum of demands related to node i . A 2-approximation has been found for the dynamic fractional case [Fingerhut et al. (1997), Gupta et al. (2001)] based on tree routing (where we route through a static tree that should be found) showing that $\frac{\text{lin}_{\text{sta-tree}}}{\text{lin}_{\text{dyn-frac}}} \leq 2$. It has been conjectured that this solution resulted in an optimal solution for the static single path routing. This question has been open for some time and has become known as the *VPN conjecture*. It was finally answered by the affirmative in [Goyal et al. (2013)]. The *asymmetric* hose polytope was also considered in many papers. An approximation algorithm is proposed to compute $\text{lin}_{\text{sta-sing}}$ within a ratio of 3.39 [Eisenbrand et al. (2007)] (or more precisely 2 plus the best approximation ratio for the Steiner tree problem). If \mathcal{D} is a *balanced* asymmetric hose polytope, i.e., $\sum_{v \in V} b_v^{\text{out}} = \sum_{v \in V} b_v^{\text{in}}$ where b_v^{in} (resp. b_v^{out}) is the upper bound for the traffic entering into (resp. going out of) v , then the best approximation factor becomes 2, see [Eisenbrand et al. (2007)]. Moreover, if we assume that $b_v^{\text{out}} = b_v^{\text{in}}$, then $\text{lin}_{\text{sta-sing}}$ is easy to compute and we get that $\text{lin}_{\text{sta-tree}} = \text{lin}_{\text{sta-sing}}$ [Olver (2010)]. In other words, there is some similarity with the case where \mathcal{D} is a symmetric hose polytope.

It has been shown [Bosman and Olver (2017)] in that the masked hose model can be solved in polynomial-time when the underlying graph is a cycle and when it is a tree of bounded degree. They have also shown that when the graph is a tree of bounded degree, a 2α -approximation can be obtained, where α is the approximation ratio for the Steiner tree problem without any bound on the degree of the tree.

1.8 Related problems

In this section, we present tightly related problems to the robust network design problems considered in this thesis.

1.8.1 Other variants on robust network design

Robust network design is a rich research area where many variants have been studied. Beyond flow restrictions and the uncertainty on traffic as presented earlier, others variants have been considered. We will present some of them in the following.

[Bertsimas and Sim (2003)] consider a robust minimum cost flow problem where the demand is fixed but the cost is uncertain. This problem takes as input an undirected graph, two source and destination nodes s, t , a demand value d that needs to be sent and

a polytope Λ of possible edge cost vectors. Let us denote X to be the set of $s - t$ flows of value d . The goal is to find a flow minimising $\min_{x \in X} \max_{\lambda \in \Lambda} \lambda^t x$. They show that this problem can be solved by solving $n + 1$ nominal flow problems $\min_{x \in X} \lambda^t x$ (with deterministic costs vectors λ).

[Lemaréchal et al. (2010)] consider a tightly related robust network design problem where a maximum budget b is given on the costs of the capacities and the goal is to minimize a weighted sum of the unroutable demands for a given $\pi \in \mathbb{R}_+^{\mathcal{H}}$. Formally the problem can be written as follows:

$$\mu = \min_{u \in \mathcal{U}(b)} \max_{d \in \mathcal{D}} \min_{d' \in \mathcal{D}(u)} \sum_{h \in \mathcal{H}} \pi_h (d_h - d'_h)^+ \quad (1.10)$$

where $\mathcal{U}(b)$ is the set of capacities of total cost less than the budget b , i.e. the set of $u \in \mathbb{R}_+^E$ satisfying $\sum_{e \in E} \lambda_e u_e \leq b$ and $\mathcal{D}(u)$ is the set of demand vectors that can be routed over capacities u . Solutions to this problem are interesting for the robust network design problem lin_{dyn} for at least two reasons. The first one is that if we can find an oracle solving Problem (1.10), we can use it to solve our lin_{dyn} using a dichotomic search. The second reason is that the inner max-min problem of (1.10) can be seen as a way to check if a capacity vector u is enough to route all demands inside the polytope \mathcal{D} . This can be used in a cutting plane procedure to compute a solution for the lin_{dyn} problem.

They proposed two procedures, one to compute a lower bound using an ingenious procedure to iteratively construct an inner approximation of the demand polyhedron \mathcal{D} and another procedure to compute an upper bound on μ based on inverting the inner min-max problem and other smart ideas. Later [Ouorou (2013)] investigated the problem from a dual perspective: given an upper bound $\bar{\mu}$ on the maximum traffic rejection they seek a capacity vector assignment. This problem can also be seen as a generalisation of the lin_{dyn} problem that we consider in this thesis. Based on the affinely adjustable robust counterpart method and other idea borrowed from earlier work they propose four tractable procedures to approximately solve this problem.

1.8.2 Two-stage adaptive optimization

At first sight it might be possible to cast Problem (1.3) as a two-stage adaptive robust optimization problem of the form (1.11). [Bertsimas and Goyal (2012)] presented the following general model:

$$\min cx + \max_{b \in \mathcal{U}} dy(b)$$

$$Ax + By(b) \geq b \quad \forall b \in \mathcal{U} \quad (1.11a)$$

$$x, y(b) \geq 0 \quad (1.11b)$$

where the set $\mathcal{U} \subseteq \mathbb{R}_+^m$ is the uncertainty set supposed to be compact, convex and full dimensional, variables $x \in \mathbb{R}_+^{n_1}$ are the first stage variables, $y(b) \in \mathbb{R}_+^{n_2}$ are the second stage variables $n_1, n_2 \in \mathbb{N}$, $A \in \mathbb{R}_+^{m \times n_1}$, $B \in \mathbb{R}^{m \times n_1}$, $c \in \mathbb{R}_+^{n_1}$, $d \in \mathbb{R}_+^{n_2}$.

When the coefficients of the matrix A are positive, [Bertsimas and Goyal (2012)] have shown that the optimal *affine* solution cannot be worse than $O(\sqrt{m})$ times the cost of the optimal dynamic solution.

While it is not trivial to cast the robust network design problem as a two-stage adaptive robust optimization problem, the ideas of [Bertsimas and Goyal (2012)] will be adapted in Chapter 3.

1.9 Thesis contributions

As previously mentioned, the thesis is dedicated to the design of robust networks where the general objective is to propose a routing strategy that can adapt to uncertainty. Since the dynamic routing strategy is generally difficult to compute and to implement, we would like to approximate it using routing strategies that can be computed in polynomial-time. Hence, we should first study the approximability/inapproximability of dynamic routing. Then, we study new affine and multi-static routing strategies that can approximate dynamic routing in a reasonable way. The contributions of the thesis are therefore mainly on approximability and algorithmic approaches.

1.9.1 Inapproximability results in the undirected case

Chapter 2 is dedicated to dynamic routing in undirected graphs. The main contributions are new inapproximability results.

- We first prove that the robust network design problem with minimum congestion cannot be approximated within any constant factor. The reduction is based on the PCP theorem and some connections with the Gap-3-SAT problem [Arora and Barak (2009)]. The same reduction also allows to show inapproximability within $\Omega(\log \frac{n}{\Delta})$ where Δ is the maximum degree in the graph and n is the number of vertices.
- Using the ETH conjecture [Impagliazzo and Paturi (2001), Impagliazzo et al. (2001)], we prove a $\Omega(\frac{\log n}{\log \log n})$ lower bound for the approximability of the ro-

bust network design problem with minimum congestion. This implies that the well-known $O(\log n)$ approximation ratio that can be obtained using the result in [Räcke (2008)] is tight.

- We show that any α -approximation algorithm for the robust network design problem with linear costs directly leads to an α -approximation for the problem with minimum congestion. The proof is based on Lagrange relaxation. We obtain that robust network design with minimum congestion can be approximated within $O(\log n)$. This was already proved in [Räcke (2008)] in a different way.
- An important consequence of the Lagrange-based reduction and our inapproximability results is that the robust network design problem with linear reservation cost cannot be approximated within any constant ratio. This answers a long-standing open question stated in [Chekuri (2007)].
- Another consequence is a new proof for the existence of instances for which the optimal static solution can be $\Omega(\log n)$ more expensive than a solution based on dynamic routing, when a linear cost is minimized. This was already proved in [Goyal et al. (2009)] in a different way.
- We show that even if only two given paths are allowed for each commodity, there is a constant k such that the robust network design problem with minimum congestion or linear costs cannot be approximated within k .

1.9.2 Approximability and inapproximability results in the directed case

Chapter 3 is dedicated to dynamic routing in directed graphs. The main contributions are a new approximation factor and some inapproximability results.

- We prove that compared to dynamic routing, when static routing is considered, congestion is multiplied by a factor less than or equal to $\sqrt{8k}$ where k is the number of commodities. This implies that the gap between static routing and dynamic routing for the congestion minimization problem is $O(\sqrt{k}) = O(n)$ where n is the number of nodes. The best-known previous bound is $O(\sqrt{kn}^{\frac{1}{4}} \log n)$ and was given by [Hajiaghayi, Kleinberg, Räcke and Leighton (2007)]. The same $\sqrt{8k}$ bound applies to the linear reservation cost problem. The new upper bound matches the $\Omega(\sqrt{k})$ lower bound of [Azar et al. (2003)] and the $\Omega(n)$ lower bound of [Ene et al. (2016)].
- We introduce a more general version of the two robust network design problems (related to congestion and linear cost) by considering some flow restrictions (each commodity h can only be routed through edges inside a given subset E_h). The

upper bound $\sqrt{8k} = O(n)$ is still valid and the static versions of the problems can still be solved in polynomial-time. We show some strong inapproximability results for this problem. More precisely, we prove that unless $NP \subseteq SUBEXP$, neither minimum dynamic congestion nor optimal linear cost can be approximated within a ratio of $k^{\frac{c}{\log \log k}}$ (resp. $n^{\frac{c}{\log \log n}}$) for some constant c . Making use of a weaker assumption, we get that unless $NP \subseteq QP$, there is no approximation within a factor of $2^{\log^{1-\epsilon} k}$ (resp. $2^{\log^{1-\epsilon} n}$) for any $\epsilon > 0$. This result improves the $\Omega(\frac{\log n}{\log \log n})$ inapproximability bound of we proved for the undirected case that also applies to the directed one.

1.9.3 Affine routing

In Chapter 4 we study the affine routing method and propose several strengthening of the original scheme. The main contributions are given below.

- We study the relationship between the original affine routing formulations [Ouurou and Vial (2007), Poss and Raack (2013)], namely the node-arc and the arc-path formulations. We show that the node-arc formulation can strictly dominate the arc-path formulation. We also derive a natural cycle-based formulation equivalent to the node-arc formulation but that uses less variables and constraints.
- We introduce two ways of relaxing the flow conservation constraints in the node-arc formulation. We prove that this leads to feasible solutions and then we see that they can both strictly dominate the standard node-arc formulation.
- We propose a cut based formulation as an improved solution over both relaxed-flow conservation formulations. However we show that, unless $P = NP$, it cannot be solved in polynomial-time.
- We combine the two relaxed-flow conservation formulations using an extended graph. We prove that this formulation dominates both relaxations and that it can be solved in polynomial-time.
- To drastically reduce the size of models and consequently the solving times, we present variants of the formulations based on the aggregation of different flows having the same source and/or sink in the node-arc formulation. Moreover, we show that this can also improve the cost of the solutions.
- We numerically test the different formulations that can be solved with polynomial-time algorithms after a reformulation using a classical duality-based method. We compare the solutions and execution times for two polyhedrons, two topologies and two objective functions.

1.9.4 Multi-static routing

In Chapter 5 we introduce a new multi-static approach. The contributions are summarized as follows.

- Instead of covering the whole polytope \mathcal{D} by polyhedral subsets, we only need that the convex hull of some polyhedral subsets cover the polytope. This leads to a natural choice of polyhedral subsets consisting of faces of the polytope. A static routing is then considered for each subset.
- One possibility to use our new multi-static approach is to take all faces of the polytope \mathcal{D} having some dimension. We show that in some cases we only need a tiny fraction of the faces to cover the polytope \mathcal{D} by their convex hulls.
- We numerically tested our new approach.

1.10 List of publications

1. Al-Najjar, Y., Ben-Ameur, W. and Leguay, J. (2022), Approximability of robust network design: The directed case, in ‘39th International Symposium on Theoretical Aspects of Computer Science (STACS)’, Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
2. Al-Najjar, Y., Ben-Ameur, W. and Leguay, J. (2021), On the approximability of robust network design, Elsevier Theoretical Computer Science. 860, 41–50.
3. Al-Najjar, Y., Ben-Ameur, W., Leguay, J. and Elias, J. (2021), Affine routing for robust network design, Wiley Journal on Networks. 79, 557– 579.
4. Al-Najjar, Y., Paris, S., Elias, J., Leguay, J. and Ben-Ameur, W. (2019), Optimal routing configuration clustering through dynamic programming, in ‘ALGOTEL 2019: 21emes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications’.
5. Al-Najjar, Y., Ben-Ameur, W. and Leguay, J. (2020), Autour de l’approximabilité du problème de dimensionnement robuste des réseaux, in ‘ALGOTEL 2020: 22èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications’.

CHAPTER 2

Robust network design problem in undirected graphs

In this chapter we present our theoretical study of some variants of the robust network design problem in undirected graphs under a general polyhedral set given as input. As mentioned in Chapter 1, polynomial-time algorithms to compute optimal static routing (with respect to either congestion or linear reservation cost) have been proposed [Aplegate and Cohen (2003), Azar et al. (2003), Ben-Ameur and Kerivin (2003, 2005)] based on either duality or cutting-plane algorithms. Additionally it is known from a result attributed to A. Gupta (c.f. [Chekuri (2007)]) that the optimal solution of lin_{sta} cannot be worse than $O(\log n)$ times the cost of the optimal solution of lin_{dyn} . It was also known that this problem is NP -hard in directed graphs [Gupta et al. (2001)] and in undirected graphs [Chekuri et al. (2007)]. Despite the consequent amount of results related to robust routing, there was still a wide gap between the best approximation algorithm and inapproximability lower bounds for the dynamic routing variants.

Some of the open questions that had some interest for the scientific community (see, e.g. [Chekuri (2007)]) are recalled below.

- Is it possible to obtain a better approximation ratio than $O(\log n)$?
- Is it possible to obtain a constant approximation?
- Is it possible to obtain a polynomial-time approximation scheme for this problem?

In this chapter we will provide some insights on the approximability of the dynamic routing variants of robust network design. More specifically, we will prove the following results:

- Unless $P = NP$, $cong_{dyn}$ cannot be approximated in polynomial-time within any constant factor.
- Unless the ETH conjecture is false, there is a lower bound of $\Omega(\log n / \log \log n)$ on the approximability of $cong_{dyn}$.
- Given an oracle that computes an α approximate solution for lin_{dyn} , it is also possible to compute an α approximate for $cong_{dyn}$ by making a polynomial number

of calls to the oracle and in polynomial-time. This implies that the inapproximability results we proved for cong_{dyn} also applies to lin_{dyn}

Those results settle the approximability status of cong_{dyn} and lin_{dyn} up to a lower order factor. We also conducted a theoretical study of the path restricted problem (1.3.2) under dynamic routing for which no complexity results were known previously. If one path per commodity is given then the dynamic routing problem can be trivially solved in polynomial time. Therefore, one may wonder whether the problem is still easy to solve when the number of possible paths is lower than some constant. In other words:

- Is it possible to find a polynomial-time algorithm for this problem when the number of possible (and given) paths is less than or equal to a constant?

However, the following result we proved answers this question by the negative:

- Even if the number of paths per commodity is at most 2 (i.e., $|\mathcal{P}_h| \leq 2$), there is a constant $c > 1$ under which this problem cannot be approximated (unless $P = NP$).

2.1 Inapproximability under some constant factor $c > 1$

In Lemma 2.1.1, we will prove that it is NP -hard to distinguish between instances where $\text{cong}_{\text{dyn}}(\mathcal{D})$ is less than or equal to 1 and those where the cut congestion $\text{cong}_{\text{cut}}(\mathcal{D})$ is greater than or equal to $1 + \rho$ for some constant $\rho > 0$.

Given a 3-SAT instance φ , $\text{val}(\varphi)$ denotes the maximum proportion of clauses that can be simultaneously satisfied (thus φ is satisfiable when $\text{val}(\varphi) = 1$). We will consider polytopes \mathcal{D} that can be expressed through linear constraints and auxiliary variables ξ , i.e., $\mathcal{D} = \{d \in \mathbb{R}^{\mathcal{H}} \mid Ad + B\xi \leq b\}$ where A and B are matrices of polynomial size (the maximum of the number of columns and the number of rows is polynomially bounded). Notice that it is important to consider polytopes that can be easily described (otherwise the difficulty of solving cong_{dyn} would be a consequence of the difficulty of describing the polytope).

Lemma 2.1.1. *For $0 < \rho < 1$, there is a polynomial-time reduction from a 3-SAT instance φ to an instance $\mathcal{I} = (G, c, \mathcal{H}, \mathcal{D})$ of cong_{dyn} :*

- If $\text{val}(\varphi) \leq 1 - \rho$ then $\text{cong}_{\text{dyn}}(\mathcal{I}) \leq 1$
- If φ is satisfiable then $\text{cong}_{\text{dyn}}(\mathcal{I}) \geq \text{cong}_{\text{cut}}(\mathcal{I}) \geq 1 + \rho$.

Furthermore, $|V(G)|$, $|E(G)|$, $|\mathcal{H}|$ and the size of the matrices A and B defining \mathcal{D} are all $O(m)$ where m is the number of clauses of φ .

Proof. Given a 3-SAT instance φ with m clauses, we build an instance of cong_{dyn} as follows. We consider two nodes: a source s and a destination t . Then, for each $i = 1, \dots, m$ we create a path from s to t containing three directed edges $e_{i,j}$ of capacity 1 for $j = 1, 2, 3$ corresponding to the i -th clause of φ . For each $i = 1, \dots, m$ and $j = 1, 2, 3$, \mathcal{H} contains a commodity $h_{i,j}$ with the same source and destination as edge $e_{i,j}$. We also add a commodity $h_{s,t}$ from s to t . The polytope \mathcal{D} is defined as follows. We set $d_{h_{s,t}} = \rho \cdot m$. For each literal l (i.e. a variable or its negation) of the 3-SAT instance φ we add an auxiliary variable ξ_l . Intuitively $\xi_l = 1$ will correspond to setting the literal l to true. For each variable v , we add the constraint $\xi_v + \xi_{\neg v} = 1$ in addition to non-negativity constraints $\xi_v \geq 0$ and $\xi_{\neg v} \geq 0$. For each $i = 1, \dots, m$ and $j = 1, 2, 3$, we consider the constraint $d_{h_{i,j}} = \xi_{l_{i,j}}$ where $l_{i,j}$ is the literal appearing in the i -th clause in the j -th position. Observe that the size of \mathcal{D} is $O(m)$. The number of nodes, edges and commodities are also $O(m)$.

Consider first the case $\text{val}(\varphi) \leq 1 - \rho$. The set of extreme points of \mathcal{D} is such that the ξ_l variables take their values in $\{0, 1\}$. The maximum dynamic congestion is attained for a demand vector of this form. Let d be such a demand vector and consider the corresponding solution of the 3-SAT instance φ . We route all demands $d_{h_{i,j}}$ on $e_{i,j}$. If for some $i = 1, \dots, m$ the i -th clause is false, then the demands $d_{h_{i,1}}, d_{h_{i,2}}, d_{h_{i,3}}$ are equal to 0 and therefore one unit of flow of the commodity $h_{s,t}$ can be routed on the path $(e_{i,1}, e_{i,2}, e_{i,3})$ with congestion less than or equal to 1. Since $\text{val}(\varphi) \leq 1 - \rho$, there are at least $m \cdot \rho$ such indices i (i.e., false clauses) and therefore the demand $d_{h_{s,t}}$ can be routed with a congestion less than or equal to 1.

We now consider the case where φ is satisfiable. Let d be the demand vector corresponding to a truth assignment satisfying φ . For each $i = 1, \dots, m$, let $j(i)$ be the position of a literal set to true in the i -th clause. Therefore we have $d_{h_{i,j(i)}} = 1$ for all $i = 1, \dots, m$. Consider the cut $C = \{e_{i,j(i)} | i = 1, \dots, m\}$. C intersects the paths related to the m demands $d_{h_{i,j(i)}}$ of value 1 in addition to demand $d_{h_{s,t}}$ of value $m \cdot \rho$. The total capacity of this cut is m while the sum of demands belonging to C is $m + m \cdot \rho$. Therefore the congestion of this cut is $\frac{m+m \cdot \rho}{m} = 1 + \rho$. \square

2.2 Inapproximability in undirected graphs

Theorem 2.2.1. *Unless $P = NP$, the minimum congestion problem cannot be approximated with a polynomial-time algorithm within any constant factor even if \mathcal{D} is given by $\{d : Ad + B\psi \leq b\}$ whose size is polynomial bounded by $|V(G)|$.*

This theorem will be an immediate consequence of NP hardness of the gap-3SAT problem and Lemma 2.2.1

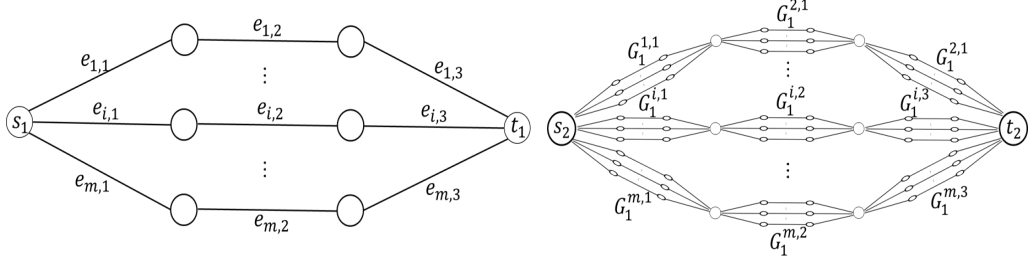


Fig. 2.1 G_1 and G_2

Lemma 2.2.1. *For every $r \in \mathbb{N}$ there is a polynomial-time mapping from a 3-SAT instance to an instance $\mathcal{I}_r = (G_r, c_r, \mathcal{H}_r, \mathcal{D}_r)$ such that:*

- $val(\varphi) < \rho \implies cong_{dyn}(\mathcal{I}_r) \leq 1.$
- $val(\varphi) = 1 \implies cong_{dyn}(\mathcal{I}_r) \geq 1 + r(1 - \rho)$

Furthermore, $|V(G)|$, $|E(G)|$, $|\mathcal{H}|$ and the size of the matrices A and B defining \mathcal{D} are all $O(m^{cr})$, for some constant c , where m is the number of clauses of φ .

Proof. For $r = 1$ we take the instance $I_1 = (G_1, c_1, \mathcal{H}_1, \mathcal{D}_1)$ constructed in Lemma 2.1.1, in this instance we define two special nodes $s_1 := s$ and $t_1 := s$ (see the proof of Lemma 2.1.1). For $r \geq 2$, having constructed $I_{r-1} = (G_{r-1}, c_{r-1}, \mathcal{H}_{r-1}, \mathcal{D}_{r-1})$ with two special nodes denoted s_{r-1}, t_{r-1} , we recursively build an instance $I_r = (G_r, c_r, \mathcal{H}_r, \mathcal{D}_r)$ as follows. We construct the graph G_r , by taking the graph G_1 and replacing each edge by a copy of the graph G_{r-1} denoted by $G_{r-1}^{i,j}$. Each copy $G_{r-1}^{i,j}$ contains a node s_{r-1} that is identified with $s(e_{i,j})$ and a node t_{r-1} identified with $t(e_{i,j})$ (see Figure 2.1). All commodities related to $G_{r-1}^{i,j}$ (belonging to \mathcal{H}_{r-1}) are also considered as commodities of \mathcal{H}_r . Let us use $d^{i,j} \in \mathbb{R}^{\mathcal{H}_{r-1}}$ to denote the related demand vector. \mathcal{H}_r also contains a non-negative commodity $h_{0,r}$ constrained by $d_{h_{0,r}} \leq m^r(1 - \rho)$. Thus $|\mathcal{H}_r| = 1 + 3m \times |\mathcal{H}_{r-1}|$.

We are going to build an uncertainty set \mathcal{D}_r as a coordinate projection of a higher-dimensional polyhedron Ξ_r , involving demand variables in addition to auxiliary non-negative variables ξ_l related to literals, and also auxiliary variables $\psi_{r-1}^{i,j}$ related to $G_{r-1}^{i,j}$ and the description of \mathcal{H}_{r-1} . We gradually explain the construction. For each variable v of the 3-SAT formula φ we add the constraint $\xi_v + \xi_{-v} = 1$. And for $e_{i,j} \in E(G_1)$, we impose the following constraint:

$$d^{i,j} \in \xi_{i,j} \mathcal{D}_{r-1} := \{\xi_{i,j} d_0 \mid d_0 \in \mathcal{D}_{r-1}\} \quad (2.1)$$

Let us explain how this can be expressed with linear constraints. Without loss of generality we can suppose that \mathcal{D}_{r-1} is given as $\mathcal{D}_{r-1} = \{d : A_{r-1}d + B_{r-1}\psi_{r-1} \leq b_{r-1}\}$ and this representation includes (among others) non-negativity constraints of all

variables in addition to the upper bound constraints $d_h \leq d_h^{max}$ where $d_h^{max} = \max_{d \in \mathcal{D}_{r-1}} d_h$. Then by writing $A_{r-1}d^{i,j} + B_{r-1}\psi^{i,j} \leq \xi_{l_{i,j}}b_{r-1}$, we can ensure that $\xi_{l_{i,j}} = 0$ implies $d^{i,j} = 0$ (because of the upper bound constraints), while $\xi_{l_{i,j}} > 0$ leads to $\frac{1}{\xi_{l_{i,j}}}d^{i,j} \in \mathcal{D}_{r-1}$. In particular when $\xi_{l_{i,j}} = 0$, from outside, the whole subgraph corresponding to $G_{r-1}^{i,j}$ acts like a single edge of capacity m^{r-1} .

We observe that all extreme points of Ξ_r are such that $\xi_l \in \{0, 1\}$ for $l \in \mathcal{L}$. To verify that, we first recall that constraints (2.1) are equivalent to $d^{i,j} \in \xi_{l_{i,j}}\mathcal{D}_{r-1}$ (in this way, the vectors $\psi_{r-1}^{i,j}$ can be ignored).

Second, let \mathcal{L}_+ be the set of literals appearing in positive form. We observe that variables ξ_l for $l \in \mathcal{L}_+$ are pairwise independent. Only variables $d^{i,j}$ such that either $l_{i,j} = l$ or $l_{i,j} = \neg l$ depend on ξ_l since $d^{i,j} \in \xi_l\mathcal{D}_{r-1}$ in the first case and $d^{i,j} \in (1 - \xi_l)\mathcal{D}_{r-1}$ in the second case. This immediately implies that given some arbitrary real vectors $q_{i,j}$ and f , minimizing $\sum_{i=1,\dots,m;j=1,2,3} q_{i,j}^T d^{i,j} + \sum_{l \in \mathcal{L}_+} f_l \xi_l$ is equivalent to minimizing $\sum_{l \in \mathcal{L}_+} \xi_l \left(f_l + \sum_{i,j:l_{i,j}=l} \min_{d^{i,j} \in \mathcal{D}_{r-1}} q_{i,j}^T d^{i,j} - \sum_{i,j:l_{i,j}=\neg l} \min_{d^{i,j} \in \mathcal{D}_{r-1}} q_{i,j}^T d^{i,j} \right)$ subject to $0 \leq \xi_l \leq 1$. It is then clear that there is an optimal solution such that ξ_l values will be either 0 or 1. Since this holds for an arbitrary linear objective function, we get the wanted result about extreme points.

Let us now show that $val(\varphi) < \rho \implies \text{cong}_{\text{dyn}}(\mathcal{I}_r) \leq 1$. Assume that $val(\varphi) < \rho$. We prove by induction that the congestion of \mathcal{I}_r is 1. Suppose that this is true for some $r - 1$. If $\xi_{l_{i,1}} = \xi_{l_{i,2}} = \xi_{l_{i,3}} = 0$ for some i , a flow of value m^{r-1} can be routed between s_r and t_r by sending a flow of value 1 on each edge of $G_{r-1}^{i,j}$ for $j = 1, 2, 3$. Since $val(\varphi) < \rho$, there are necessarily at least $m(1 - \rho)$ such i , thus we can send the whole demand $m^{r-1}m(1 - \rho) = m^r(1 - \rho)$ this way. For the indices i, j such that $\xi_{l_{i,j}} = 1$, by the induction hypothesis ($\text{cong}_{\text{dyn}}(\mathcal{I}_{r-1}) \leq 1$), the demands inside $G_{r-1}^{i,j}$ can be routed without sending more than one unit of flow on each edge of $G_{r-1}^{i,j}$.

Since φ is satisfiable, there is a truth assignment represented by ξ variables (the auxiliary variables) such that for each $i = 1, \dots, m$ there is a $j(i)$ such that $\xi_{l_{i,j(i)}} = 1$. By considering the graph $G_{r-1}^{i,j(i)}$ and using the induction hypothesis, we can build a cut $\delta(C_{r-1}^i)$ separating the node $s(e_{i,j(i)})$ and $t(e_{i,j(i)})$ and containing m^{r-1} edges. We also build a demand vector $d^{i,j(i)} \in \mathcal{D}_{r-1}$ such that the sum of demands traversing the cut is greater than or equal to $m^{r-1}(1 + (r - 1)(1 - \rho))$ (still possible by induction). By taking the union of these m disjoint cuts we get a cut $\delta(C_r)$ that is separating s_r and t_r having the required number of edges. A demand vector d can be built by combining the vectors $d^{i,j(i)}$ and the demand $d_{h_0,r}$ taken equal to $m^r(1 - \rho)$. Since the demand from s_r to t_r is also traversing the cut, the total demand through $\delta(C_r)$ is greater than or equal to $m^r(1 - \rho) + m \cdot m^{r-1}(1 + (r - 1)(1 - \rho)) = m^r(1 + r(1 - \rho))$.

Notice that to show that all traffic vectors of \mathcal{D}_r can be routed with congestion 1, we considered demand vectors corresponding with $\{0, 1\}$ ξ variables. The result shown above about extreme points is useful here since it allows us to say that each extreme point of \mathcal{D}_r can be routed with congestion less than or equal to 1 implying that each demand vector inside \mathcal{D}_r can also be routed with congestion less than or equal to 1.

Let us now show that $val(\varphi) = 1 \implies \text{cong}(\mathcal{I}_r) \geq 1 + r(1 - \rho)$. We are going to use induction to build a cut $\delta(C_r)$ where C_r is set of vertices of $V(G_r)$ containing s_r and not containing t_r . The number of edges of the cut will be m^r and each edge has a capacity equal to 1. We also show the existence of a demand vector $d \in \mathcal{D}_r$ such that the sum of the demands traversing the cut is greater than or equal to $m^r(1 + r(1 - \rho))$. This would show that there is at least one edge that carries at least $1 + r(1 - \rho)$ units of flow. □

Lemma 2.2.1 can be further exploited in different ways since there are many possible connections between the value $1 + r(1 - \rho)$ and the characteristics of the undirected graph built in the proof of the lemma. Observe, for example, that by a simple induction we get that the number of vertices $|V(G_r)| = 2 + 2m \frac{(3m)^r - 1}{3m - 1}$ leading to $|V(G_r)| \simeq 2 \times 3^{r-1} m^r$ (when m goes to infinity). We also have $\Delta(G_r)$ equal to m^r where $\Delta(\cdot)$ denotes the maximum degree in the graph. Consequently, $\log\left(\frac{|V(G_r)|}{\Delta(G_r)}\right) \simeq r \log 3 + \log 2/3$. Then by taking any constant k such that $k \times \log 3 < (1 - \rho)$ where ρ is the constant in the PCP Theorem we get a lower bound on the approximability ratio. This is stated in the following corollary.

