

LUIZA BERNARDES REAL

**DOIS PROBLEMAS DE LOCALIZAÇÃO DE
CONCENTRADORES PARA SISTEMAS DE
TRANSPORTE**

Belo Horizonte
28 de fevereiro de 2020

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**DOIS PROBLEMAS DE LOCALIZAÇÃO DE
CONCENTRADORES PARA SISTEMAS DE
TRANSPORTE**

Tese apresentada ao Curso de Pós-Graduação em Engenharia de Produção da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Engenharia de Produção.

Área de concentração: Pesquisa Operacional e Intervenção em Sistemas Sociotécnicos.

Linha de pesquisa: Otimização de Sistemas Logísticos e de Grande Porte.

Orientador: Ricardo Saraiva de Camargo.

LUIZA BERNARDES REAL

Belo Horizonte
28 de fevereiro de 2020

UNIVERSIDADE FEDERAL DE MINAS GERAIS
SCHOOL OF ENGINEERING
POSTGRADUATE PROGRAM IN PRODUCTION ENGINEERING

TWO HUB LOCATION PROBLEMS FOR TRANSPORTATION SYSTEMS

Thesis presented to the Graduate Program in Production Engineering of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Production Engineering.

Concentration area: Operational Research and Intervention in Sociotechnical Systems.

Line of research: Optimization of Logistics and Large-Scale Systems.

Advisor: Ricardo Saraiva de Camargo.

LUIZA BERNARDES REAL

Belo Horizonte
February 28, 2020

R288d Real, Luiza Bernardes.
Dois problemas de localização de concentradores para sistemas de transporte [recurso eletrônico] / Luiza Bernardes Real. - 2020.
1 recurso online (viii, 78 f. : il., color.) : pdf.

Orientador: Ricardo Saraiva de Camargo.

Tese (doutorado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Bibliografia: f. 73-78.

Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia de produção - Teses. 2. Algoritmos - Teses. 3. Economia de escala - Teses. 4. Transportes - Teses. I. Camargo, Ricardo Saraiva de. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 658.5(043)



FOLHA DE APROVAÇÃO

Two Hub Location Problems for Transportation Systems

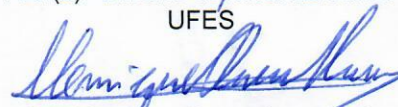
LUIZA BERNARDES REAL


Tese submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA DE PRODUÇÃO, como requisito para obtenção do grau de Doutor em ENGENHARIA DE PRODUÇÃO, área de concentração PESQUISA OPERACIONAL E INTERVENÇÃO EM SISTEMAS SOCIOTÉCNICOS, linha de pesquisa Otimização e Simulação de Sistemas Logíst. e de Grande Porte.


Aprovada em 28 de fevereiro de 2020, pela banca constituída pelos membros:


Prof(a). Ricardo Saraiva de Camargo - Orientador
UFMG


Prof(a). Gilberto de Miranda Junior
UFES


Prof(a). Henrique Pacca Loureiro Luna
UFAL


Prof(a). Rodney Rezer de Saldanha
UFMG


Prof(a). Nelson Maculan Filho
Universidade Federal do Rio de Janeiro

Belo Horizonte, 28 de fevereiro de 2020.

Resumo

Redes do tipo eixo-raio são normalmente utilizadas em sistemas de transporte com o intuito de rotear eficientemente commodities e passageiros entre vários pares de origem e destino. Duas variações do problema de localização de concentradores são propostas. Programas inteiros mistos são apresentados e resolvidos através de métodos exatos e heurísticos. O primeiro problema estudado propõe o desenho da malha aérea global, com base na localização de gateways em redes do tipo eixo-raio, diferenciando passageiros domésticos e internacionais. Uma formulação de programação inteira mista é desenvolvida e dois algoritmos baseados no método de decomposição de Benders são implementados para resolver o problema. Enquanto a versão monolítica não consegue resolver instâncias médias dentro de um tempo máximo, os algoritmos propostos são capazes de resolver instâncias maiores em um tempo razoável. A segunda variação do problema de localização de concentradores estudado otimiza o desenho de uma rede de transporte genérica considerando rotas flexíveis. Nessa versão, além de localizar nós concentradores e alocar nós não-concentradores a nós concentradores, as rotas dos veículos são definidas. Enquanto a maioria dos estudos até então presentes na literatura consideram um fator de desconto fixo para representar economias de escala em links entre hubs e uma topologia específica para as redes, esse trabalho introduz uma formulação inteira mista, em que economias de escalas dependem da tecnologia de transporte escolhida para operar as rotas e a topologia da rede é determinada endogenamente. Duas metaheurísticas são implementadas para achar boas soluções para o problema em tempos computacionais razoáveis.

Keywords: Problema de localização de concentradores, gateway, economia de escala, rotas flexíveis.

Abstract

Hub-and-spoke networks are frequently employed in transportation systems to efficiently route commodities and passengers between many origins and destinations. We propose two variants for the Hub Location Problem and introduce mixed-integer programs solved by exact and heuristic techniques. The first studied problem focuses on locating gateway facilities on hub networks to design global air transport systems, by differentiating international from domestic passengers. A mixed-integer programming formulation is developed and two algorithms based on Benders decomposition method are devised to solve the problem. While the monolithic version fails to solve medium instances, the proposed algorithms can solve large instances in a reasonable time. The second variant of the Hub Location Problem introduced here aims to design a generic transport network with flexible routes. In this version, besides locating hub facilities and allocating non-hub nodes to hubs, vehicle routes are defined. Whereas most previous studies consider a fixed discount factor to represent economies of scale in inter-hub links and a specific network topology is imposed, we introduce a mixed integer formulation, in which scale economy depends on the transport technology chosen to operate the route and the network topology is endogenously determined. Two metaheuristics are implemented to find good solutions to the problem in reasonable computational time.

Keywords: Hub location problem, gateway, scale economy, flexible routes.

Acknowledgments

I would like to express my gratitude to my advisor, Professor Dr. Ricardo Saraiva de Camargo, for all his support during my years as a graduate student at UFMG. I truly value his patience guidance and full time enrollment during the entire process. He has been a constant source of inspiration, a role model, a mentor. I feel lucky to have an advisor so big in love, in person and in profession. He is the one professor who definitely made a difference in my life.

I would also like to thank Professor Dr. Gilberto de Miranda for his technical knowledge and helpful suggestions.

I am also very thankful to Professor Dr. Morton O’Kelly for his advice and insightful comments.

Special thanks to Professor Dr. Ivan Contreras and Professor Dr. Jean-François Cordeau from Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) for their supervision through my inter-university exchange doctorate in Montreal - Canada and for their deep knowledge and experience that shared with me. I feel deeply honored for this collaboration.

I had the chance to meet great friends in the Industrial Engineering department and at CIRRELT. Thank you for each one that in his way made this process softer. A very special thanks goes out to Afonso Henrique Sampaio and Bruno Salezze Vieira for helping me out with technical doubts and for having spent endless hours looking after the innumerable details in the production of this work.

This would not be complete without the mention of the most important people of my life. I thank my parents for having taught me the importance of studies and hard work and for always being a force of strength. I thank my siblings for their friendship and encouragement. I thank Guilherme de Souza Ferreira for his love, patience, emotional and technical support and for constantly believing in me.

Most importantly, I thank God for giving me the opportunity and the ability to make this far.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Main Contributions of this Thesis	4
1.3	Thesis Outline	5
2	The Gateway Hub Location Problem	6
2.1	Introduction	6
2.2	Notation, definitions and formulation	11
2.3	Benders decomposition algorithms	15
2.3.1	Benders subproblem and master problem	15
2.3.2	A basic Benders decomposition algorithm outline	19
2.3.3	Adding Pareto Optimal Cuts	19
2.3.4	Adding multiple Benders cuts	21
2.3.5	Solving Benders subproblem	22
2.3.6	Repairing infeasible primal solutions	23
2.4	Computational experiments	24
2.5	Insights from this study	33
2.5.1	Measuring the Influence of Economies of Scale and Transporta- tion Costs on the Optimal Topologies	34
2.5.2	An illustrative example	37
2.6	Conclusion and future research	40
3	Hub Network Design with Flexible Routes	41
3.1	Introduction	41
3.2	Problem definition and formulation	46
3.3	Solution algorithms	49
3.3.1	Initial solution	52
3.3.2	Destroy operators	53
3.3.3	Repair operators	54
3.3.4	Verifying the connectivity of the solution	55

3.3.5	Hub configuration and capacity-feasibility check	55
3.3.6	Selection of operators	59
3.3.7	Penalizing infeasible solutions	59
3.3.8	Acceptance criterion	59
3.4	Computational experiments	60
3.4.1	Test instances	61
3.4.2	Solving small instances	62
3.4.3	Solving large instances	65
3.5	Conclusion	67
4	Final Remarks	71
	Bibliography	73

List of Figures

1.1	Types of network	2
2.1	Example with the original setting and with a three layer configuration. . .	12
2.2	Optimal air network for instance global-141 setting I	37
3.1	Example of a hub network with flexible routes	45
3.2	TTT plot for 25AP-030	66
3.3	TTT plot for 25AP-080	66
3.4	Performance profile of the devised algorithms.	67

List of Tables

2.1	Description of the global instances created	25
2.2	Five settings	25
2.3	Results for smaller instances - Setting I	26
2.4	Results for smaller instances - Setting II	26
2.5	Results for smaller instances - Setting III	27
2.6	Results for smaller instances - Setting IV	27
2.7	Results for smaller instances - Setting V	27
2.8	Results for bigger instances - Setting I	31
2.9	Results for bigger instances - Setting II	31
2.10	Results for bigger instances - Setting III	31
2.11	Results for bigger instances - Setting IV	32
2.12	Results for bigger instances - Setting V	32
2.13	Optimal network under different economies of scale values	35
2.14	Optimal network under different unitary operational costs	36
2.15	Total passenger traffic 2015	38
2.16	Total international passenger traffic 2015	39
3.1	Related literature to the HNBP	45
3.2	Capacity of the vehicles and potential economies of scale for the random instances	62
3.3	Capacity of the vehicles and potential economies of scale for the AP instances ($\tau^{II} = 3,000$)	62
3.4	Results for small instances: 6R, 7R, 8R, 9R, 10R and 10AP.	63
3.5	Results for medium instances: 20R, 15AP and 20AP	64
3.6	Results for large instances: 30R, 40R, 50R, 25AP, 30AP, 35AP, 40AP, 45AP and 50AP.	69
3.7	Results of the normality analysis.	70
3.8	Results of the Wilcoxon tests.	70

List of Algorithms

1	Basic Benders decomposition	20
2	Papadakos based Benders decomposition	29
3	Repair Benders decomposition	30
4	Basic steps of TDALNS	51
5	Basic steps of BUALNS	68

Chapter 1

Introduction

1.1 Motivation

Transport plays an important role in economic and social development, bringing opportunities and enabling economies to be more competitive. Transport infrastructure connects people to jobs, education, and health services; permits the supply of goods and services all over the world; allows people to interact and ideas to be spread. In doing so it has a catalytic effect on the global economy, whether it is tourism, whether it is goods and services, or whether it is trade.