Corollary 2.2.1. *Under conditions of Theorem 2.2.1, for any constant $c < \frac{1-\rho}{\log 3}$, it is not possible to approximate the minimum congestion problem in polynomial-time within a ratio of $c \log \frac{|V(G)|}{\Delta(G)}$.*

2.2.1 A $\Omega\left(\frac{\log n}{\log \log n}\right)$ approximability lower bound

To get an approximability lower bound, we will use the well-known ETH conjecture that is recalled below.

Conjecture 2.2.1 (Exponential Time Hypothesis). *[Impagliazzo and Paturi (2001), Impagliazzo et al. (2001)] There is a constant δ such that no algorithm can solve 3-SAT instances in time $O(2^{\delta m})$, where m is the number of clauses.*

Let us use n to denote the number of vertices of the graph.

Theorem 2.2.2. *Under Conjecture 2.2.1, no polynomial-time algorithm can solve the minimum congestion problem with the approximation ratio $\Omega\left(\frac{\log n}{\log \log n}\right)$.*

Proof. The combination of PCP Theorem and ETH Conjecture 2.2.1 implies that distinguishing between 3-SAT instances such that $val(\varphi) < \rho$ and $val(\varphi) = 1$ cannot be done in time $O(2^{m^\beta})$ for some constant $\beta > 0$ (a slightly better bound is $O(2^{m/\log^c m})$ for some constant c , but this will not help us to improve the lower bound of Theorem 2.2.2).

Suppose that there is an algorithm that solves the minimum congestion problem with an approximation factor $\alpha(n)$ and a running time $O(n^{c_1})$ for some constant c_1 . Given a 3-SAT instance and a function $r : \mathbb{N} \rightarrow \mathbb{N}$ we can construct a minimum congestion instance $\mathcal{I}_{r(m)}$ as in Lemma 2.2.1 in time $O(m^{c_2 r(m)})$ and where the number of vertices of the instance is $m^{r(m)}$. Then by running the approximation algorithm for minimum congestion we get a total time of $O(m^{c_3 r(m)})$ where $c_3 = \max\{c_1, c_2\}$. Thus by choosing $r(m) = \frac{m^\beta}{c_3 \log m}$ we get an algorithm that runs in time $O(2^{m^\beta})$. And if the approximation factor $\alpha(n)$ is small enough, that is if $\alpha(m^{r(m)}) < 1 + (1 - \rho)r(m)$ for a big enough m , we get an algorithm solving Gap-3-SAT and thus contradicting Conjecture 2.2.1. This is the case for $c_4 \frac{\log n}{\log \log n}$ for some constant c_4 . To see this, we can observe that:

$\frac{1+(1-\rho)r(m)}{\alpha(m^{r(m)})} = \frac{1+(1-\rho)\frac{m^\beta}{c_3 \log m}}{\frac{m^\beta/c_3}{c_4 \beta \log m - \log c_3}} \simeq \frac{\beta(1-\rho)}{c_4}$. By taking $c_4 < \beta(1 - \rho)$ we get the wanted inapproximability result. \square

2.3 From minimum congestion to linear cost

In this section we are going to prove that given an oracle that computes lin_{dyn} within an approximation factor of α we can construct an α approximate solution for $cong_{\text{dyn}}$ problem. A consequence of this result is that inapproximability results for $cong_{\text{dyn}}$ can be directly transferred to lin_{dyn} .

Proposition 2.3.1. *Let $\mathcal{I}' = (G, \lambda, \mathcal{H}, \mathcal{D})$ and assume that $lin_{\text{sta}}(\mathcal{I}') \leq \alpha lin_{\text{dyn}}(\mathcal{I}')$ for some $\alpha \geq 1$ and for **any** cost vector $\lambda \in \mathbb{R}_+^{\mathcal{H}}$. Then $cong_{\text{sta}}(\mathcal{I}) \leq \alpha cong_{\text{dyn}}(\mathcal{I})$ where $\mathcal{I} = (G, c, \mathcal{H}, \mathcal{D})$ for **any** capacity vector $c \in \mathbb{R}_+^{\mathcal{H}}$. Moreover, any β -approximation ($\beta \geq 1$) for lin_{dyn} leads to a β -approximation for $cong_{\text{dyn}}$.*

Given any $\lambda \geq 0$, the robust network design problem with linear costs is simply the following

$$\min_{u \in \mathcal{U}(\mathcal{D})} \lambda^T u. \quad (2.2)$$

Where $\mathcal{U}(\mathcal{D}) \subseteq \mathbb{R}_+^E$ is the set of capacity vectors such that all demand in \mathcal{D} can be (dynamically) routed without exceeding the capacities.

Assume that there exists a number $\alpha \geq 1$ such that Problem (2.2) can be solved in polynomial-time within an approximation ratio α . More precisely, we have a polynomial-time oracle that takes as input a non-negative linear cost $\lambda \in \mathbb{R}^{E(G)}$ and outputs a

$u^{ap}(\lambda) \in \mathcal{U}(\mathcal{D})$ such that $\lambda^T u(\lambda) \leq \lambda^T u^{ap}(\lambda) \leq \alpha \lambda^T u(\lambda)$ where $u(\lambda) \in \mathcal{U}(\mathcal{D})$ is the optimal solution of (2.2).

Recall that the congestion problem is given by

$$\begin{aligned} \min_{\beta, u} \beta & \tag{2.3} \\ u_e &\leq c_e \beta, \forall e \in E(G) \\ u &\in \mathcal{U}(\mathcal{D}) \end{aligned}$$

where β and u are optimization variables.

Let us consider a Lagrange relaxation of (2.3) by dualizing the capacity constraints and using λ for the dual multipliers. The dual problem is then given by $\max_{\lambda \geq 0} \min_{\beta, u \in \mathcal{U}(\mathcal{D})} \beta + \sum_{e \in E(G)} \lambda_e (u_e - \beta c_e)$ (where β is an optimization variable). If λ is chosen such $\sum_e \lambda_e c_e \neq 1$, then the value of the inner minimum would be $-\infty$. Thus in an optimal solution, we will always have $\sum_e \lambda_e c_e = 1$. The problem is then equivalent to:

$$\max_{\substack{\lambda \geq 0 \\ \sum_{e \in E(G)} \lambda_e c_e = 1}} \min_{u \in \mathcal{U}(\mathcal{D})} \sum_{e \in E(G)} \lambda_e u_e = \max_{\substack{\lambda \geq 0 \\ \sum_{e \in E(G)} \lambda_e c_e = 1}} \lambda^T u(\lambda). \tag{2.4}$$

Since $\mathcal{U}(\mathcal{D})$ is polyhedral and all constraints and the objective function are linear, there is no duality gap between (2.3) and (2.4).

Observe that (2.4) can be expressed as follows:

$$\max_{\beta, \lambda \geq 0} \beta \tag{2.5a}$$

$$\beta \leq \sum_{e \in E(G)} \lambda_e u_e, \forall u \in \mathcal{U}(\mathcal{D}) \tag{2.5b}$$

$$1 = \sum_{e \in E(G)} \lambda_e c_e \tag{2.5c}$$

We are going to approximately solve (2.5) using a cutting-plane algorithm where inequalities (2.5b) are iteratively added by using the α -approximation oracle. Let (β', λ') be a potential solution of (2.5), we can run the α -approximation of robust network design problem (2.2) with the cost vector λ' to get a solution $u^{ap}(\lambda')$. If $\beta' > \sum_{e \in E(G)} \lambda'_e u_e^{ap}(\lambda')$ we return the inequality $\beta \leq \sum_{e \in E(G)} \lambda'_e u_e^{ap}(\lambda')$, otherwise the algorithm stops and returns (β', λ') . We know from the separation-optimization equivalence theorem [Grötschel et al. (1988)] that (2.5) can be solved by making a polynomial number of calls to the separation oracle leading a globally polynomial-time algorithm. Notice that this happens if the separation oracle is exact. In our case, the oracle is only an approximate one,

implying that the cutting plane algorithm might be prematurely interrupted before obtaining the true optimum of (2.5). Observe however that this implies that the computing time is polynomially bounded. Let $(\tilde{\beta}, \tilde{\lambda})$ be the solution returned by the cutting-plane algorithm. Let (β^*, λ^*) be the true optimal solution of (2.5). The next lemma states that the returned solution is an α -approximation of the optimal solution.

Lemma 2.3.1. *The cutting-plane algorithm computes in polynomial-time a solution $\tilde{\beta}$ satisfying:*

$$\beta^* \leq \tilde{\beta} \leq \alpha\beta^*. \quad (2.6)$$

Proof. Observe that $\beta^* = \lambda^{*T}u(\lambda^*)$. Moreover, since (2.5) is equivalent to (2.4), we get that $\lambda^{*T}u(\lambda^*) = \beta^* \geq \tilde{\lambda}^T u(\tilde{\lambda})$. From the approximation factor of the oracle, one can write that $\tilde{\lambda}^T u^{ap}(\tilde{\lambda}) \leq \alpha \tilde{\lambda}^T u(\tilde{\lambda})$. Using the fact that no inequalities can be added for $(\tilde{\beta}, \tilde{\lambda})$, we get that $\tilde{\beta} \leq \tilde{\lambda}^T u^{ap}(\tilde{\lambda})$. Finally, since (β^*, λ^*) is feasible for (2.5), we obviously have $\tilde{\beta} \geq \beta^*$. Combining the 4 previous inequalities leads to (2.6). \square

The above lemma has many consequences.

Theorem 2.3.1. *Unless $P = NP$, the robust network design problem with linear costs cannot be approximated in polynomial-time within any constant ratio. Unless the ETH conjecture is false, the robust network design problem with linear costs cannot be approximated within $\Omega(\frac{\log n}{\log \log n})$.*

Proof. The result is an immediate consequence of Theorems 2.2.1, 2.2.2 and Lemma 2.3.1. \square

The theorem above answers a long-standing open question of [Chekuri (2007)]. All other inapproximability results proved for the congestion problem directly hold for the robust network design problem with linear cost.

Another important consequence is that the congestion problem can be approximated within $O(\log n)$. This result was already proved in [Räcke (2008)] using other techniques. In our case, the result is an immediate consequence of the $O(\log n)$ -approximation algorithm for the robust network design problem with linear cost provided by [Gupta (2004), Gupta and Könemann (2011)] and fully described in [Chekuri (2007), Goyal et al. (2009)].

Theorem 2.3.2. *[Räcke (2008)] Congestion can be approximated within $O(\log n)$.*

Notice that Theorem 2.2.2 tells us that the ratio $O(\log n)$ is tight.

Starting from the results of [Maggs et al. (1997), Bartal and Leonardi (1999)] showing the existence of instances for which the ratio $\frac{\text{cong}_{\text{sta}}}{\text{cong}_{\text{dyn}}}$ is $\Omega(\log n)$, one can also use

the reduction above to prove, differently from [Goyal et al. (2009)], the existence of instances for which the ratio $\frac{\text{lin}_{\text{sta}}}{\text{lin}_{\text{dyn}}}$ is $\Omega(\log n)$ where a linear cost is minimized.

Theorem 2.3.3. [Goyal et al. (2009)] *There are instances for which $\frac{\text{lin}_{\text{sta}}}{\text{lin}_{\text{dyn}}}$ is $\Omega(\log n)$.*

Proof. Similarly to $\mathcal{U}(\mathcal{D})$ defined when dynamic routing is considered, let $\mathcal{U}_{\text{sta}}(\mathcal{D})$ be the set of capacity vectors for which there exists a static fractional routing satisfying all demand vectors of \mathcal{D} . $\mathcal{U}_{\text{sta}}(\mathcal{D})$ is obviously a polyhedral set. The mathematical programs (2.3), (2.4) and (2.5) can be considered in the same way: we only have to replace $\mathcal{U}(\mathcal{D})$ by $\mathcal{U}_{\text{sta}}(\mathcal{D})$. All results stated above about the equivalence of (2.3), (2.4) and (2.5) still hold in the static case. Consider an instance from [Maggs et al. (1997), Bartal and Leonardi (1999)] for which $\frac{\text{cong}_{\text{sta}}}{\text{cong}_{\text{dyn}}}$ is $\Omega(\log n)$. cong_{sta} is computed from (2.5). Then there is at least one vector $\lambda^{\text{sta}} \geq 0$ and one vector $u^{\text{sta}} \in \mathcal{U}_{\text{sta}}(\mathcal{D})$ such that $\text{cong}_{\text{sta}} = \sum_{e \in E(G)} \lambda_e^{\text{sta}} u_e^{\text{sta}}$ and $\sum_{e \in E(G)} \lambda_e^{\text{sta}} c_e = 1$. This implies that u^{sta} is an optimal solution of the linear problem where we minimize $\sum_{e \in E(G)} \lambda_e^{\text{sta}} u_e$ under the condition $u \in \mathcal{U}_{\text{sta}}(\mathcal{D})$. We consequently have $\text{cong}_{\text{sta}} = \text{lin}_{\text{sta}}$ for the considered instance.

Let $u' \in \mathcal{U}(\mathcal{D})$ be an optimal solution minimizing the linear cost $\sum_{e \in E(G)} \lambda_e^{\text{sta}} u_e$ under dynamic routing. In other words, $\text{lin}_{\text{dyn}} = \sum_{e \in E(G)} \lambda_e^{\text{sta}} u'_e$ when the coefficients of the objective function are λ^{sta} . Moreover, we know from (2.5) that cong_{dyn} is obtained by maximizing through λ , implying that $\text{cong}_{\text{dyn}} \geq \sum_{e \in E(G)} \lambda_e^{\text{sta}} u'_e = \text{lin}_{\text{dyn}}$. Using that $\frac{\text{cong}_{\text{sta}}}{\text{cong}_{\text{dyn}}}$ is $\Omega(\log n)$, we get that $\frac{\text{lin}_{\text{sta}}}{\text{lin}_{\text{dyn}}}$ is $\Omega(\log n)$ for the same instance where the linear objective function is defined through λ^{sta} . \square

Although we consider only undirected graphs in this section, proposition 2.3.1 can be proved also for directed graphs by the same proof without modifications.

2.4 Inapproximability with restrictions on a set of given paths

In this section we consider the path restricted problem presented in the introduction 1.3.2 where for each commodity $h \in \mathcal{H}$ we are given a set of path \mathcal{P}_h going from $s(h)$ to $t(h)$ in input. And the flow of the commodity must be routed along only those paths.

First, observe that in the proof of Lemma 2.1.1, the minimum congestion instances built there are such that some commodities can be routed along many paths. For example, in graph G_1 (Figure 2.1), commodity h_0 (between s and t) can use up to m paths. Second, consider an instance of the minimum congestion problem where only one path is given for each commodity. Then computing the minimum congestion is easy since we only have to compute $\max_{d \in \mathcal{D}} \sum_{h \in \mathcal{H}_e} d_h$ where \mathcal{H}_e denotes the set of commodities routed through e . The congestion is just given by $\max_{e \in E(G)} \frac{1}{c_e} \max_{d \in \mathcal{D}} \sum_{h \in \mathcal{H}_e} d_h$.

Combining these two observations, one can wonder whether the difficulty of the congestion problem is simply due to the number of possible paths that can be used by

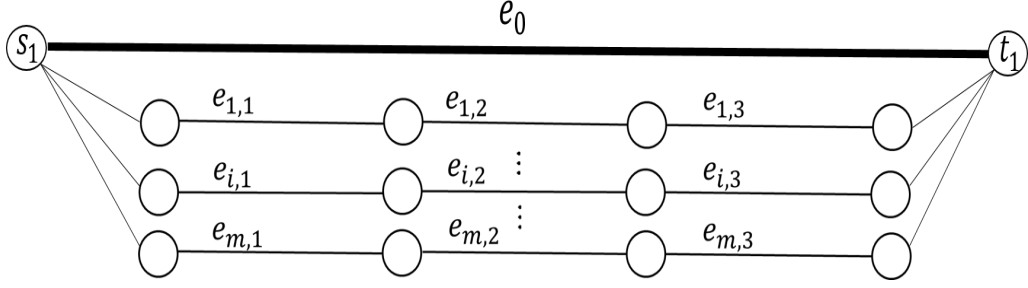


Fig. 2.2 The graph G' related to the proof of Theorem 2.4.1

each commodity. We will show that the problem is still difficult even if each commodity can be routed along at most two fixed given paths.

Theorem 2.4.1. *Unless $P = NP$, for some constant $c > 1$, minimum congestion cannot be approximated within a ratio c even if each commodity can be routed along at most two given paths.*

Proof. The proof is a simple modification of the proof of Lemma 2.1.1). We are going to slightly modify graph G in such a way that at most 2 paths are allowed for each commodity. Given a 3-SAT formula φ with m clauses, we construct $G', \mathcal{H}', \mathcal{D}'$ as follows. We first create two nodes s_1 and t_1 and an edge e_0 between s_1 and t_1 of capacity $m\rho$ (ρ is the constant in PCP theorem). Then for each clause index $i = 1, \dots, m$, as in Lemma 2.1.1, we create 3 consecutive edges $e_{i,j}$ ($j = 1, 2, 3$) and a commodity $h_{i,j}$ between $s(e_{i,j})$ and $t(e_{i,j})$ that is allowed to be routed only through the path using edge $e_{i,j}$. We also add one edge between $s(e_{i,1})$ and s_1 and one edge connecting t_1 and $t(e_{i,3})$ of infinite capacity and a commodity $h_{i,0}$ between $s(e_{i,1})$ and $t(e_{i,3})$ with a demand $d_{h_{i,0}} = 1$. $h_{i,0}$ is allowed to be routed only through the path P_i containing the edges $(e_{i,1}, e_{i,2}, e_{i,3})$ and the path going through s_1, e_0 and t_1 (see Figure 2.2). We consider auxiliary variables ξ_l for each literal l . We add constraints $\xi_l + \xi_{\neg l} = 1$ and $d_{h_{i,j}} = \xi_{l_{i,j}}$.

If $val(\varphi) < \rho$ there are at least $m(1 - \rho)$ commodities $h_{i,0}$ that can be routed on the paths P_i and the remaining $m\rho$ can be routed on the edge e_0 . This implies that each extreme point of \mathcal{D}' can be routed with congestion ≤ 1 . Notice that the observation made in the proof of Lemma 2.1.1 about extreme points is still valid here: extreme points correspond to 0 – 1 values of the variables ξ_l .

If $val(\varphi) = 1$, then there is a cut and a demand vector d (corresponding to the truth assignment satisfying φ) such that the capacity of the cut is $m\rho + m$ and the demand that needs to cross the cut is $2m$. There is consequently at least one edge of congestion greater than or equal to $\frac{2m}{(1+\rho)m} = \frac{2}{1+\rho}$. By taking $c < \frac{2}{1+\rho}$ we get the wanted result. \square

Finally, observe that the result above can also be stated for the linear cost case using again the Lagrange based reduction of the previous section.

Corollary 2.4.1. *Unless $P = NP$, for some positive constant k , robust network design with linear costs is difficult to approximate within a ratio k even if each commodity can be routed along at most two given paths.*

CHAPTER 3

Robust network design problem in directed graphs

In the previous chapter we have seen some results related to inapproximability of the robust network design problem in undirected graphs. In this chapter, we consider directed graphs. We first consider the static/dynamic gap. As already mentioned in Chapter 1, an upper bound of $O(\sqrt{k}n^{1/4} \log n)$ from [Hajiaghayi, Kleinberg, Räcke and Leighton (2007)] has been proven for the static to dynamic gap. On the other hand, a lower bound of $\Omega(\sqrt{k})$ [Azar et al. (2003)] and [Ene et al. (2016)] is also known. The main question we are focusing in this chapter is the following.

- Is it possible to improve the upper or the lower bound on the static/dynamic gap?

To answer the question, we first tried to obtain an improved upper bound by exhibiting simple classes of demand polytopes where the static routing solution is as good as an optimal dynamic routing solution. Then we approximate an arbitrary demand polytope by those simpler polytope. We found two classes of polytopes having this property: down monotone boxes and down monotone simplices. We can obtain an upper bound of k on the static/dynamic gap by approximating the demand polytope by a box or by a simplex. The class of demand polytopes with the worst approximation by a simplex are the boxes and conversely the polytopes with the worst approximation by a boxes are the simplex. This suggested that by using the best of those two bounds we might obtain an improved upper bound on the static/dynamic gap. In the special case of demand polytopes that remain invariant by permutation of the commodities this strategy leads to a proof that the static/dynamic gap can be at most \sqrt{k} . However, as we will see, this strategy does not permit to obtain an upper bound better than $\frac{k}{2}$ in general.

Nonetheless, the insights and intermediate results gained during this first attempt proved to be crucial ingredients for proof of the $O(\sqrt{k})$ upper bound for the general case. The next step to improve the upper bound was to reinterpret a result of [Bertsimas and Goyal (2013)] in geometric terms. This result can be reformulated as follows: “An arbitrary down monotone polytope \mathcal{D} can be approximated by a product of a box and a simplex within a factor of $O(\sqrt{k})$ where k is the dimension of the polytope”. Using this result we were able to prove that the congestion of an optimal solution under static routing cannot worse than $O(\sqrt{k})$ times the congestion under a dynamic routing. This

upper bound on the static dynamic gap can also be proved in the same way for the variants of the problem with flow restrictions presented in Chapter 1 in Section 1.3 (except for the tree routing restriction).

In the second part of the chapter, we investigate the approximability status of the edge restricted problem presented in Section 1.3.3. Since this problem can be seen as a generalization of the robust network design problem without flow restriction, the $\Omega(\log \log n / \log n)$ lower bound proved in the previous chapter also holds for this problem. On the other hand the best approximation factor for this problem is the $O(\sqrt{k})$ that we will prove in this section. Despite those results, there were still a wide gap between those two bounds. To obtain better inapproximability lower bounds, we will introduce a product between edge restricted instances. This will allow us to prove inapproximability results for this edge restricted problem, specifically we are going to prove the following.

- Unless $NP \subseteq QP$ (resp. $NP \subseteq SUBEXP$) this problem cannot be approximated within a factor better than $\Omega(2^{\log^{1-\epsilon} k})$ for any $\epsilon > 0$ (resp. $\Omega(k^{\frac{c'}{\log \log k}})$ for some $c' > 0$).

3.1 Preliminaries

In this section, we recall the formal definitions of the *robust network design* problems considered in this chapter. The notation introduced in Chapter 1 is used.

Remember that an instance \mathcal{I} of the congestion minimization problem is denoted by $\mathcal{I} = (G, c, \mathcal{H}, \mathcal{D})$, while $\text{cong}(x, d)$ still denotes the maximum congestion over all links $e \in E$, i.e. $\text{cong}(x, d) = \max_{e \in E} \sum_{h \in \mathcal{H}} \frac{x_{h,e} d_h}{c_e}$. Two problems can be considered depending on whether the routing can be adapted to each demand vector d in \mathcal{D} or if only one fixed routing $x \in \mathcal{X}$ can be used. The dynamic congestion is formally defined as: $\text{cong}_{\text{dyn}}(\mathcal{I}) = \max_{d \in \mathcal{D}} \min_{x \in \mathcal{X}} \text{cong}(x, d)$. In the second case, the routing is said to be *static* (or *oblivious*). This static congestion is formally defined as: $\text{cong}_{\text{sta}}(\mathcal{I}) = \min_{x \in \mathcal{X}} \max_{d \in \mathcal{D}} \text{cong}(x, d)$. When clear from the context, we might use notation $\text{cong}_{\text{dyn}}(\mathcal{D})$ and $\text{cong}_{\text{sta}}(\mathcal{D})$ to insist on the dependency on \mathcal{D} when all other parameters of the instance \mathcal{I} are fixed.

In the same way, the robust linear reservation problem is defined where a positive cost vector $(\lambda_e)_{e \in E}$ is given, and we aim to reserve a capacity $u_e \geq 0$ on each link e such that $\sum_{e \in E} \lambda_e u_e$ is minimized and $\sum_{h \in \mathcal{H}} x_{h,e} d_h \leq u_e$ holds for any demand vector d . An instance can then be denoted by $(G, \lambda, \mathcal{H}, \mathcal{D})$. We also have two variants depending on routing. The optimal cost is then denoted by $\text{lin}_{\text{dyn}}(\mathcal{I})$ (or $\text{lin}_{\text{dyn}}(\mathcal{D})$) and $\text{lin}_{\text{sta}}(\mathcal{I})$ (or $\text{lin}_{\text{sta}}(\mathcal{D})$).

For concise notation, the four variants of the robust optimization problems considered in this chapter will simply be denoted by lin_{sta} , lin_{dyn} , cong_{sta} and cong_{dyn} .

All previous definitions still make sense even when \mathcal{D} is not a polytope. However, the next lemma tells us that the optimal objective value does not increase when the uncertainty set \mathcal{S} is replaced by its convex-hull (this lemma can be considered as a folklore result that is implicitly used in many robust optimization papers).

Lemma 3.1.1. *Let $\mathcal{S} \subset \mathbf{R}_+^{\mathcal{H}}$ be a compact set. Then $\text{cong}_{\text{sta}}(\mathcal{S}) = \text{cong}_{\text{sta}}(\text{conv}(\mathcal{S}))$, $\text{cong}_{\text{dyn}}(\mathcal{S}) = \text{cong}_{\text{dyn}}(\text{conv}(\mathcal{S}))$, $\text{lin}_{\text{sta}}(\mathcal{S}) = \text{lin}_{\text{sta}}(\text{conv}(\mathcal{S}))$, and $\text{lin}_{\text{dyn}}(\mathcal{S}) = \text{lin}_{\text{dyn}}(\text{conv}(\mathcal{S}))$.*

Proof. Since $\mathcal{S} \subseteq \text{conv}(\mathcal{S})$, we have $\text{cong}_{\text{sta}}(\mathcal{S}) \leq \text{cong}_{\text{sta}}(\text{conv}(\mathcal{S}))$ and $\text{cong}_{\text{dyn}}(\mathcal{S}) \leq \text{cong}_{\text{dyn}}(\text{conv}(\mathcal{S}))$. The same holds for the robust linear cost problem. Moreover, given a static routing solution x and the corresponding reservation vector u , we have $\sum_{h \in \mathcal{H}} x_{h,e} d_h \leq u_e$ for any $d \in \mathcal{S}$. Consider any point d' of $\text{conv}(\mathcal{S})$ written as $d' = \sum_{d \in \mathcal{S}} \alpha_d d$ ($\alpha_d \geq 0, \sum_{d \in \mathcal{S}} \alpha_d = 1$). By multiplying the previous inequalities by α_d and summing them all, we get that $\sum_{h \in \mathcal{H}} x_{h,e} d'_h \leq u_e$ implying that x and u are feasible. Therefore, we have $\text{lin}_{\text{sta}}(\mathcal{S}) = \text{lin}_{\text{sta}}(\text{conv}(\mathcal{S}))$. The proof can be easily extended to the dynamic routing version and to the congestion objective function. \square

Let us now focus on the connection between the congestion problem and the linear cost problem.

We know from [Goyal et al. (2009)] (Proposition 1.5.1) that if the static to dynamic ratio is less than or equal to α for the congestion problem, then the same applies to the robust linear reservation problem.

A converse result is presented in Chapter 2 (Proposition 2.3.1). While the proof in Chapter 2 (and [Al-Najjar et al. (2021)]) was given in the context of undirected graphs, it can be repeated verbatim for the directed case.

To close this section, let us recall some notation and assumptions that will be used in the rest of the chapter. The uncertainty set (i.e., the set of demand vectors) \mathcal{D} is assumed to be polyhedral and down monotone (i.e., if $d \in \mathcal{D}$, then $d' \in \mathcal{D}$ for any $0 \leq d' \leq d$). Let $d^{\text{max}}(\mathcal{D})$ be the vector representing the maximum commodity values (i.e., $d_h^{\text{max}}(\mathcal{D}) = \max_{d \in \mathcal{D}} d_h$). We will naturally assume that $d_h^{\text{max}} > 0$ for any $h \in \mathcal{H}$ since otherwise the commodity can just be ignored. When the polytope \mathcal{D} is clear from the context, we just write d^{max} (instead of $d^{\text{max}}(\mathcal{D})$).

Let I, J be some set of indices. For a vector $v \in \mathbf{R}^{I \times J}$ and $i \in I$ we denote by $v_{i,\cdot}$ the vector $w \in \mathbf{R}^J$ defined by $w_j = v_{i,j}$. Given a set $X \in \mathbf{R}^I$ and $\lambda \geq 0$, we denote by λX the set $\{\lambda x | x \in X\}$.

3.2 Gap between static and dynamic routing strategies

We are going to prove Theorem 3.2.1 stating that compared to dynamic routing, when static routing is considered, congestion is multiplied by a factor less than or equal to $\sqrt{8k}$. This result improves the upper bound $O(\sqrt{k}n^{1/4} \log n)$ from [Hajiaghayi, Kleinberg, Räcke and Leighton (2007)]. It implies that the gap between static and dynamic congestion is $O(\sqrt{k}) = O(n)$. By combining Proposition 1.5.1 with Theorem 3.2.1, we also obtain similar results for the minimization of a linear reservation cost, i.e., that $\text{lin}_{\text{sta}}(\mathcal{D}) \leq \sqrt{8k} \cdot \text{lin}_{\text{dyn}}(\mathcal{D})$.

Theorem 3.2.1. $\text{cong}_{\text{sta}}(\mathcal{D}) \leq \sqrt{8k} \cdot \text{cong}_{\text{dyn}}(\mathcal{D})$. Therefore $\frac{\text{cong}_{\text{sta}}(\mathcal{D})}{\text{cong}_{\text{dyn}}(\mathcal{D})} = O(n)$.

To derive an upper bound for the ratio $\text{cong}_{\text{sta}}(\mathcal{D})/\text{cong}_{\text{dyn}}(\mathcal{D})$, our strategy first consists in approximating the uncertainty set either by a box or a simplex where $\text{cong}_{\text{sta}}(\mathcal{D}) = \text{cong}_{\text{dyn}}(\mathcal{D})$. While this method yields an $O(k)$ upper bound, we obtain further improvement by partitioning the set of commodities into two sets $\mathcal{H}_1, \mathcal{H}_2$ and considering a box approximation for \mathcal{D}_1 and a simplex approximation for \mathcal{D}_2 , where \mathcal{D}_1 and \mathcal{D}_2 are respectively the projections of \mathcal{D} on $\mathbb{R}^{\mathcal{H}_1}$ and $\mathbb{R}^{\mathcal{H}_2}$.

To prove Theorem 3.2.1, we first present some preliminary lemmas.

Lemma 3.2.1 states that if the uncertainty set \mathcal{D} can be well approximated by another set \mathcal{D}' for which $\text{cong}_{\text{sta}}(\mathcal{D}') = \text{cong}_{\text{dyn}}(\mathcal{D}')$, then $\text{cong}_{\text{sta}}(\mathcal{D})$ gives a good approximation of $\text{cong}_{\text{dyn}}(\mathcal{D})$.

Lemma 3.2.1. *Let \mathcal{D} and \mathcal{D}' be two compact subsets of $\mathbb{R}_+^{\mathcal{H}}$ and $\alpha \in \mathbb{R}_+$ such that $\mathcal{D}' \subseteq \mathcal{D} \subseteq \alpha\mathcal{D}'$ and $\text{cong}_{\text{sta}}(\mathcal{D}') = \text{cong}_{\text{dyn}}(\mathcal{D}')$. Then $\text{cong}_{\text{sta}}(\mathcal{D}) \leq \alpha \cdot \text{cong}_{\text{dyn}}(\mathcal{D})$.*

Proof. The proof of this lemma relies on two simple facts. The first one is that if we scale the demand values by a factor α , then the congestion (either static or dynamic) is also scaled by the same factor α . The second fact is that cong_{dyn} and cong_{sta} are increasing in \mathcal{D} . In other words, if \mathcal{D}_1 and \mathcal{D}_2 are two subsets of $\mathbb{R}_+^{\mathcal{H}}$ such that $\mathcal{D}_1 \subseteq \mathcal{D}_2$, then $\text{cong}_{\text{dyn}}(\mathcal{D}_1) \leq \text{cong}_{\text{dyn}}(\mathcal{D}_2)$ and $\text{cong}_{\text{sta}}(\mathcal{D}_1) \leq \text{cong}_{\text{sta}}(\mathcal{D}_2)$. Combining the two facts, we can write the following:

$$\text{cong}_{\text{sta}}(\mathcal{D}) \leq \text{cong}_{\text{sta}}(\alpha\mathcal{D}') = \alpha \cdot \text{cong}_{\text{sta}}(\mathcal{D}') = \alpha \cdot \text{cong}_{\text{dyn}}(\mathcal{D}') \leq \alpha \cdot \text{cong}_{\text{dyn}}(\mathcal{D}) \quad (3.1)$$

□

We now provide in Lemmas 3.2.2 and 3.2.3 two classes of polytopes, based on box and simplex sets, for which $\text{cong}_{\text{sta}}(\mathcal{D}) = \text{cong}_{\text{dyn}}(\mathcal{D})$.

For a vector $d^{\text{max}} \in \mathbb{R}_+^{\mathcal{H}}$, let $\mathcal{B}(d^{\text{max}})$ be the box set defined by $\{d \in \mathbb{R}^{\mathcal{H}} \mid 0 \leq d \leq d^{\text{max}}\}$.

Lemma 3.2.2. *Let $\mathcal{D} = \mathcal{B}(d^{\text{max}})$ for some $d^{\text{max}} \in \mathbb{R}_+^{\mathcal{H}}$. Then $\text{cong}_{\text{dyn}}(\mathcal{D}) = \text{cong}_{\text{sta}}(\mathcal{D})$.*

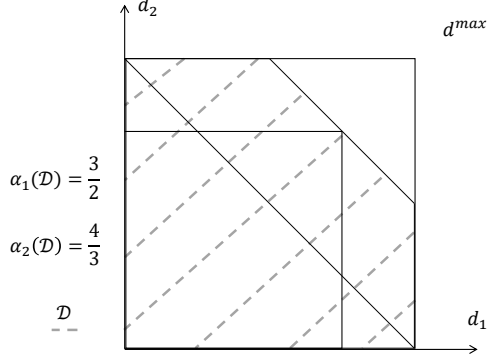


Fig. 3.1 Approximation using boxes and simplices: example of a 2-dimensional demand polytope \mathcal{D} .

Proof. For a routing $x \in \mathcal{X}$ and a demand vector $d \in \mathcal{D}$, we have $\text{cong}(x, d) \leq \text{cong}(x, d^{\max})$. Since $d^{\max} \in \mathcal{D}$, it implies that $\max_{d \in \mathcal{D}} \text{cong}(x, d) = \text{cong}(x, d^{\max})$. Minimizing both sides of the equality over $x \in \mathcal{X}$, we get that $\text{cong}_{\text{sta}}(\mathcal{D}) = \min_{x \in \mathcal{X}} \text{cong}(x, d^{\max})$. We can also write that $\min_{x \in \mathcal{X}} \text{cong}(x, d) \leq \min_{x \in \mathcal{X}} \text{cong}(x, d^{\max})$. Taking the maximum over all $d \in \mathcal{D}$ leads to $\text{cong}_{\text{dyn}}(\mathcal{D}) = \max_{d \in \mathcal{D}} \min_{x \in \mathcal{X}} \text{cong}(x, d) \leq \min_{x \in \mathcal{X}} \text{cong}(x, d^{\max})$. Since $d^{\max} \in \mathcal{D}$, the previous inequality becomes $\text{cong}_{\text{dyn}}(\mathcal{D}) = \min_{x \in \mathcal{X}} \text{cong}(x, d^{\max})$. \square

For a vector $d \in \mathbb{R}_+^{\mathcal{H}}$, let $\Delta(d)$ be the simplex set whose vertices are the zero vector and the k vectors $d_h e_h$ where e_h denotes the vector in $\mathbb{R}_+^{\mathcal{H}}$ with a component of 1 for commodity h and 0 otherwise. Formally, we have $\Delta(d) = \text{conv}(\{d_h e_h | h \in \mathcal{H}\} \cup \{0\})$.

Lemma 3.2.3. *Let $\mathcal{D} = \Delta(d^{\max})$ where $d^{\max} \in \mathbb{R}_+^{\mathcal{H}}$. Then $\text{cong}_{\text{dyn}}(\mathcal{D}) = \text{cong}_{\text{sta}}(\mathcal{D})$.*

Proof. Assume that $\text{cong}_{\text{dyn}}(\mathcal{D})$ has been computed and consider the obtained dynamic routing. The extreme points of \mathcal{D} are the demand vectors $\{d_h^{\max} e_h | h \in \mathcal{H}\} \cup \{0\}$. For each demand vector $d_h^{\max} e_h$, we consider the flow $x_{h,\cdot}$ representing its routing. Let us build a static routing x just by routing each commodity h in accordance to $x_{h,\cdot}$. By construction, taking the extreme points of \mathcal{D} , we have $\text{cong}_{\text{sta}}(\{d_h^{\max} e_h | h \in \mathcal{H}\} \cup \{0\}) = \text{cong}_{\text{dyn}}(\{d_h^{\max} e_h | h \in \mathcal{H}\} \cup \{0\})$. By considering the convex-hulls and applying Lemma 3.1.1, we get that $\text{cong}_{\text{dyn}}(\mathcal{D}) = \text{cong}_{\text{sta}}(\mathcal{D})$. \square

Let $\alpha_1(\mathcal{D}) = \max_{d \in \mathcal{D}} \sum_{h \in \mathcal{H}} \frac{d_h}{d_h^{\max}}$ (remember that $d_h^{\max} = \max_{d \in \mathcal{D}} d_h$). It is then clear that $\Delta(d^{\max}) \subseteq \mathcal{D} \subseteq \alpha_1(\mathcal{D}) \Delta(d^{\max})$. Consider the box $\mathcal{B}(d^{\max})$ and let $\alpha_2(\mathcal{D})$ be the smallest factor α such that d^{\max}/α belongs to \mathcal{D} . In other words, $\alpha_2(\mathcal{D})$ represents the best approximation ratio that can be obtained through boxes. We obviously have $\frac{1}{\alpha_2(\mathcal{D})} \mathcal{B}(d^{\max}) \subseteq \mathcal{D} \subseteq \mathcal{B}(d^{\max})$. Figure 3.1 illustrates the approximations by boxes and simplices for a 2-dimensional demand polytope \mathcal{D} .

Since $\frac{1}{k} \mathcal{B}(d^{\max}) \subseteq \Delta(d^{\max}) \subseteq \mathcal{D} \subseteq \mathcal{B}(d^{\max})$, $\alpha_2(\mathcal{D})$ is always less than or equal to k . And by definition, $\alpha_1(\mathcal{D})$ is also less than or equal to k .

It is easy to check that the upper bound k is reached since $\alpha_1(\mathcal{B}(d^{max})) = k$ and $\alpha_2(\Delta(d^{max})) = k$. In other words, using box and simplex approximations with the approach above, we cannot expect to prove a better upper bound for the ratio $\text{cong}_{\text{sta}}(\mathcal{D})/\text{cong}_{\text{dyn}}(\mathcal{D})$ for arbitrary uncertainty sets.

A more refined strategy is to take the best of the two bounds $\alpha_1(\mathcal{D})$, $\alpha_2(\mathcal{D})$. The next proposition states that a better bound is obtained if \mathcal{D} is permutation-invariant (i.e., by permuting the components of any vector d of \mathcal{D} we always get a vector inside \mathcal{D}).

Proposition 3.2.1. *If \mathcal{D} is permutation-invariant then $\min\{\alpha_1(\mathcal{D}), \alpha_2(\mathcal{D})\} \leq \sqrt{k}$.*

Proof. Let d^* be the demand maximizing $\max_{d \in \mathcal{D}} \sum_{h \in \mathcal{H}} \frac{d_h}{d^{max}_h}$. Since \mathcal{D} is permutation-invariant, $d_h^{max} = d_{h'}^{max}$ for all $h, h' \in \mathcal{H}$ and d^* can be chosen such that $d_h^* = d_{h'}^*$ for all $h, h' \in \mathcal{H}$. Consequently, we have $\alpha_1(\mathcal{D}) \leq k \frac{d_{h_0}^*}{d^{max}_{h_0}}$. Moreover, since $\frac{d_{h_0}^*}{d^{max}_{h_0}} \mathcal{B}(d^{max}) \subseteq \mathcal{D} \subseteq \mathcal{B}(d^{max})$ we also have $\alpha_2(\mathcal{D}) \leq \frac{d_{h_0}^{max}}{d_{h_0}^*}$. Therefore, using notation $x = \frac{d_{h_0}^*}{d_{h_0}^{max}}$, we get that $\min\{\alpha_1(\mathcal{D}), \alpha_2(\mathcal{D})\} \leq \min\{kx, \frac{1}{x}\}$ and x is such that $0 \leq x \leq 1$. To conclude, observe that $\max_{0 \leq x \leq 1} \min\{kx, \frac{1}{x}\} = \sqrt{k}$. \square

One can wonder whether a general $O(\sqrt{k})$ bound can be obtained by trying to find a better upper bound for $\min\{\alpha_1(\mathcal{D}), \alpha_2(\mathcal{D})\}$. The following example, on a specific polytope \mathcal{D} , shows that this is not possible. Let \mathcal{D} be the product of a box $\mathcal{B}(d^1)$ of dimension $k/2$ and a simplex $\Delta(d^2)$ of the same dimension. Using the remark above we know that $\alpha_1(\mathcal{B}(d^1)) = k/2$ and $\alpha_2(\Delta(d^2)) = k/2$ implying that $\alpha_1(\mathcal{D}) \geq k/2$ and $\alpha_2(\mathcal{D}) \geq k/2$.

To overcome this difficulty, we are going to partition the set of commodities \mathcal{H} into two well-chosen subsets \mathcal{H}_1 and \mathcal{H}_2 , then we approximate \mathcal{D}_1 (resp. \mathcal{D}_2) defined as the projection of \mathcal{D} on $\mathbb{R}^{\mathcal{H}_1}$ (resp. $\mathbb{R}^{\mathcal{H}_2}$) using a simplex (resp. a box). The algorithm used to partition the set of commodities is an adaptation of an algorithm of [Bertsimas and Goyal (2012)] proposed in a different context. We will also slightly improve the analysis of this algorithm ($\sqrt{8k}$ instead of $3\sqrt{k}$).

Let us start with Lemma 3.2.4 where we show how an approximation of cong_{dyn} in \mathcal{D} can be obtained from cong_{sta} using the approximations related to \mathcal{D}_1 and \mathcal{D}_2 .

Lemma 3.2.4. *Let $\mathcal{H}_1, \mathcal{H}_2$ be a partition of \mathcal{H} and $\mathcal{D}_1, \mathcal{D}_2$ be the projection of \mathcal{D} on $\mathbb{R}^{\mathcal{H}_1}$ and $\mathbb{R}^{\mathcal{H}_2}$. Suppose that for some $\alpha_1, \alpha_2 \geq 1$ we have $\text{cong}_{\text{sta}}(\mathcal{D}_1) \leq \alpha_1 \text{cong}_{\text{dyn}}(\mathcal{D}_1)$ and $\text{cong}_{\text{sta}}(\mathcal{D}_2) \leq \alpha_2 \text{cong}_{\text{dyn}}(\mathcal{D}_2)$, then $\text{cong}_{\text{sta}}(\mathcal{D}) \leq (\alpha_1 + \alpha_2) \text{cong}_{\text{dyn}}(\mathcal{D})$.*

Proof. We first show that we have $\text{cong}_{\text{sta}}(\mathcal{D}) \leq \text{cong}_{\text{sta}}(\mathcal{D}_1) + \text{cong}_{\text{sta}}(\mathcal{D}_2)$. Let us

denote \mathcal{X}_1 (resp. \mathcal{X}_2) as the set of routing for the commodities in \mathcal{H}_1 (resp. \mathcal{H}_2).

$$\begin{aligned}
\text{cong}_{\text{sta}}(\mathcal{D}) &= \min_{x \in \mathcal{X}} \max_{d \in \mathcal{D}} \text{cong}(x, d) \\
&\leq \min_{x \in \mathcal{X}} \max_{d^1 \in \mathcal{D}_1, d^2 \in \mathcal{D}_2} \text{cong}(x, d^1) + \text{cong}(x, d^2) \\
&= \min_{x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2} (\max_{d^1 \in \mathcal{D}_1} \text{cong}(x_1, d^1) + \max_{d^2 \in \mathcal{D}_2} \text{cong}(x_2, d^2)) \\
&= \min_{x_1 \in \mathcal{X}_1} \max_{d^1 \in \mathcal{D}_1} \text{cong}(x_1, d^1) + \min_{x_2 \in \mathcal{X}_2} \max_{d^2 \in \mathcal{D}_2} \text{cong}(x_2, d^2) \\
&= \text{cong}_{\text{sta}}(\mathcal{D}_1) + \text{cong}_{\text{sta}}(\mathcal{D}_2)
\end{aligned}$$

We now prove the lemma: $\text{cong}_{\text{sta}}(\mathcal{D}) \leq \text{cong}_{\text{sta}}(\mathcal{D}_1) + \text{cong}_{\text{sta}}(\mathcal{D}_2) \leq \alpha_1 \text{cong}_{\text{dyn}}(\mathcal{D}_1) + \alpha_2 \text{cong}_{\text{dyn}}(\mathcal{D}_2) \leq (\alpha_1 + \alpha_2) \text{cong}_{\text{dyn}}(\mathcal{D})$. \square

Let us now present Algorithm 1 that can be seen as a direct adaptation of the partitioning algorithm of [Bertsimas and Goyal (2012)] (Algorithm \mathcal{A} , Figure 1) for our dynamic routing problem. It has initially been introduced for the analysis of affine policies in a class of two-stage adaptive linear optimization problems. The main idea of Algorithm 1 is to partition the set of commodities into two sets \mathcal{H}_1 and \mathcal{H}_2 and to produce a vector $\beta \in \mathbb{R}_+^{\mathcal{H}}$ such that $\max_{d \in \mathcal{D}} \sum_{h \in \mathcal{H}_1} \frac{d_h}{d_h^{\max}} \leq \gamma \sqrt{k}$ (i.e., $\alpha_1(\mathcal{D}_1) \leq \gamma \sqrt{k}$ for $\gamma > 0$) and $\beta_h \geq d_h^{\max}$ for any $h \in \mathcal{H}_2$. The returned vector β is built as a sum of at most Z points of \mathcal{D} where Z is the number of iterations of the algorithm. Since the vector $\frac{1}{Z} \beta$ belongs to \mathcal{D} , we deduce that $\alpha_2(\mathcal{D}_2) \leq Z$. We will show in Lemma 3.2.5 that Z is less than or equal to $2 \frac{\sqrt{k}}{\gamma}$ leading to $\alpha_2(\mathcal{D}_2) \leq 2 \frac{\sqrt{k}}{\gamma}$. Notice that γ is equal to 1 in the original algorithm of [Bertsimas and Goyal (2012)]. Let us describe more precisely the different steps of Algorithm 1. At iteration i , $\mathcal{H}_1^i, \mathcal{H}_2^i$ denote the current partitions of commodities while $\mathcal{D}_1^i, \mathcal{D}_2^i$ denote the projections of \mathcal{D} on $\mathbb{R}^{\mathcal{H}_1^i}$ and $\mathbb{R}^{\mathcal{H}_2^i}$. A vector b^i is also defined and used to update $\mathcal{H}_1^i, \mathcal{H}_2^i$. We start with $\mathcal{H}_1^0 = \mathcal{H}, \mathcal{H}_2^0 = \emptyset$ and $b^0 = 0$.

If $\alpha_1(\mathcal{D}_1^i) > \gamma \sqrt{k}$ then we consider a traffic vector u^i maximizing $\sum_{h \in \mathcal{H}_1^{i-1}} \frac{d_h}{d_h^{\max}}$, otherwise a partition is returned. The vector u^i is then used to update b^i (lines 5-7). Observe that only the components related to commodities inside \mathcal{H}_1^{i-1} are updated while the others do not change. This means that the returned vector $\beta = \sum_{1 \leq i \leq Z} u^i$ (line 19) is such that $\beta \geq b^Z$. The sets \mathcal{H}_1^i and \mathcal{H}_2^i are updated by moving each commodity $h \in \mathcal{H}_1^{i-1}$ to \mathcal{H}_2^i if $b_h^i \geq d_h^{\max}$ (lines 8-15). Notice that we always have $\mathcal{H}_1^i \subseteq \mathcal{H}_1^{i-1}$. It is then clear that when the algorithm stops, the obtained partition satisfies what is announced above. The only fact that remains to be proved is that the number of iterations Z is bounded by $2 \frac{\sqrt{k}}{\gamma}$.

Lemma 3.2.5. *For any $\gamma > 0$, the commodity set \mathcal{H} can be partitioned in two subsets $\mathcal{H}_1, \mathcal{H}_2$ such that $\alpha_1(\mathcal{D}_1) \leq \gamma \sqrt{k}$ and $\alpha_2(\mathcal{D}_2) \leq \frac{2\sqrt{k}}{\gamma}$ where $\mathcal{D}_1, \mathcal{D}_2$ are the projections of \mathcal{D} on $\mathbb{R}^{\mathcal{H}_1}$ and $\mathbb{R}^{\mathcal{H}_2}$.*

Algorithm 1 Commodity partitioning algorithm (adapted from [Bertsimas and Goyal (2012)])

```

1: Initialize  $i \leftarrow 0, \mathcal{H}_1^0 \leftarrow \mathcal{H}, \mathcal{H}_2^0 \leftarrow \emptyset, b^0 \leftarrow 0$ 
2: while  $\alpha_1(\mathcal{D}_1^i) > \gamma\sqrt{k}$  do
3:    $i \leftarrow i + 1$ 
4:    $u^i \in \arg \max_{d \in \mathcal{D}} \sum_{h \in \mathcal{H}_1^{i-1}} \frac{d_h}{d_h^{max}}$ 
5:   for all  $h \in \mathcal{H}$  do
6:      $b_h^i = \begin{cases} b_h^{i-1} + u_h^i & \text{if } h \in \mathcal{H}_1^{i-1} \\ b_h^{i-1} & \text{otherwise} \end{cases}$ 
7:   end for
8:   for all  $h \in \mathcal{H}_1^{i-1}$  do
9:     if  $b_h^i \geq d_h^{max}$  then
10:       $\mathcal{H}_1^i \leftarrow \mathcal{H}_1^{i-1} \setminus \{h\}$ 
11:       $\mathcal{H}_2^i \leftarrow \mathcal{H}_2^{i-1} \cup \{h\}$ 
12:     else
13:       $\mathcal{H}_1^i \leftarrow \mathcal{H}_1^{i-1}, \mathcal{H}_2^i \leftarrow \mathcal{H}_2^{i-1}$ 
14:     end if
15:   end for
16: end while
17:  $Z \leftarrow i, \mathcal{H}_1 \leftarrow \mathcal{H}_1^Z, \mathcal{H}_2 \leftarrow \mathcal{H}_2^Z$ 
18:  $\beta \leftarrow \sum_{1 \leq i \leq Z} u^i$ 

```

Proof. We only have to prove that $Z \leq \frac{2\sqrt{k}}{\gamma}$. This can be done by slightly modifying the proof of Lemma 10 of [Bertsimas and Goyal (2012)].

We first argue that $b_h^Z \leq 2d_h^{max}$ for all $h \in \mathcal{H}$. For $h \in \mathcal{H}$, let $i(h)$ be the last iteration number when $h \in \mathcal{H}_1^i$. Therefore we have $b_h^{i(h)-1} \leq d_h^{max}$. Also $u^{i(h)} \leq d_h^{max}$ leading to $b_h^{i(h)} \leq 2d_h^{max}$. Now for $i \geq i(h)$ we have $b_h^i = b_h^{i(h)} = b_h^{i(h)}$ implying that, $\sum_{h \in \mathcal{H}} \frac{b_h^Z}{d_h^{max}} \leq$

$$2k. \text{ Alternatively, } \sum_{h \in \mathcal{H}} \frac{b_h^Z}{d_h^{max}} = \sum_{h \in \mathcal{H}} \sum_{i=1}^Z \frac{b_h^i - b_h^{i-1}}{d_h^{max}} = \sum_{i=1}^Z \sum_{h \in \mathcal{H}} \frac{b_h^i - b_h^{i-1}}{d_h^{max}} = \sum_{i=1}^Z \sum_{h \in \mathcal{H}_1^{i-1}} \frac{u_h^i}{d_h^{max}} \geq$$

$\sum_{i=1}^Z \gamma\sqrt{k} = Z\gamma\sqrt{k}$. Therefore we have that $Z\gamma\sqrt{k} \leq \sum_{h \in \mathcal{H}} \frac{b_h^Z}{d_h^{max}} \leq 2k$ which implies that $Z \leq \frac{2\sqrt{k}}{\gamma}$. Since β is the sum of Z points in \mathcal{D} , we have $\mathcal{B}(\frac{1}{Z}\beta) \subseteq \mathcal{D}$. Moreover, the projection β^2 of β on $\mathbb{R}^{\mathcal{H}_2}$ satisfies $\mathcal{D}_2 \subseteq \mathcal{B}(\beta^2)$ and thus $\alpha_2(\mathcal{D}_2) \leq \frac{2\sqrt{k}}{\gamma}$. \square

To prove Theorem 3.2.1, we only have to take $\gamma = \sqrt{2}$, use Lemma 3.2.5, and then invoke Lemma 3.2.4 to conclude. Using $k = O(n^2)$, we get that the ratio $\frac{\text{cong}_{\text{sta}}}{\text{cong}_{\text{dyn}}}$ is $O(n)$.

Observe that when there are no coupling constraints between the routing of the network commodities, then the $O(\sqrt{k})$ upper bound on the static/dynamic gap is still valid.

3.3 Inapproximability for the problem with edge restrictions

Let us consider a more general variant of the robust congestion problem where each commodity can only be routed on a subset of allowed edges $E_h \subseteq E$. These restrictions seem to be quite natural to ensure quality of service requirements such as delay constraints. Observe first that cong_{sta} can still be computed in polynomial-time for this variant.

Notice that the $O(\sqrt{k})$ upper bound on the static/dynamic gap proved in the previous section applies for this new variant.

The $\Omega(\frac{\log n}{\log \log n})$ inapproximability bound shown for the undirected case [Al-Najjar et al. (2021)] (under ETH assumption) still applies to the directed case (with and without flow restrictions). It is however quite far from the $O(\sqrt{k})$ approximation ratio deduced from Section 3.2. We will prove stronger inapproximability results for the generalisation of cong_{dyn} with flow restrictions under some classical complexity conjectures.

A standard way to prove this kind of results is to first prove that the problem is inapproximable under some constant and then to amplify this constant, see for example [Haviv and Regev (2007)].

Let us first introduce some additional notations. Taking into account the flow restrictions and given a subset of edges $C \subseteq E$, let $\mathcal{H}_C \subseteq \mathcal{H}$ be the set commodities such that each valid path related to any commodity $h \in \mathcal{H}_C$ intersects C . Even if C is not necessarily a cut in the standard sense of graph theory, C is called a cut in what follows. Given a demand vector $d \in \mathcal{D}$ and a cut C , $\sum_{h \in \mathcal{H}_C} d_h / \sum_{e \in C} c_e$, is obviously a lower bound of $\text{cong}_{\text{dyn}}(\mathcal{D})$. The maximum over all demand vectors $d \in \mathcal{D}$ and all cuts C of the ratio $\sum_{h \in \mathcal{H}_C} d_h / \sum_{e \in C} c_e$ is called cut congestion and denoted by $\text{cong}_{\text{cut}}(\mathcal{D})$. We also use $E_{\mathcal{H}}$ to denote the set of all flow restrictions: $E_{\mathcal{H}} = (E_h)_{h \in \mathcal{H}}$. An instance of cong_{dyn} with flow restrictions is then defined by $(G, c, \mathcal{H}, \mathcal{D}, E_{\mathcal{H}})$.

In Lemma 2.1.1, we proved that it is *NP*-hard to distinguish between instances where $\text{cong}_{\text{dyn}}(\mathcal{D})$ is less than or equal to 1 and those where the cut congestion $\text{cong}_{\text{cut}}(\mathcal{D})$ is greater than or equal to $1 + \rho$ for some constant $\rho > 0$.

In Lemma 3.3.1, we will show that given two instances of this problem, it is possible to build some kind of product instance whose dynamic congestion is less than or equal to the product of the dynamic congestion of the two instances and the cut congestion is greater than or equal to the product of the cut congestion of the two initial instances. Finally, by repetitively using the product of Lemma 3.3.1 on the instance of Lemma 2.1.1, we can amplify the gap leading to some strong inapproximability results.

Lemma 3.3.1. *Given two instances of cong_{dyn} with flow restrictions $\mathcal{I}_1 = (G_1, c_1, \mathcal{H}_1, \mathcal{D}_1, E_{\mathcal{H}_1})$ and $\mathcal{I}_2 = (G_2, c_2, \mathcal{H}_2, \mathcal{D}_2, E_{\mathcal{H}_2})$, we can build a new instance $\mathcal{I} = \mathcal{I}_1 \times \mathcal{I}_2 = (G, c, \mathcal{H}, \mathcal{D}, E_{\mathcal{H}})$ such that:*

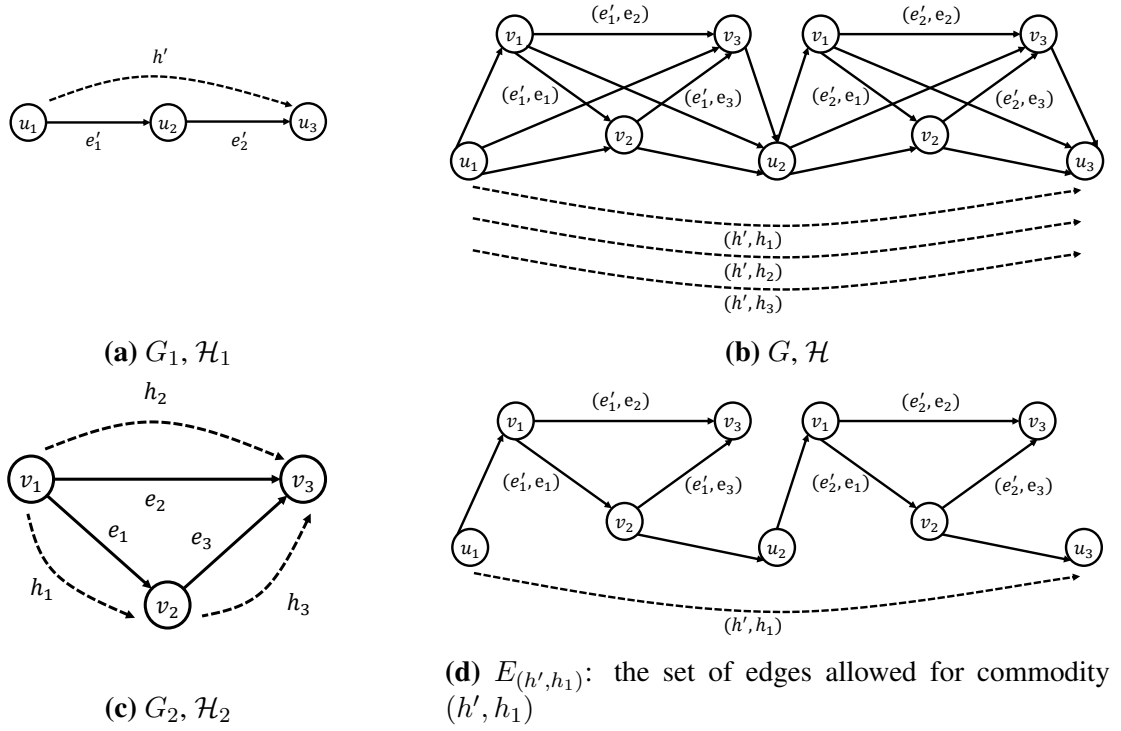


Fig. 3.2 Illustration of the construction of the product instance.

- $\text{cong}_{\text{dyn}}(\mathcal{I}) \leq \text{cong}_{\text{dyn}}(\mathcal{I}_1) \cdot \text{cong}_{\text{dyn}}(\mathcal{I}_2)$
- $\text{cong}_{\text{cut}}(\mathcal{I}) \geq \text{cong}_{\text{cut}}(\mathcal{I}_1) \cdot \text{cong}_{\text{cut}}(\mathcal{I}_2)$.

Furthermore, we have $|E(G)| = |E(G_1)| \cdot (|E(G_2)| + 2|V(G_2)|)$, $|V(G)| = |V(G_1)| + |V(G_2)| \cdot |E(G_1)|$, $|\mathcal{H}| = |\mathcal{H}_1| \cdot |\mathcal{H}_2|$ and the size of \mathcal{D} is less than or equal to the product of the sizes of \mathcal{D}_1 and \mathcal{D}_2 .

Proof. Lemma 3.3.1

Let $\mathcal{I}_1 = (G_1, c_1, \mathcal{H}_1, \mathcal{D}_1, E_{\mathcal{H}_1})$ and $\mathcal{I}_2 = (G_2, c_2, \mathcal{H}_2, \mathcal{D}_2, E_{\mathcal{H}_2})$ be two instances of cong_{dyn} with edge restrictions. We denote by G'_2 the graph obtained from G_2 by adding two nodes $s(G_2)$ and $t(G_2)$ to G_2 , an edge from $s(G_2)$ to each node of G_2 having an infinite capacity (i.e., $|V(G_2)|$ edges), and an edge from each node of G_2 to $t(G_2)$ having also an infinite capacity (i.e., $|V(G_2)|$ edges). We build a graph G by replacing each edge e of G_1 by a copy of G'_2 while identifying the node $s(e)$ (resp. $t(e)$) with the node $s(G_2)$ (resp. $t(G_2)$). Figure 3.2 illustrates the construction of the product instance. We denote by (e_1, e_2) the edge e_2 in G'_2 corresponding to the copy of G'_2 related to $e_1 \in E(G_1)$. The capacity of the edge (e_1, e_2) is the product of the capacity of edges e_1 and e_2 : $c_{(e_1, e_2)} = c_{e_1} \cdot c_{e_2}$.

We create a set of commodities \mathcal{H} in G by taking $\mathcal{H} = \mathcal{H}_1 \times \mathcal{H}_2$ and assuming that $s(h_1, h_2) = s_{h_1}$ and $t(h_1, h_2) = t_{h_1}$ for $(h_1, h_2) \in \mathcal{H}$. We also assume that edges of type $(s(G_2) = s(e), v)$ can only be used by a commodity $(h_1, h_2) \in \mathcal{H}$ if $s(h_2) = v$. Similarly, edges of type $(v, t(G_2) = t(e))$ can only be used by (h_1, h_2) if $t(h_2) = v$.

In other words, when a commodity (h_1, h_2) is routed through the copy of G_2 related to an edge $e \in E(G_1)$, then it should enter from $s(h_2)$ and leave at $t(h_2)$ (cf. Figure 3.2). Other edge restrictions are added by considering the restrictions related to \mathcal{I}_1 and \mathcal{I}_2 . If $h' \in \mathcal{H}_1$ is not allowed to use edge $e' \in E(G_1)$, then all commodities (h', h_2) are not allowed to be routed through the e' copy of G'_2 . Moreover, if $e_2 \in E(G_2)$ does not belong to E_{h_2} for some $h_2 \in \mathcal{H}_2$, then for each $e_1 \in E(G_1)$ and each $h_1 \in \mathcal{H}_1$, (e_1, e_2) cannot be used to route commodity (h_1, h_2) .