The transport industry directly employs around 10 million people and accounts for about 5% of gross domestic product (GDP). Logistics, such as transport and storage, account for 10–15% of the cost of a finished product for European companies. The quality of transport services has a major impact on people's quality of life. On average 13.2% of every household's budget is spent on transport goods and services. In this sense, the development of methodologies and tools to increase the efficiency of the whole transport system becomes more and more crucial. (Directorate-General for Mobility and Transport (European Commission) 2019).

To contribute to the cost-efficient design of transportation networks, two different problems are studied in this thesis. The first problem is focused on the air transportation system, aiming to design an integrated global air network, offering opportunities for people to fly and also facilitating economic development all over the world. Air transport provides a significant improvement to economic development, by bringing people together, transporting vital items faster, facilitating the exchange of experiences and ideas, supporting trade and enabling business to access global markets. Although more than 99% of global trade, measured by weight, is transported by surface, more than one-third of global trade, measured by value, is transported by air. Moreover, air activities raise jobs in the air transport sector and in its supply chain. Aviation

supports 65 million jobs and \$2.7 trillion in global GDP. (International Air Transport Association 2019).

While the first problem is focused on a specific context, the second problem studied here proposes the design of a more generic network, being applied in a wide range of transport systems. The objective of the second problem is to design a network in a way that flows and resources are routed at a minimal cost, without imposing restrictions of any type to route the vehicles and to serve the demands. A concrete application of this problem arises in liner shipping. According to the United Nations Conference on Trade and Development (UNCTAD), total maritime trade has grown more than fourfold since 1970. In 2017, global maritime trade grew by 4.0 percent over the previous year, which is higher than the 2.6 percent increase recorded in 2016. UNCTAD estimated that 11.8 billion tons of cargo were transported over water in 2017. (United Nations Conference on Trade and Development 2018).

Hub-and-spoke networks are frequently employed in transportation systems to efficiently route commodities between many origins and destinations. One of the key features of these networks is that direct connections between origin-destination (O-D) pairs can be replaced by fewer, indirect but privileged connections by using transshipment, consolidation, or sorting points, called hub facilities. In this type of network, the flow of different origins is aggregated in the hubs. Then, a hub network directs the flow to its destination. Overall transportation costs may decrease due to the bundling or consolidating of flows through inter-hub arcs. Figures 1.1a and 1.1b show an example of a point-to-point network and a hub-and-spoke network, respectively.

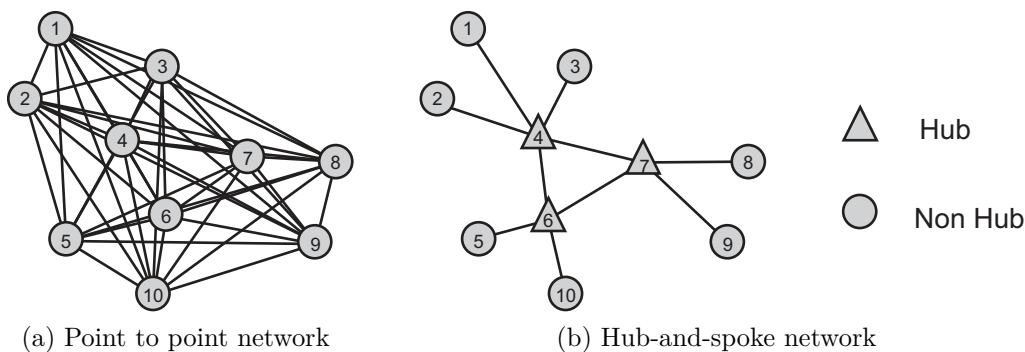


Figure 1.1: Types of network

Hub location problems (HLPs) deal with the design of hub-and-spoke networks and typically arise when commodities must be transported between every pair of origin-destination, but it could be expensive to make this transport from every single point to another point directly. In general, these problems consist of choosing the sites of hub facilities and allocating demand nodes to hubs to effectively route the traffic be-

tween origin-destination pairs. O’Kelly (1987) was the first to propose a mathematical programming formulation and two heuristics for this problem.

Some application areas of the hub location problem are logistics systems such as trucking, liner shipping and, airline industries. Several classes of HLPs have been studied. The various applications within each class give rise to variants that differ in terms of assumptions, such as the required topological structure, the allocation pattern of nodes to hubs, and the existence of capacity constraints on the hub nodes or arcs. Various mathematical models (Ernst and Krishnamoorthy 1998, Labbé and Yaman 2004, Hamacher et al. 2004, Contreras and Fernández 2014, de Camargo et al. 2017) and specialized solution algorithms (Ernst and Krishnamoorthy 1998, Labbé et al. 2005, Çetiner et al. 2010, Martins de Sá et al. 2015, de Sá et al. 2018, de Carvalho et al. 2017) were developed to solve real-size instances.

This thesis discusses two different HLPs: (i) A Gateway Hub Location Problem (GHLP), and; (ii) Hub Network Design Problem with Flexible Routes (HNDPs). Both of them share the decisions of locating transshipment points and routing the flow through the designed network at a minimal cost.

The first problem introduced here focuses on designing a global air transport system, differentiating international passengers from domestic passengers and international hubs from domestic hubs. In a typical hub-and-spoke network, we have two connection levels: hub level (the connection between hubs) and spoke level (the connection between non hubs and hubs). The local and global flow are not differentiated. The complexity involved in routing global flow is ignored (for example, the existence of an agreement between the countries is necessary). To design a global hub-and-spoke network, articulating global system with the domestic system, three connection levels are needed: international hub level (the connection between international hubs, referred as gateways), domestic hub level (the connection between domestic hubs) and spoke level (the connection between non hubs and domestic hubs). A mixed-integer programming formulation for the GHLP is presented. Variants based on Benders decomposition method are devised to solve the problem.

While GHLP focus on strategic decisions locating international and domestic hubs, in the HNDPs, strategic decisions are taken simultaneously with tactical decisions. In this second variant, besides establishing hub locations and routing demand flow through the designed network, vehicle routes are defined. In a typical hub location-routing problem, routing decisions happen only at the access level, and the hub network is assumed to be fully interconnected with one vehicle serving each pair of hubs. Normally they consider a given exogenous network topology and a homogeneous fleet that can only pass a limited number of times through each demand node, while the path of a demand flow is restricted by the prescribed network topology. Moreover, these typical

problems completely disregard that the economies of scale derive from the technology chosen to operate the routes and its utilization. To overcome this simplification, the HNDPs proposes the design of a more general network with flexible routes. The routes are flexible in the sense that they may or may not have hubs on them. No restriction of any kind is imposed on the path of a demand flow. That is, demand flow can be routed from its origin to its destination by using a single vehicle route or by using a set of vehicle routes. However, commodity transfers can only be performed at hubs. Most important of all, instead of considering a fixed discount factor to represent economies of scale in inter-hub links, the HNDPs stand on the idea that economies of scale are obtained based on the costs of the vehicles that operate on each arc of the network and also on the amount of flow carried on the vehicles. These assumptions render the HNDP more difficult to formulate and solve than classical hub network design problems. A mixed-integer programming formulation for the HNDPs is presented. Two metaheuristics are developed to solve large instances in reasonable computation time.

For the purposes of this thesis, we aim to expand bibliographic content that deals with variants of HLPs, presenting problems that consider reasonable assumptions and provide good approximation with reality. In the two problems studied here, the idea is to propose efficient algorithms to solve benchmark instances, in which commercial solvers have difficulties solving optimality, in a desired computational time.

1.2 Main Contributions of this Thesis

The main contributions of this thesis are presented below, divided by chapters:

Chapter 2: A Gateway Hub Location Problem

- To propose a more explicit formulation that incorporates local and global flows to design air transportation networks with a hub and spoke structure.
- To devise Benders variants algorithms capable of solving large-scale instances in reasonable time.

Chapter 3: Aircraft Routing and Scheduling Hub Location Problem

- To propose a more explicit formulation that designs a generic transportation network with flexible routes.
- To model economies of scale according to the transport technology chosen to operate the routes and their actual utilization.
- To consider strategic and tactical decisions at the same time.

- To develop two metaheuristics capable of finding good solutions for large instances in reasonable computational time.

1.3 Thesis Outline

Chapters 2 and 3 of this thesis are structured in the format of two scientific articles. Chapter 2 describes the GHLP, presents a mixed integer programming formulation for the problem and some algorithms based on Benders decomposition method to solve the problem. This article is published to Journal of Air Transport Management, jcr 2.357. Chapter 3 introduces the HNDPs, presents a mixed integer programming formulation for the problem and two metaheuristics to find good solutions to the problem.

Chapter 2

The Gateway Hub Location Problem

Abstract

We introduce the Gateway Hub Location Problem (GHLP) to design global air transportation systems. Relying on a three-level hub network structure and on having nodes located in different geographic regions, the GHLP consists of locating international gateways and domestic hubs, activating arcs to induce a connected gateway and hub network, and routing flows within the network at minimum cost. Most previous studies focus on a typical hub-and-spoke network, in which local and global flows are not differentiated. Here to better represent a world wide air transportation system, global flows can only leave or enter a given geographic region by means of a gateway, while local flows can only use hubs within their respective region. As routing local or global flows involved different agents, this study presents a mixed integer programming formulation that exploits these differences to model both the local and global flows. Due to the formulation's characteristics, two algorithm variants based on Benders decomposition method are devised to solve the problem. A new repair procedure produces optimality Benders cuts whenever feasibility Benders cuts would rather be expected. While the monolithic version failed to solve medium size instances, our algorithms solved larger ones in reasonable time.

Key words: Air transport; Hub-and-spoke networks; Gateway; Benders decomposition method.

2.1 Introduction

By the year 2034, global air traffic is expected to double reaching over seven billion passengers annually transported (IATA 2015), being Africa, Middle East, Asian and

Latin America the geographic areas with the largest percentage growths till then. This rapid demand growth is pressuring airlines and air transport management agencies to extend and expand the existent networks to accommodate new markets, new players, new infra-structures, and new flight connections to serve both increasing domestic (local) and international (global) passenger flows.

Modeling and understanding these local and global passenger flows are generally done separately in the literature (Preis et al. 2013, Mao et al. 2015), or are usually considered to be non differentiable when designing networks for many-to-many air transportation systems with a hub-and-spoke structure (Campbell et al. 2002, Alumur and Kara 2008, Campbell and O’Kelly 2012, Farahani et al. 2013). However, there are some differences between domestic and international passengers that might justify differentiating them.

From the perspective of service quality, reliability was ranked by international passengers as the most important dimension, whereas domestic passengers value more assurance dimension (Arslan et al. 2011). According to the Resource Manual for Airport In-Terminal Concessions (2011), international passengers, on average, arrive at the airport earlier and spend more time in terminals. Thereby, their needs for food, reading materials, travel accessories and other amenities are larger than domestic passengers’ needs. They also tend to be more sophisticated with higher average incomes. This represents a higher potential revenue for international airports, which usually have a better infrastructure for shopping, eating and even resting. Then contrasting to domestic airports. In this way, even though they might share some resources, and affect each other’s routing design decisions, when designing air passenger networks, local and global flows are required to be routed through different facility types over the network.