We define \mathcal{D} as the set of vectors $d \in \mathbb{R}_+^{\mathcal{H}_1 \times \mathcal{H}_2}$ such that there is a vector $d^1 \in \mathcal{D}_1$ satisfying $d_{h_1, \cdot} \in d_{h_1}^1 \mathcal{D}_2$ for all $h_1 \in \mathcal{H}_1$. The constraint $d_{h_1, \cdot} \in d_{h_1}^1 \mathcal{D}_2$ can be enforced with linear inequalities as follows. Suppose that $\mathcal{D}_2 = \{d^2 \in \mathbb{R}^{\mathcal{H}_2} \mid A_2 d^2 + B_2 \xi \leq b_2\}$ for some matrices A_2, B_2 . We also assume that this description contains the constraints $d_{h_2}^2 / d_{h_2}^{2, \max} \leq 1$ for all $h_2 \in \mathcal{H}_2$ in addition to the non-negativity constraints of demand values $d_{h_2}^2$. Then we can write the constraint $d_{h_1, \cdot} \in d_{h_1}^1 \mathcal{D}_2$ as $A_2 d_{h_1, \cdot} + B_2 \xi' - d_{h_1}^1 b_2 \leq 0$. Indeed, $d_{h_1}^1 = 0$ implies $d_{h_1, \cdot} = 0$ while for $d_{h_1}^1 > 0$ we have $A_2 d_{h_1, \cdot} + B_2 \xi' - d_{h_1}^1 b_2 \leq 0$ if and only if $d_{h_1, \cdot} / d_{h_1}^1 \in \mathcal{D}_2$. Polytope \mathcal{D} is then defined by constraints $A_2 d_{h_1, \cdot} + B_2 \xi^{h_1} - d_{h_1}^1 b_2 \leq 0$ for each $h_1 \in \mathcal{H}_1$ in addition to $A_1 d^1 + B_1 \xi \leq b_1$. Observe that a subscript h_1 is added to express the fact that the auxiliary variables ξ^{h_1} depend on $h_1 \in \mathcal{H}_1$. Notice also that the size of the matrices defining \mathcal{D} is less than or equal to the product of the sizes of the matrices defining \mathcal{D}_1 and \mathcal{D}_2 .

We will now prove that $\text{cong}_{\text{dyn}}(\mathcal{I}) \leq \text{cong}_{\text{dyn}}(\mathcal{I}_1) \cdot \text{cong}_{\text{dyn}}(\mathcal{I}_2)$. Let d be a vector in \mathcal{D} and let $d^1 \in \mathcal{D}_1$ be a vector such that $d_{h_1, \cdot} \in d_{h_1}^1 \mathcal{D}_2$. For h_1 in \mathcal{H}_1 , we define $d^{2, h_1} \in \mathcal{D}_2$ by $d_{h_2}^{2, h_1} = \frac{d_{h_1, h_2}}{d_{h_1}^1}$ if $d_{h_1}^1 \neq 0$ and $d^{2, h_1} = 0$ if $d_{h_1}^1 = 0$. We clearly have $d_{h_1, h_2} = d_{h_1}^1 \cdot d_{h_2}^{2, h_1}$ for all $h_1 \in \mathcal{H}_1, h_2 \in \mathcal{H}_2$.

Let x^1, x^{2, h_1} be the optimal routing schemes for $d^1 \in \mathbb{R}^{\mathcal{H}_1}$ and $d^{2, h_1} \in \mathbb{R}^{\mathcal{H}_2}$ for $h_1 \in \mathcal{H}_1$. To route commodity (h_1, h_2) , we consider the following multi-commodity flow in G defined by $x_{(h_1, h_2), (e_1, e_2)} = x_{h_1, e_1}^1 x_{h_2, e_2}^{2, h_1}$. The total flow on the edge (e_1, e_2) is then given by:

$$\begin{aligned} \sum_{(h_1, h_2) \in \mathcal{H}_1 \times \mathcal{H}_2} d_{h_1}^1 d_{h_2}^{2, h_1} x_{(h_1, h_2), (e_1, e_2)} &= \sum_{h_1 \in \mathcal{H}_1} d_{h_1}^1 x_{h_1, e_1}^1 \sum_{h_2 \in \mathcal{H}_2} d_{h_2}^{2, h_1} x_{h_2, e_2}^{2, h_1} \\ &\leq \sum_{h_1 \in \mathcal{H}_1} d_{h_1}^1 x_{h_1, e_1}^1 \text{cong}_{\text{dyn}}(\mathcal{I}_2) c_{e_2} \\ &\leq \text{cong}_{\text{dyn}}(\mathcal{I}_1) \cdot \text{cong}_{\text{dyn}}(\mathcal{I}_2) \cdot c_{e_1} \cdot c_{e_2} \\ &= \text{cong}_{\text{dyn}}(\mathcal{I}_1) \cdot \text{cong}_{\text{dyn}}(\mathcal{I}_2) \cdot c_{(e_1, e_2)}. \end{aligned}$$

Since this holds for any edge (e_1, e_2) of G (the other edges of G have an infinite capacity), we deduce that $\text{cong}_{\text{dyn}}(\mathcal{I}) \leq \text{cong}_{\text{dyn}}(\mathcal{I}_1) \cdot \text{cong}_{\text{dyn}}(\mathcal{I}_2)$.

Let us now show that $\text{cong}_{\text{cut}}(\mathcal{I}) \geq \text{cong}_{\text{cut}}(\mathcal{I}_1) \cdot \text{cong}_{\text{cut}}(\mathcal{I}_2)$. Let C_1 (resp. C_2) be a cut of G_1 (resp. G_2) achieving the maximal congestion $\text{cong}_{\text{cut}}(\mathcal{I}_1)$ (resp. $\text{cong}_{\text{cut}}(\mathcal{I}_2)$),

and let $d^1 \in \mathcal{D}_1$ (resp. $d^2 \in \mathcal{D}_2$) be a demand vector for which the maximal cut congestion is obtained. In other words, we have $\sum_{h_1 \in \mathcal{H}_{C_1}} d_{h_1}^1 / \sum_{e_1 \in C_1} c_{e_1} = \text{cong}_{\text{cut}}(\mathcal{I}_1)$ and

$$\sum_{h_2 \in \mathcal{H}_{C_2}} d_{h_2}^2 / \sum_{e_2 \in C_2} c_{e_2} = \text{cong}_{\text{cut}}(\mathcal{I}_2).$$

Observe that the set of edges $C_1 \times C_2$ is a cut of G that is intersecting all demands of $\mathcal{H}_{C_1} \times \mathcal{H}_{C_2}$. Notice that the flow restrictions that have been considered are crucial here to guarantee the previous fact. Let $d \in \mathbb{R}^{\mathcal{H}}$ be the demand defined by $d_{(h_1, h_2)} = d_{h_1}^1 \cdot d_{h_2}^2$. Since $d^1 \in \mathcal{D}_1$ and $d^2 \in \mathcal{D}_2$, we also have $d \in \mathcal{D}$. The congestion on the cut $C_1 \times C_2$ is given by:

$$\begin{aligned} \frac{\sum_{(h_1, h_2) \in \mathcal{H}_{C_1} \times \mathcal{H}_{C_2}} d_{(h_1, h_2)}}{\sum_{(e_1, e_2) \in C_1 \times C_2} c_{(e_1, e_2)}} &= \frac{\sum_{h_1 \in \mathcal{H}_{C_1}} d_{h_1}^1}{\sum_{e_1 \in C_1} c_{e_1}} \cdot \frac{\sum_{h_2 \in \mathcal{H}_{C_2}} d_{h_2}^2}{\sum_{e_2 \in C_2} c_{e_2}} \\ &= \text{cong}_{\text{cut}}(\mathcal{I}_1) \cdot \text{cong}_{\text{cut}}(\mathcal{I}_2). \end{aligned}$$

This clearly implies that $\text{cong}_{\text{cut}}(\mathcal{I}) \geq \text{cong}_{\text{cut}}(\mathcal{I}_1) \cdot \text{cong}_{\text{cut}}(\mathcal{I}_2)$. □

Combining the previous lemma with lemma 2.1.1, one can amplify the gap as follows.

Lemma 3.3.2. *For some $0 < \rho < 1$ and each $r \in \mathbb{N}$, each 3-SAT instance φ can be mapped to an instance $\mathcal{I}^r = (G^r, c^r, \mathcal{H}^r, \mathcal{D}^r, E_{\mathcal{H}}^r)$ of cong_{dyn} with flow restrictions where $\mathcal{D}^r = \{d \in \mathbb{R}^{\mathcal{H}^r} \mid A^r d + B^r \xi \leq b^r\}$ such that:*

- *If $\text{val}(\varphi) \leq 1 - \rho$ then $\text{cong}_{\text{dyn}}(\mathcal{I}^r) \leq 1$*
- *If φ is satisfiable then $\text{cong}_{\text{cut}}(\mathcal{I}^r) \geq (1 + \rho)^r$.*

Furthermore, there exists a positive constant θ such that $|V(G^r)|$, $|E(G^r)|$, $|\mathcal{H}^r|$ and the size of the matrices A^r and B^r defining \mathcal{D}^r are all less than or equal to $(\theta m)^r$ where m is the number of clauses of φ .

Proof. Let \mathcal{I}^1 be the instance defined in Lemma 2.1.1. We recursively build \mathcal{I}^r as the product of \mathcal{I}^{r-1} and \mathcal{I}^1 . Using notation of Lemma 3.3.1, we take $\mathcal{I}_1 = \mathcal{I}^{r-1}$, $\mathcal{I}_2 = \mathcal{I}^1$ and $\mathcal{I}^r = \mathcal{I} = \mathcal{I}_1 \times \mathcal{I}_2$. Using what is already known about the size of the instance \mathcal{I}^1 of Lemma 2.1.1 and the results of Lemma 3.3.1, a simple induction proves the existence of a constant θ such that $(\theta m)^r$ is an upper bound of the number of vertices, number of edges, number of commodities and the size of the matrices defining the polytope \mathcal{D}^r . □

By making use of some standard complexity assumptions, inapproximability results can be directly deduced from the previous lemma.

Proposition 3.3.1. *Unless $NP \subseteq SUBEXP$, cong_{dyn} with flow restrictions cannot be approximated within a factor of $k^{\frac{c'}{\log \log k}}$ (resp. $n^{\frac{c'}{\log \log n}}$) for some constant c' .*

Proof. *SUBEXP* is the class of problems that can be solved in 2^{n^ϵ} time for all $\epsilon > 0$. Therefore, if $NP \not\subseteq SUBEXP$ then there is a constant $\epsilon_0 > 0$ such that no algorithm can solve the Gap-3-SAT problem in time $O(2^{m^{\epsilon_0}})$ where m is the number of clauses of the 3-SAT instance.

Let $\epsilon_1 < \epsilon_0$ and let $r(m) = \frac{m^{\epsilon_1}}{\log m}$. The size of the instance $\mathcal{I}^{r(m)}$ is polynomial in $m^{r(m)}$. Therefore if we run a polynomial approximation algorithm on the instance $\mathcal{I}^{r(m)}$, the running time will be $m^{c_1 r(m)}$ for some constant c_1 . Furthermore, $m^{c_1 r(m)} = m^{c_1 \frac{m^{\epsilon_1}}{\log m}} = 2^{c_1 m^{\epsilon_1}} < 2^{m^{\epsilon_0}}$ for big enough m .

The number of commodities k in the instance $\mathcal{I}^{r(m)}$ is bounded by $(\theta m)^{r(m)}$. We consequently have $\log k \leq \log(\theta m) \frac{m^{\epsilon_1}}{\log m}$ implying that $m > a \log^{\frac{1}{\epsilon_1}} k$ for some constant a and big enough m .

The gap between the congestion of the instances $\mathcal{I}^{r(m)}$ corresponding to a 3-SAT instance for which $val(\varphi) < 1 - \rho$ and those for which $val(\varphi) = 1$ is:

$$(1 + \rho)^{r(m)} > (1 + \rho)^{r(a \log^{\frac{1}{\epsilon_1}} k)} = (1 + \rho)^{\frac{a^{\epsilon_1} \log k}{\frac{1}{\epsilon_1} \log a \log k}} > k^{\frac{c'}{\log \log k}} \text{ for some constant } c'.$$

Hence, if a polynomial-time algorithm could solve cong_{dyn} with flow restrictions within an approximation ratio of $O(k^{\frac{c'}{\log \log k}})$, we could use it to solve the Gap-3-SAT problem in $O(2^{m^{\epsilon_0}})$ time. The same proof applies if parameter n (the number of vertices) is considered instead of k . \square

A slightly weaker inapproximability result is obtained using a weaker complexity assumption, that $NP \neq QP$ where QP is the class of problems solvable in quasi-polynomial time, i.e. in time $O(n^{\log^c n})$ for some constant c and where n is the size of the instance.

Proposition 3.3.2. *Unless $NP \subseteq QP$, cong_{dyn} with flow restrictions cannot be approximated within a factor of $2^{\log^{1-\epsilon} k}$ (resp. $2^{\log^{1-\epsilon} n}$) for any $\epsilon > 0$.*

Proof. Let us take $r(m) = \log^{c_1}(m)$ for an arbitrary constant c_1 . If we run a polynomial-time algorithm on instance the instance $\mathcal{I}^{r(m)}$, we get an algorithm running in quasi-polynomial time. The number of commodities k is bounded by $(\theta m)^{r(m)}$. Thus $\log k \leq \log^{c_1} m \log \theta m < \log^{c_1+2} m$ for big enough m and therefore $m > \exp(\log^{\frac{1}{c_1+2}} k)$.

The gap between the congestion of the instances $\mathcal{I}^{r(m)}$ corresponding to 3-SAT instances such that $val(\varphi) < 1 - \rho$ and those such that $val(\varphi) = 1$ is:

$$\begin{aligned} (1 + \rho)^{r(m)} &> (1 + \rho)^{r(\log^{\frac{1}{c_1+2}} k)} \\ &= (1 + \rho)^{\log^{\frac{c_1}{c_1+2}} k} \\ &> (1 + \rho)^{\log^{1-\epsilon} k} \end{aligned}$$

for any $\epsilon > 0$ if we take c_1 such that $\frac{c_1}{c_1+2} > 1 - \epsilon$. The $(1 + \rho)$ term can be replaced by 2 by observing that $2^{\log^{1-\epsilon'} k} = o((1 + \rho)^{\log^{1-\epsilon} k})$ for any $\epsilon' < \epsilon$. The same proof applies if parameter n is considered instead of k . \square

Using the last part of Proposition 2.3.1, all inapproximability results stated for the congestion problem cong_{dyn} are also valid for lin_{dyn} .

CHAPTER 4

Affine routing for the robust network design problem

In this chapter we study the affine routing method both for the non restricted flow problem and the problem with a given set of paths. The motivations of the work are the following. Both for the static and for dynamic routing problems, several equivalent formulations are possible such as node-arc formulations and arc-path formulations. One might ask whether this still holds when affine decision rules are applied. Another goal might be to try to improve these formulations without losing the polynomial-time solvability aspects of affine formulations.

- When affine routing is considered, is there an equivalence between node-arc formulations and arc-path formulations?
- Is it possible to get stronger affine formulations that can still be solved in polynomial time?

Instead of requiring that the flow conservation constraints are strictly satisfied, we will consider a relaxed flow conservation formulation by allowing some flows to be created on intermediary nodes but requiring that at least the needed amount of flow is sent at the source. Alternatively, we can also allow some flows to be lost at intermediary nodes but we require that at least the needed amount of flow arrives at the destination. We can also wonder what is the relationship between the edge formulation and the path formulation where all paths between each source and destination are allowed. Other formulations can be also be obtained by aggregating several commodities by sources and/or destinations.

In this chapter we systematically study those variants of affine routing formulations. We provide toy examples showing that when an affine routing is considered each of those variants can (quite surprisingly) strictly improve the cost of the solution. We also numerically test those variants on realistic network instances. We observed that even in this case, relaxed formulations can provide an improvement of the solution. We also show that relaxed flow conservation variants are equivalent to adding *fictive* edges in the graph for each commodity. This allows us to derive a formulation that is even better than relaxed flow conservation variants.

4.1 Possible affine formulations

We start by recalling some standard node-arc formulations that we will improve later in Section 4.2. Then we recall an arc-path formulation that might be more practical when paths can be enumerated easily and we show that it can be strictly dominated by node-arc formulations. Finally, we close this section by proposing a natural cycle-based formulation that is equivalent to node-arc formulations but with slightly less variables and constraints.

4.1.1 Initial node-arc formulation

Let's consider a directed graph $G = (V, E)$ representing a communication network. The traffic is characterized by a set of commodities $h \in \mathcal{H}$ associated to different node pairs. For a given commodity h , the traffic originates at $s(h)$ and ends at $t(h)$. As introduced in [Poss and Raack (2013)], for each demand scenario, the flow $f_{h,e}(d)$ related to commodity h and edge $e \in E$, is restricted to affinely depend on the demand vector d . It represents the capacity reservation in the robust network design problem. This flow $f_{h,e}(d)$ is of the form $f_{h,e}(d) = x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'}$ where coefficients $x_{h,e}^0$ and $x_{h,e}^{h'}$ are subject to optimization.

The affine routing with congestion minimization can be then modeled as follows:

$$\begin{aligned} & \min m \\ & \sum_{e \in \delta_+(v)} \left(x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'} \right) - \sum_{e \in \delta_-(v)} \left(x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'} \right) = \begin{cases} d_h, & \text{if } v = s(h) \\ -d_h, & \text{if } v = t(h) \\ 0 & \text{otherwise} \end{cases} \\ & \forall h \in \mathcal{H}, v \in V, d \in \mathcal{D} \end{aligned} \tag{4.1a}$$

$$\sum_{h \in \mathcal{H}} \left(x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'} \right) \leq c_e m, \quad \forall e \in E, d \in \mathcal{D} \tag{4.1b}$$

$$x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'} \geq 0, \quad \forall e \in E, \forall h \in \mathcal{H}, d \in \mathcal{D} \tag{4.1c}$$

Constraints (4.1a) are standard flow conservation constraints, while (4.1b) express capacity limitation. Finally, constraints (4.1c) impose positivity on capacity reservations. $\delta_-(v)$ and $e \in \delta_+(v)$ respectively denote incoming and outgoing edges from node $v \in V$. Notice that the flow conservation constraint related to $v = t(h)$ can be skipped since it can be obtained by summing the constraints related to the other vertices.

\mathcal{D} is supposed to be fully dimensional (it contains a ball). This assumption is not really restrictive since, in practice, one should not expect that there is any linear equation satisfied by all demands vectors. Moreover, if the assumption is not satisfied, then one can eliminate some of the coefficients $x_e^{hh'}$. For example, if we always have

$d_{h'} = \sum_{h''} \alpha_{h''} d_{h''}$ then there is clearly no need to consider any dependency on h' implying that coefficients $x_e^{hh'}$ are useless. Under this hypothesis, two affine functions a_1, a_2 are equal over \mathcal{D} (i.e. $a_1(d) = a_2(d), \forall d \in \mathcal{D}$), if and only if, $a_1 = a_2$ (i.e. all coefficients of the affine functions a_1, a_2 are equal). Using this fact, as proposed in [Poss and Raack (2013)], we can replace the flow conservation constraints (4.1a) by the following equivalent reformulation:

$$\sum_{e \in \delta_+(v)} x_{h,e}^{h'} - \sum_{e \in \delta_-(v)} x_{h,e}^{h'} = \begin{cases} 1, & \text{if } v = s(h) \text{ and } h = h' \\ -1, & \text{if } v = t(h) \text{ and } h = h' \\ 0 & \text{otherwise (including } h' = 0) \end{cases} \quad (4.2)$$

The obtained formulation proposed by [Poss and Raack (2013)] is given below.

min m

$$\sum_{e \in \delta_+(v)} x_{h,e}^{h'} - \sum_{e \in \delta_-(v)} x_{h,e}^{h'} = \begin{cases} 1, & \text{if } v = s(h) \text{ and } h = h' \\ -1, & \text{if } v = t(h) \text{ and } h = h' \\ 0 & \text{otherwise} \end{cases} \quad (4.3a)$$

$$\forall h \in \mathcal{H}, h' \in \mathcal{H} \cup \{0\}, v \in V \quad (4.3a)$$

$$\sum_{h \in \mathcal{H}} \left(x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'} \right) \leq c_e m, \quad \forall e \in E, d \in \mathcal{D} \quad (4.3b)$$

$$x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'} \geq 0, \quad \forall e \in E, h \in \mathcal{H}, d \in \mathcal{D} \quad (4.3c)$$

As will be recalled in Section 4.6, when some linear constraints with uncertain coefficients need to be considered, we can handle uncertainty either by using constraint generation or by duality-based reformulation techniques. This should be done for each constraint. The main advantage of (4.3) is that the uncertainty appearing in the flow conservation constraints (4.1a) is already handled using (4.2). However, the other constraints (4.1b) and (4.1c) still need to be treated using the techniques briefly recalled in Section 4.6 and used for the numerical evaluation.

4.1.2 Arc-path formulation

Another natural formulation is the one obtained by considering path variables. This might lead to solution methods that are easier to implement in communication networks when only a small number of paths is used for each commodity or the total number of paths that could be handled by each router/node is limited. As proposed in [Ouorou and Vial (2007)], the flow on each path affinely depends on the demand vector d . This leads to the following model where \mathcal{P}_h is a set of (possibly all) paths from $s(h)$ to $t(h)$.

$\min m$

$$\sum_{h \in \mathcal{H}} \sum_{p \in \mathcal{P}_h: p \ni e} \left(x_{p,h}^0 + \sum_{h' \in \mathcal{H}} x_{p,h}^{h'} d_{h'} \right) \leq c_e m, \quad \forall e \in E, d \in \mathcal{D} \quad (4.4a)$$

$$\sum_{p \in \mathcal{P}_h} \left(x_{p,h}^0 + \sum_{h' \in \mathcal{H}} x_{p,h}^{h'} d_{h'} \right) = d_h, \quad \forall h \in \mathcal{H}, d \in \mathcal{D} \quad (4.4b)$$

$$x_{p,h}^0 + \sum_{h' \in \mathcal{H}} x_{p,h}^{h'} d_{h'} \geq 0, \quad \forall h \in \mathcal{H}, p \in \mathcal{P}_h, d \in \mathcal{D} \quad (4.4c)$$

Observe that constraints (4.4c) impose the non-negativity of the flow on each path. Notice that when there is no uncertainty (i.e, when \mathcal{D} contains only one demand vector), then (4.1) and (4.4) are equivalent when \mathcal{P}_h contains all possible paths. One might wonder whether this still holds for any \mathcal{D} . Each solution of (4.4) can obviously be transformed into a feasible solution of (4.1) and (4.3). However, the following example shows that (4.4) can be strictly dominated by (4.1) even if all paths are considered.

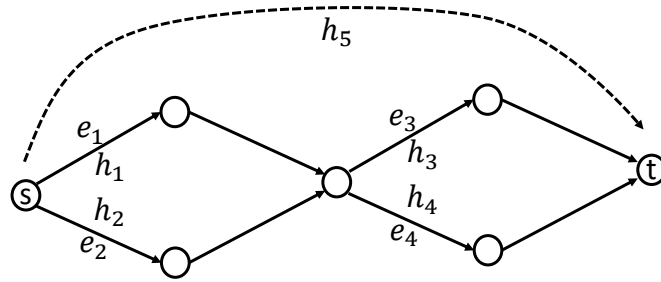


Fig. 4.1 An example with five commodities: h_i with $i = 1, 2, 3, 4$ having the same source and sink than edges e_i , and h_5 from source node s to sink node t . In this example we show that formulation (4.1) is strictly better than formulation (4.4).

Proposition 4.1.1. *Formulation (4.1) is less conservative than formulation (4.4).*

Proof. In the example of Figure 4.1, there is a demand h_1 (resp. h_2, h_3 , and h_4) having the same source and sink than edge e_1 (resp. e_2, e_3 , and e_4) and a demand h_5 having the node s as a source and the node t as sink. The demand polyhedron \mathcal{D} is defined as the set of $d \in \mathbb{R}^5$ satisfying the equations $d_{h_1} + d_{h_2} = 1, d_{h_3} + d_{h_4} = 1$ and $d_{h_5} = 1$ in addition to non-negativity constraints. The capacity of each edge is equal to 1.

This demand polyhedron can be routed with model (4.1) without exceeding one unit of flow on each edge. This can be seen by considering the following solution: $f_{h_i, e_i}(d) = d_{h_i}, f_{h_5, e_i}(d) = 1 - d_{h_i}, \forall i = 1, \dots, 4$ and $f_{h_i, e_j}(d) = 0 \forall i, j = 1, \dots, 4, i \neq j$. By taking $m = 1$, all capacity constraints and flow conservation constraints of (4.1) are satisfied.

Now we are going to show that a solution to model (4.4) necessarily uses strictly more than one unit of flow on at least one edge. By contradiction, suppose that there exists a solution of (4.4) such that $m = 1$. First observe that given the equalities defining the demand polyhedron \mathcal{D} , it is sufficient that the affine flow function depends on the demand's values d_{h_1} and d_{h_3} . Let p_1 denote the path that uses edges e_1 and e_3 , while p_2 contains edges e_1 and e_4 , p_3 includes edges e_2 and e_3 and p_4 goes through edges e_2 and e_4 . Since the total flow that uses edge e_i must be less than 1 and the demand h_i must necessarily use edge e_i for $i = 1, \dots, 4$ we have the following inequalities:

$$1 - d_{h_1} \geq f_{h_5, p_1}(d) + f_{h_5, p_2}(d) \quad (4.5a)$$

$$d_{h_1} \geq f_{h_5, p_3}(d) + f_{h_5, p_4}(d) \quad (4.5b)$$

$$1 - d_{h_3} \geq f_{h_5, p_1}(d) + f_{h_5, p_3}(d) \quad (4.5c)$$

$$d_{h_3} \geq f_{h_5, p_2}(d) + f_{h_5, p_4}(d) \quad (4.5d)$$

By summing inequalities (4.5), we get $2 \geq 2 \times \sum_{i=1, \dots, 4} f_{h_5, p_i}(d)$. Since $\sum_{i=1, \dots, 4} f_{h_5, p_i}(d) = d_{h_5} = 1$, all inequalities (4.5) should be equalities. Remember that for each path p_i , we have $f_{h_5, p_i}(d) = x_{h_5, p_i}^0 + x_{h_5, p_i}^{h_1} d_{h_1} + x_{h_5, p_i}^{h_3} d_{h_3}$. Writing the four equalities above for the vector d where $d_{h_1} = d_{h_3} = 0$, we get: $1 = x_{h_5, p_1}^0 + x_{h_5, p_2}^0$, $0 = x_{h_5, p_3}^0 + x_{h_5, p_4}^0$, $1 = x_{h_5, p_1}^0 + x_{h_5, p_3}^0$, $0 = x_{h_5, p_2}^0 + x_{h_5, p_4}^0$. This implies that $x_{h_5, p_2}^0 = x_{h_5, p_3}^0 = -x_{h_5, p_4}^0$. Now using the fact that $f_{h_5, p_i}(d) \geq 0$ (still for the same demand vector), we get that all variables x_{h_5, p_1}^0 , x_{h_5, p_2}^0 , x_{h_5, p_3}^0 and x_{h_5, p_4}^0 are non-negative. This obviously leads to $x_{h_5, p_1}^0 = 1$, $x_{h_5, p_i}^0 = 0$, $\forall i = 2, 3, 4$.

Furthermore, by considering the positivity constraint (4.4c) for path p_2 and the demand vector where $d_{h_1} = 1$ and $d_{h_3} = 0$, we can deduce that $f_{h_5, p_2}(d) = x_{h_5, p_2}^0 + x_{h_5, p_2}^{h_1} = x_{h_5, p_2}^{h_1} \geq 0$. Writing equality (4.5a) leads to $-1 = x_{h_5, p_1}^{h_1} + x_{h_5, p_2}^{h_1}$. Combination with the previous inequality implies that $x_{h_5, p_1}^{h_1} \leq -1$. Similarly, by considering the demand vector where $d_{h_3} = 1$ and $d_{h_1} = 0$, the positivity constraint related to path p_3 and equality(4.5c) lead to $x_{h_5, p_1}^{h_3} \leq -1$. Let us now consider the case where $d_{h_3} = 1$ and $d_{h_1} = 1$. The positivity of $f_{h_5, p_1}(d)$ is equivalent to $x_{h_5, p_1}^0 + x_{h_5, p_1}^{h_1} + x_{h_5, p_1}^{h_3} \geq 0$ implying that $x_{h_5, p_1}^{h_1} + x_{h_5, p_1}^{h_3} \geq -1$. This is clearly not possible since $x_{h_5, p_1}^{h_1} \leq -1$ and $x_{h_5, p_1}^{h_3} \leq -1$, and it ends the proof. \square

4.1.3 Elementary cycle-based formulation

We have seen that the arc-path formulation can be sometimes strictly dominated by the node-arc formulation. The main reason for that is the positivity constraint imposed for each path and each $d \in \mathcal{D}$. Then, if one tries to relax these positivity constraints and replace them by constraints saying that the total flow on each directed edge is non-negative (i.e., $f_{h, e}(d) \geq 0$ for each h, e and d), then we will get a new formulation where

circulations appear. However, an easy way to present the new formulation is proposed below starting from formulation (4.3).

Let us first recall a basic result [Gondran et al. (1984)] about circulation decomposition as a sum of elementary circulations through elementary cycles. We fix a spanning tree T of the graph G (supposed to be connected). For each $e \in E(G) \setminus E(T)$, there is a unique elementary cycle σ in $T \cup \{e\}$. We note χ_σ this circulation that has a value of 1 on the edges oriented in the same direction than edge e , -1 in the other direction and 0 on the edges outside σ . We denote by $\Sigma(T)$ the set of cycles. It is well-known [Gondran et al. (1984)] that every circulation ϕ can be (uniquely) written as: $\phi = \sum_{\sigma \in \Sigma(T)} x_\sigma \chi_\sigma$ for some scalars $x_\sigma \in \mathbb{R}$.

Let us now go back to constraints (4.3a). For each commodity h , let p^h be any arbitrary fixed undirected path connecting $s(h)$ and $t(h)$ in T and let χ_{p^h} be the flow of value 1 on p^h and zero elsewhere (the value of the flow on each edge is either 1 or -1 depending on the direction of the edge). Since $x_{h,\cdot}^h - \chi_{p^h}$ is a circulation, $x_{h,\cdot}^h$ can be written as: $x_{h,\cdot}^h = \chi_{p^h} + \sum_{\sigma \in \Sigma(T)} x_{h,\sigma}^h \chi_\sigma$. Furthermore, for $h \neq h'$, $x_{h,\cdot}^{h'}$ is a circulation and thus it can be written as $x_{h,\cdot}^{h'} = \sum_{\sigma \in \Sigma(T)} x_{h,\sigma}^{h'} \chi_\sigma$. We then obtain the new model (4.6) by substituting $x_{h,e}^h$ in model (4.1) by $\chi_{p^h,e} + \sum_{\sigma \in \Sigma(T)} x_{h,\sigma}^h \chi_{\sigma,e}$ and replacing $x_{h,e}^{h'}$ by

$$\sum_{\sigma \in \Sigma(T)} x_{h,\sigma}^{h'} \chi_{\sigma,e}.$$

$$\begin{aligned} & \min m \\ & \sum_{h \in \mathcal{H}} \left(\sum_{\sigma \ni e} \left(x_{h,\sigma}^0 \chi_{\sigma,e} + \sum_{h' \in \mathcal{H}} x_{h,\sigma}^{h'} \chi_{\sigma,e} d_{h'} \right) + \sum_{p^h \ni e} \chi_{p^h,e} d_h \right) \leq c_e m, \quad \forall e \in E, d \in \mathcal{D} \\ & \sum_{\sigma \ni e} \left(x_{h,\sigma}^0 \chi_{\sigma,e} + \sum_{h' \in \mathcal{H}} x_{h,\sigma}^{h'} \chi_{\sigma,e} d_{h'} \right) + \sum_{p^h \ni e} \chi_{p^h,e} d_h \geq 0, \quad \forall e \in E, h \in \mathcal{H}, d \in \mathcal{D} \\ & x_{h,\sigma}^{h'} \in \mathbb{R}, \quad \forall \sigma \in \Sigma(T), h \in \mathcal{H}, h' \in \mathcal{H} \cup \{0\} \end{aligned} \tag{4.6}$$

Observe that since the number of elementary cycles $|\Sigma(T)|$ is equal to the cyclic number $|E| - |V| + 1$, the number of x variables in (4.6) is equal to $(|\mathcal{H}| + 1) \times |\mathcal{H}| \times (|E| - |V| + 1)$ whereas formulation (4.3) has $|\mathcal{H}| \times (|\mathcal{H}| + 1) \times |E|$ variables (there are also variables related to duality to take into account uncertainty as will be recalled in Section 4.6 but their number is the same in both formulations). Then Formulation (4.6) has around $|\mathcal{H}|^2 \times |V|$ less variables than formulation (4.3). Formulation (4.6) has also around $|\mathcal{H}|^2 \times |V|$ less constraints than formulation (4.3) due to constraints (4.3a). Formulation (4.6) is obviously equivalent to formulations (4.3) and (4.1) since it was obtained from (4.3) using the decomposition result.

4.2 Relaxing the flow conservation constraints

In this section, we present some improvements of the node-arc formulation (4.1) described in Section 4.1, by relaxing the flow conservation constraints (4.1a). Such improvements permit to further reduce the congestion and minimize the gap with the solution given by the dynamic routing. The standard formulation (4.1) might be denoted by $\mathcal{F}_=$ (“=” means that we have equalities in constraints (4.1a)). Let \mathcal{F}_+ be the formulation obtained from (4.1) by replacing (4.1a) by the following inequalities.

$$\sum_{e \in \delta_+(v)} f_{h,e}(d) - \sum_{e \in \delta_-(v)} f_{h,e}(d) \begin{cases} \geq d_h, & \text{if } v = s(h) \\ \geq 0 & \text{if } v \neq s(h), t(h) \end{cases} \quad (4.7)$$

Notice that by summing all inequalities (for some h) we get that $\sum_{e \in \delta_+(v)} f_{h,e}(d) - \sum_{e \in \delta_-(v)} f_{h,e}(d) \leq -d_h$ for $v = t(h)$. Since the quantities $f_{h,e}(d)$ no longer satisfy flow conservation constraints, we have no more the notion of flow. However, we can interpret $f_{h,e}(d)$ as being the amount of resources that is reserved for commodity h on edge e when the demand scenario d is considered. We will prove that for each demand vector d , it is possible to route each commodity h without exceeding the capacity $f_{h,e}(d)$ of edge e . We will then say that \mathcal{F}_+ is valid.

Proposition 4.2.1. \mathcal{F}_+ is valid.

Proof. Consider any commodity $h \in \mathcal{H}$ and any cut $\delta_+(C)$ separating $s(h)$ and $t(h)$ (so $C \subset V$, $s(h) \in C$ and $t(h) \notin C$). By summing all constraints (4.7) for vertices inside C , we get

$$\sum_{e \in \delta_+(C)} f_{h,e}(d) - \sum_{e \in \delta_-(C)} f_{h,e}(d) = \sum_{v \in C} \left(\sum_{e \in \delta_+(v)} f_{h,e}(d) - \sum_{e \in \delta_-(v)} f_{h,e}(d) \right) \geq d_h.$$

Using the positivity constraint on capacities $f_{h,e}(d)$, we deduce that $\sum_{e \in \delta_+(C)} f_{h,e}(d) \geq d_h$. Since this holds for any separating cut, it is possible by the maximum-flow minimum-cut theorem to send from $s(h)$ to $t(h)$ a flow of value d_h using the capacities $f_{h,e}(d)$. □

Since \mathcal{F}_+ is obtained from $\mathcal{F}_=$ by relaxing some constraints, the congestion m computed by \mathcal{F}_+ is less than or equal to the congestion given by $\mathcal{F}_=$. One might wonder whether there is any gain by considering \mathcal{F}_+ instead of $\mathcal{F}_=$. The example given below positively answers the question.

Proposition 4.2.2. Formulation \mathcal{F}_+ is less conservative than formulation $\mathcal{F}_=$.

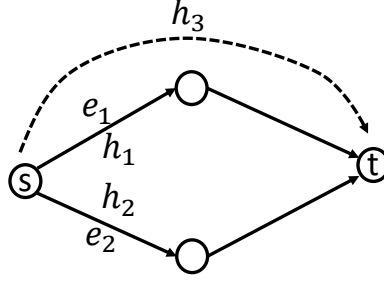


Fig. 4.2 An example with three commodities: h_i with $i = 1, 2$ having the same source and sink than edges e_i , and h_3 from source node s to sink node t . In this example we show that \mathcal{F}_+ is strictly better than formulation $\mathcal{F}_=$.

Proof. Figure 4.2 illustrates a simple graph with two commodities $h_{i,i=1,2}$ having the same source and sink than edges $e_{i,i=1,2}$. There is an additional commodity h_3 having node s as source and node t as sink. All edges have a capacity equal to 1. The polyhedron \mathcal{D} is defined as the set of $d \in \mathbb{R}_+^3$ satisfying the two inequalities $d_{h_1} + d_{h_2} \leq 1$ and $d_{h_3} \leq 1$ in addition to non-negativity constraints.

First, observe that the solution given by $f_{h_3,e_1}(d) = 1 - d_{h_1}$, $f_{h_3,e_2}(d) = d_{h_1}$, $f_{h_1,e_1}(d) = d_{h_1}$, $f_{h_1,e_2}(d) = 0$, $f_{h_2,e_2}(d) = d_{h_2}$, $f_{h_2,e_1}(d) = 0$, satisfies the constraints of \mathcal{F}_+ . Consequently, $m = 1$ is the optimal congestion found by \mathcal{F}_+ .

Let us now show that any solution of $\mathcal{F}_=$ should necessarily use more than one unit of flow on at least one edge. By contradiction, let's assume that there exists a solution of $\mathcal{F}_=$ such that $m = 1$. First, observe that if the demand for a commodity, let say h_3 , is equal to zero then the flow for this commodity must also be 0 in model (4.1) (i.e., $\mathcal{F}_=$). This is due to the fact that we are dealing here with flows and there are no directed cycles in the graph. Consequently, for each edge $e \in E$ we have $f_{h_3,e}(0) = x_e^{h_3 0} = 0$. Considering the demand vector $\mathbb{1}_{h_1}$ where $d_{h_1} = 1$ and the two other demands are 0, we get that $f_{h_3,e}(\mathbb{1}_{h_1}) = x_{h_3,e}^0 + x_{h_3,e}^{h_1} = 0$ leading to $x_{h_3,e}^{h_1} = 0$. Similarly, by considering the demand vector $\mathbb{1}_{h_2}$, we prove that $x_{h_3,e}^{h_2} = 0$. Combining the previous facts leads to $f_{h_3,e}(d) = x_{h_3,e}^{h_3} d_{h_3}$.

Let us consider the demand vector $\mathbb{1}_{h_1} + \mathbb{1}_{h_3}$ ($d_{h_1} = 1$, $d_{h_2} = 0$, $d_{h_3} = 1$). Then $f_{e_1}^{h_1}(\mathbb{1}_{h_1} + \mathbb{1}_{h_3}) = 1$ since the only path to route the demand h_1 is through e_1 . Moreover, the assumption $m = 1$ implies that $f_{h_3,e_1}(\mathbb{1}_{h_1} + \mathbb{1}_{h_3}) + f_{h_1,e_1}(\mathbb{1}_{h_1} + \mathbb{1}_{h_3}) \leq 1$. Combining the two propositions leads to $f_{h_3,e_1}(\mathbb{1}_{h_1} + \mathbb{1}_{h_3}) \leq 0$. Using the positivity constraint, we simply get $f_{h_3,e_1}(\mathbb{1}_{h_1} + \mathbb{1}_{h_3}) = 0$. Using the fact that $f_{h_3,e_1}(d) = x_{h_3,e_1}^{h_3} d_{h_3}$, we finally deduce that $x_{h_3,e_1}^{h_3} = 0$.

Using a similar argument, we can get that $f_{e_2}^{h_2}(\mathbb{1}_{h_2} + \mathbb{1}_{h_3}) = x_{e_2}^{h_3 h_3} = 0$. Hence, all coefficients related to commodity h_3 are zero which is nonsense. \square

Another way to relax the flow conservation constraints consists in replacing (4.1a)

by the following inequalities:

$$\sum_{e \in \delta_+(v)} f_{h,e}(d) - \sum_{e \in \delta_-(v)} f_{h,e}(d) \begin{cases} \leq -d_h, & \text{if } v = t(h) \\ \leq 0 & \text{if } v \neq s(h), t(h) \end{cases} \quad (4.8)$$

The obtained formulation can then be called \mathcal{F}_- formulation which is completely symmetrical to \mathcal{F}_+ . The validity of \mathcal{F}_- can be shown in almost the same way (the proof of Proposition 4.2.1 can be modified by summing inequalities (4.8) through all vertices belonging to $V \setminus C$). Proposition 4.2.2 also holds for \mathcal{F}_- (the same example provided in the proof can still be used). While both \mathcal{F}_- and \mathcal{F}_+ dominate $\mathcal{F}_=$, they are not comparable (for some instances \mathcal{F}_- provides better results than \mathcal{F}_+ and vice-versa).

4.3 A cut-based formulation

To show the validity of either \mathcal{F}_+ or \mathcal{F}_- , we just proved that every cut separating the source and the sink of a commodity has enough capacity to carry the demand. This suggests the advantage of proposing a formulation based on cuts. More precisely, for each commodity $h \in \mathcal{H}$ and each cut $\delta_+(C)$ separating $s(h)$ and $t(h)$ ($C \subset V$, $s(h) \in C$ and $t(h) \notin C$) we require that

$$\sum_{e \in \delta_+(C)} f_{h,e}(d) \geq d_h. \quad (4.9)$$

The cut formulation denoted by \mathcal{F}_{cut} is then obtained from (4.1) by replacing flow conservation constraints (4.1a) by (4.9). \mathcal{F}_{cut} is obviously valid.

Since solutions of \mathcal{F}_+ and \mathcal{F}_- satisfy the constraints of \mathcal{F}_{cut} , the solution provided by \mathcal{F}_{cut} is at least as good as those of either \mathcal{F}_+ or \mathcal{F}_- . We provide below an example showing that \mathcal{F}_{cut} can dominate \mathcal{F}_+ and \mathcal{F}_- .

Proposition 4.3.1. *Formulation \mathcal{F}_{cut} is less conservative than \mathcal{F}_+ and \mathcal{F}_- .*

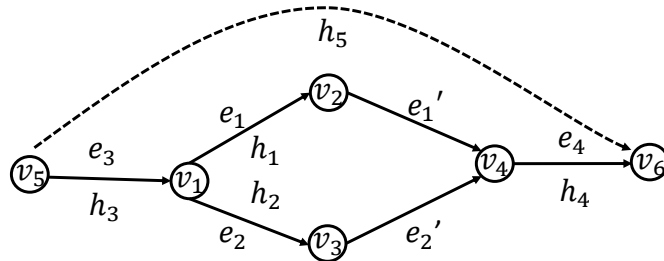


Fig. 4.3 An example with five commodities: h_i with $i = 1, 2, 3, 4$ having the same source and sink than edges e_i , and h_5 from source node v_5 to sink node v_6 . In this example we show that \mathcal{F}_{cut} is strictly better than \mathcal{F}_+ and \mathcal{F}_- .

Proof. (sketch) Consider the graph of Figure 4.3. It contains 6 directed edges each of capacity 1: $e_1 = v_1v_2$, $e_2 = v_1v_3$, $e'_1 = v_2v_4$, $e'_2 = v_3v_4$, $e_3 = v_5v_1$ and $e_4 = v_4v_6$. 5 commodities have to be carried through the network: $h_i, i = 1, 2, 3, 4$ having the same source and sink than edges e_i , and h_5 from source node v_5 to sink node v_6 . The polyhedral uncertainty set \mathcal{D} is defined by constraints: $d_{h_1} + d_{h_2} \leq 1$, $d_{h_3} + d_{h_5} \leq 1$ and $d_{h_4} + d_{h_5} \leq 1$ in addition to non-negativity constraints. By solving either \mathcal{F}_+ or \mathcal{F}_- , we get the following solution: $f_{h_5, e_1}(d) = \frac{1}{3} - \frac{d_{h_1}}{3} + \frac{d_{h_5}}{3}$, $f_{h_5, e_2}(d) = \frac{d_{h_1}}{3} + \frac{d_{h_5}}{3}$, $f_{h_5, e_3}(d) = f_{h_5, e_4}(d) = \frac{1}{3} + \frac{2}{3}d_{h_5}$ while the assignment related to the other commodities is obvious since only one path is available for each of them. The congestion m related to this solution is $m = \frac{4}{3}$. The fact that there is no solution of either \mathcal{F}_+ or \mathcal{F}_- with $m < \frac{4}{3}$ was numerically checked by solving the problems using the algorithms of Section 4.6. On the other hand, the optimal solution of \mathcal{F}_{cut} has a congestion $m = 1$ since we can consider the following solution: $f_{h_5, e_1}(d) = f_{h_5, e'_1}(d) = 1 - d_{h_1}$, $f_{h_5, e_2}(d) = f_{h_5, e'_2}(d) = 1 - d_{h_2}$ and $f_{h_5, e_3}(d) = f_{h_5, e_4}(d) = d_{h_5}$. To show the validity of the solution, observe that a cut separating v_5 and v_6 contains either e_3 or e_4 . Its capacity is obviously greater than or equal to d_{h_5} . Moreover, a separating cut containing neither e_3 nor e_4 will necessarily contain either e_1 or e'_1 and either e_2 or e'_2 . The capacity of the cut will then be at least $2 - d_{h_1} - d_{h_2} \geq 1 \geq d_{h_5}$. \square

Let us now study the complexity of \mathcal{F}_{cut} . If the number of separating cuts in the graph is polynomial (in fact one should only consider those not included in larger cuts), then \mathcal{F}_{cut} can still be solved using standard robust optimization techniques (See Section 4.6). However, we will show that solving \mathcal{F}_{cut} is unfortunately NP-hard.

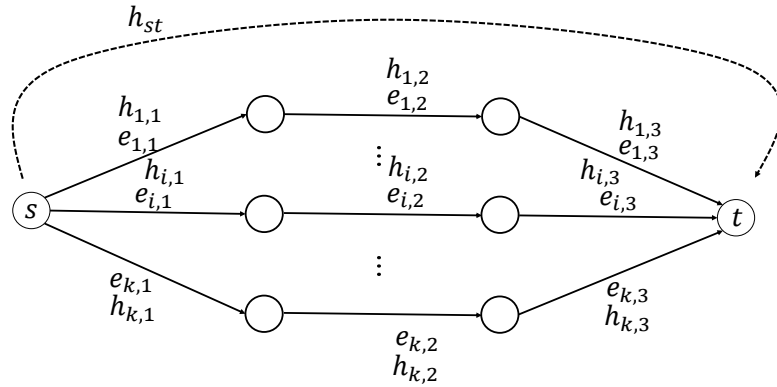


Fig. 4.4 An example with $3k+1$ commodities: $h_{i,j}$ with $i = 1, \dots, k, j = 1, 2, 3$ having the same source and sink than edges $e_{i,j}$, and h_{st} from source node s to sink node t . In this example we prove that it is NP-hard to solve \mathcal{F}_{cut} .

Proposition 4.3.2. *It is NP-hard to solve \mathcal{F}_{cut} .*

Proof. We are going to propose a reduction from the 3-SAT problem. Let us consider a 3-SAT formula φ with k clauses and r variables. We note $\mathcal{L} = \{l_1, \dots, l_r, \neg l_1, \dots, \neg l_r\}$

the set of the literals appearing in formula φ and $l_{i,j}$ the literal appearing in the i -th clause C_i at the j -th position for $i = 1, \dots, k$ and $j = 1, 2, 3$. We create a polyhedron Ξ by adding for each literal $l \in \mathcal{L}$ a non-negative variable ξ_l and for $p = 1, \dots, r$, we add the constraint $\xi_{l_p} + \xi_{\neg l_p} = 1$.

We build as follows a graph G , a set of commodities \mathcal{H} and a polyhedral uncertainty set \mathcal{D} . For each $i = 1, \dots, k$, $j = 1, 2, 3$ we add 3 consecutive directed edges $e_{i,j}$ (see Figure 4.4) and 3 commodities $h_{i,j}$ with $s(h_{i,j}) = s(e_{i,j})$ and $t(h_{i,j}) = t(e_{i,j})$, and $d_{h_{i,j}} \leq \xi_{l_{i,j}}$. We impose that all nodes $s(e_{i,1})$ (resp. $t(e_{i,3})$) for $i = 1, \dots, k$ are equal to a single node noted s (resp. t) (see Figure 4.4). We consider an additional commodity h_{st} from s to t whose value satisfies $d_{h_{st}} \leq 1$. The uncertainty polyhedron \mathcal{D} is then obtained by projecting Ξ on the space of d_h variables. Finally, the capacity c_e of each edge e is here equal to 1 ($c_e = 1$).

Let us now prove that the optimal objective value of \mathcal{F}_{cut} is $m = 1$ if and only if the 3-SAT formula φ is not satisfiable. If φ is satisfiable, then there is a demand vector (induced by the truth assignment) such that for each path between s and t (there is one path corresponding to each clause), at least one commodity whose endpoints are on the path is equal to 1 (a commodity corresponding to a true literal). This implies that all paths are blocked and thus $m > 1$ since one has to route commodity h_{st} through the network. If φ is not satisfiable, then for each extreme point of \mathcal{D} , there is at least one free path to route the demand $d_{h_{st}}$. In other words, each extreme demand vector of \mathcal{D} can be routed through the network. Since each demand vector d inside \mathcal{D} can be written as a convex combination of the extreme points of \mathcal{D} , d can also be routed through the network without requiring more than one unit of capacity on each edge. It is then clear that the solution defined by $m = 1$ and $f_{h_{st}, e_{i,j}}(d) = 1 - d_{h_{i,j}}$ is feasible for \mathcal{F}_{cut} . The optimal congestion is hence equal to 1. \square

Notice that the separation problem related to the polyhedron \mathcal{D} introduced in the proof above (i.e., given some vector d , check whether d belongs to \mathcal{D} or provide a cut separating d from \mathcal{D}) can obviously be solved in polynomial-time [Grötschel et al. (2012)]. Otherwise the NP-hardness of solving \mathcal{F}_{cut} would be a direct consequence of the difficulty of the separation problem related to \mathcal{D} .

We know from proposition 4.3.1 that \mathcal{F}_{cut} can be strictly better than \mathcal{F}_+ and \mathcal{F}_- . We just proved that \mathcal{F}_{cut} is NP-hard to solve. We also recalled in Section 1 that the robust network design problem is NP-hard. This is called *dynamic routing* in several papers where there are no restrictions related to the choice of $f_{h,e}(d)$ which can be any function of d that allows the routing of each commodity. One can then wonder whether there is any difference between \mathcal{F}_{cut} and the dynamic routing formulation. The following proposition answers this question.

For sake of completeness we provide a possible formulation \mathcal{F}_{dyn} related to dynamic

routing.

$$\begin{aligned}
& \min m \\
& \sum_{e \in \delta_+(C)} f_{h,e}(d) \geq d_h, \forall d \in \mathcal{D}, h \in \mathcal{H}, C \subset V, s(h) \in V, t(h) \notin V \\
& \sum_{h \in \mathcal{H}} f_{h,e}(d) \leq c_e m, \quad \forall e \in E, d \in \mathcal{D} \\
& f_e^h(d) \geq 0, \quad \forall e \in E, h \in \mathcal{H}, d \in \mathcal{D}
\end{aligned} \tag{4.10}$$

The difference between \mathcal{F}_{cut} and \mathcal{F}_{dyn} lies in the affine form of $f_{h,e}(d)$ that is imposed only for \mathcal{F}_{cut} .

Proposition 4.3.3. \mathcal{F}_{dyn} is strictly better than \mathcal{F}_{cut} on some instances.

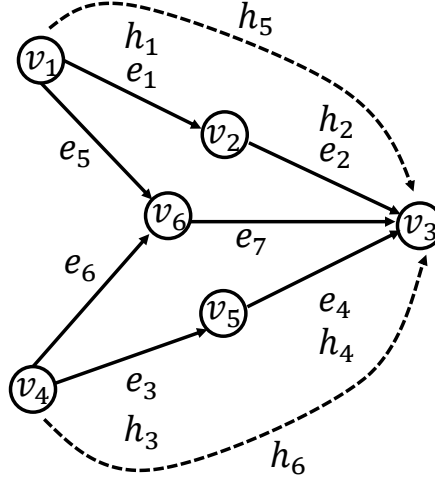


Fig. 4.5 An example with six commodities: h_i with $i = 1, 2, 3, 4$ having the same source and sink than edges e_i , h_5 from source node v_1 to sink node v_3 , and h_6 from source node v_4 to sink node v_3 . In this example we prove that \mathcal{F}_{dyn} is strictly better than \mathcal{F}_{cut} .

Proof. Let us consider again the example of Figure 4.5 containing 6 vertices and the 7 directed edges: $e_1 = v_1v_2$, $e_2 = v_2v_3$, $e_3 = v_4v_5$, $e_4 = v_5v_3$, $e_6 = v_4v_6$, $e_5 = v_1v_6$ and $e_7 = v_6v_3$, of capacity 1 each. It also contains 6 commodities (see Figure 4.5): h_i with $i = 1, 2, 3, 4$ having the same source and sink than edges e_i , h_5 from source node v_1 to sink node v_3 , and h_6 from source node v_4 to sink node v_3 . The uncertainty set \mathcal{D} is here defined by the constraints: $d_{h_1} + d_{h_3} \leq 1$, $d_{h_1} + d_{h_4} \leq 1$, $d_{h_2} + d_{h_3} \leq 1$, $d_{h_2} + d_{h_4} \leq 1$, $d_{h_5} \leq 1$ and $d_{h_6} \leq 1$ in addition to non-negativity constraints.

To show that the optimal congestion provided by \mathcal{F}_{dyn} is equal to 1 we only have to prove that each extreme point of \mathcal{D} can be routed without using more than 1 unit of capacity. It is clear that the more constraining scenarios are those where $d_{h_5} = d_{h_6} = 1$. We also either have $d_{h_1} = d_{h_2} = 1$ and $d_{h_3} = d_{h_4} = 0$ or $d_{h_1} = d_{h_2} = 0$ and $d_{h_3} = d_{h_4} = 1$. By symmetry, we can just focus on the first case ($d_{h_5} = d_{h_6} = 1$, $d_{h_1} =$

$d_{h_2} = 1$ and $d_{h_3} = d_{h_4} = 0$) where one can clearly route commodity h_5 completely through the path containing e_5 and e_7 while h_6 is routed through e_3 and e_4 . In other words, $m = 1$ for formulation (4.10).

Let us now prove by contradiction that the congestion obtained by \mathcal{F}_{cut} is strictly greater than 1. Assume it to be equal to 1. Observe that when either $d_{h_1} = 1$ or $d_{h_2} = 1$, commodity h_5 is necessarily routed through e_7 . This implies that $f_{h_5,e_7}(\mathbb{1}_{h_1} + \mathbb{1}_{h_5} + \mathbb{1}_{h_6}) = f_{h_5,e_7}(\mathbb{1}_{h_2} + \mathbb{1}_{h_5} + \mathbb{1}_{h_6}) = f_{h_5,e_7}(\mathbb{1}_{h_1} + \mathbb{1}_{h_2} + \mathbb{1}_{h_5} + \mathbb{1}_{h_6}) = 1$, where $\mathbb{1}_{h_i}$ denotes the demand vector where all demands are equal to 0 while $d_{h_i} = 1$. Since $f_{h_5,e_7}(d) = x_{h_5,e_7}^0 + x_{h_5,e_7}^{h_1} d_{h_1} + x_{h_5,e_7}^{h_2} d_{h_2} + x_{h_5,e_7}^{h_3} d_{h_3} + x_{h_5,e_7}^{h_4} d_{h_4} + x_{h_5,e_7}^{h_5} d_{h_5} + x_{h_5,e_7}^{h_6} d_{h_6}$, the previous equalities imply that $x_{h_5,e_7}^{h_1} = 0$ and $x_{h_5,e_7}^{h_2} = 0$. Observe that when either d_{h_3} or d_{h_4} is equal to 1 and demand $d_{h_6} = 1$ then commodity h_6 is fully routed through e_7 which requires that h_5 does not use e_7 . Consequently, we have $f_{h_5,e_7}(\mathbb{1}_{h_3} + \mathbb{1}_{h_5} + \mathbb{1}_{h_6}) = f_{h_5,e_7}(\mathbb{1}_{h_4} + \mathbb{1}_{h_5} + \mathbb{1}_{h_6}) = f_{h_5,e_7}(\mathbb{1}_{h_3} + \mathbb{1}_{h_4} + \mathbb{1}_{h_5} + \mathbb{1}_{h_6}) = 0$. These equalities lead to $x_{h_5,e_7}^{h_3} = 0$ and $x_{h_5,e_7}^{h_4} = 0$. From $f_{h_5,e_7}(\mathbb{1}_{h_3} + \mathbb{1}_{h_4} + \mathbb{1}_{h_5} + \mathbb{1}_{h_6}) = 0$, we get that $x_{h_5,e_7}^{h_5} + x_{h_5,e_7}^{h_6} = 0$, while $f_{h_5,e_7}(\mathbb{1}_{h_1} + \mathbb{1}_{h_2} + \mathbb{1}_{h_5} + \mathbb{1}_{h_6}) = 1$ leads to the contradictory equality $x_{h_5,e_7}^{h_5} + x_{h_5,e_7}^{h_6} = 1$. \square

4.4 Extended graph formulation

We have seen that formulations \mathcal{F}_- and \mathcal{F}_+ can be strictly tighter than $\mathcal{F}_=$ (i.e., closer to \mathcal{F}_{dyn}). The difference between \mathcal{F}_- and \mathcal{F}_+ lies in the sign of the terms $\sum_{e \in \delta_+(v)} f_{h,e}(d) - \sum_{e \in \delta_-(v)} f_{h,e}(d)$ for $v \in V \setminus \{s(h), t(h)\}$ required to be negative for \mathcal{F}_- and positive for \mathcal{F}_+ . Our first trial to improve both \mathcal{F}_- and \mathcal{F}_+ led to formulation \mathcal{F}_{cut} . However, \mathcal{F}_{cut} is generally NP-hard to solve. We would like to propose a stronger formulation that is still easy to solve, where the features of \mathcal{F}_- and \mathcal{F}_+ are combined in some way.

We propose the following. For each commodity $h \in \mathcal{H}$, and for each vertex $v \in V \setminus \{s(h), t(h)\}$, we add to G the two directed edges $t(h)v$ and $vs(h)$. We also add an edge directed from $t(h)$ to $s(h)$. For each commodity h , an $s(h)t(h)$ flow f^h is considered in the extended graph. Notice that the extra edges we added $t(h)v$, $vs(h)$ and $t(h)s(h)$ can only be used by commodity h . Flow conservation constraints can be expressed as follows.

$$f_{h,vs(h)}(d) + \sum_{e \in \delta_+(v)} f_{h,e}(d) - f_{h,t(h)v}(d) - \sum_{e \in \delta_-(v)} f_{h,e}(d) = 0 \text{ if } v \neq s(h), t(h) \quad (4.11)$$

$$\sum_{e \in \delta_+(s(h))} f_{h,e}(d) - \sum_{v \in V \setminus \{s(h)\}} f_{h,vs(h)}(d) = d_h. \quad (4.12)$$

Notice that $\delta_+(v)$ and $\delta_-(v)$ contain only edges belonging to G .

For sake of completeness, we give below the new formulation $\overline{\mathcal{F}}$.

min m

$$\sum_{e \in \delta_+(v) \cup \{vs(h)\}} \left(x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'} \right) - \sum_{e \in \delta_-(v) \cup \{t(h)v\}} \left(x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'} \right) = 0$$

$$\forall h \in \mathcal{H}, v \in V \setminus \{s(h), t(h)\}, d \in \mathcal{D} \quad (4.13a)$$

$$\sum_{e \in \delta_+(s(h))} \left(x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'} \right) - \sum_{v \in V \setminus \{s(h)\}} \left(x_{h,vs(h)}^0 + \sum_{h' \in \mathcal{H}} x_{h,vs(h)}^{h'} d_{h'} \right) = d_h$$

$$\forall h \in \mathcal{H}, d \in \mathcal{D}$$

$$\sum_{h \in \mathcal{H}} \left(x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'} \right) \leq c_e m, \quad \forall e \in E, d \in \mathcal{D} \quad (4.13b)$$

$$x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'} \geq 0, \quad \forall d \in \mathcal{D}, h \in \mathcal{H}, e \in E \cup \{t(h)s(h)\} \cup_{v \in V \setminus \{s(h), t(h)\}} \{vs(h), t(h)v\}$$

$$(4.13c)$$

Observe that there are no explicit capacity limitations for the edges not belonging to E (the added edges of type $vs(h)$ and $t(h)v$). However, non-negativity is required for the flow on these edges.

It is easy to see that \mathcal{F}_+ (resp. \mathcal{F}_-) is a special case of $\overline{\mathcal{F}}$ since the term $\sum_{e \in \delta_+(v)} f_{h,e}(d) - \sum_{e \in \delta_-(v)} f_{h,e}(d)$ (resp. $\sum_{e \in \delta_-(v)} f_{h,e}(d) - \sum_{e \in \delta_+(v)} f_{h,e}(d)$) is positive in \mathcal{F}_+ (resp. \mathcal{F}_-) and can be seen as the flow going through an additional edge $t(h)v$ (resp. $vs(h)$). In other words, by considering only edges of type $t(h)v$ (resp. $vs(h)$) and solving $\overline{\mathcal{F}}$ we get \mathcal{F}_+ (resp. \mathcal{F}_-).

Let us now prove that $\overline{\mathcal{F}}$ is valid.

Proposition 4.4.1. $\overline{\mathcal{F}}$ is valid.

Proof. For each commodity $h \in \mathcal{H}$ and for each $d \in \mathcal{D}$ the solution induced by $\overline{\mathcal{F}}$ is a $s(h)t(h)$ flow in the extended graph of value d_h . Consequently, each cut of the extended graph that separates $s(h)$ and $t(h)$ has necessarily a capacity greater than or equal to d_h . Observe however that additional edges of type $t(h)v$, $vs(h)$ and $t(h)s(h)$ do not belong to any separating cut. This means that any separating cut in the extended graph contains only edges from the original graph. We can thus deduce that any separating cut in G has a capacity greater than or equal to d_h . By the maximum-flow minimum-cut theorem, it is then possible to route commodity h using the capacities $f_{h,e}(d)$ on the edges of G . \square

We give below an example showing that $\overline{\mathcal{F}}$ can be strictly better than both \mathcal{F}_- and \mathcal{F}_+ .

Proposition 4.4.2. $\overline{\mathcal{F}}$ is less conservative than \mathcal{F}_- and \mathcal{F}_+ .