Local flows are routed via domestic hubs (hubs), while global flows go through international gateways (gateways) to leave from or to enter into a different geographic region. Hubs allow passengers to do connections and airplanes along their routes, whereas gateways are critical for connecting wide regions, such as continents, and for performing customs, immigration and security checks. Since a global passenger flow may be routed via some hubs before going through some gateways to reach its destination, or vice versa, a global flow may share then some inter-hub connections with other local flows, showing thus how both flow types are intertwined.

To articulate both local and global flows, three connection levels are needed: international gateway level with inter-gateway connections, domestic hub level with inter-hub connections, and spoke level with regional airports linked to hubs or gateways. Inter-hub connections are usually done by large carriers, while inter-gateway connections are performed by even larger, long-range airplanes. Further regional airports are

usually linked to hubs or gateways by middle to small size planes. This three level setting can be seen as a three-tier hierarchical hub-and-spoke network structure.

Hub-and-spoke systems are commonly used in many-to-many transport applications to lower transportation costs by exploiting scale economies whenever large carriers can be used to carry consolidated flows over the network (O’Kelly 1987, Jr. 2012). A typical hub-and-spoke network uses two connection levels instead of three: hub level with inter-hub connections, and spoke level with flow exchanging nodes (regional airports) linked to hubs. Scale economies are usually achieved on the hub level by bulk transportation on inter-hub connections. A myriad of applications and topologies have been modeled as hub-and-spoke networks as can be seen in Campbell et al. (2002), Alumur and Kara (2008), Campbell and O’Kelly (2012), Farahani et al. (2013).

In the past 20 years the global airline industry has undergone major changes. The notion of international airlines collaborating for creating cost and revenue synergies through the formation of strategic alliances (such as Star Alliance, Oneworld and Skyteam) has been gained credibility (Schosser and Wittmer 2015). Thereby, design air network from a global perspective becomes necessary. However, only in the last decade, the idea of differentiating local from global flows has attracted some attention from the research community (Adler and Smilowitz 2007, Sasaki et al. 2009, Yaman 2009, Catanzaro et al. 2011).

Adler and Smilowitz (2007) analyze global alliances and mergers in an airline industry under competition. They present a game-theoretic competitive merger framework that allows airlines to choose partners with their installed gateways, inter-gateway connections, and regional networks so that mergers can be proposed and profits maximized. Selection is based on cost and revenue analyses by considering information of a given airlines and its competitors. Local and global flows are differentiated, but treated separately on a two stage approach. First hubs are installed to route local flows, then, assuming that global flows are temporarily aggregated at each installed hub, rather than in their original locations, one gateway per region is selected within these installed hubs. As the trace of each demand flow exchange can only be performed after the network is designed, transportation costs are poorly underestimated, questioning thus the quality of the achieved network configurations.

Disregarding the many-to-many nature of the local and global flows, Sasaki et al. (2009) develop a gateway and hub location model based on a two level p -median facility location problem. From a candidate set, a fixed number of gateways and hubs are selected so that each regional airport is served by a hub, and each installed hub is linked to a gateway at minimum allocation cost. By not considering flow demands happening between pairs of origin-destination nodes, the problem’s complexity is greatly reduced at the expense of having ill-formed air networks.

Yaman (2009) does not distinguish between local and global flows, he considers the design of a hierarchical hub and spoke network which consists of locating a fixed number of gateways and hubs, such that regional airports and hubs are single allocated to hubs and gateways, respectively, to form a star sub-network for each gateway. The simpler strict formulation imposes gateways to be fully interconnected, and prevent hubs to directly interact with each other. Given the single allocation policy, undesirable long distances are perceived by the demand flows in the attained solutions. Further, a fully interconnected gateway hub is not always possible to be assumed in an air network design, since airlines tend to avoid flying for long ranges over water without communication, or over conflict zones.

Finally, Catanzaro et al. (2011) investigate a particular variant of a hub location problem which partitions a given network into sub-networks, and locates at most a fixed number of gateways, but with at least one gateway in each sub-network. Sub-networks are supposed to have at least (at most) a minimum (maximum) number of nodes to exist. The problem's objective is to split the network into regions and then route flows at minimum transportation cost. A flow can only enter or leave a sub-network through a installed gateway, and once it leaves a sub-network, it can only be routed through gateways until it reaches its destination sub-network, when then it can use the available hubs and local links. Hubs and all network connections are assumed to be given beforehand, i.e. costs incurred from installing hubs, gateways, and inter-hub and inter-gateway connections are not considered.

Until now, the literature has acknowledged the importance of differentiating local from global flows, but, as aforementioned, has made assumption compromises that resulted into over-simplified problems or models. Here a more explicit formulation that incorporates local and global flows is proposed for the air transportation network design. Hubs, gateways, and inter-hub and inter-gateway connections are decided so that the induced network can route local and global flows at minimal transportation and installation costs. Different scale economies are granted for installed inter-hub and inter-gateway connections to mimic lower transportation costs due to consolidated flows. Regional airports can be linked to any installed hub or gateway within its region and within aircraft range, i.e. local airports can be multiple allocated to hubs and gateways. This provides greater flexibility to route flows at the expense of demanding a more elaborated model. Further fixed costs for establishing hubs, gateways, and inter-hub and inter-gateway connections are assumed to be known, and continental and country divisions are adopted as natural regions.

Because our aim is to consider the design of air network from a global perspective, we made some simplifications for now. The current study ignored, for example: airline competition, passenger's behavior and choice of routes, congestion transshipment air-

ports, the effects of frequency on service quality and schedule delay. These issues have been well studied in the literature (Hansen 1990, Hong and Harker 1992, Hsu and Wen 2003, Adler 2005).

The addressed air transportation network design is modeled as a multi-commodity flow based hub and spoke system, given rise to a gateway hub location problem or a three-level hub location problem. Given its large scale multi-commodity nature and its induced decomposable matrix structure, the devised formulation is solved by two specialized Benders decomposition algorithms (Benders 1962) which incorporate two features that greatly speed up the method: a repair procedure which allows to generate Benders optimality cuts from unbounded dual subproblems, and a tailored dual subproblem solution algorithm which calculates the optimal dual values to produce Benders optimality cuts without relying on a Simplex solver. In order to evaluate and assess the efficiency and limitations of the devised Benders algorithms, computational experiments were performed and compared with a general purpose solver (IBM CPLEX) on solving the proposed formulation. Both algorithms clearly out-performed the general purpose solver when solving large instance sizes.

To be clear from the outset, the focus here is not on reproducing the current air network, rather, we wish to use network design tools that contribute to improve air transport systems. The proposed model of how things should be can be used to contrast to actual systems. We believe that this analysis is needed and should be of concern. There are many broad participants interested in the efficiency of the world's aviation system as World Bank, FAA (Federal Aviation Administration), Eurocontrol, mainframe manufacturers (Boeing, Airbus, Embraer) and probably many others. The rational planning of the air network has implications consistent with the strategic objectives of ICAO (International Civil Aviation Organization) on supporting the growth of air transport. Analyses of the efficient of aviation could or ought to be used to harmonize the air transport framework focused on the development of an economically viable aviation infrastructure. This study suggests a move towards a more rational conception of air network design, based on differentiating local and global air passenger flows.

Operations research (OR) has played a critical role in helping the airline industry designs its air network, plans its scheduling, routing and crew assignment (Barnhart et al. 2003). Since 1961, there is a professional society dedicated to the advancement and application of Operational Research within the airline industry, the Airline Group of the International Federation of Operational Research Societies (AGIFORS). More than 500 airlines and air transport associations are currently represented in AGIFORS.

The contribution of this study is twofold: To propose a more explicit formulation that incorporates local and global flows to design air transportation networks with a

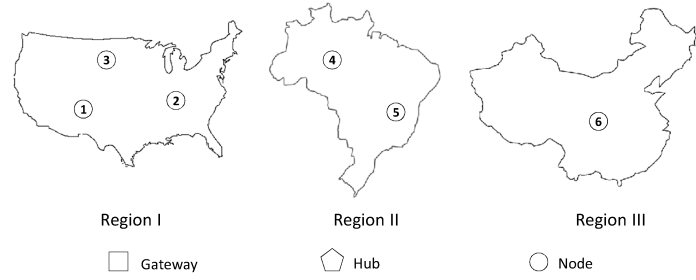
hub and spoke structure, and to devise exact algorithms capable of solving large-scale instances in reasonable time. The remainder of this article is organized as follows. §2.2 introduces the used notation and the proposed mathematical formulation; while §2.3 presents the devised Benders algorithms to solve the problem. §§2.4 and 2.5 reports the carried out computational experiments, and the insights provided by this study, respectively. §2.6 discusses achieved conclusions.

2.2 Notation, definitions and formulation

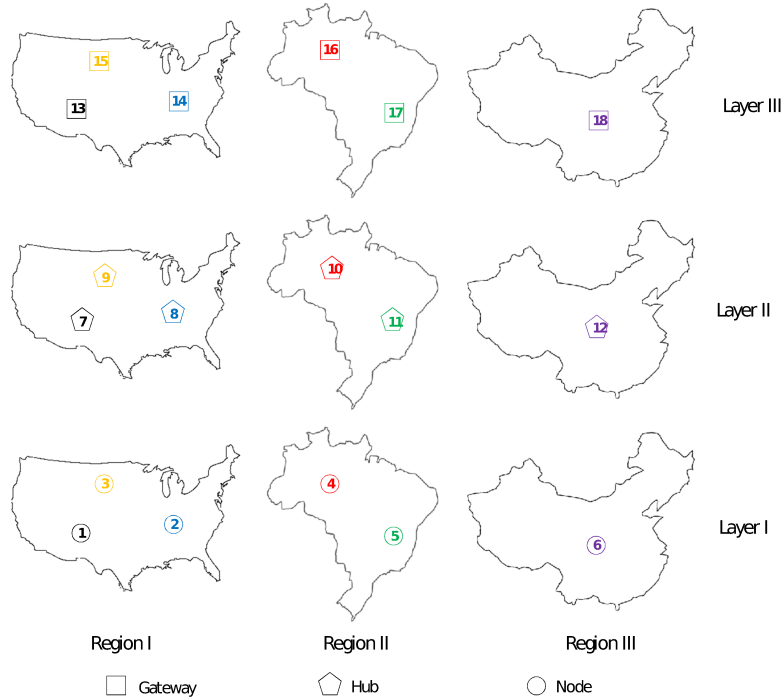
Given different geographic regions with airports which exchange flows between them, the addressed gateway hub location problem consists of locating hubs and gateways, and inter-hub and inter-gateway connections so that a hub and spoke based air network is designed, and local and global flows can be routed at minimal transportation and fixed costs. Local flows are not required to go through hubs to be routed, i.e. direct connections between local airports are allowed. Fixed costs for direct connections are disregarded, but transportation costs are accounted for them. However scale economies are only granted on inter-hub and inter-gateway connections which are required to be established. Global flows or flow exchanges between airports of different regions, pass through at least two gateways: one in the origin's region, and another in the destination's region. Consequently, each region must have at least one gateway. A gateway also acts as a hub, but a hub does not operate as a gateway unless one is installed at it. Local airports can be connected to more than one hub or gateway within their region, in other words, multiple allocations are allowed. Further the gateway level form a connected incomplete network.