Proof. (sketch)

We will consider again the example of Figure 4.3. We have seen that the congestion obtained by both \mathcal{F}_- and \mathcal{F}_+ is equal to $\frac{4}{3}$. We give here a feasible solution of $\overline{\mathcal{F}}$ for which the congestion is only 1.25. We only have to determine the assignments related to h_5 . Consider the solution defined by:

- $f_{h_5, e_1}(d) = f_{h_5, e'_1}(d) = 0.5 + 0.25d_{h_5} - 0.5d_{h_1}$,
- $f_{h_5, e_2}(d) = f_{h_5, e'_2}(d) = 0.25d_{h_5} + 0.5d_{h_1}$, $f_{h_5, e_3}(d) = 0.25 + 0.75d_{h_5}$,
- $f_{h_5, e_4}(d) = 0.5 + 0.5d_{h_5} - 0.25d_{h_4}$, $f_{h_5, v_6 v_5}(d) = 0.25 - 0.25d_{h_5} - 0.25d_{h_4}$,
- $f_{h_5, v_4 v_5}(d) = 0.25d_{h_4}$, $f_{h_5, v_6 v_1}(d) = 0.25 - 0.25d_{h_5}$,

while $f_{h_5, e}(d) = 0$ for all other edges. Observe that flow conservation constraints are satisfied in the extended graph. The three last edges mentioned above (i.e., $v_6 v_5$, $v_4 v_5$ and $v_6 v_1$) do not belong to G (they are of type $vs(h)$ and $t(h)v$). The fact that they appear in the solution means that $\sum_{e \in \delta_+(v)} f_{h, e}(d) - \sum_{e \in \delta_-(v)} f_{h, e}(d)$ will be always positive for $v = v_1$ and negative for $v = v_4$ (this is to say that this solution is neither feasible for \mathcal{F}_- nor for \mathcal{F}_+). Observe also that the total capacity used on e_1 is given by $f_{h_5, e_1}(d) + d_{h_1} = 0.5 + 0.25d_{h_5} + 0.5d_{h_1} \leq 1.25$. Similarly, the capacity used on e_2 is equal to $0.25d_{h_5} + 0.5d_{h_1} + d_{h_2} = 0.25d_{h_5} + 0.5(d_{h_1} + d_{h_2}) + 0.5d_{h_2} \leq 1.25$ (since $d_{h_1} + d_{h_2} \leq 1$). One can check that the same holds for edges e'_1 and e'_2 . The positivity of the terms $f_{h_5, e}(d)$ is also easy to check for each edge e using the definition of \mathcal{D} . \square

Since \mathcal{F}_{cut} is the best formulation that one can get when $f_{h, e}(d)$ is assumed to be affine and is generally NP-hard to solve while $\overline{\mathcal{F}}$ can be solved in polynomial-time, there are cases where $\overline{\mathcal{F}}$ is strictly dominated by \mathcal{F}_{cut} . This is shown by the example used above (Figure 4.3) for which we already proved in proposition 4.3.1 that \mathcal{F}_{cut} leads to a congestion equal to 1 while $\overline{\mathcal{F}}$ gives a congestion equal to 1.25.

To close this section, we would like to add that formulations (4.3) and (4.6) that were proposed as an alternative to formulation (4.1) (i.e., $\mathcal{F}_=$), can also be expressed in the context of the extended graph. To write the cycle-based formulation (4.6), we only have to take into account the fact that the set of cycles will here depend on the commodities (since for each commodity h we added some edges that can only be used by this commodity).

4.5 Aggregation

One standard way to solve classical linear multi-commodity problems in a more efficient way consists in aggregating commodities either by source or by sink [Ahuja et al. (1993)]. Let us then try to do the same in the context of polyhedral uncertainty and

affine decision rules.

4.5.1 Aggregation for $\mathcal{F}_=$

Let $S \subset V$ and $T \subset V$ be two subsets such that for each $h \in \mathcal{H}$ we either have $s(h) \in S$ or $t(h) \in T$. All commodities having s as a source ($s \in S$) will be aggregated and considered as one commodity having a source s and several sinks. Similarly, all commodities having t as a sink ($t \in T$) are aggregated into one commodity having several sources and one sink t . It may happen that $s(h) \in S$ and $t(h) \in T$ simultaneously occur, then we arbitrarily decide whether h is aggregated by source or by sink. For each $s \in S$ (resp. $t \in T$), let us use \mathcal{H}_s (resp. \mathcal{H}^t) to denote the set of commodities having s (resp. t) as a source (resp. sink) and aggregated by source (resp. sink). For any $h \in \mathcal{H}$, it will be more convenient here to use $d_{h_{s(h)t(h)}}$ to denote the demand value of the commodity (there is no ambiguity since we can assume that there are no demands having exactly the same source and the same sink).

We also define for each $s \in S$ the set $T(s) = \{v : h \in \mathcal{H}_s, v = t(h)\}$ to denote the set of vertices v such that there is a commodity aggregated by source s and having v as a sink. Similarly, for $t \in T$, let $S(t) = \{v : h \in \mathcal{H}^t, v = s(h)\}$.

Applying this aggregation for $\mathcal{F}_=$ leads to the following aggregated formulation \mathcal{F}_{agg} .

$$\begin{aligned} \min m \\ \sum_{e \in \delta_+(v)} \left(x_{s,e}^0 + \sum_{h' \in \mathcal{H}} x_{s,e}^{h'} d_{h'} \right) - \sum_{e \in \delta_-(v)} \left(x_{s,e}^0 + \sum_{h' \in \mathcal{H}} x_{s,e}^{h'} d_{h'} \right) &= \begin{cases} -d_{h_{sv}} & \text{if } v \in T(s) \\ 0 & \text{otherwise} \end{cases} \\ \forall s \in S, v \in V \setminus \{s\}, d \in \mathcal{D} \end{aligned} \quad (4.14a)$$

$$\begin{aligned} \sum_{e \in \delta_+(v)} \left(x_{t,e}^0 + \sum_{h' \in \mathcal{H}} x_{t,e}^{h'} d_{h'} \right) - \sum_{e \in \delta_-(v)} \left(x_{t,e}^0 + \sum_{h' \in \mathcal{H}} x_{t,e}^{h'} d_{h'} \right) &= \begin{cases} d_{h_{vt}} & \text{if } v \in S(t) \\ 0 & \text{otherwise} \end{cases} \\ \forall t \in T, v \in V \setminus \{t\}, d \in \mathcal{D} \end{aligned} \quad (4.14b)$$

$$\sum_{s \in S} \left(x_{s,e}^0 + \sum_{h' \in \mathcal{H}} x_{s,e}^{h'} d_{h'} \right) + \sum_{t \in T} \left(x_{t,e}^0 + \sum_{h' \in \mathcal{H}} x_{t,e}^{h'} d_{h'} \right) \leq c_e m, \quad \forall e \in E, d \in \mathcal{D} \quad (4.14c)$$

$$x_{s,e}^0 + \sum_{h' \in \mathcal{H}} x_{s,e}^{h'} d_{h'} \geq 0, \quad \forall e \in E, s \in S, d \in \mathcal{D} \quad (4.14d)$$

$$x_{t,e}^0 + \sum_{h' \in \mathcal{H}} x_{t,e}^{h'} d_{h'} \geq 0, \quad \forall e \in E, t \in T, d \in \mathcal{D} \quad (4.14e)$$

Observe that all variables (except m) are either indexed by a source s or a sink t . The number of variables is almost proportional to $|S| + |T|$. Then to minimize the number of variables, one has to minimize $|S| + |T|$ which can obviously be done by computing a minimum vertex cover in a bipartite graph (the demand graph) and is equal

to the cardinality of a maximum matching (by König's theorem) [Günlük (2002)].

The validity of \mathcal{F}_{agg} is a direct consequence of the validity of aggregation when there is no uncertainty.

It is also obvious that any solution of $\mathcal{F}_=$ can be used to build a solution for \mathcal{F}_{agg} having the same congestion. For each $s \in S$ (resp. $t \in T$), we only have to sum the variables related to commodities belonging to \mathcal{H}_s (resp. \mathcal{H}^t) to get those related to s (resp. t).

We know that when there is no uncertainty, $\mathcal{F}_=$ and \mathcal{F}_{agg} are equivalent. One may wonder whether they are equivalent when polyhedral uncertainty is considered. The next proposition states that \mathcal{F}_{agg} dominates $\mathcal{F}_=$.

Proposition 4.5.1. *Formulation \mathcal{F}_{agg} is less conservative than formulation $\mathcal{F}_=$.*

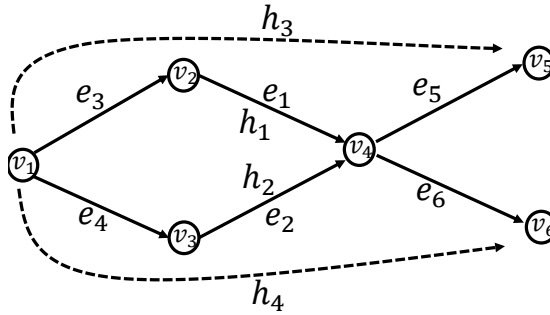


Fig. 4.6 An example with four commodities: h_i with $i = 1, 2$ having the same source and sink than edges e_i , h_3 from source node v_1 to sink node v_5 , and h_4 from source node v_1 to sink node v_6 . In this example we prove that formulation \mathcal{F}_{agg} is strictly better than formulation $\mathcal{F}_=$.

Proof. Let us consider the graph of Figure 4.6 containing 6 edges of capacity 1 each. There are 4 commodities: h_i with $i = 1, 2$ having the same source and sink than edges e_i , h_3 from source node v_1 to sink node v_5 , and h_4 from source node v_1 to sink node v_6 .

The polyhedron \mathcal{D} is defined as the set of demands $d \in \mathbb{R}_+^4$ satisfying the two equations $d_{h_1} + d_{h_2} = 1$ and $d_{h_3} + d_{h_4} = 1$. Due to the equalities defining \mathcal{D} , we can assume without generality loss that there is affine dependence on only d_{h_1} and d_{h_3} .

First, let us consider formulation \mathcal{F}_{agg} with only source aggregation. We will then aggregate commodities h_3 and h_4 . Consider the solution of \mathcal{F}_{agg} defined by: $f_{v_1, e_1}(d) = f_{v_1, e_3}(d) = 1 - d_{h_1}$, $f_{v_1, e_2}(d) = f_{v_1, e_4}(d) = d_{h_1}$, $f_{v_1, e_5}(d) = d_{h_3}$ and $f_{v_1, e_6}(d) = 1 - d_{h_3}$. Variables related to the two other sources v_2 and v_3 are fixed in an obvious way. This solution allows a congestion equal to 1.

Let us now assume that there is a feasible solution of $\mathcal{F}_=$ with congestion equal to 1. Let f be such a solution. For each edge e , we have $f_{h_3, e}(d) = x_{h_3, e}^0 + x_{h_3, e}^{h_1} d_{h_1} + x_{h_3, e}^{h_3} d_{h_3}$. Since the graph is acyclic and f_{h_3} is a positive flow, if the demand for a commodity h_3

is zero then the flow for this commodity must also be zero in model (4.1). Then when d is the demand vector where $d_{h_3} = d_{h_1} = 0$, we should have $f_{h_3,e}(d) = 0$ implying that $x_{h_3,e}^0 = 0$ for each edge e . Similarly, when d is such that $d_{h_3} = 0$ and $d_{h_1} = 1$ we also have $f_{h_3,e}(d) = 0$ leading to $x_{h_3,e}^{h_1} = 0$.

Let us now focus on edge e_1 . When $d_{h_1} = d_{h_3} = 1$, e_1 already carries commodity h_1 whose value is here equal to 1. Then there are no more resources that can be used by commodity h_3 implying that $f_{h_3,e_1}(d) \leq 0$. Using the positivity constraint we can deduce that $f_{h_3,e_1}(d) = 0$ when $d_{h_1} = d_{h_3} = 1$. Thus, $x_{h_3,e_1}^{h_3} = 0$. In other words, $f_{h_3,e_1}(d) = 0$ for any d implying that commodity h_3 is never routed through e_1 . It is then fully routed through e_2 . This is of course not possible without violating the capacity constraint of e_2 since commodity h_2 is already routed through e_2 . \square

According to Proposition 4.5.1 it should be understood that aggregation is not only interesting for accelerating problem solving (as is the case for problems without uncertainty), but it also leads to better solutions since we are getting closer to \mathcal{F}_{dyn} . In fact, by aggregating commodities and solving \mathcal{F}_{agg} , the capacities reserved for each aggregated commodity is affine while the capacities used by each individual commodity making up the aggregated one are not necessarily affine in d .

4.5.2 Sink aggregation for \mathcal{F}_+

Since \mathcal{F}_+ dominates $\mathcal{F}_=$, it would be interesting to perform some kind of aggregation to be able to solve larger problems and further reduce congestion.

We consider the aggregated formulation \mathcal{F}_{agg+} given below where only sink aggregation is possible (so $S = \emptyset$). Observe also that equality constraints (4.14b) are replaced by inequalities (4.15a).

$$\begin{aligned} \min m \\ \sum_{e \in \delta_+(v)} \left(x_{t,e}^0 + \sum_{h' \in \mathcal{H}} x_{t,e}^{h'} d_{h'} \right) - \sum_{e \in \delta_-(v)} \left(x_{t,e}^0 + \sum_{h' \in \mathcal{H}} x_{t,e}^{h'} d_{h'} \right) &\geq \begin{cases} d_{h_{vt}} & \text{if } v \in S(t) \\ 0 & \text{otherwise} \end{cases} \\ \forall t \in T, v \in V \setminus \{t\}, d \in \mathcal{D} & \end{aligned} \quad (4.15a)$$

$$\sum_{t \in T} \left(x_{t,e}^0 + \sum_{h' \in \mathcal{H}} x_{t,e}^{h'} d_{h'} \right) \leq c_e m, \quad \forall e \in E, d \in \mathcal{D} \quad (4.15b)$$

$$x_{t,e}^0 + \sum_{h' \in \mathcal{H}} x_{t,e}^{h'} d_{h'} \geq 0, \quad \forall e \in E, t \in T, d \in \mathcal{D} \quad (4.15c)$$

The validity of \mathcal{F}_{agg+} is less obvious than the validity of \mathcal{F}_{agg} , and it is demonstrated hereafter.

Proposition 4.5.2. \mathcal{F}_{agg+} is valid.

Proof. Consider any feasible solution f of \mathcal{F}_{agg+} . Let us select any traffic vector $d \in \mathcal{D}$ and any sink $t \in T$. For each vertex $v \in V \setminus \{t\}$ we add to G the edge tv of infinite capacity. Let us also add to G a “virtual” node s_t and an edge from s_t to each vertex $v \in S(t)$ of capacity $d_{h_{vt}}$. Then, starting from $f_{t,e}(d) = x_{t,e}^0 + \sum_{h' \in \mathcal{H}} x_{t,e}^{h'} d_{h'}$ for each edge $e \in E$, $f_{t,\cdot}$ can be extended to a positive flow from s_t to t by taking

$$f_{t,tv}(d) = \begin{cases} \sum_{e \in \delta_+(v)} f_{t,e}(d) - \sum_{e \in \delta_-(v)} f_{t,e}(d) & \text{if } v \in V \setminus S(t) \\ \sum_{e \in \delta_+(v)} f_{t,e}(d) - \sum_{e \in \delta_-(v)} f_{t,e}(d) - d_{h_{vt}} & \text{if } v \in S(t) \end{cases}$$

and $f_{t,s_tv}(d) = d_{h_{vt}}$ for $v \in S(t)$. We are then sending a flow of value $\sum_{v \in S(t)} d_{h_{vt}}$ from s_t to t . Directed cycles can be cancelled in a standard way by decreasing flow on the edges of each directed cycle. We can therefore assume that the set of edges for which $f_{t,e}(d) > 0$ does not contain any directed cycle. Since the flow on the “virtual” edges s_tv is exactly equal to $d_{h_{vt}}$, the $s_t t$ -flow induces $|S(t)|$ simultaneous positive flows, each from a vertex $v \in S(t)$ to t and of value exactly equal to $d_{h_{vt}}$. Each vt -flow ($v \in S(t)$) uses only original edges of G . This clearly implies that it is possible to simultaneously route the demand $d_{h_{vt}}$ for each $v \in S(t)$. Since this holds for any $t \in T$ and any $d \in \mathcal{D}$, the validity of the formulation is proved. \square

One can easily modify the example of Figure 4.6 to show that \mathcal{F}_{agg+} strictly dominates \mathcal{F}_+ . It is also easy to see that \mathcal{F}_{agg+} dominates \mathcal{F}_{agg} when only sink aggregation is considered to build \mathcal{F}_{agg} (i.e., when $|S| = 0$).

Finally, we should mention that source aggregation cannot be used in combination with \mathcal{F}_+ even if there is no uncertainty. Consider, for example, a graph having 4 vertices, s , v , t_1 and t_2 and 2 edges st_1 and vt_2 having some capacity. Assume that we have two commodities h_1 and h_2 of value 1 each from s to t_1 and from s to t_2 . Observe that there is even no path from s to t_2 so the network design problem has no solution. However, by taking $f_{s,vt_2} = 1$ and $f_{s,st_1} = 2$, we can ensure that all constraints related to the aggregated commodity will be satisfied (the flow entering t_1 is greater than 1, the flow going out of v is greater than what is going into v , the flow reaching t_2 is greater than 1, and we even have that what comes out of s is greater than the sum of the two demands).

4.5.3 Source aggregation for \mathcal{F}_-

Aggregation can also be considered in combination with \mathcal{F}_- . However, only source aggregation can be used. The obtained formulation denoted by \mathcal{F}_{agg-} would be the

following.

$$\begin{aligned} \min m \\ \sum_{e \in \delta_+(v)} \left(x_{s,e}^0 + \sum_{h' \in \mathcal{H}} x_{s,e}^{h'} d_{h'} \right) - \sum_{e \in \delta_-(v)} \left(x_{s,e}^0 + \sum_{h' \in \mathcal{H}} x_{s,e}^{h'} d_{h'} \right) \leq \begin{cases} -d_{h_{sv}} & \text{if } v \in T(s) \\ 0 & \text{otherwise} \end{cases} \\ \forall s \in S, v \in V \setminus \{s\}, d \in \mathcal{D} \end{aligned} \quad (4.16a)$$

$$\sum_{s \in S} \left(x_{s,e}^0 + \sum_{h' \in \mathcal{H}} x_{s,e}^{h'} d_{h'} \right) \leq c_e m, \quad \forall e \in E, d \in \mathcal{D} \quad (4.16b)$$

$$x_{s,e}^0 + \sum_{h' \in \mathcal{H}} x_{s,e}^{h'} d_{h'} \geq 0, \quad \forall e \in E, s \in S, d \in \mathcal{D} \quad (4.16c)$$

The proof of validity of \mathcal{F}_{agg-} is very similar to the proof of validity of \mathcal{F}_{agg+} . One can also build examples where \mathcal{F}_{agg-} strictly dominates \mathcal{F}_- . It is also easy to see that \mathcal{F}_{agg-} dominates \mathcal{F}_{agg} when only source aggregation is considered to build \mathcal{F}_{agg} (i.e., when $|T| = 0$).

Finally, we would like to mention that aggregation can also be considered in the context of formulation \mathcal{F}_{cut} . However, since solving \mathcal{F}_{cut} is NP-hard and aggregation would not change the theoretical complexity, we are not going to study this kind of aggregation.

4.6 Solution Methods and Numerical Evaluation

Figure 4.7 summarizes the main domination relations between the models introduced or recalled in the chapter. However, a numerical evaluation is needed to quantify the difference in terms of performance between these variants.

In this section, we begin by presenting the two types of uncertainty sets considered in the evaluations. For the sake of completeness, we briefly recall in Section 4.6.2 standard duality-based methods to solve the introduced formulations. Data instances considered for evaluation are described in Section 4.6.3 and, finally, we present all the results in Section 4.6.4.

4.6.1 Uncertainty sets

For the numerical evaluation, we consider two different uncertainty sets. We first use the *Budget* uncertainty of [Bertsimas and Sim (2003)] (recalled in Chapter 1) and defined as follows.

$$\mathcal{D} = \{d \in \mathbb{R}^{\mathcal{H}} : d_h = \bar{d}_h + z^h \hat{d}_h, \sum_{h \in \mathcal{H}} z_h \leq \Gamma, 0 \leq z_h \leq 1, \forall h \in \mathcal{H}\} \quad (4.17)$$

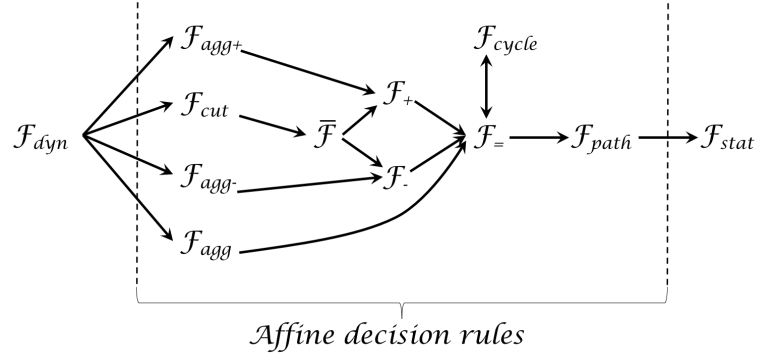


Fig. 4.7 Domination relations between the models introduced or recalled in the chapter.

where \bar{d}^h is the nominal demand for commodity h , \hat{d}_h is the maximum possible deviation from \bar{d}_h , and Γ is a parameter that specifies a limit (the budget) on the deviations of all demands at the same time with respect to the nominal values.

Second, we consider the *All Routable Demands* uncertainty set [Azar et al. (2003)] which contains all demand vectors that can be routed through a given network where capacities are fixed and routing can be adapted to each demand vector (See Chapter 1 for more details).

The two uncertainty sets introduced above are easy to handle (i.e., the separation problem related to each set can be solved in polynomial-time).

4.6.2 Problem solving methods

The models introduced in this chapter involve constraints that must be satisfied for all traffic vectors $d \in \mathcal{D}$.

When \mathcal{D} is a polytope having a polynomial-number of extreme points, some formulations such as \mathcal{F}_{dyn} can be solved by considering the constraints related to each extreme point. However, for most of polytopes considered in literature (such as those described above), the number of extreme points is not polynomial. Then there are mainly two methods to handle constraints involving d : either cutting-plane algorithms where traffic vectors are generated in iterative way [Ben-Ameur and Kerivin (2003)] or duality-based approaches [Ben-Tal et al. (2009)]. While the cutting-plane approach can be applied for any tractable polytope (i.e., for which separation is polynomial), the second approach is recommended when the polyhedral set can be described using a limited number of variables and constraints [Ben-Ameur et al. (2012), Ben-Tal et al. (2009)].

We are then going to use duality-based approaches to solve the problems introduced in the chapter. Duality allows to obtain equivalent compact linear programs of the original problems [Ben-Tal et al. (2009)]. In the following we will describe, as an example, how this is done for model (4.3) and the *Budget* uncertainty set \mathcal{D} described in

(4.17). The same method can be (quite) straightforwardly applied to the other models. The numerical results that we will present later in the section are obtained using this method.

For each edge e , constraint (4.3b) (recall that this latter is given as:

$$\sum_{h \in \mathcal{H}} \left(x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'} \right) \leq c_e m$$

is satisfied for all traffic vectors $d \in \mathcal{D}$ if and only if the constraint

$$\max_{d \in \mathcal{D}} \sum_{h \in \mathcal{H}} \left(x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} d_{h'} \right) \leq c_e m$$

is satisfied. Thus by writing the polyhedron \mathcal{D} in a more explicit form we obtain that a given solution (x, m) of (4.3) satisfies this constraint if and only if the solution of the following linear program gives a capacity reservation/congestion value that is less than $c_e m$.

$$\begin{aligned} \max_z \sum_{h \in \mathcal{H}} \left(x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} (\bar{d}_{h'} + z^{h'} \hat{d}_{h'}) \right) \\ \sum_{h' \in \mathcal{H}} z^{h'} \leq \Gamma \end{aligned} \tag{4.18a}$$

$$0 \leq z^{h'} \leq 1 \quad \forall h' \in \mathcal{H} \tag{4.18b}$$

By linear programming duality theory, model (4.18) has an optimal solution of value less than $c_e m$ if and only if the following dual linear program has a feasible solution of value less than $c_e m$.

$$\begin{aligned} \min_{\pi, \mu} \sum_{h \in \mathcal{H}} \left(x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} \bar{d}_{h'} \right) + \Gamma \pi_e + \sum_{h' \in \mathcal{H}} \mu_e^{h'} \\ \pi_e + \mu_e^{h'} \geq \hat{d}_{h'} \sum_{h \in \mathcal{H}} x_{h,e}^{h'} \quad \forall h' \in \mathcal{H} \\ \mu_e^{h'} \geq 0, \pi_e \geq 0 \quad \forall h' \in \mathcal{H} \end{aligned}$$

where π_e and $\mu_e^{h'}$ are the dual variables corresponding to constraints (4.18a) and (4.18b), respectively.

We can thus replace each constraint (4.3b) by the following inequalities.

$$\begin{aligned}
\sum_{h \in \mathcal{H}} \left(x_{h,e}^0 + \sum_{h' \in \mathcal{H}} x_{h,e}^{h'} \bar{d}_{h'} \right) + \Gamma \pi_e + \sum_{h' \in \mathcal{H}} \mu_e^{h'} &\leq c_e m \\
\pi_e + \mu_e^{h'} &\geq \hat{d}_{h'} \sum_{h \in \mathcal{H}} x_{h,e}^{h'} && \forall h' \in \mathcal{H} \\
\mu_e^{h'} \geq 0, \pi_e &\geq 0 && \forall h' \in \mathcal{H}
\end{aligned}$$

Constraints (4.3c) can also be dualized in a very similar way.

Finally, we should mention that in formulations $\mathcal{F}_=$, \mathcal{F}_+ , \mathcal{F}_- , \mathcal{F}_{agg+} , \mathcal{F}_{agg-} , \mathcal{F}_{agg} and $\overline{\mathcal{F}}$ the flow conservation constraints are handled as done in (4.3). For example, to solve \mathcal{F}_+ , by adding virtual edges of type $t(h)v$, we recover again equalities in the flow conservation constraints that should be satisfied for each $d \in \mathcal{D}$. These equalities are then replaced by a set of equalities that is similar to (4.3a).

The static routing approach (noted as \mathcal{F}_{stat} in the result tables) consists in choosing a fixed routing for all demand scenarios (i.e., $f_{h,e}(d) = x_e^{h0}$ for any $d \in \mathcal{D}$). It is described in [Ben-Ameur and Kerivin (2003), Azar et al. (2003)]. Here we solve this problem with the same duality-based techniques.

Further details about the solution methods are provided in Appendix A.

4.6.3 Network instances

We consider Abilene and Geant, two publicly available directed network topologies taken from the SNDlib [Orlowski et al. (2007)] library and commonly used in the networking community for numerical evaluations. The former is of medium size (12 nodes and 30 links) while the latter is of larger size (24 nodes and 72 links). The arc capacities are those present in the SNDlib instances.

We compare the affine routing and static routing formulations considering the minimization of two classical objective functions: 1) the Maximum Link Utilization (denoted as *MLU* which is nothing but the optimal value of m) and 2) a linear reservation cost (denoted as *Linear*). The *MLU* is expressed as $\max_{e \in E} \frac{u_e}{c_e}$ where c_e is the capacity of edge e and u_e is the reserved capacity on such edge. While the *Linear* objective is expressed as $\sum_{e \in E} \lambda_e u_e$ where λ_e are scalars corresponding to the unit cost of underlying resources.

To generate different sets of commodities on each instance with an increasing number of demands, we begin with the set \mathcal{H}_0 consisting of all the possible commodities between sources and destinations. We generate a subset $\mathcal{H}_1 \subseteq \mathcal{H}_0$ by selecting commodities from \mathcal{H}_0 with a uniform probability distribution. We re-iterate this process until \mathcal{H}_1 is of the size we desire. Next, we build a subset \mathcal{H}_2 of \mathcal{H}_1 with the same procedure to get a smaller set of demands. Successively, we obtain a sequence of demand

sets of decreasing size that are successively included in each other.

For each topology and each objective function we also consider the two uncertainty sets, *Budget* and *All Routable Demands*. The parameters for the *Budget* polyhedra are set as follow: $\bar{d}^h = \hat{d}^h = \min_{e \in E} c_e$ for all commodity $h \in \mathcal{H}$ and $\Gamma = \sqrt{|\mathcal{H}|}$.

4.6.4 Numerical results

We first analyze the solutions from the different formulations. Then, we compare solution times and model sizes.

We compare static routing (denoted as *Static*) with affine routing formulations that can be categorized into 5 groups: (1) those based on the relaxation of flow conservation constraints (\mathcal{F}_- and \mathcal{F}_+) of affine routing ($\mathcal{F}_=$), (2) the one based on aggregation (\mathcal{F}_{agg}), (3) those using a mix of the two former (\mathcal{F}_{agg-} and \mathcal{F}_{agg+}), (4) the one using the elementary cycle formulation (\mathcal{F}_{cycle}), and (5) the one using the extended graph formulation ($\bar{\mathcal{F}}$).

In our implementation, we used the CPLEX solver version 12.6.3 on servers having four Intel Xeon E5-4627 v2 3.3 GHz CPU cores and 512GB of memory. In all our computations, CPLEX is configured without a time limit and the default optimality gap. We used Julia to model problems and interface with the solver.

4.6.4.1 Comparison of objective values

We present two series of tables, Tables 4.1 and 4.2, with the solution of all formulations on Abilene and Geant, respectively. For each topology, we consider the two polyhedra and the two objective functions described above.

For each case, the table is organized as follows. The second row gives the number of demands (or commodities) $|\mathcal{H}|$ ranging from 10 to 30 demands for the instances related to *Budget* polyhedra (4.17), and from 5 to 15 for the more computationally expensive *All Routable Demands* polyhedra, except for Geant and the MLU objective (Table 4.2d) where the instances with 15 demands become prohibitively expensive to compute. The subsequent rows report the value of objective function at the optimum for all affine routing variants presented in this chapter: $\mathcal{F}_=$, \mathcal{F}_- , \mathcal{F}_+ , \mathcal{F}_{agg} , \mathcal{F}_{agg-} , \mathcal{F}_{agg+} , \mathcal{F}_{cycle} , and $\bar{\mathcal{F}}$. We also give the cost of static routing solution \mathcal{F}_{stat} . The last row contains the gap between the original affine routing formulation $\mathcal{F}_=$ and $\bar{\mathcal{F}}$ denoted by Best Gap (BG) in the tables and computed as $\frac{OF_{\mathcal{F}_=} - OF_{\bar{\mathcal{F}}}}{OF_{\mathcal{F}_=}}$ where $OF_{\mathcal{F}_=}$ and $OF_{\bar{\mathcal{F}}}$ are the costs of the solutions of $\mathcal{F}_=$ and $\bar{\mathcal{F}}$.

First of all, as expected, we can see that, in almost all the tables, all the variants of affine routing exhibit better solutions compared to the static routing, especially when the number of demands is large (this was also observed in [Poss and Raack (2013)]). Also, the solution given by $\mathcal{F}_=$ is, on one hand, almost always strictly dominated by the

solution obtained by $\overline{\mathcal{F}}$. The solution given by $\overline{\mathcal{F}}$ seems to be the best one with respect to all the other variants of affine routing and static routing in all the considered scenarios (e.g. see Tables 4.2a and 4.2c at the 30 commodities column). We also observe that the solution of \mathcal{F}_+ , and \mathcal{F}_- can give strictly better solutions than $\mathcal{F}_=$ (e.g. see Table 4.2d). Furthermore, observe that the solutions of \mathcal{F}_{agg} , \mathcal{F}_{agg+} , and \mathcal{F}_{agg-} can give slightly strictly better solutions than $\mathcal{F}_=$, \mathcal{F}_+ , and \mathcal{F}_- respectively (see, for example, Table 4.2a with 30 commodities for \mathcal{F}_- and \mathcal{F}_+ , and Table 4.2c with 30 commodities for $\mathcal{F}_=$).

Let us now look more closely at solutions from the different formulations and in particular compare $\mathcal{F}_=$, the original affine formulation, and $\overline{\mathcal{F}}$. For instance, for Abilene with *MLU* and *All Rutable Demands* we obtain a percentage gap up to 9.914 % between $\mathcal{F}_=$ and $\overline{\mathcal{F}}$, and up to 10.538 % between \mathcal{F}_{stat} and $\overline{\mathcal{F}}$. Similarly, for Geant, we have the same trend, with slightly lower percentage gaps (up to 4.458 % and 5.259 %). In practice the objective function can have a different structure and higher gaps may be observed. For instance, a step cost function (often approximated with a piece-wise linear function) can be used to model the investment in additional unitary physical resources.

Table 4.1 Solutions on scenarios with Abilene topology.

(a) *Budget, Linear objective.*

	Nb demands				
	10	15	20	25	30
\mathcal{F}_{stat}	44.0	69.87	90.47	115.0	141.91
$\mathcal{F}_=$	44.0	69.18	89.68	112.0	134.57
\mathcal{F}_{cycle}	44.0	69.18	89.68	112.0	134.57
\mathcal{F}_+	44.0	69.08	89.68	112.0	134.57
\mathcal{F}_-	44.0	68.93	89.68	112.0	134.57
$\overline{\mathcal{F}}$	44.0	67.63	87.97	108.0	129.82
\mathcal{F}_{agg}	44.0	69.18	89.68	112.0	134.57
\mathcal{F}_{agg+}	44.0	69.08	89.6	112.0	134.57
\mathcal{F}_{agg-}	44.0	68.93	89.68	112.0	134.57
BG	0.0 %	2.24 %	1.91 %	3.57 %	3.53 %

(b) *All Rutable Demands, Linear objective.*

	Nb demands		
	5	10	15
\mathcal{F}_{stat}	21.0	30.5	35.5
$\mathcal{F}_=$	21.0	30.5	35.5
\mathcal{F}_{cycle}	21.0	30.5	35.5
\mathcal{F}_+	21.0	30.5	35.5
\mathcal{F}_-	21.0	30.5	35.5
$\overline{\mathcal{F}}$	21.0	29.6	34.43
\mathcal{F}_{agg}	21.0	30.5	35.5
\mathcal{F}_{agg+}	21.0	30.5	35.5
\mathcal{F}_{agg-}	21.0	30.5	35.5
BG	0.0 %	2.951 %	3.014 %

(c) *Budget, MLU objective.*

	Nb demands				
	10	15	20	25	30
\mathcal{F}_{stat}	3.0	4.936	6.236	6.5	7.739
$\mathcal{F}_=$	3.0	4.936	6.236	6.5	7.739
\mathcal{F}_{cycle}	3.0	4.936	6.236	6.5	7.739
\mathcal{F}_+	3.0	4.936	6.236	6.5	7.739
\mathcal{F}_-	3.0	4.936	6.236	6.5	7.739
$\overline{\mathcal{F}}$	3.0	4.936	6.236	6.5	7.739
\mathcal{F}_{agg}	3.0	4.936	6.236	6.5	7.739
\mathcal{F}_{agg+}	3.0	4.936	6.236	6.5	7.739
\mathcal{F}_{agg-}	3.0	4.936	6.236	6.5	7.739
BG	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %

(d) *All Rutable Demands, MLU objective.*

	Nb demands		
	5	10	15
\mathcal{F}_{stat}	1.3	1.424	1.491
$\mathcal{F}_=$	1.291	1.42	1.486
\mathcal{F}_{cycle}	1.291	1.42	1.486
\mathcal{F}_+	1.245	1.39	1.452
\mathcal{F}_-	1.24	1.381	1.448
$\overline{\mathcal{F}}$	1.163	1.312	1.38
\mathcal{F}_{agg}	1.291	1.42	1.486
\mathcal{F}_{agg+}	1.244	1.379	1.446
\mathcal{F}_{agg-}	1.24	1.38	1.447
BG	9.914 %	7.592 %	7.137 %

4.6.4.2 Comparison of model sizes and solving times

We now present two series of tables displaying the solving times and the model sizes. In the first series (Table 4.3) we compare the two polyhedron on the Abilene topology with

Table 4.2 Solutions on scenarios with Geant topology.**(a)** Budget, Linear objective.

	Nb demands				
	10	15	20	25	30
\mathcal{F}_{stat}	48.0	71.87	94.0	118.0	131.48
$\mathcal{F}_=$	47.07	70.25	92.47	114.0	127.75
\mathcal{F}_{cycle}	47.07	70.25	92.47	114.0	127.75
\mathcal{F}_+	47.01	70.04	92.47	114.0	127.5
\mathcal{F}_-	46.96	69.83	91.96	113.71	127.39
$\overline{\mathcal{F}}$	46.94	69.2	90.83	110.53	123.22
\mathcal{F}_{agg}	47.07	70.25	92.47	114.0	127.75
\mathcal{F}_{agg+}	46.94	69.78	91.96	113.5	126.99
\mathcal{F}_{agg-}	46.94	69.46	91.79	113.5	127.04
BG	0.29 %	1.49 %	1.78 %	3.05 %	3.55 %

(b) All Rutable Demands, Linear objective.

	Nb demands		
	5	10	15
\mathcal{F}_{stat}	33.0	52.0	63.0
$\mathcal{F}_=$	33.0	52.0	63.0
\mathcal{F}_{cycle}	33.0	52.0	63.0
\mathcal{F}_+	33.0	52.0	63.0
\mathcal{F}_-	33.0	51.0	63.0
$\overline{\mathcal{F}}$	33.0	50.52	62.5
\mathcal{F}_{agg}	33.0	52.0	63.0
\mathcal{F}_{agg+}	33.0	51.0	63.0
\mathcal{F}_{agg-}	33.0	50.5	62.5
BG	0.0 %	2.846 %	0.79 %

(c) Budget, MLU objective.

	Nb demands				
	10	15	20	25	30
\mathcal{F}_{stat}	1.621	2.311	3.174	3.252	3.836
$\mathcal{F}_=$	1.515	2.218	2.868	3.0	3.427
\mathcal{F}_{cycle}	1.515	2.218	2.868	3.0	3.427
\mathcal{F}_+	1.505	2.218	2.868	3.0	3.422
\mathcal{F}_-	1.515	2.218	2.868	3.0	3.427
$\overline{\mathcal{F}}$	1.505	2.218	2.868	3.0	3.422
\mathcal{F}_{agg}	1.515	2.218	2.868	3.0	3.422
\mathcal{F}_{agg+}	1.505	2.218	2.868	3.0	3.422
\mathcal{F}_{agg-}	1.515	2.218	2.868	3.0	3.422
BG	0.621 %	0.0 %	0.0 %	0.0 %	0.165 %

(d) All Rutable Demands, MLU objective.

	Nb demands	
	5	10
\mathcal{F}_{stat}	1.154	1.312
$\mathcal{F}_=$	1.133	1.301
\mathcal{F}_{cycle}	1.133	1.301
\mathcal{F}_+	1.091	1.282
\mathcal{F}_-	1.111	1.285
$\overline{\mathcal{F}}$	1.091	1.243
\mathcal{F}_{agg}	1.133	1.301
\mathcal{F}_{agg+}	1.091	1.282
\mathcal{F}_{agg-}	1.111	1.285
BG	3.743 %	4.418 %

the *MLU* objective. And in the second series (Table 4.4) we compare both topologies with the *All Rutable Demands* polyhedron and the *Linear* objective. The solving times are in seconds. For the size of models, we display for each formulation the number of columns (i.e. variables), denoted as #col, and the number of rows (i.e., constraints), denoted as #row.

We can observe that, in general, the computation time for the scenarios with *All Rutable Demands* (Table 4.3b) can be several hundred times longer than with *Budget* (Table 4.3a). This can be explained by the fact that the *All Rutable Demands* polyhedron leads to a larger model size in terms of number of variables and constraints. Observe, for example, that the model related to $\overline{\mathcal{F}}$ with *All Rutable Demands* polyhedron and 15 demands on the Abilene topology has five times more variables and six times more constraints than the model with the *Budget* polyhedron (Tables 4.3c and 4.3d).

We further observe that increasing the number of demands greatly increases the required solving time for the affine variants without aggregation (e.g. $\mathcal{F}_=$, \mathcal{F}_- , \mathcal{F}_+ , \mathcal{F}_{cycle} , and $\overline{\mathcal{F}}$). The aggregation technique for the affine routing that we introduced (e.g. \mathcal{F}_{agg} , \mathcal{F}_{agg-} , \mathcal{F}_{agg+}) permits to alleviate this drawback for big enough commodity set size (Tables 4.3a and 4.3c). This is somewhat explained by the fact that the number of x variables varies quadratically with the number of demands in the non-aggregated model while it is linear with the number of demands in the aggregated models.

Let us now focus on a given scenario, for instance, Abilene with *Budget* and the *Linear* objective (Table 4.1a) to compare the solving times. It can be observed that $\overline{\mathcal{F}}$, the routing scheme based on the extended graph formulation, takes the longest time to compute the optimal solution with respect to all the other approaches when varying the demand in the range [5-30]. This is indeed expected since the formulation of $\overline{\mathcal{F}}$ is more complex: for each commodity h , we added to the original graph \mathcal{G} directed edges from the sink of h (i.e., $t(h)$) to each node $v \in V$ of \mathcal{G} , from each node v to the source of h (i.e., $s(h)$), and from $t(h)$ to $s(h)$. The complexity of this approach naturally increases with the number of demands and number of nodes. However, as mentioned above, $\overline{\mathcal{F}}$ shows the best solutions among other formulations.

Table 4.3 Scenarios with *Abilene* topology and *MLU* objective: Impact of the polyhedron on the solving time.

(a) *Budget*, Solving times (s).

	Nb demands				
	10	15	20	25	30
\mathcal{F}_{stat}	< 1	< 1	< 1	< 1	< 1
$\mathcal{F}_=$	< 1	1	5	12	29
\mathcal{F}_{cycle}	< 1	3	5	41	76
\mathcal{F}_+	< 1	2	7	23	89
\mathcal{F}_-	1	2	7	20	37
$\overline{\mathcal{F}}$	< 1	3	12	30	61
\mathcal{F}_{agg}	< 1	< 1	2	4	6
\mathcal{F}_{agg+}	< 1	< 1	2	5	6
\mathcal{F}_{agg-}	< 1	1	2	4	6

(b) *All Routable Demands*, Solving times (s).

	Nb demands		
	5	10	15
\mathcal{F}_{stat}	< 1	< 1	1
$\mathcal{F}_=$	6	447	2929
\mathcal{F}_{cycle}	9	170	2421
\mathcal{F}_+	9	415	5284
\mathcal{F}_-	9	402	3806
$\overline{\mathcal{F}}$	29	1136	11405
\mathcal{F}_{agg}	6	132	590
\mathcal{F}_{agg+}	8	340	2342
\mathcal{F}_{agg-}	6	189	2127

(c) *Budget*, Size of the models.

		Nb demands				
		10	15	20	25	30
\mathcal{F}_{stat}	#col	961	1411	1861	2311	2761
	#row	770	1125	1480	1835	2190
$\mathcal{F}_=$	#col	6931	14881	25831	39781	56731
	#row	4950	10560	18270	28080	39990
\mathcal{F}_{cycle}	#col	5721	12241	21211	32631	46501
	#row	3630	7680	13230	20280	28830
\mathcal{F}_+	#col	9791	21121	36751	56681	80911
	#row	6380	13680	23730	36530	52080
\mathcal{F}_-	#col	9791	21121	36751	56681	80911
	#row	6380	13680	23730	36530	52080
$\overline{\mathcal{F}}$	#col	14851	32161	56071	86581	123691
	#row	7590	16320	28350	43680	62310
\mathcal{F}_{agg}	#col	4291	9121	15751	19501	23251
	#row	3036	6384	10962	13572	16182
\mathcal{F}_{agg+}	#col	8251	17761	30871	38221	45571
	#row	3828	8112	13986	17316	20646
\mathcal{F}_{agg-}	#col	9571	17761	28351	35101	41851
	#row	4411	8112	12873	15938	19003

(d) *All Routable Demands*, Size of the models.

		Nb demands		
		5	10	15
\mathcal{F}_{stat}	#col	5251	8701	12151
	#row	9415	18770	28125
$\mathcal{F}_=$	#col	16201	49501	100801
	#row	28440	103950	226560
\mathcal{F}_{cycle}	#col	15871	48291	98161
	#row	28080	102630	223680
\mathcal{F}_+	#col	22116	69131	141946
	#row	38580	144380	317430
\mathcal{F}_-	#col	22116	69131	141946
	#row	38580	144380	317430
$\overline{\mathcal{F}}$	#col	27841	88381	182521
	#row	47160	178590	394320
\mathcal{F}_{agg}	#col	16201	44971	75481
	#row	28410	94389	169696
\mathcal{F}_{agg+}	#col	22741	64843	109669
	#row	37770	127977	231208
\mathcal{F}_{agg-}	#col	18703	51367	90793
	#row	31152	101611	191712

Table 4.4 shows the impact of the topology on the problem size and the solving time. The problems related to Geant topology clearly have larger size than those related to Abilene and require much more time to be solved. While the number of nodes and edges is twice bigger in Geant compared to Abilene, the number of variables and constraints for models can be between five and seven times larger.

Table 4.4 Scenarios with *Linear* objective and *All Routable Demands* polyhedrons: Impact of the topology on the solving time.

(a) *Abilene*, Solving time (s).

	Nb demands		
	5	10	15
\mathcal{F}_{stat}	< 1	< 1	< 1
$\mathcal{F}_=$	< 1	17	122
\mathcal{F}_{cycle}	< 1	19	213
\mathcal{F}_+	1	33	495
\mathcal{F}_-	1	32	606
$\overline{\mathcal{F}}$	2	457	3292
\mathcal{F}_{agg}	< 1	14	60
\mathcal{F}_{agg+}	1	30	272
\mathcal{F}_{agg-}	< 1	27	145

(b) *Geant*, Solving times (s).

	Nb demands		
	5	10	15
\mathcal{F}_{stat}	0.271	0.536	1.34
$\mathcal{F}_=$	13.57	406.57	3094.01
\mathcal{F}_{cycle}	18.5	340.56	3339.84
\mathcal{F}_+	33.05	1628.02	11155.9
\mathcal{F}_-	49.87	2220.21	12307.3
$\overline{\mathcal{F}}$	85.24	7480.31	53291.0
\mathcal{F}_{agg}	12.07	265.31	787.58
\mathcal{F}_{agg+}	18.0	1030.97	4561.98
\mathcal{F}_{agg-}	58.59	819.89	6972.41

(c) *Abilene*, Size of the models.

		Nb demands		
		5	10	15
\mathcal{F}_{stat}	#col	5281	8731	12181
	#row	9415	18770	28125
$\mathcal{F}_=$	#col	16231	49531	100831
	#row	28440	103950	226560
\mathcal{F}_{cycle}	#col	15901	48321	98191
	#row	28080	102630	223680
\mathcal{F}_+	#col	22146	69161	141976
	#row	38580	144380	317430
\mathcal{F}_-	#col	22146	69161	141976
	#row	38580	144380	317430
$\overline{\mathcal{F}}$	#col	27871	88411	182551
	#row	47160	178590	394320
\mathcal{F}_{agg}	#col	16231	45001	75511
	#row	28410	94389	169696
\mathcal{F}_{agg+}	#col	22771	64873	109699
	#row	37770	127977	231208
\mathcal{F}_{agg-}	#col	18733	51397	90823
	#row	31152	101611	191712

(d) *Geant*, Size of the models.

		Nb demands		
		5	10	15
\mathcal{F}_{stat}	#col	25921	41401	56881
	#row	52809	105474	158139
$\mathcal{F}_=$	#col	78697	231337	463177
	#row	158772	581372	1267872
\mathcal{F}_{cycle}	#col	78067	229027	458137
	#row	158112	578952	1262592
\mathcal{F}_+	#col	99742	298727	602212
	#row	200862	749502	1645992
\mathcal{F}_-	#col	99742	298727	602212
	#row	200862	749502	1645992
$\overline{\mathcal{F}}$	#col	119872	363187	735202
	#row	241122	910322	2007672
\mathcal{F}_{agg}	#col	52345	189145	289081
	#row	105786	475536	792144
\mathcal{F}_{agg+}	#col	65611	246521	378379
	#row	129942	604192	1009152
\mathcal{F}_{agg-}	#col	100807	246521	495193
	#row	199002	604192	1319232

4.7 Conclusion

We have presented variants of the original affine routing formulation to further improve the solutions of the robust network design problem. We proposed a formulation \mathcal{F}_{cycle} based on cycle decomposition that is equivalent to the initial node-arc formulation $\mathcal{F}_=$ of [Poss and Raack (2013)]. We also proved that a formulation based on paths is dominated by $\mathcal{F}_=$. Then two main ideas have been proposed: relaxation of flow conservation constraints and aggregation. The first idea led to \mathcal{F}_- and \mathcal{F}_+ that have been combined into a stronger formulation $\overline{\mathcal{F}}$ by considering an extended graph. All these formulations are less conservative than $\mathcal{F}_=$. The second idea allowed us to build new formulations \mathcal{F}_{agg} , \mathcal{F}_{agg+} and \mathcal{F}_{agg-} that are respectively dominating formulations $\mathcal{F}_=$, \mathcal{F}_+ and \mathcal{F}_- . The striking fact is that aggregation simultaneously reduces the size of formulations as well as the solution's cost. Furthermore, we have proposed a cut-based formulation \mathcal{F}_{cut} that improves over formulation $\overline{\mathcal{F}}$ but is generally NP-hard to solve. Finally, we illustrated our results with a numerical evaluation on two popular network topologies, two objective functions and two polyhedrons.

Despite the efficiency of the new proposed formulations, several challenges remain. To solve larger size problems and reduce solution's cost, it would be nice to find some way to combine aggregation with the extended-graph-based formulation $\overline{\mathcal{F}}$. Combining the uncertainty partitioning techniques (i.e., [Ben-Ameur (2007), Al Najjar et al. (2019), Ben-Ameur and Żotkiewicz (2011)]) recalled in introduction with some of the formulations introduced in the chapter would be another challenge.

CHAPTER 5

A new multi-static routing approach

A *multi-static* approach has been proposed in [Ben-Ameur (2007)] where the uncertainty set is partitioned using some linear inequalities and routing is restricted to be static over each partition.

On the other hand, using the fact that each point of a compact convex set can be written as a convex combination of extreme points, one can see the dynamic approach as a multi-static approach where a routing is imposed for each extreme point of the uncertainty set.

One might ask the following question.

- Is it possible to generalize the multi-static framework in a natural way to obtain a strategy enclosing dynamic, static and multi-static approaches ?

A first attempt to achieve this was the multipolar approach of [Ben-Ameur and Żotkiewicz (2013b), Ben-Ameur et al. (2018b)] (recalled in Chapter 1). We are going to propose a different approach.

We propose a strategy where several subpolytopes contained in the uncertainty set (and more precisely faces) and containing all non-dominated extreme points of this set are defined. A static routing is considered for each subset. We show how these subpolytopes can be found for at least two popular uncertainty sets: the budget polytope and the hose polytope. We describe the approach and some preliminary numerical results in the following sections.

5.1 Decomposition of the uncertainty set

Assume that $\mathcal{D} = \{d \in \mathbb{R}^H : a_i d \leq b_i, \forall i \in I\}$ where I is a finite set. We also assume that \mathcal{D} is bounded. For $J \subset I$, let $\mathcal{D}_J = \{d \in \mathcal{D} : a_i d = b_i, \forall i \in J\}$. If $\mathcal{D}_J \neq \emptyset$, then it is a face of \mathcal{D} . Let us consider subsets of I denoted by J_1, J_2, \dots, J_r such that each non-dominated extreme point of \mathcal{D} belongs to at least one \mathcal{D}_{J_l} for $l \in \{1, \dots, r\}$. Recall that a non-dominated point d of \mathcal{D} is a point such that the only vector $d' \geq d$ belonging to \mathcal{D} is $d' = d$. It is well-known that we only have to ensure that all non-dominated traffic vectors can be routed. Since such a non-dominated vector is a convex combination of

non-dominated extreme points of \mathcal{D} , it is then sufficient to route such extreme points. This can be achieved by considering a static routing for each \mathcal{D}_{J_l} . The problem can then be modeled as follows:

$$\min \sum_{e \in E} \lambda_e u_e \quad (5.1a)$$

$$\begin{aligned} u_e^l &\leq u_e & \forall l \in \{1, \dots, r\}, e \in E \\ \sum_{h \in \mathcal{H}} x_{h,e}^l d_h &\leq u_e^l & \forall e \in E, l \in \{1, \dots, r\}, d \in \mathcal{D}_{J_l} \end{aligned} \quad (5.1b)$$

$$\sum_{e \in \delta_+(v)} x_{h,e}^l - \sum_{e \in \delta_-(v)} x_{h,e}^l = \begin{cases} 1, & \text{if } v = s(h) \\ -1, & \text{if } v = t(h) \\ 0 & \text{otherwise} \end{cases} \quad \forall h \in \mathcal{H}, v \in V, l \in \{1, \dots, r\} \quad (5.1c)$$

$$0 \leq x_{h,e}^l \leq 1 \quad \forall e \in E, h \in \mathcal{H}, l \in \{1, \dots, r\} \quad (5.1d)$$

where $s(h)$ (resp. $t(h)$) denotes the source (resp. sink) of commodity h , and $\delta_+(v)$ (resp. $\delta_-(v)$) is the set of edges outgoing of (resp. ingoing to) v . Variables u_e^l represent the capacity required on link e to route demands inside \mathcal{D}_{J_l} , while $x_{h,e}^l$ denotes the fraction of commodity h that is routed through e when the static routing related to \mathcal{D}_{J_l} is considered. Constraints (5.1a) and (5.1b) ensure that capacities are sufficient to route traffic, while (5.1c) are classical flow-conservation constraints.

Observe that in the classical multi-static approach, the uncertainty set is partitioned and a static routing is considered for each subset. In the new proposal, we only require $\bigcup_{l=1}^r \mathcal{D}_{J_l}$ to contain the set of non-dominated extreme points implying that $\text{conv}(\bigcup_{l=1}^r \mathcal{D}_{J_l})$ contains all non-dominated vectors of \mathcal{D} .

Observe that since each extreme point is defined by a set of equations among the inequalities defining \mathcal{D} , the proposed approach can theoretically reach the dynamic routing approach. However, as the number of extreme points can be exponential, the approach would not be practical. Our challenge is then to find a better decomposition defined by J leading to a good approximate of the optimal dynamic solution in a reasonable computing time.

Generation of the faces \mathcal{D}_{J_l}

We will now describe a method to generate the sets \mathcal{D}_{J_l} . Suppose that there is a subset of inequalities $a_i d \leq b_i$ for $i \in I_0 \subset I$ such that for all non-dominated extreme points d^{ext} of \mathcal{D} at most Γ inequalities can be non-active at the same time (i.e., $|\{i \in I_0 | a_i d^{ext} < b_i\}| \leq \Gamma$). We partition the set I_0 into $\Gamma + 1$ subsets $J_1, \dots, J_{\Gamma+1}$. In the numerical experiment we conducted we arbitrarily constructed the partition composed

of sets of approximately the same size. A non-dominated extreme points d^{ext} of \mathcal{D} is necessarily contained in one of the \mathcal{D}_{J_l} . Otherwise we would have an i_l in each J_l such that $a_{i_l} d^{ext} < b_{i_l}$ and thus $\Gamma + 1$ such indices and this would contradict the hypothesis that $|\{i \in I_0 | a_i d^{ext} < b_i\}| \leq \Gamma$. Consider, for example, the budget polytope defined by $d_h = \underline{d}_h + \xi_h(\overline{d}_h - \underline{d}_h)$, $0 \leq \xi_h \leq 1$, and $\sum_{h \in \mathcal{H}} \xi_h \leq \Gamma$ where \underline{d}_h (resp. \overline{d}_h) is a non-negative lower (resp. upper) bound of d_h and Γ represents an uncertainty budget. In this case, one can use constraints $-d_h \leq -\underline{d}_h$ to generate the faces \mathcal{D}_{J_l} .

The same approach can also be applied for the hose polytope defined by constraints $\sum_{h \in \mathcal{H}:s(h)=v} d_h \leq b^{out}(v)$ and $\sum_{h \in \mathcal{H}:t(h)=v} d_h \leq b^{in}(v)$ where $b^{in}(v)$ and $b^{out}(v)$ are positive constants, in addition to non-negativity constraints. Since at most $2|V|$ constraints related to incoming and outgoing traffic would be active, at most $2|V|$ non-negativity constraints would be non-active. Therefore, non-negativity constraints $d_h \geq 0$ can be used to get the sets \mathcal{D}_{J_l} .

The approach can be iteratively applied for sets \mathcal{D}_{J_l} . Instead of considering a static routing for \mathcal{D}_{J_l} , one can consider a decomposition covering the non-dominated extreme points of \mathcal{D}_{J_l} . This will generally improve the solution quality at the expense of a larger model to solve. In the following, two types of strategies are considered to iteratively build the subsets J_l , a non-selective strategy and some selective ones.

Non-selective Strategy

First, the set I_0 is arbitrarily partitioned into $\Gamma + 1$ subsets. Then, for each polytope \mathcal{D}_{J_l} with $l \in \{1, \dots, \Gamma + 1\}$, the procedure is repeated replacing I_0 by $I_0 \setminus J_l$ and \mathcal{D} by \mathcal{D}_{J_l} . Since each non-dominated extreme point of \mathcal{D}_{J_l} is also a non-dominated extreme point of \mathcal{D} , the property $|\{i \in I_0 | a_i d^{ext} < b_i\}| \leq \Gamma$ is still satisfied for each extreme point d^{ext} which allows application of the decomposition process. By doing so the number of sets of \mathcal{D}_{J_l} increases (almost multiplied by $\Gamma + 1$) and the size of the sets J_l increases as well (since more inequalities are turned to equations). One can continue iterating until the number of \mathcal{D}_{J_l} reaches some fixed value to keep computational complexity under control. Notice that the number of distinct sets \mathcal{D}_{J_l} that can be obtained is bounded by $\binom{|I_0|}{|I_0| - \Gamma - 1} = \binom{|I_0|}{\Gamma + 1}$. Once the decomposition is obtained, we have to solve Problem (5.1) to get u_e values.

Selective Strategies

Instead of systematically decomposing each \mathcal{D}_{J_l} , we select the sets that will be further decomposed based on $\sum_{e \in E} \pi_e^l u_e^l$ where π_e^l is the dual variable related to constraint (5.1a). More specifically, two selective strategies will be considered. In the first

one, all polytopes \mathcal{D}_{J_l} for which $\sum_{e \in E} \pi_e^l u_e^l > 0$ are decomposed, while only the set maximizing $\sum_{e \in E} \pi_e^l u_e^l$ is considered in the second strategy. In fact, when constraint (5.1a) is relaxed in a Lagrangian way using π_e^l multipliers, the objective function becomes $\sum_l \sum_{e \in E} \pi_e^l (u_e^l - u_e) + \sum_{e \in E} \lambda_e u_e = \sum_{e \in E} u_e (\lambda_e - \sum_l \pi_e^l) + \sum_l \sum_{e \in E} \pi_e^l u_e^l$. From Lagrange theory, we have to solve $\max_{\pi_e^l} \min_{u_e, u_e^l, x_{h,e}^l} \sum_{e \in E} u_e (\lambda_e - \sum_l \pi_e^l) + \sum_l \sum_{e \in E} \pi_e^l u_e^l$. To get a finite minimum, we should have $\lambda_e - \sum_l \pi_e^l = 0$ and the objective function to minimize becomes $\sum_l \sum_{e \in E} \pi_e^l u_e^l$. It is then natural to select subsets \mathcal{D}_{J_l} for which $\sum_{e \in E} \pi_e^l u_e^l$ is large and to try to decompose them in order to reduce the objective value.

Numerical solution for problem (5.1)

Problem (5.1) can be seen as a robust network design problem. It can be solved using a standard dualization approach where constraint (5.1b) is integrated through duality (see, e.g., Appendix A).

5.2 Some preliminary numerical experiments

We first evaluated our methods on the Abilene topology using a set of 15 randomly chosen demands. We used the budget polytope with $\underline{d}_h = 1$ and $\overline{d}_h = 2$ for all $h \in \mathcal{H}$ and $\Gamma = 4$. In Figure 5.1 we plotted the objective function (vertical axis) and the solving time in seconds (horizontal axis) for several levels of decomposition (sets \mathcal{D}_{J_l}). After the first decomposition all 3 methods give the same solution with a cost of 68.873 in less than 1 second. This solution already outperforms the static routing solution of cost 69.87 and the affine solution of cost 69.18 (solved in 2 seconds). After additional steps, the 3 multi-static approaches further reduce the solution cost. As observed in Figure 5.1, the non-selective approach seems to dominate both selective approaches for numbers of decomposition steps requiring less than 50 seconds. However after 50 seconds the relationship between the 3 methods is reversed. Selective methods provide better solutions than the non-selective one while using less computing time. We also considered the Geant topology with 15 demands and the same budget polytope as before. Results are also shown on the right part of Figure 5.1. In this case, the cost of the optimal static solution is 71.87 while the best solution cost for affine routing is 70.25. Almost the same conclusions can be drawn from simulations related to Geant topology. While we have shown that our new multi-static approach can improve the solution over static and affine routing, further simulations should be performed to better evaluate the multi-static strategies introduced in this chapter and confirm the observations made above. One might also consider more sophisticated strategies (e.g., affine strategy) for each face to get better solutions.

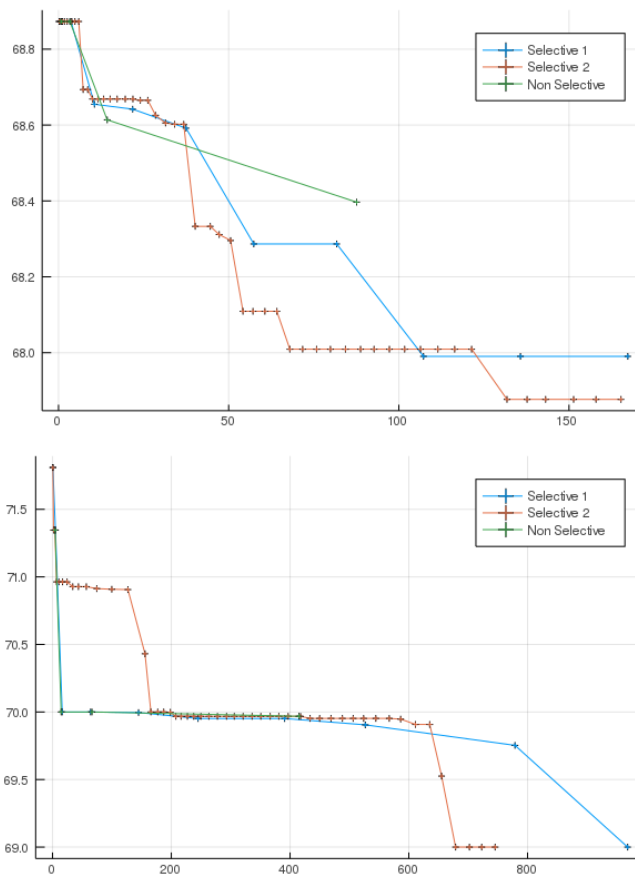


Fig. 5.1 Comparison of the 3 strategies on Abilene (top) and Geant (bottom).

CHAPTER 6

Conclusion and perspectives

6.1 Summary of contributions

Given the dynamic nature of traffic, we investigate the robust network design problem where we have to determine the capacity to reserve on each link so that each demand vector belonging to a polyhedral set can be routed. The objective is either to minimize congestion or a linear cost.

Chapter 2. When routing is assumed to be fractional and dynamic (i.e., dependent on the current traffic vector), we first prove that the robust network design problem with minimum congestion cannot be approximated within any constant factor. Then, using the ETH conjecture, we get a $\Omega(\frac{\log n}{\log \log n})$ lower bound for the approximability of this problem. This implies that the well-known $O(\log n)$ approximation ratio established by Räcke in 2008 is tight up to a lower order factor. Using Lagrange relaxation, we obtain a new proof of the $O(\log n)$ approximation. An important consequence of the Lagrange-based reduction and our inapproximability results is that the robust network design problem with linear reservation cost cannot be approximated within any constant ratio. This answers a long-standing open question of Chekuri (2007). We also give another proof of the result of Goyal et al. (2009) stating that the optimal linear cost under static routing can be $\Omega(\log n)$ more expensive than the cost obtained under dynamic routing. Finally, we show that even if only two given paths are allowed for each commodity, the robust network design problem with minimum congestion or linear cost is hard to approximate within some constant.