To model the aforementioned problem, three different layers can be used to represent each airport operation type: layer *I* has all the actual local airports of each region, layer *II* contains hub candidate airports, while layer *III* has gateway candidate airports. To illustrate this idea, please refer to Figures 2.1a. The original configuration represented by Figure 2.1b has three different regions with six local airports, which compose layer *I*. Assuming that each local airport is a hub and a gateway candidate for this example, then, by copying each local airport of layer *I* and re-indexing and re-labeling it accordingly, it is possible to set layers *II* (airports 7 – 12) and *III* (airports 13 – 18). Please see Figure 2.1.

These three layers can be represented by a digraph $\mathbb{G} = (N, A)$ in which N and A are the airport and the arc sets, respectively. Set $N = L \cup H \cup G$ is formed by disjoint subsets L (local airport set), H (hub candidate airport set), and G (gateway candidate airport set)– note that set N has its airports re-indexed as aforementioned – while set



(a) Original setting with three regions and six local airports.



(b) Setting with three layers.

Figure 2.1: Example with the original setting and with a three layer configuration.

A has to be carefully assembled. Let $\mathcal{R} = \{1, \dots, n_{\mathcal{R}}\}$ be the region set, in which $n_{\mathcal{R}}$ is the number of regions, and $\phi(i)$ and $R(i)$ are two mathematical functions that return the original airport index and the region of airport $i \in N$, respectively. In the example of Figure 2.1, $\mathcal{R} = \{1, 2, 3\}$, $L = \{1, \dots, 6\}$, $H = \{7, \dots, 12\}$, and $G = \{13, \dots, 18\}$, and as e.g. $\phi(2) = \phi(8) = \phi(14) = 2$ and $R(6) = R(12) = R(18) = 3$. Set A is then composed of different arc sets or $A = A^L \cup A^H \cup A^G \cup A^{LH} \cup A^{HG}$, in which $A^L = \{(i, j) : i, j \in L \wedge R(i) = R(j) \wedge i \neq j\}$ is the local arc set, $A^H = \{(i, j) : i, j \in H \wedge R(i) = R(j) \wedge i \neq j\}$ is the hub arc set, $A^G = \{(i, j) : i, j \in G \wedge i \neq j\}$ is the gateway arc set. Note that gateway arcs can connect gateways within the same region or from different regions. Sets $A^{LH} = \{(i, j) : (i \in L \wedge j \in H) \vee (i \in H \wedge j \in L) \wedge \phi(i) = \phi(j)\}$ and $A^{HG} = \{(i, j) : (i \in H \wedge j \in G) \vee (i \in G \wedge j \in H) \wedge \phi(i) = \phi(j)\}$ have arcs connecting local airports to their associated hub candidates, and likewise arcs linking

hub candidates to their associated gateway candidates, respectively.

Further definitions are required to model the problem. Let $W = W^l \cup W^g$ be the demand set with pairs of airports (i, j) exchanging w_{ij} units of flow, and that is formed by two subsets. $W^l = \{(i, j) : i, j \in L \wedge R(i) = R(j) \wedge w_{ij} > 0\}$ and $W^g = \{(i, j) : i, j \in L \wedge R(i) \neq R(j) \wedge w_{ij} > 0\}$ are sets with pairs of airports which exchange flows on a local and global level, respectively. Let also c_{uv} be a non-negative unitary cost of arc $(u, v) \in A$ given as:

$$c_{uv} = \begin{cases} \tilde{c}_{uv} & \forall (u, v) \in A^L \\ \alpha^H \tilde{c}_{uv} & \forall (u, v) \in A^H \\ \alpha^G \tilde{c}_{uv} & \forall (u, v) \in A^G \\ b^H & \forall (u, v) \in A^{LH} \\ b^G & \forall (u, v) \in A^{HG} \end{cases}$$

in which \tilde{c}_{uv} is the unitary transportation cost of arc (u, v) , and $0 < \alpha^H < 1$ and $0 < \alpha^G < 1$ are granted discount factors to represent local and global scale economies, respectively. Parameters b^H and b^G are unitary operational costs due to baggage handling and custom and immigration checks done at hub and gateway levels, respectively.

To simplify notation, let $E = E^H \cup E^G$ be the edge set in which $E^H = \{(i, j) \in A^H : i < j\}$ and $E^G = \{(i, j) \in A^G : i < j\}$ are the edge sets associated to hub and gateway levels. Whenever an edge $(u, v) \in E$ is established to allow or inter-hub or inter-gateway flows a fixed cost q_{uv} given as:

$$q_{uv} = \begin{cases} \ell^H \tilde{q}_{uv} & \forall (u, v) \in E^H \\ \ell^G \tilde{q}_{uv} & \forall (u, v) \in E^G \end{cases}$$

is incurred, in which \tilde{q}_{uv} is the fixed cost of edge (u, v) , and ℓ^H and ℓ^G are fixed scaling factors for setting inter-hub and inter-gateway connections, respectively. Usually $\ell^H < \ell^G$. A fixed cost a_u is set for installing a hub or a gateway at airport $u \in H \cup G$.

With the aforementioned notation and definitions, and with the implementation of the layers, it is now possible to model the problem as multi-commodity flow based formulation with the help of the following variables: Let $y_u \in \{0, 1\}$ be equal to 1, if a hub or a gateway $u \in H \cup G$ is installed, 0 otherwise. Further let $x_{u,v} \in \{0, 1\}$ be equal to 1, if edge $(u, v) \in E$ is activated, 0, otherwise, and let $f_{uv}^{ij} \geq 0$ represent the flow percentage of demand w_{ij} , $(i, j) \in W$, that goes through arc $(u, v) \in A$. The formulation for the gateway hub location problem can now be written as:

$$\min \sum_{u \in H \cup G} a_u y_u + \sum_{(u,v) \in E} q_{uv} x_{uv} + \sum_{(i,j) \in W} \sum_{(u,v) \in A} w_{ij} c_{uv} f_{uv}^{ij} \quad (2.1)$$

$$\text{s.t.: } \sum_{(i,v) \in A} f_{iv}^{ij} = 1 \quad \forall (i,j) \in W \quad (2.2)$$

$$- \sum_{\substack{(v,u) \in A \\ j \neq v}} f_{vu}^{ij} + \sum_{\substack{(u,v) \in A \\ i \neq v}} f_{uv}^{ij} = 0 \quad \forall (i,j) \in W, u \in N : i \neq u, j \neq u \quad (2.3)$$

$$- \sum_{(u,j) \in A} f_{uj}^{ij} = -1 \quad \forall (i,j) \in W \quad (2.4)$$

$$f_{uv}^{ij} + f_{vu}^{ij} \leq x_{uv} \quad \forall (i,j) \in W, (u,v) \in E \quad (2.5)$$

$$\sum_{(u,v) \in A} f_{uv}^{ij} \leq y_u \quad \forall (i,j) \in W, u \in H \cup G \quad (2.6)$$

$$\sum_{\substack{(u,v) \in E^G \\ u \in S \\ v \in G \setminus S}} x_{uv} + \sum_{\substack{(v,u) \in E^G \\ u \in G \setminus S \\ v \in S}} x_{vu} \geq y_k + y_m - 1 \quad \forall k \in G \setminus S, m \in S : S \subset \mathbb{S} \wedge R(k) \neq R(m) \quad (2.7)$$

$$y_u \in \{0, 1\} \quad \forall u \in H \cup G \quad (2.8)$$

$$x_{u,v} \in \{0, 1\} \quad \forall (u,v) \in E \quad (2.9)$$

$$f_{uv}^{ij} \geq 0 \quad \forall (i,j) \in W, (u,v) \in A \quad (2.10)$$

in which $\mathbb{S} = \{S : S \subset G, |S| \geq 2\}$.

The objective function (2.1) minimizes the total cost consisted of the transportation costs, and the fixed installation costs for hubs and gateways, and for inter-hub and inter-gateway connections. Constraints (2.2)-(2.4) are the flow balancing equations. The constraints (2.5) guarantee that flows only go through inter-hub and inter-gateway connections if their respective edges are set. Constraints (2.6) ensure that flows can only go through a hub or a gateway $u \in H \cup G$, if u is set. Constraints (2.7) are the well-known sub-tour elimination constraints (SECs) which ensure that the gateway level is always connected. They can be disregarded whenever there are global flows from at least one region to all the other regions, otherwise they are required to ensure connectivity on the gateway level. Constraints (2.8)-(2.10) set the variables' domain.

The matrix associated with the constraints' set of formulation (2.1)-(2.10) has a stair-case shape regarding the large scale variables f , but which is coupled by the integer variables y and x . This feature makes the whole system amenable to a decomposition approach. For valid fixed values for variables y and x , the resulting subproblem decomposes into subsystems which are actually instances of shortest path problems, one for each $(i,j) \in W$. Hence a coordination scheme, akin Benders decomposition method (Benders 1962) which iteratively proposes valid values for variables y and x and solves shortest path instance problems, has a great appeal, since it will most likely surpass an approach that directly solves the whole formulation (2.1)-(2.10) at once. It

is also important to remark that this model is closely related to the model displayed in Camargo et. al. de Camargo et al. (2017), with several simplifications due to the size of the problems and instances under analysis.

2.3 Benders decomposition algorithms

The Benders decomposition technique (Benders 1962) is a classical exact method suitable to solve mixed integer linear programs with a stair-case matrix structure. In general terms, the method partitions the original problem into two simpler problems of smaller dimensions: a master problem (MP) and a subproblem (SP). The MP is a relaxed version of the original problem having only its integer variables and their respective constraints, but having the continuous variables projected out. These variables are replaced by an auxiliary variable, responsible for sub-estimating the objective function of the projected subsystem, and by associated cutting planes known as Benders cuts. While the SP is the original problem with the integer variables temporarily fixed by the MP. The algorithm iterates by solving the MP followed by the SP, while Benders cuts are separated from the SP and added to the MP at each iteration, until the lower bound (LB) and the upper bound (UB) converge to an optimal solution, if one exists. The LB is provided by the MP, whereas a UB is readily available from the SPs' solutions.

The Benders decomposition technique has been successfully applied to different problems and other hub location variants (Geoffrion and Graves 1974, Magnanti and Wong 1981, Birge and Louveaux 1988, Leung et al. 1990, Cordeau et al. 2000, Costa 2005, Gelareh and Nickel 2011, Contreras et al. 2011, Gelareh et al. 2015), being its performance closely related to the problem's structure, to how easily the SP is solved, and to the model's linear programming relaxation. A closer look at formulation (2.1)-(2.10) reveals that it shares these features. Hence, in this section, a Benders reformulation for the gateway hub location problem is shown, as well as a Benders algorithm to solve it. Further, a specialized efficient procedure that selects suitable dual optimal values among the multiple possible ones due to the dual SP's degeneracy is presented. This procedure is capable generates strong Benders cuts for the MP.

2.3.1 Benders subproblem and master problem

Let $\mathbb{Y} = \{(y, x) \in \mathbb{B}^{|H \cup G| \times |E|}\}$ be the set of feasible integer solutions associated to variables y and x for formulation (2.1)-(2.10). After parameterizing variables y and x , i.e. for $(\bar{y}, \bar{x}) \in \mathbb{Y}$, the following primal SP (PSP) is obtained:

$$(PSP) \quad v(\bar{y}, \bar{x}) = \min \sum_{(i,j) \in W} \sum_{(u,v) \in A} w_{ij} c_{uv} f_{uv}^{ij} \quad (2.11)$$

$$\text{s.t.:} \quad \sum_{(i,v) \in A} f_{iv}^{ij} = 1 \quad \forall (i,j) \in W \quad (2.12)$$

$$- \sum_{\substack{(v,u) \in A \\ j \neq v}} f_{vu}^{ij} + \sum_{\substack{(u,v) \in A \\ i \neq v}} f_{uv}^{ij} = 0 \quad \forall (i,j) \in W, u \in N : i \neq u, j \neq u \quad (2.13)$$

$$- \sum_{(u,j) \in A} f_{uj}^{ij} = -1 \quad \forall (i,j) \in W \quad (2.14)$$

$$- f_{uv}^{ij} - f_{vu}^{ij} \geq -\bar{x}_{uv} \quad \forall (i,j) \in W, (u,v) \in E \quad (2.15)$$

$$- \sum_{(u,v) \in A} f_{uv}^{ij} \geq -\bar{y}_u \quad \forall (i,j) \in W, u \in H \cup G \quad (2.16)$$

$$f_{uv}^{ij} \geq 0 \quad \forall (i,j) \in W, (u,v) \in A \quad (2.17)$$

Let $\pi_{iju} \in \mathbb{R}$, and $\beta_{ijuv} \geq 0$ and $\rho_{iju} \geq 0$ be the dual variables associated with constraints (2.12)-(2.14), and (2.15) and (2.16) respectively. Then the dual SP (DSP) can be written as:

$$(DSP) \quad v(\bar{y}, \bar{x}) = \max \sum_{(i,j) \in W} \left[\pi_{iji} - \pi_{ijj} - \sum_{u \in H \cup G} \bar{y}_u \rho_{iju} - \sum_{(u,v) \in E} \bar{x}_{uv} \beta_{ijuv} \right] \quad (2.18)$$

$$\text{s.t.:} \pi_{iju} - \pi_{ijv} - \rho_{iju} \leq w_{ij} c_{uv} \quad \forall (i,j) \in W, (u,v) \in A^{LH} \cup A^{HG} \quad (2.19)$$

$$\pi_{iju} - \pi_{ijv} - \beta_{ijuv} - \rho_{iju} \leq w_{ij} c_{uv} \quad \forall (i,j) \in W, (u,v) \in A^H \cup A^G : u < v \quad (2.20)$$

$$\pi_{iju} - \pi_{ijv} - \beta_{ijvu} - \rho_{iju} \leq w_{ij} c_{uv} \quad \forall (i,j) \in W, (u,v) \in A^H \cup A^G : u > v \quad (2.21)$$

$$\pi_{iju} - \pi_{ijv} \leq w_{ij} c_{uv} \quad \forall (i,j) \in W, (u,v) \in A^L \quad (2.22)$$

Observe that the feasible space (2.19)-(2.22) is invariable for any (\bar{y}, \bar{x}) value, and that the null vector is always a feasible solution to DSP, since $w_{ij} c_{uv} \geq 0$ for any $(i,j) \in W$ and $(u,v) \in A$. Further, the DSP is either bounded or unbounded, which, from strong duality, implies that either the PSP is feasible or infeasible, respectively. Hence, it is important to recognize when a $(\bar{y}, \bar{x}) \in \mathbb{Y}$ vector renders into a feasible PSP, i.e. into a bounded DSP. The condition under which such vector exists is given by Proposition 1.

Proposition 1. *The primal and dual SPs are feasible and bounded, for any $(\bar{y}, \bar{x}) \in \mathbb{Y}$ such that constraints (2.7), $\sum_{u \in G: R(u)=r} y_u \geq 1$ for all $r \in \mathcal{R}$ (i.e. there is at least one*

gateway per region), and $y_u \leq y_v$, for all $u \in G$ and $v \in H$, such that $\phi(u) = \phi(v)$ (i.e. a gateway can only be installed if its respective associated hub is also set), are respected.

Proof. Since by the problem's definition about the pre-existence of the local arcs given by set A^l , there is at least one path for each local flow $(i, j) \in W^l$ within its region. Further constraints $\sum_{u \in G: R(u)=r} y_u \geq 1$ for all $r \in \mathcal{R}$, and (2.7) assure that all regions has at least an installed gateway, and are also linked forming a connected graph, i.e. there is a path consisted of gateway arcs connecting any pair of regions. Moreover as constraints $y_u \leq y_v$ for all $u \in G$ and $v \in H$, such that $\phi(u) = \phi(v)$, guarantee that a gateway can only exists if its respective associated hub is also installed, and, by the problem's definition about the local arc set A^l that establishes that there is always a local arc from a local airport to a installed hub, then there is at least a path for any global flow $(i, j) \in W^G$. As $w_{ij}c_{uv}$ are finite for $(i, j) \in W$ and $(u, v) \in A$, and due to constraints (2.12)-(2.16) then any feasible solution to the PSP must be bounded. Hence, by strong duality, the DSP must also be feasible and bounded. \square

Let \mathcal{D} be the set of extreme points associated to DSP. It follows from Proposition 1 that the whole dual SP can then be expressed as:

$$v(\bar{y}, \bar{x}) = \max_{(\pi, \beta, \rho) \in \mathcal{D}} \sum_{(i,j) \in W} \left[\pi_{iji} - \pi_{ijj} - \sum_{u \in H \cup G} \bar{y}_u \rho_{iju} - \sum_{(u,v) \in E} \bar{x}_{uv} \beta_{ijuv} \right]$$

which, with the help of an auxiliary variable $\eta \geq 0$ to sub-estimate the routing costs, allows to reformulate formulation (2.1)-(2.10) as the following Benders MP:

$$(BMP) \quad \min \sum_{u \in H \cup G} a_u y_u + \sum_{(u,v) \in E} q_{uv} x_{uv} + \eta \quad (2.23)$$

$$\text{s.t.: (2.7) - (2.9)}$$

$$\eta \geq \sum_{(i,j) \in W} \left[\bar{\pi}_{iji} - \bar{\pi}_{ijj} - \sum_{u \in H \cup G} \bar{\rho}_{iju} y_u - \sum_{(u,v) \in E} \bar{\beta}_{ijuv} x_{uv} \right] \quad \forall (\bar{\pi}, \bar{\beta}, \bar{\rho}) \in \mathcal{D} \quad (2.24)$$

$$\sum_{\substack{u \in G \\ R(u)=r}} y_u \geq 1 \quad r \in \mathcal{R} \quad (2.25)$$

$$y_u \leq y_v \quad \forall u \in G, v \in H : \phi(u) = \phi(v) \quad (2.26)$$

$$\eta \geq 0 \quad (2.27)$$

Constraints (2.24) are known as Benders optimality cuts. There is one associated to each extreme point of the feasibility space of the DSP. As established by Proposition 1, the DSP is always bounded due to constraints (2.7) and (2.25) and (2.26), ergo there is no need for Benders feasibility cuts associated with the DSP's extreme rays to be added to the BMP. However if those constraints were to be disregarded, then Benders feasibility cuts like the following would be required:

$$0 \geq \sum_{(i,j) \in W} \left[\bar{\pi}_{iji} - \bar{\pi}_{ijj} - \sum_{u \in H \cup G} \bar{\rho}_{iju} y_u - \sum_{(u,v) \in E} \bar{\beta}_{ijuv} x_{uv} \right] \quad \forall (\bar{\pi}, \bar{\beta}, \bar{\rho}) \in \mathbb{E} \quad (2.28)$$

where \mathbb{E} is the set of extreme rays associated with the DSP's feasibility space. In this study, both approaches are evaluated and assessed on the computational experiments. To aid in resolution, some auxiliary valid inequalities are also added to the BMP:

$$x_{uv} \leq y_u \quad \forall (u, v) \in E \quad (2.29)$$

$$x_{uv} \leq y_v \quad \forall (u, v) \in E \quad (2.30)$$

$$\sum_{(u,v) \in E} x_{uv} + \sum_{(v,u) \in E} x_{vu} \geq y_u \quad \forall u \in G \quad (2.31)$$

$$\sum_{\substack{(u,v) \in E \\ R(u)=r \wedge R(v) \neq r}} x_{uv} + \sum_{\substack{(v,u) \in E \\ R(v) \neq r \wedge R(u)=r}} x_{vu} \geq 1 \quad \forall r \in \mathcal{R} \quad (2.32)$$

$$\sum_{\substack{(u,v) \in E \\ R(u) \neq R(v)}} x_{u,v} \geq n_{\mathcal{R}} - 1 \quad (2.33)$$

Constraints (2.29) and (2.30) ensure that an inter-hub or inter-gateway connection is established only if the respective associated hubs or gateways are also set. Constraints (2.31) guarantee that each installed gateway is connected to at least another gateway. Constraints (2.32) insure that each region is connected to at least another region. Finally, constraint (2.33) determines that at least $(n_{\mathcal{R}} - 1)$ inter-gateway connections linking different regions are installed.

Problem (2.1)-(2.10) was then reformulated into an equivalent mixed integer program with fewer variables, having only the integer variables y and x , and one continuous variable η . Though the BMP has a smaller dimension than the original problem, it has now two constrain sets with an exponential size that must be managed in a suitable fashion. All but a few of the Benders optimality constraints and the SECs are initially disregarded to be iteratively added to the BMP, on demand, till an optimal solution is attained for the original problem.

2.3.2 A basic Benders decomposition algorithm outline

Let UB and LB be the current upper and lower bounds, respectively, and h the current iteration index, $(\bar{y}^h, \bar{x}^h, \bar{\eta}^h)$ and $v(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h)$ the current optimal solutions for the BMP and the DSP for iteration h , respectively. Let also $\bar{\mathcal{D}}, \bar{\mathcal{E}},$ and $\bar{\mathcal{S}}$ be the restricted sets of extreme points and rays of the DSP's feasible space, and of disconnected components within G generated so far, up to iteration h . An outline of the Benders decomposition technique steps is presented in Algorithm 1. This algorithm results in two different versions, called here as Alg-1-v1 and Alg-1-v2. While in Alg-1-v1 the DSP is solved at each iteration, in Alg-1-v2, a procedure detects whether the solution from (\bar{y}, \bar{x}) results in an infeasible or feasible solution to formulation (2.1)-(2.10) and the DSP is solved only when a feasible (\bar{y}^h, \bar{x}^h) solution to formulation (2.1)-(2.10) is generated. Although both versions consider the addition of optimality Benders cuts, Alg-1-v1 adds feasibility Benders cut (2.28) and Alg-1-v2 adds SECs (2.7) when BMP solution is infeasible. The performance of Algorithm 1 is greatly affected by the computational effort spend on solving the BMP and DSP, and the total number of iterations required to attain optimality. These issues are addressed in the next sections.