Chapter 3. We focus on the variant in which the underlying graph is directed. We prove that an $O(\sqrt{k}) = O(n)$ -approximation can be obtained by solving the problem under static routing, where k is the number of commodities and n is the number of nodes. This improves previous results of Hajiaghayi et al. [SODA'2005] and matches the $\Omega(n)$ lower bound of Ene et al. [STOC'2016] and the $\Omega(\sqrt{k})$ lower bound of Azar et al. [STOC'2003]. Finally, we introduce a slightly more general problem version where some flow restrictions can be added. We show that it cannot be approximated within a

ratio of $k^{\frac{c}{\log \log k}}$ (resp. $n^{\frac{c}{\log \log n}}$) for some constant c . Making use of a weaker complexity assumption, we prove that there is no approximation within a factor of $2^{\log^{1-\epsilon} k}$ (resp. $2^{\log^{1-\epsilon} n}$) for any $\epsilon > 0$.

Chapter 4. While a common heuristic for the robust network design problem is to compute, in polynomial-time, an optimal static routing, affine routing can be used to obtain better solutions also with polynomial-time algorithms. It consists in restricting the routing to depend on the demands in an affine way. We first show that a node-arc formulation is less conservative than an arc-path formulation. We also provide a natural cycle-based formulation that is shown to be equivalent to the node-arc formulation. To further reduce the solution's cost, several new formulations are proposed. They are based on the relaxation of flow conservation constraints. The obtained formulations have been further improved through aggregation. As might be expected, aggregation allows us to reduce the size of formulations. A more striking result is that aggregation reduces the solution's cost. Finally, some numerical experiments are presented.

Chapter 5. We finally propose an intermediate strategy between static and dynamic routing that can be seen as a new variant of multi-static routing. We consider some faces of the uncertainty set whose union contains all non-dominated extreme points of the polytope. A static routing is then assumed for each face. We show how these faces can be found for at least two popular uncertainty sets: the budget polytope and the hose polytope. Some preliminary numerical results are also provided..

6.2 Thesis outlook

A first perspective is to study the approximability of the dynamic routing problem for special cases. This can include the following restrictions of the problem:

- Some special polytopes (Hose, Budget).
- Single source problem (i.e. all commodity have the same source).
- Single path routing. Most of known general results related to static single path routing consider undirected graphs and linear cost objective function. An $O(\log(n))$ approximation is known (due to Gupta) and an $\Omega(\log^{\frac{1}{4}-\epsilon} n)$ lower bound has been proven by Olver and Shepherd (2014). It might be interesting to try to prove some bounds in other cases (congestion and/or directed graphs).

A second line of work is related to the derivation of better bounds using:

- The affine routing method either in its first version proposed by Poss and Raack (2013), Ouorou and Vial (2007) or in one of its strengthened forms of Chapter 4.

- A result of Barvinok (2014) stating that a down-monotone convex body can be approximated with a factor of $\epsilon\sqrt{k}$ (for any $\epsilon > 0$) by a polytope \mathcal{D}' having a polynomial number of non dominated extreme points. In this case, it may be possible to construct such polytope in polynomial-time. A better polytope approximation than the one presented in Chapter 3 might then be deduced. Suppose that we are given a demand polytope $\mathcal{D}, \mathcal{D}' \subseteq \mathbb{R}_+^{\mathcal{H}}$ such that $\mathcal{D}' \subseteq \mathcal{D} \subseteq \alpha\mathcal{D}'$ for some $\alpha \geq 1$. Then computing $lin_{dyn,frac}(\alpha\mathcal{D}')$ provides an α approximation of $lin_{dyn,frac}(\mathcal{D})$. Furthermore if \mathcal{D}' contains a polynomial number of non dominated extreme points then $lin_{dyn,frac}(\alpha\mathcal{D}')$ can be computed in polynomial-time. This can potentially provide a way to improve upon the $\sqrt{8k}$ approximation obtained from static routing (Chapter 3).

Finally, further investigations are needed to study the performance of the new multi-static approach in Chapter 5. This can be done not only for robust network design but also in the context of other multi-stage robust optimization problems.

REFERENCES

- Ahuja, R. K., Magnanti, T. L. and Orlin, J. B. (1993), *Network Flows: Theory, Algorithms, and Applications*, Prentice hall.
- Ahuja, S. S., Gupta, V., Dangui, V., Bali, S., Gopalan, A., Zhong, H., Lapukhov, P., Xia, Y. and Zhang, Y. (2021), Capacity-efficient and uncertainty-resilient backbone network planning with hose, in ‘Proceedings of the 2021 ACM SIGCOMM 2021 Conference’, SIGCOMM ’21, Association for Computing Machinery, New York, NY, USA, p. 547–559.
- Al-Najjar, Y., Ben-Ameur, W. and Leguay, J. (2021), ‘On the approximability of robust network design’, *Theoretical Computer Science* **860**, 41–50.
- Al Najjar, Y., Paris, S., Elias, J., Leguay, J. and Ben-Ameur, W. (2019), Optimal routing configuration clustering through dynamic programming, in ‘ALGOTEL 2019: 21èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications’.
- Applegate, D. and Cohen, E. (2003), Making intra-domain routing robust to changing and uncertain traffic demands: Understanding fundamental tradeoffs, in ‘Proc. ACM SIGCOMM’.
- Arora, S. and Barak, B. (2009), *Computational Complexity: A Modern Approach*, 1st edn, Cambridge University Press, New York, NY, USA.
- Ayoub, J. and Poss, M. (2016), ‘Decomposition for adjustable robust linear optimization subject to uncertainty polytope’, *Computational Management Science* **13**.
- Azar, Y., Cohen, E., Fiat, A., Kaplan, H. and Racke, H. (2003), Optimal oblivious routing in polynomial time, in ‘Proc. ACM STOC’.
- Bartal, Y. and Leonardi, S. (1999), ‘Ondashline routing in all-optical networks’, *Theoretical Computer Science* **221**(1), 19 – 39.
- Barvinok, A. (2014), ‘Thrifty approximations of convex bodies by polytopes’, *International Mathematics Research Notices* **2014**(16), 4341–4356.
- Ben-Ameur, W. (2007), Between fully dynamic routing and robust stable routing, in ‘International Workshop on Design and Reliable Communication Networks’.

- Ben-Ameur, W. and Kerivin, H. (2003), ‘New economical virtual private networks’, *Commun. ACM* **46**(6), 69–73.
- Ben-Ameur, W. and Kerivin, H. (2005), ‘Routing of uncertain traffic demands’, *Optimization and Engineering* **6**(3), 283–313.
- Ben-Ameur, W. and Ouorou, A. (2006), ‘Mathematical models of the delay constrained routing problem’, *Algorithmic Operations Research* **1**(2), 94–103.
- Ben-Ameur, W., Ouorou, A., Wang, G. and Żotkiewicz, M. (2018a), ‘Multipolar robust optimization’, *EURO Journal on Computational Optimization* **6**(4), 395–434.
- Ben-Ameur, W., Ouorou, A., Wang, G. and Żotkiewicz, M. (2018b), ‘Multipolar robust optimization’, *EURO Journal on Computational Optimization* **6**(4), 395–434.
- Ben-Ameur, W., Ouorou, A. and Żotkiewicz, M. (2012), ‘Robust routing in communication networks’, *Progress in combinatorial optimization* pp. 353–390.
- Ben-Ameur, W. and Żotkiewicz, M. (2011), ‘Robust routing and optimal partitioning of a traffic demand polytope’, *International Transactions in Operational Research* **18**(3), 307–333.
- Ben-Ameur, W. and Żotkiewicz, M. (2013a), ‘Multipolar routing: where dynamic and static routing meet’, *Electronic Notes in Discrete Mathematics* **41**, 61–68.
- Ben-Ameur, W. and Żotkiewicz, M. (2013b), ‘Multipolar routing: where dynamic and static routing meet’, *Electronic Notes in Discrete Mathematics* **41**, 61 – 68.
- Ben-Tal, A., El Ghaoui, L. and Nemirovski, A. (2009), *Robust optimization*, Princeton University Press.
- Ben-Tal, A. and Nemirovski, A. (2000), ‘Robust solutions of linear programming problems contaminated with uncertain data’, *Mathematical programming* **88**(3), 411–424.
- Bertsimas, D. and Caramanis, C. (2010), ‘Finite adaptability in multistage linear optimization’, *IEEE Transactions on Automatic Control* **55**(12), 2751–2766.
- Bertsimas, D. and Dunning, I. (2016), ‘Multistage robust mixed-integer optimization with adaptive partitions’, *Operations Research* **64**(4), 980–998.
- Bertsimas, D. and Goyal, V. (2012), ‘On the power and limitations of affine policies in two-stage adaptive optimization’, *Mathematical programming* **134**(2), 491–531.
- Bertsimas, D. and Goyal, V. (2013), ‘On the approximability of adjustable robust convex optimization under uncertainty’, *Mathematical Methods of Operations Research* **77**.

- Bertsimas, D. and Sim, M. (2003), ‘Robust discrete optimization and network flows’, *Mathematical programming* **98**(1), 49–71.
- Bertsimas, D. and Sim, M. (2004), ‘The price of robustness’, *Operations research* **52**(1), 35–53.
- Bhamare, D., Jain, R., Samaka, M. and Erbad, A. (2016), ‘A survey on service function chaining’, *Journal of Network and Computer Applications* **75**, 138–155.
- Bosman, T. and Olver, N. (2017), Exploring the tractability of the capped hose model, in ‘25th Annual European Symposium on Algorithms (ESA 2017)’, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- Charnes, A., Cooper, W. W. and Symonds, G. H. (1958), ‘Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil’, *Management science* **4**(3), 235–263.
- Chekuri, C. (2007), ‘Routing and network design with robustness to changing or uncertain traffic demands’, *SIGACT News* **38**(3), 106–129.
- Chekuri, C., Shepherd, F., Oriolo, G. and Scutellá, M. (2007), ‘Hardness of robust network design’, *Networks* **50**(1), 50–54.
- Dantzig, G. B. (1955), ‘Linear programming under uncertainty’, *Management science* **1**(3-4), 197–206.
- Duffield, N. G., Goyal, P., Greenberg, A., Mishra, P., Ramakrishnan, K. K. and van der Merive, J. E. (1999), A flexible model for resource management in virtual private networks, in ‘Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication’, pp. 95–108.
- Dunning, I. R. (2016), Advances in robust and adaptive optimization: algorithms, software, and insights, PhD thesis, Massachusetts Institute of Technology.
- Eisenbrand, F., Grandoni, F., Oriolo, G. and Skutella, M. (2007), ‘New approaches for virtual private network design’, *SIAM Journal on Computing* **37**(3).
- Ene, A., Miller, G., Pachocki, J. and Sidford, A. (2016), Routing under balance, in ‘Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing’, STOC ’16, Association for Computing Machinery, New York, NY, USA, p. 598–611.
- Fingerhut, J., Suri, S. and Turner, J. S. (1997), ‘Designing least-cost nonblocking broadband networks’, *Journal of Algorithms* **24**(2), 287 – 309.

- Fréchet, A., Shepherd, F. B., Thottan, M. K. and Winzer, P. J. (2014), ‘Shortest path versus multihub routing in networks with uncertain demand’, *IEEE/ACM Transactions on Networking* **23**(6), 1931–1943.
- Gondran, M., Minoux, M. and Vajda, S. (1984), *Graphs and Algorithms*, John Wiley and Sons, Inc., USA.
- Goyal, N., Olver, N. and Shepherd, F. (2009), Dynamic vs. oblivious routing in network design, in ‘European Symposium on Algorithms’, Springer, pp. 277–288.
- Goyal, N., Olver, N. and Shepherd, F. B. (2013), ‘The vpn conjecture is true’, *Journal of the ACM (JACM)* **60**(3), 1–17.
- Grötschel, M., Lovász, L. and Schrijver, A. (1988), *Geometric Algorithms and Combinatorial Optimization*, Vol. 2 of *Algorithms and Combinatorics*, Springer.
- Grötschel, M., Lovász, L. and Schrijver, A. (2012), *Geometric algorithms and combinatorial optimization*, Vol. 2, Springer Science and Business Media.
- Günlük, O. (2002), A new min-cut max-flow ratio for multicommodity flows, in ‘Proc. IPCO Conference’.
- Gupta, A. (2004), ‘Personal communication 2004’.
- Gupta, A., Kleinberg, J., Kumar, A., Rastogi, R. and Yener, B. (2001), Provisioning a virtual private network: A network design problem for multicommodity flow, in ‘Proceedings of the thirty-third annual ACM symposium on Theory of computing’, pp. 389–398.
- Gupta, A. and Könemann, J. (2011), ‘Approximation algorithms for network design: A survey’, *Surveys in Operations Research and Management Science* **16**(1), 3 – 20.
- Hajiaghayi, M., Kleinberg, R. and Leighton, T. (2007), Semi-oblivious routing: Lower bounds, in ‘Proc. ACM-SIAM SODA’.
- Hajiaghayi, M. T., Kleinberg, R. D., Räcke, H. and Leighton, T. (2007), ‘Oblivious routing on node-capacitated and directed graphs’, *ACM Trans. Algorithms* **3**(4).
- Haviv, I. and Regev, O. (2007), Tensor-based hardness of the shortest vector problem to within almost polynomial factors, in ‘Proceedings of the thirty-ninth annual ACM symposium on Theory of computing’, pp. 469–477.
- Hijazi, H., Bonami, P. and Ouorou, A. (2013), ‘Robust delay-constrained routing in telecommunications’, *Annals of Operations Research* **206**(1), 163–181.

- Impagliazzo, R. and Paturi, R. (2001), ‘On the complexity of k-sat’, *Journal of Computer and System Sciences* **62**(2), 367 – 375.
- Impagliazzo, R., Paturi, R. and Zane, F. (2001), ‘Which problems have strongly exponential complexity?’, *Journal of Computer and System Sciences* **63**(4), 512 – 530.
- Infanger, G. (1992), Planning under uncertainty solving large-scale stochastic linear programs, Technical report, Stanford Univ., CA (United States). Systems Optimization Lab.
- Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M. et al. (2013), ‘B4: Experience with a globally-deployed software defined wan’, *ACM SIGCOMM Computer Communication Review* **43**(4), 3–14.
- Koster, A. M., Kutschka, M. and Raack, C. (2013), ‘Robust network design: Formulations, valid inequalities, and computations’, *Networks* **61**(2), 128–149.
URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/net.21497>
- Kumar, P., Yuan, Y., Yu, C., Foster, N., Kleinberg, R., Lapukhov, P., Lim, C. L. and Soulé, R. (2018), Semi-oblivious traffic engineering: The road not taken, in ‘15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)’, USENIX Association, pp. 157–170.
- Lemaréchal, C., Ouorou, A. and Petrou, G. (2010), ‘Robust network design in telecommunications under polytope demand uncertainty’, *European Journal of Operational Research* **206**(3), 634–641.
- Lim, H. and Kim, C. (2000), Multicast tree construction and flooding in wireless ad hoc networks, in ‘Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems’, pp. 61–68.
- Luna, H. P. L. (2006), Network planning problems in telecommunications, in ‘Handbook of Optimization in Telecommunications’, Springer, pp. 213–240.
- Maggs, B. M., Meyer auf der Heide, F., Vöcking, B. and Westermann, M. (1997), Exploiting locality for data management in systems of limited bandwidth, in ‘Proc. FOCS’.
- Mattia, S. (2013), ‘The robust network loading problem with dynamic routing’, *Computational Optimization and Applications* **54**(3), 619–643.
- Michel, O. and Keller, E. (2017), SDN in wide-area networks: A survey, in ‘2017 Fourth International Conference on Software Defined Systems (SDS)’, IEEE, pp. 37–42.

- Olver, N. (2010), Robust network design, PhD thesis, McGill University Library.
- Olver, N. (2018), ‘Designing networks under uncertain demands (draft)’.
- Olver, N. and Shepherd, F. B. (2014), ‘Approximability of robust network design’, *Mathematics of Operations Research* **39**(2), 561–572.
- Orlowski, S., Pióro, M., Tomaszewski, A. and Wessály, R. (2007), SNDlib 1.0–Survivable Network Design Library, in ‘Proc. INOC’.
- Ouorou, A. (2013), ‘Tractable approximations to a robust capacity assignment model in telecommunications under demand uncertainty’, *Computers and Operations Research* **40**(1), 318–327.
- Ouorou, A. and Vial, J. (2007), A model for robust capacity planning for telecommunications networks under demand uncertainty, in ‘Workshop on Design and Reliable Communication Networks’.
- Paolucci, F., Cugini, F., Giorgetti, A., Sambo, N. and Castoldi, P. (2013), ‘A survey on the path computation element (pce) architecture’, *IEEE Communications Surveys and Tutorials* **15**(4), 1819–1841.
- Poss, M. and Raack, C. (2013), ‘Affine recourse for the robust network design problem: Between static and dynamic routing’, *Networks* **61**(2), 180–198.
- Postek, K. and Hertog, D. d. (2016), ‘Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set’, *INFORMS Journal on Computing* **28**(3), 553–574.
- Prékopa, A. (2013), *Stochastic programming*, Vol. 324, Springer Science and Business Media.
- Raack, C., Koster, A. M., Orlowski, S. and Wessály, R. (2011), ‘On cut-based inequalities for capacitated network design polyhedra’, *Networks* **57**(2), 141–156.
- Räcke, H. (2002), Minimizing congestion in general networks, in ‘The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.’, pp. 43–52.
- Räcke, H. (2008), Optimal hierarchical decompositions for congestion minimization in networks, in ‘40th annual ACM symposium on Theory of Computing’, pp. 255–264.
- Rahimian, H. and Mehrotra, S. (2022), ‘Frameworks and Results in Distributionally Robust Optimization’, *Open Journal of Mathematical Optimization* **3**.
URL: <https://ojmo.centre-mersenne.org/articles/10.5802/ojmo.15/>

Scarf, H. (1958), 'A min-max solution of an inventory problem', *Studies in the mathematical theory of inventory and production* .

Silva, M., Poss, M. and Maculan, N. (2018), 'k-adaptive routing for the robust network loading problem', *Electronic Notes in Discrete Mathematics* **64**, 95–104.

Soyster, A. L. (1973), 'Convex programming with set-inclusive constraints and applications to inexact linear programming', *Operations research* **21**(5), 1154–1157.

Wang, N., Ho, K. H., Pavlou, G. and Howarth, M. (2008), 'An overview of routing optimization for internet traffic engineering', *IEEE Communications Surveys and Tutorials* **10**(1), 36–56.

Zeng, B. and Zhao, L. (2013), 'Solving two-stage robust optimization problems using a column-and-constraint generation method', *Operations Research Letters* **41**(5), 457–461.

Appendices

Appendix A

A modular approach for solving robust design problems

We present in this appendix a general solving approach we implemented for several routing strategies: static routing, several variants of affine routing, the new multi-static approach we presented and potentially others. The goal is to handle several demand polytopes such as the hose model, budget, all routable demands and potentially others. A straightforward implementation would be to explicitly write by hand the compact reformulation of the model and implement a cutting plane procedure for each routing method and each demand polytope. However, this solution is not practical. We now describe below an alternative approach, more modular, where an abstract representation, independent of the demand polytope, of a the robust optimization problem is first constructed. And then we implemented two procedures that take as input a representation of a demand polytope and of the robust problem. The first one solves the robust optimization problem using a cutting plane algorithm and the second procedure automatically constructs a compact reformulation by dualization of the problem. This design choice is inspired by how the robust optimization library presented in Dunning (2016), Chapter 6, is structured.

Consider the following quite general single stage robust optimization problem:

$$\min \sum_{i \in I} c_i x_i$$

$$b_i(d, x) \leq 0, \quad \forall i \in I, d \in \mathcal{D}_i \tag{A.1a}$$

$$a'_{i'}(x) \leq 0, \quad \forall i' \in I' \tag{A.1b}$$

$$x_j \geq 0 \quad j \in J \tag{A.1c}$$

Where b_i are bi-affine functions in d and x , i.e., functions given as:

$$b_i(d, x) = a_{0,i}(x) + \sum_{h \in \mathcal{H}} d_h a_{h,i}(x)$$

and $a_{h,i}$ are affine functions in x i.e., functions of the form:

$$a_{h,i}(x) = a_{h,i}^0 + \sum_{j \in J} a_{h,i}^j x_j$$

and $a'_{i'}$ are affine functions in x i.e., functions of the form:

$$a'_{i'}(x) = a'_{i'}^0 + \sum_{j \in J} a'_{h,i}^j x_j$$

Constraints (A.1b) are deterministic constraints i.e. constraints that does not involve the uncertainty on demands d . Those constraints can represent flow conservation constraints in the static routing problem for example. Constraints (A.1a) are uncertain constraints, i.e., constraints that do depend on the uncertainty on demands d . Those constraints can represent capacity constraints in the static routing problem for example or the flow conservation constraints in the affine routing model.

The variables of this model are x_j for j in J for some indices set J . In our case the variables x_j will be the routing variables $x_{h,e}$ in the static routing problem, the coefficient of the affine flow function $x_{h,e}^{h'}$ in the affine routing problem or the routing variables $x_{h,e}^i$ for different subpolytope \mathcal{D}_i in the multi-static approach. Other variables also related to reservations u_e . The polytopes \mathcal{D}_i can be all equal in the case of static routing and affine routing. They can be different for multi-static.

A.1 Abstract representation of a robust optimization problem

We represent the single stage robust optimization problem, independently of the uncertainty sets \mathcal{D}_i as follows:

- A classical linear program composed of only the deterministic constraints (A.1b) i.e., the following linear program:

$$\min \sum_{i \in I} c_i x_i$$

$$a'_{i'}(x) \leq 0, \quad \forall i' \in I' \tag{A.2a}$$

$$x \geq 0 \tag{A.2b}$$

- An array composed of objects representing the bi-affine expressions $b_i(d, x)$. We internally represented a bi-affine expression as a dictionary with entries in the set $\{0\} \cup \mathcal{H}$. The value in the dictionary for the entry h for $h \in \{0\} \cup \mathcal{H}$ is the affine expression $a_{h,i}(x)$ where x is an array composed of the variables in model (A.2). If a key h is not in the dictionary this means that $a_{h,i}(x) = 0$.

A.2 Demand polytope representation

A demand polytope \mathcal{D}_i is represented as follows:

- A linear program:

$$\begin{aligned} \max & 0 \\ & Ad + By \leq 0, \end{aligned} \tag{A.3a}$$

$$y \geq 0, \tag{A.3b}$$

- An array composed of the variables d_h of model (A.3). To be able to know which variable corresponds to each commodity h .

A.2.1 Cutting plane procedure

We will now describe a cutting plane procedure that takes as input an abstract representation of a robust optimization problem and the representation of a demand polytope \mathcal{D}_i for each $i \in I$. It outputs a feasible solution x to Problem (A.1). This can be done by iteratively doing the following steps:

- Solve linear program (A.2) and query the optimal solution x^{opt} to the current model.
- For each $i \in I$ check if one of the constraint $b_i(d, x^{opt}) \leq 0$ is violated by some $d \in \mathcal{D}_i$ and solve the linear problem $\min_{d \in \mathcal{D}_i} \sum_{h \in \mathcal{H}} d_h a_{h,i}(x^{opt})$ and query the optimal solution d^{opt} . If $a_{0,i}(x^{opt}) + \sum_{h \in \mathcal{H}} d_h^{opt} a_{h,i}(x^{opt}) > 0$ add the following constraint to the linear program (A.2).

$$a_{0,i}(x) + \sum_{h \in \mathcal{H}} d_h^{opt} a_{h,i}(x) \leq 0$$

- Stop if none of the constraints $b_i(d, x^{opt}) \leq 0$ are violated.

A.3 Automatic compact reformulation procedure

We will now describe a compact reformulation procedure that take as input an abstract representation of a robust optimization problem and the representation of a demand polytope \mathcal{D}_i for each $i \in I$. It outputs a new linear program having the same optimal solution x than Problem (A.1).

The constraint $b_i(d, x) \leq 0, \forall d \in \mathcal{D}$ is equivalent to

$$\max_{d \in \mathcal{D}} b_i(d, x) = a_{0,i}(x) + \max_{d \in \mathcal{D}} \sum_{h \in \mathcal{H}} d_h a_{h,i}(x) \leq 0$$

We note A_l (resp B_l) the l -th column of the matrix A (resp. B) defining the polytope D .

The problem $\max_{d \in \mathcal{D}} \sum_{h \in \mathcal{H}} d_h a_{h,i}(x)$ can be written as the following linear program (if we consider x fixed):

$$\begin{aligned} \max_{d,y} \quad & \sum_{h \in \mathcal{H}} d_h a_{h,i}(x) \\ A_i d + B_i y & \leq g, \end{aligned} \tag{A.4a}$$

$$d, y \geq 0, \tag{A.4b}$$

By duality this linear program is equivalent to:

$$\begin{aligned} \min_{\lambda_i} \quad & g^t \lambda_i \\ A_i^{h,t} \lambda_i & \leq a_{h,i}(x), \quad \forall h \in \mathcal{H} \end{aligned} \tag{A.5a}$$

$$B_i^{l,t} \lambda_i \leq 0, \quad \forall h \in \mathcal{H} \tag{A.5b}$$

$$\lambda_i \geq 0, \quad \forall l \in \mathbb{N} : 1 \leq l \leq \dim(y) \tag{A.5c}$$

Therefore the constraint $\max_{d \in \mathcal{D}} b_i(d, x) \leq 0$ is equivalent to the following linear programming constraints (linear in λ_i and x):

$$\begin{aligned} a_{0,i}(x) + g^t \lambda_i & \leq 0 \\ A_i^{h,t} \lambda_i & \leq a_{h,i}(x), \quad \forall h \in \mathcal{H} \end{aligned} \tag{A.6a}$$

$$B_i^{l,t} \lambda_i \leq 0, \quad \forall l \in \mathbb{N} : 1 \leq l \leq \dim(y) \tag{A.6b}$$

$$\lambda_i \geq 0 \tag{A.6c}$$

The compact reformulation procedure works as follows:

- For each $i \in I$, add a vector of variable λ_i of dimension the number of constraints in Model (A.3). Add the following Constraints (A.6) to Model (A.2). Recall that $a_{h,i}(x)$ are affine expressions in the x variables of Model (A.2).

The compact reformulation procedure returns the following linear program:

$$\min \sum_{i \in I} c_i x_i \tag{A.7a}$$

$$a'_{i'}(x) \leq 0, \quad \forall i' \in I' \tag{A.7a}$$

$$a_{0,i}(x) + g^t \lambda_i \leq 0 \quad \forall i \in I$$

$$A_i^{h,t} \lambda_i \leq a_{h,i}(x), \quad \forall h \in \mathcal{H}, i \in I \tag{A.7b}$$

$$B_i^{l,t} \lambda_i \leq 0, \quad \forall l \in \mathbb{N} : 1 \leq l \leq \dim(y), i \in I \tag{A.7c}$$

$$\lambda_i \geq 0, \quad \forall i \in I \tag{A.7d}$$

$$x \geq 0 \tag{A.7e}$$

A vector x is an optimal solution to Model (A.7) if, and only if, it is also an optimal solution of Model (A.1).

Titre : Optimisation Robuste pour la Conception de Réseaux

Mots clés : optimisation robuste adaptative, complexité, approximation, routage, conception de réseaux

Résumé : Étant donnée la nature dynamique du trafic, nous étudions la variante du problème de dimensionnement robuste de réseaux qui consiste à déterminer la capacité à réserver sur chaque lien d'un réseau de telle sorte que chaque demande appartenant à un polytope donné puisse être routée. L'objectif est soit de minimiser la congestion soit un coût linéaire. Nous étudions tout d'abord l'approximabilité de la variante avec un routage fractionnaire et dynamique dans des graphes non dirigés. Nous prouvons tout d'abord que, sauf si $P = NP$ le problème de minimisation de la congestion ne peut être approché en dessous d'aucun facteur constant répondant ainsi à une question ouverte de Chekuri (2007). Ensuite, en utilisant la conjecture ETH, nous prouverons une borne inférieure de $\Omega(\log n / \log \log n)$ sur l'approximabilité de ce problème. Nous portons ensuite notre attention sur la variante avec un graphe dirigé. Nous montrons qu'une solution avec un routage statique optimal donne une $O(\sqrt{k}) = O(n)$ -approximation du routage dynamique optimal, où k est le nombre de commodités et n le nombre de noeuds. Nous montrons ensuite qu'une généralisation naturelle du problème ne peut être approché en dessous d'un fac-

teur de $k^{\frac{c}{\log \log k}}$ pour une certaine constante c (resp. $2^{\log^{1-\epsilon} k}$ pour tout $\epsilon > 0$) sauf si $NP \subseteq SUBEXP$ (resp. $NP \subseteq QP$). Nous étudions également plusieurs reformulations du problème de dimensionnement robuste de réseaux permettant d'améliorer la méthode de routage affine. Nous montrons tout d'abord que la formulation par noeud-arc peut être moins restrictive que la formulation par arc-chemin. Nous fournissons également une formulation naturelle équivalente à la formulation par noeud-arc. Nous étudions ensuite plusieurs formulations basées sur des relaxations des contraintes de conservation de flot. Ensuite, nous étudions des formulations basées sur des agrégations de commodité par source ou par destination. Enfin nous proposons une stratégie intermédiaire entre l'approche statique et l'approche dynamique pour s'approcher encore plus du dynamique tout en contrôlant la complexité. Il s'agit d'une approche qu'on pourrait qualifier de multi-statique. L'idée est de choisir un ensemble de faces du polytope représentant l'ensemble d'incertitude de telle manière que l'union des ces faces contienne tous les points extrêmes non-dominés de cet ensemble. Un routage statique est considéré pour chacune de ces faces.

Title : Robust Optimization for Network Design

Keywords : adjustable robust optimization, complexity, approximation, routing, network design

Abstract : Given the dynamic nature of traffic, we investigate the robust network design problem where we have to determine the capacity to reserve on each link so that each demand vector belonging to a polyhedral set can be routed. The objective is either to minimize congestion or a linear cost. When routing is assumed to be fractional and dynamic (i.e., dependent on the current traffic vector), we first prove that the robust network design problem with minimum congestion cannot be approximated within any constant factor, settling an open question by Chekuri (2007). Then, using the ETH conjecture, we get a $\Omega(\frac{\log n}{\log \log n})$ lower bound for the approximability of this problem. We next focus on the variant in which the underlying graph is directed. We prove that an $O(\sqrt{k}) = O(n)$ -approximation can be obtained by solving the problem under static routing, where k is the number of commodities and n is the number of nodes. We show that a natural generalization of the problem cannot be approximated within a ratio of $k^{\frac{c}{\log \log k}}$ for some constant c (resp. $2^{\log^{1-\epsilon} k}$ for any $\epsilon > 0$) un-

less $NP \subseteq SUBEXP$ (resp. $NP \subseteq QP$). Affine routing can be used to obtain better solutions also with polynomial-time algorithms. It consists in restricting the routing to depend on the demands in an affine way. We first show that a node-arc formulation is less conservative than an arc-path formulation. We also provide a natural cycle-based formulation that is shown to be equivalent to the node-arc formulation. To further reduce the solution's cost, several new formulations are proposed. They are based on the relaxation of flow conservation constraints. The obtained formulations have been further improved through aggregation. As might be expected, aggregation allows us to reduce the size of formulations. A more striking result is that aggregation reduces the solution's cost. We finally propose an intermediate strategy between static and dynamic routing that can be seen as a new variant of multi-static routing. We consider some faces of the uncertainty set whose union contains all non-dominated extreme points of the polytope. Then a static routing is considered for each of these faces.