2.3.3 Adding Pareto Optimal Cuts

One way to speed up the BMP's solution and shorten the required number of iterations for the Benders decomposition algorithm to reach optimality is by generating stronger, non dominated Benders cuts or Pareto-optimal cuts (Magnanti and Wong 1981). A cut constructed from a dual solution $(\bar{\pi}^1, \bar{\beta}^1, \bar{\rho}^1)$ is said to dominate a cut assembled from another dual solution $(\bar{\pi}^2, \bar{\beta}^2, \bar{\rho}^2)$ if and only if $\sum_{(i,j) \in W} [\bar{\pi}_{iji}^1 - \bar{\pi}_{ijj}^1 - \sum_{u \in H \cup G} \bar{\rho}_{iju}^1 y_u - \sum_{(u,v) \in E} \bar{\beta}_{ijuv}^1 x_{uv}] \geq \sum_{(i,j) \in W} [\bar{\pi}_{iji}^2 - \bar{\pi}_{ijj}^2 - \sum_{u \in H \cup G} \bar{\rho}_{iju}^2 y_u - \sum_{(u,v) \in E} \bar{\beta}_{ijuv}^2 x_{uv}]$, for all $(y, x) \in \mathbb{Y}$ with a strict inequality for at least one vector. A cut is said to be Pareto Optimal if it is not dominated by any other cut. To separate a Benders Pareto Optimal cut, Magnanti and Wong (1981) use a core-point or a reference point $(y^c, x^c) \in ri(\mathbb{P})$ belonging to the relative interior of the polyhedron $\mathbb{P} = \{(y, x) : (2.7), (2.25), (2.26), 0 \leq y_u \leq 1, \forall u \in H \cap G, \text{ and } 0 \leq x_{uv} \leq 1, \forall (u, v) \in E\}$ when solving an additional dual SP, besides the regular DSP. In other words, two different SPs are solved at each iteration: one associated to the current BMP's solution (\bar{y}^h, \bar{x}^h) , and another related to the core point (y^c, x^c) .

As this additional dual SP has a further dense equality constraint, it poses as a much harder problem to solve than the DSP and prone to numerical instabilities. Papadakos (2008) then proposes a lighter version to generate Benders Pareto Optimal cuts by solving the same DSP but with the core point $(y^c, x^c) \in ri(\mathbb{P})$ in place of the current

Algorithm 1 Basic Benders decomposition

```

UB ← +∞, LB ← -∞, stop ← false,  $\bar{\mathcal{D}} \leftarrow \emptyset$ ,  $\bar{\mathbb{E}} \leftarrow \emptyset$ ,  $\bar{\mathbb{S}} \leftarrow \emptyset$ 
while (stop = false) do
  {solve MP}
  (LB,  $\bar{y}^h$ ,  $\bar{x}^h$ ,  $\bar{\eta}^h$ ) ← BMP( $\bar{\mathcal{D}}$ ,  $\bar{\mathbb{E}}$  or  $\bar{\mathbb{S}}$ )
  if (UB = LB) then
    stop ← true
  else
    Alg-1-v1: {solve DSP}
     $v(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h) \leftarrow$  DSP( $\bar{y}^h$ ,  $\bar{x}^h$ )
    if ( $v(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h) < \infty$ ) then
      {bounded DSP}
      {add optimality Benders cuts}
       $\mathcal{D} \leftarrow \mathcal{D} \cup \{\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h\}$ 
      UB = min(UB, LB -  $\bar{\eta}^h$  +  $v(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h)$ )
    else
       $\bar{\mathbb{E}} \leftarrow \bar{\mathbb{E}} \cup \{(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h)\}$ 
    end if
    OR
    Alg-1-v2: {test the feasibility of BMP solution}
    if BMP solution is infeasible then
      {add SECs}
       $s \leftarrow$  find disconnected components within  $\{u \in G : \bar{y}_u^h = 1\}$ 
       $\bar{\mathbb{S}} \leftarrow \bar{\mathbb{S}} \cup \{s\}$ 
    else
      {solve DSP}
       $v(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h) \leftarrow$  DSP( $\bar{y}^h$ ,  $\bar{x}^h$ )
      {bounded DSP}
      {add optimality Benders cuts}
       $\mathcal{D} \leftarrow \mathcal{D} \cup \{\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h\}$ 
      UB = min(UB, LB -  $\bar{\eta}^h$  +  $v(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h)$ )
    end if
  end if
   $h \leftarrow h + 1$ 
end while

```

BMP's solution (\bar{y}^h, \bar{x}^h) instead, or:

$$(PDSP) \quad \max_{(\pi, \beta, \rho) \in \mathcal{D}} \sum_{(i,j) \in W} \left[\pi_{iji} - \pi_{ijj} - \sum_{u \in H \cup G} y_u^c \rho_{iju} - \sum_{(u,v) \in E} x_{uv}^c \beta_{ijuv} \right] \quad (2.34)$$

but updating the core point at each iteration in which the DSP renders a bounded solution. The core point is updated by a linear convex combination of the current BMP's solution (\bar{y}^h, \bar{x}^h) with the core point or $(y^c, x^c) = \lambda(y^c, x^c) + (1 - \lambda)(\bar{y}^h, \bar{x}^h)$, in

which $0 < \lambda < 1$. Empirically $\lambda = 0.5$ provides the best overall results (Papadakos 2008, Mercier et al. 2005). An initial core point is then required for starting the Benders decomposition algorithm. A valid one which respects the definition of $ri(\mathbb{P})$ can be given as:

$$(ICP) \quad y_u^c = \frac{1}{|\{u \in G : R(u) = r\}|} \quad \forall r \in \mathcal{R} \quad (2.35)$$

$$y_u^c = y_v^c \quad \forall u \in H, v \in G : \phi(u) = \phi(v) \quad (2.36)$$

$$x_{uv}^c = \min(y_u^c, y_v^c) \quad \forall (u, v) \in E \quad (2.37)$$

Proposition 2. *The vector (y^c, x^c) given by ICP is a valid core point, i.e. $(y^c, x^c) \in ri(\mathbb{P})$.*

Proof. By construction, constraints (2.25) and (2.26) are respected by equalities (2.35) and (2.36), respectively, and $0 < (y^c, x^c) < 1$. Finally, the SECs are also attended since $\sum_{\substack{(u,v) \in E^G \\ u \in S \\ v \in G \setminus S}} \min(y_u^c, y_v^c) + \sum_{\substack{(v,u) \in E^G \\ u \in G \setminus S \\ v \in S}} \min(y_u^c, y_v^c) \geq y_k^c + y_m^c - 1$ for all $k \in G \setminus S$, $m \in S$ such that $S \subset \mathbb{S}$ and $R(k) \neq R(m)$. Recall the k and m belong to different regions ($R(k) \neq R(m)$), hence their associated edge connection, i.e variable x_{km}^c if $k < m$, or x_{mk}^c if $k > m$, will appear on left hand side of its associated SEC. Further as $0 < y_k^c < 1$ and $0 < y_m^c < 1$ and $x_{km}^c = \min(y_k^c, y_m^c)$ by construction, then $\min(y_k^c, y_m^c) > y_k^c + y_m^c - 1$ for any pair of gateways located in different regions. \square

Algorithm 2 shows an outline of the Papadakos' Benders decomposition approach. Note that the PDSP is solved before the BMP, while using the current core point, and that, whenever the DSP is bounded, the core point is updated by a linear convex combination. Furthermore, this algorithm expound two variants of the Benders decomposition method, Alg-2-v1 and Alg-2-v2. Similarly to Algorithm 1, Alg-2-v1 solves the DSP at each iteration, adding feasibility Benders cuts when necessary; while Alg-2-v2 solves the DSP only when an infeasible (\bar{y}^h, \bar{x}^h) solution to formulation (2.1)-(2.10) is generated, separating SECs to be added to BMP. This particular solution ordering might speed up the convergence of the Benders decomposition algorithm (Papadakos 2008).

2.3.4 Adding multiple Benders cuts

For feasible integer solutions $(x, y) \in \mathbb{Y}$ for formulation (2.1)-(2.10), the PSP can be decomposed into $|W|$ independent subproblems, one for each $(i, j) \in W$. Each independent subproblem consists of a minimum shortest problem which can be easily

solved by Dijkstra's algorithm. Further it allows the separation of Benders optimality cuts from each independent subproblem that can then be added to the BMP. This greatly reduces the number of iterations required by the method to reach optimality (Birge and Louveaux 1988). Let \mathcal{D}_{ij} be the set of extreme points associated to DSP regarding the independent subproblem (i, j) , which, after redefining the variables η_{ij} accordingly, allows the BMP to be rewritten into a stronger form:

$$\min \sum_{u \in H \cup G} a_u y_u + \sum_{(u,v) \in E} q_{uv} x_{uv} + \sum_{(i,j) \in W} \eta_{ij} \quad (2.38)$$

s.t.: (2.7) – (2.9) and (2.25) and (2.26)

$$\eta_{ij} \geq \bar{\pi}_{iji} - \bar{\pi}_{ijj} - \sum_{u \in H \cup G} \bar{\rho}_{iju} y_u - \sum_{(u,v) \in E} \bar{\beta}_{ijuv} x_{uv} \quad \forall (i, j) \in W, (\bar{\pi}, \bar{\beta}, \bar{\rho}) \in \mathcal{D}_{ij} \quad (2.39)$$

$$\eta_{ij} \geq 0 \quad \forall (i, j) \in W \quad (2.40)$$

2.3.5 Solving Benders subproblem

Regarding the subproblem to find SECS within the installed hubs and gateways at an iteration h , a maximum flow problem (Ahuja et al. 1993, page 240), or the Tarjan's depth-first search (Tarjan 1972) to detect strong components can be used. Further, instead of relying on a linear programming solver to solve the DSP or the $|W|$ independent subproblems – one associated to each \mathcal{D}_{ij} – induced by feasible solutions $(y, x) \in \mathbb{Y}$ to formulation (2.1)-(2.10), a specialized procedure is devised to exploit the shortest path structure of the primal subproblem to efficiently generate dual optimal values. Let \mathcal{H}_u^{ij} be the length of the shortest path from i to node u on the shortest path from i to j for $(i, j) \in W$ induced by a feasible integer solution $(\bar{y}^h, \bar{x}^h) \in \mathbb{Y}$ to formulation (2.1)-(2.10) at iteration h . Note that $\mathcal{H}_j^{ij} = \sum_{(u,v) \in A} w_{ij} c_{uv} f_{uv}^{ij}$ corresponds to the shortest path from i and j for $(i, j) \in W$ in the rendered network by (\bar{y}^h, \bar{x}^h) . Let also \mathcal{T}_u^{ij} be the length of the shortest path from u to node j in the shortest path from i to j , but considering all hubs and gateways, and inter-hub and inter-gateway connections installed, i.e. all ones vector $(y, x) = \mathbf{1}$. Define $\delta_u^{ij} = \mathcal{H}_j^{ij} - \mathcal{T}_u^{ij}$, for all $u \in N$. Given these definitions, the optimal dual solution for $(i, j) \in W$ can be calculated as:

$$\pi_{iji} = 0 \quad (2.41)$$

$$\pi_{ijj} = \mathcal{H}_j^{ij} \quad (2.42)$$

$$\pi_{iju} = \min(\mathcal{H}_u^{ij}, \delta_u^{ij}) \quad \forall u \in N : u \neq i \wedge u \neq j \quad (2.43)$$

$$\rho_{iju} = \max(0, \pi_{iju} - \pi_{ijv} - w_{ij}c_{uv}) \quad \forall (u, v) \in A : u \in H \cap G \quad (2.44)$$

$$\beta_{ijuv} = \max(0, \pi_{iju} - \pi_{ijv} - \rho_{iju} - w_{ij}c_{uv}) \quad \forall (u, v) \in A^H \cap A^G : u < v \quad (2.45)$$

$$\beta_{ijvu} = \max(0, \pi_{iju} - \pi_{ijv} - \rho_{iju} - w_{ij}c_{uv}) \quad \forall (u, v) \in A^H \cap A^G : u > v \quad (2.46)$$

Proposition 3. *Equations (2.41)-(2.46) compute optimal dual values for \mathcal{D}_{ij} for feasible integer solutions $(y, x) = (\bar{y}^h, \bar{x}^h)$ to formulation (2.1)-(2.10) at iteration h .*

Proof. By construction, equations (2.44)-(2.46) insure that the set (π, ρ, β) is feasible for \mathcal{D}_{ij} . Since equations (2.41) and (2.42) set $\pi_{iji} = 0$ and $\pi_{ijj} = \mathcal{H}_j^{ij}$ by definition, respectively, this allows to interpret variables β_{ijuv} and ρ_{iju} as the reduction in the shortest path length from node i to j if the inter-hub or inter-gateway connection $(u, v) \in E$, or the hub or gateway airport $u \in H \cap G$ are installed. Hence, when $\bar{x}_{uv}^h = 1$ or $\bar{y}_u^h = 1$ imply that $\beta_{ijuv} = 0$ or $\rho_{iju} = 0$ because of the complementary slackness condition to ensure that $\pi_{iji} = 0$ and $\pi_{ijj} = \mathcal{H}_j^{ij}$, guarantee therefore that the optimal dual solution for \mathcal{D}_{ij} will be \mathcal{H}_j^{ij} . \square

2.3.6 Repairing infeasible primal solutions

Whenever the DSP is unbounded at an iteration h , no optimality Benders cuts are separated, no core point is updated, no Pareto Optimal cuts are generated, and no improvement on the Benders decomposition algorithm's lower bound might be perceived on the next iteration. Hence no contribution to the method's convergence is observed. To dwindle this effect, a simple, but effective repair procedure to find a feasible solution from a (\bar{y}, \bar{x}) infeasible solution to formulation (2.1)-(2.10) is developed. For iterations with infeasible network generated by the BMP, beyond the separation of SECs, the repair procedure is called followed by the specialized algorithm to solve the DSP so that optimality Benders cuts can be separated. With this procedure, the core points can be updated at each iteration and consequently, Pareto Optimal Benders cuts can be added even when a (\bar{y}, \bar{x}) infeasible solution to formulation (2.1)-(2.10) is found.

Since all local airports are directly connected, the origin of an infeasible solution is from global demand. Therefore, make a network feasible implies in guarantee that all regions are connected. Since the regions are connected by gateways arcs, this sub-problem considers the arcs $(u, v) \in A^G$. The global demand, $(i, j) \in W^g$, is aggregated on gateways, resulting in an auxiliary demand matrix, W^{aux} . Each node $i \in L$ is associated to the nearest gateway $u \in G$, such that $R(i) = R(u)$, and the demand of this node $i \in I$ is added on the gateway $u \in G$ demand. To illustrate, assume that there is a demand of 10 units from node 1 ($R(1) = 1$ and closer to gateway 14, $R(14) = 1$) to

node 2 ($R(2) = 2$ and closer to gateway 17, $R(17) = 2$). Thus, the new demand from gateway 14 to gateway 17 will be at least equal 10 units.

Using the subdigraph $\mathbb{G}' = (G, A^G)$ of Section 2.2, and redefining the unitary transportation cost c_{uv} for an infeasible (\bar{y}^h, \bar{x}^h) solution to formulation (2.1)-(2.10) at iteration h as:

$$c_{uv} = \begin{cases} q_{uv} & \forall (u, v) \in A^G : ((u, v) \vee (v, u)) \in E \wedge \bar{x}_{uv}^h = 0 \wedge \bar{y}_u^h = 1 \wedge \bar{y}_v^h = 1 \\ q_{uv} + a_u & \forall (u, v) \in A^G : ((u, v) \vee (v, u)) \in E \wedge \bar{x}_{uv}^h = 0 \wedge \bar{y}_u^h = 0 \wedge \bar{y}_v^h = 1 \\ q_{uv} + a_v & \forall (u, v) \in A^G : ((u, v) \vee (v, u)) \in E \wedge \bar{x}_{uv}^h = 0 \wedge \bar{y}_u^h = 1 \wedge \bar{y}_v^h = 0 \\ q_{uv} + a_u + a_v & \forall (u, v) \in A^G : ((u, v) \vee (v, u)) \in E \wedge \bar{x}_{uv}^h = 0 \wedge \bar{y}_u^h = 0 \wedge \bar{y}_v^h = 0 \\ 0 & \forall (u, v) \in A^G : ((u, v) \vee (v, u)) \in E \wedge \bar{x}_{uv}^h = 1 \\ 0 & \forall (u, v) \in A^G : ((u, v) \vee (v, u)) \in E \wedge \bar{x}_{vu}^h = 1 \end{cases}$$

Then, for each $(i, j) \in W^{aux}$, a Dijkstra's algorithm is called on the a subdigraph $\mathbb{G}' = (G, A^G)$. The (\bar{y}^h, \bar{x}^h) solution is repaired at the end, by activating the hubs and gateways, and inter-hub and inter-gateway connections that are part of the attained shortest path, but were not originally present in the (\bar{y}^h, \bar{x}^h) solution. The specialized algorithm of Section 2.3.5 calculates the optimal dual values for the repaired solution and a new optimality Benders cut is added to the BMP, as can be seen in Algorithm 3. Note that if Pareto Optimal Benders cuts are not considered, we can have two versions of Algorithm 3, Alg-3-v1 and Alg3-v2. In Alg-3-v1, only optimality Benders cuts and SECs are separated, while in Alg-3-v2, Pareto Optimal Benders cuts are also added at each iteration.

2.4 Computational experiments

In order to test the proposed variants of the Benders decomposition for the GHLP, we have generated an instance set, called here as global set. To compose the instance set, 141 big cities have been selected. Each city has its latitude, longitude, population and Gross Domestic Product (GDP). Depending on its location, they have been divided in regions.

Assuming that p_i is the population of city i divided by 100000, g_i the factor representing the GDP of city i , d_{ij} the distance between cities i and j , and the demand from city i to city j w_{ij} is expressed as follows:

$$w_{ij} = p_i p_j g_i g_j e^{-0.01 d_{ij}} \quad (2.47)$$

Cities are placed in descending order of population, and then, the n most populous cities are selected to compose a new instance. Hence, if $k > m$, an instance of k nodes has all m nodes presented in the m nodes instance and also more $k - m$ different nodes. In other words, global-19 instance has the same nodes of global-12 instance and other 7 different cities. Different problems consisting of the first n nodes have been generated for $n = 12, 19, 29, 37, 43, 48, 59, 74, 100$ and 141. Table (2.1) relates the instance's name with its number of nodes, candidates to become a hub, candidates to become a gateway and regions.

Table 2.1: Description of the global instances created

Instances	# Nodes	# Hubs	# Gateways	# Regions
global-12	12	11	11	8
global-19	19	17	17	10
global-29	29	24	24	14
global-37	37	28	28	15
global-43	43	32	32	15
global-48	48	35	35	15
global-59	59	40	40	15
global-74	74	48	48	16
global-100	100	60	60	18
global-141	141	67	67	19

On one hand, the fixed costs for an airport becoming hub or gateway do not differ from one airport to another. On the other hand, the fixed costs to install arcs between domestic or international airports are a weighted function of the arc length. The round of experiments aims to evaluate how instances become harder by adopting more aggressive fixed costs. Table (2.2) shows five different settings evaluated. In setting I, the lowest fixed costs are considered, while, in setting V, the largest fixed costs are taken into account. In settings II and IV, the cost of activating hubs and gateways is assumed to be the same. In settings I, III and V, activating gateways is more expensive than opening hubs. We always assume that installing hub arcs is cheaper than activating gateway arcs.

Table 2.2: Five settings

Values	Setting I	Setting II	Setting III	Setting IV	Setting V
Hub fixed cost	10^3	10^4	10^4	10^5	10^5
Gateway fixed cost	10^4	10^4	10^5	10^5	10^6
Weight of domestic arcs	10^{-1}	10^0	10^0	10	10
Weight of international arcs	10^3	10^3	10^4	10^4	10^5

All computational tests have been carried out on a Dell PowerEdge T620 workstation, equipped with two Intel Xeon E5-2600v2 processors and 96 GB of RAM memory.

Table 2.3: Results for smaller instances - Setting I

Versions	global-12			global-19			global-29			global-37		
	# Iters	GAP(%)	Time[s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]
Monolithic	-	0.00	0	-	0.00	3	-	0.00	26	-	0.00	349
Alg-1-v1	11	0.00	4	12	0.00	28	16	0.00	207	17	0.00	770
Alg-1-v2-C	13	0.00	3	17	0.00	23	23	0.00	189	27	0.00	1230
Alg-1-v2-I	12	0.00	0	16	0.00	5	22	0.00	31	18	0.00	45
Alg-3-v1	7	0.00	0	11	0.00	4	10	0.00	14	10	0.00	26
Alg-2-v1	8	0.00	3	9	0.00	18	11	0.00	161	10	0.00	494
Alg-2-v2-C	13	0.00	4	12	0.00	27	9	0.00	122	11	0.00	322
Alg-2-v2-I	13	0.00	2	13	0.00	14	12	0.00	67	14	0.00	169
Alg-3-v2	8	0.00	3	8	0.00	15	8	0.00	77	8	0.00	189

Table 2.4: Results for smaller instances - Setting II

Versions	global-12			global-19			global-29			global-37		
	# Iters	GAP(%)	Time[s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]
Monolithic	-	0.00	0	-	0.00	2	-	0.00	24	-	0.00	396
Alg-1-v1	9	0.00	3	13	0.00	29	15	0.00	193	17	0.00	1055
Alg-1-v2-C	14	0.00	3	20	0.00	31	24	0.00	213	32	0.00	984
Alg-1-v2-I	14	0.00	1	19	0.00	10	23	0.00	59	23	0.00	114
Alg-3-v1	10	0.00	1	15	0.00	11	13	0.00	35	13	0.00	68
Alg-2-v1	7	0.00	3	8	0.00	26	12	0.00	221	11	0.00	527
Alg-2-v2-C	13	0.00	4	12	0.00	27	10	0.00	145	13	0.00	326
Alg-2-v2-I	13	0.00	2	13	0.00	14	12	0.00	68	14	0.00	170
Alg-3-v2	8	0.00	3	8	0.00	15	8	0.00	77	8	0.00	191

Also, all the algorithms have been implemented in C++ using Concert Technology (CPLEX 12.5).

We have generated 8 variants of the Benders decomposition by changing the methods used to solve the DSP. Alg-1-v2 and Alg-2-v2, in fact, become Alg-1-v2-C, Alg-1-v2-I, Alg-2-v2-C and Alg-2-v2-I. The letter "C" indicates that the DSP is solved by CPLEX, while the letter "I" points out the implementation of the specialized algorithm of Section 2.3.5. The monolithic version corresponds to equations (2.1) - (2.10).

To assess the implemented versions, in Table (2.3)-(2.12), we report the number of iterations required until convergence for the Benders proposed variants algorithms (# Iters), the total time required to attain an optimal solution (Time [s]), up to a limit of 259200 seconds (3 days), and the GAP when an optimal solution is not found within this time (GAP) were recorded. The symbol '-' in GAP column represents that no integer solution was found in the maximum time determined, and in time column that the algorithm ran until the maximum time allowed.

Two sets of experiments were executed. On the first one, comparison between all the variants of the Benders decomposition method is made for the smaller instances (until 37 airports). Tables (2.3), (2.4), (2.5), (2.6) and (2.7) illustrate the results obtained for settings I, II, III, IV and V, respectively.

As can be seen on Tables (2.3), (2.4), (2.5), (2.6) and (2.7), in all settings, except in setting V, the Monolithic version presents the best behavior for smaller instances

Table 2.5: Results for smaller instances - Setting III

Versions	global-12			global-19			global-29			global-37		
	# Iters	GAP(%)	Time[s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]
Monolithic	-	0.00	0	-	0.00	13	-	0.00	241	-	0.00	959
Alg-1-v1	10	0.00	4	16	0.00	67	28	0.00	6513	35	0.09	-
Alg-1-v2-C	13	0.00	4	24	0.00	67	34	0.00	4228	40	0.08	-
Alg-1-v2-I	12	0.00	1	25	0.00	42	34	0.00	1179	39	0.00	7008
Alg-3-v1	8	0.00	1	41	0.00	19	24	0.00	1040	24	0.00	6218
Alg-2-v1	7	0.00	4	11	0.00	41	20	0.00	888	22	0.00	5513
Alg-2-v2-C	9	0.00	3	14	0.00	33	18	0.00	257	22	0.00	576
Alg-2-v2-I	10	0.00	2	22	0.00	29	24	0.00	144	23	0.00	317
Alg-3-v2	6	0.00	2	10	0.00	26	13	0.00	161	15	0.00	452

Table 2.6: Results for smaller instances - Setting IV

Versions	global-12			global-19			global-29			global-37		
	# Iters	GAP(%)	Time[s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]
Monolithic	-	0.00	0	-	0.00	14	-	0.00	275	-	0.00	1182
Alg-1-v1	10	0.00	5	19	0.00	262	34	0.00	93707	19	7.43	-
Alg-1-v2-C	14	0.00	5	29	0.00	390	46	0.00	72244	32	6.30	-
Alg-1-v2-I	26	0.00	10	66	0.00	1918	81	0.33	-	44	7.30	-
Alg-3-v1	16	0.00	8	50	0.00	1682	66	1.08	-	32	5.76	-
Alg-2-v1	8	0.00	4	15	0.00	74	19	0.00	1098	19	0.00	5287
Alg-2-v2-C	8	0.00	4	12	0.00	33	15	0.00	232	26	0.00	898
Alg-2-v2-I	8	0.00	2	16	0.00	20	26	0.00	158	33	0.00	439
Alg-3-v2	6	0.00	2	13	0.00	27	14	0.00	188	15	0.00	492

Table 2.7: Results for smaller instances - Setting V

Versions	global-12			global-19			global-29			global-37		
	# Iters	GAP(%)	Time[s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]
Monolithic	-	0.00	2	-	0.00	195	-	0.00	4647	-	0.00	20405
Alg-1-v1	9	0.00	4	18	0.00	116	28	0.00	8232	41	0.00	194300
Alg-1-v2-C	13	0.00	4	23	0.00	127	35	0.00	3493	51	0.00	231401
Alg-1-v2-I	16	0.00	3	42	0.00	365	70	0.00	41388	66	0.02	-
Alg-3-v1	12	0.00	3	39	0.00	362	61	0.00	26466	48	0.00	134488
Alg-2-v1	7	0.00	4	9	0.00	44	18	0.00	645	21	0.00	3436
Alg-2-v2-C	8	0.00	3	11	0.00	28	16	0.00	200	16	0.00	506
Alg-2-v2-I	10	0.00	2	12	0.00	18	18	0.00	107	19	0.00	290
Alg-3-v2	9	0.00	3	10	0.00	26	11	0.00	149	13	0.00	441

(global-12 and global-19). For instances global-29 and global-37, there are always a variant of the Benders decomposition method that performs better than the Monolithic version. As fixed costs increase, so does the performance difference between the best version of Benders and the Monolithic version for a given instance. For example, while monolithic version requires 13 times more computational effort than Alg-3-v1 to solve global-37 instance in setting I, in setting V, monolithic version requires 70 times more computational effort than Alg-2-v2-I to solve the same instance.

It is possible to see that the instances become harder as fixed costs increase. The Monolithic version spends about 58 times longer to find an optimal solution for the global-37 instance in scenario V than in scenario I. Moreover, while the global-37 instance is solved within 26 seconds by Alg-3-v1 in setting I, in setting V, Alg-2-

v2-I requires 290 seconds to converge to an optimal solution. When fixed costs are higher, the influence of transportation costs decreases and the problem becomes more combinatorial.

Furthermore, the versions that consider the solution of DSP by using CPLEX and/or the addition of feasibility cuts instead of SECs (Alg-1-v1, Alg-1-v2-C, Alg-2-v1, Alg-2-v2-C) perform worse than the other versions. This implies that CPLEX requires more time to solve the DSP than the method described in 2.3.5. Although the versions Alg-2-v2-I and Alg-3-v1 are not able to find an optimal solution within the time limit for some instances in settings IV and V, they performed very well in settings I and II, in which fixed costs are lower. As expected, the separation of Pareto Optimal cuts presents a better behavior for harder instances.

Algorithm 2 Papadakos based Benders decomposition

```

UB  $\leftarrow$   $+\infty$ , LB  $\leftarrow$   $-\infty$ , stop  $\leftarrow$  false,  $\bar{\mathcal{D}} \leftarrow \emptyset$ ,  $\bar{\mathbb{E}} \leftarrow \emptyset$ ,  $\bar{\mathbb{S}} \leftarrow \emptyset$ 
bounded  $\leftarrow$  true
 $(y^c, x^c) \leftarrow ICP$ 
while (stop = false) do
  if (bounded = true) then
    {solve PDSP}
     $v(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h) \leftarrow \text{PDSP}(y^c, x^c)$ 
     $\mathcal{D} \leftarrow \mathcal{D} \cup \{\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h\}$ 
  end if
  {solve MP}
   $(\text{LB}, \bar{y}^h, \bar{x}^h, \bar{\eta}^h) \leftarrow \text{BMP}(\bar{\mathcal{D}}, \bar{\mathbb{E}} \text{ or } \bar{\mathbb{S}})$ 
  if (UB = LB) then
    stop  $\leftarrow$  true
  else
    Alg-2-v1: {solve DSP}
     $v(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h) \leftarrow \text{DSP}(\bar{y}^h, \bar{x}^h)$ 
    if ( $v(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h) < \infty$ ) then
      {bounded DSP}
      bounded  $\leftarrow$  true
       $(y^c, x^c) \leftarrow \lambda(y^c, x^c) + (1 - \lambda)(\bar{y}^h, \bar{x}^h)$ ,
      {add optimality Benders cuts}
       $\mathcal{D} \leftarrow \mathcal{D} \cup \{\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h\}$ 
      UB = min(UB, LB -  $\bar{\eta}^h$  +  $v(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h)$ )
    else
      {unbounded DSP}
      bounded  $\leftarrow$  false
       $\bar{\mathbb{E}} \leftarrow \bar{\mathbb{E}} \cup \{(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h)\}$ 
    end if
    OR
    Alg-2-v2: {test the feasibility of BMP solution}
    if BMP solution is infeasible then
      {add SECs}
       $s \leftarrow$  find disconnected components within  $\{u \in G : \bar{y}_u^h = 1\}$ 
       $\bar{\mathbb{S}} \leftarrow \bar{\mathbb{S}} \cup \{s\}$ 
    else
      {solve DSP}
       $v(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h) \leftarrow \text{DSP}(\bar{y}^h, \bar{x}^h)$ 
      {bounded DSP}
      bounded  $\leftarrow$  true
       $(y^c, x^c) \leftarrow \lambda(y^c, x^c) + (1 - \lambda)(\bar{y}^h, \bar{x}^h)$ ,
      {add optimality Benders cuts}
       $\mathcal{D} \leftarrow \mathcal{D} \cup \{\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h\}$ 
      UB = min(UB, LB -  $\bar{\eta}^h$  +  $v(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h)$ )
    end if
  end if
   $h \leftarrow h + 1$ 
end while

```

Algorithm 3 Repair Benders decomposition

```

UB  $\leftarrow$   $+\infty$ , LB  $\leftarrow$   $-\infty$ , stop  $\leftarrow$  false,  $\bar{\mathcal{D}} \leftarrow \emptyset$ ,  $\bar{\mathbb{E}} \leftarrow \emptyset$ ,  $\bar{\mathbb{S}} \leftarrow \emptyset$ 
 $(y^c, x^c) \leftarrow ICP$ 
while (stop = false) do
  {solve PDSP}
   $v(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h) \leftarrow \text{PDSP}(y^c, x^c)$ 
   $\mathcal{D} \leftarrow \mathcal{D} \cup \{\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h\}$ 
  {solve MP}
   $(\text{LB}, \bar{y}^h, \bar{x}^h, \bar{\eta}^h) \leftarrow \text{BMP}(\bar{\mathcal{D}}, \bar{\mathbb{E}} \text{ or } \bar{\mathbb{S}})$ 
  if (UB = LB) then
    stop  $\leftarrow$  true
  else
    {test the feasibility of BMP solution}
    if BMP solution is infeasible then
      {add SECs}
       $s \leftarrow$  find disconnected components within  $\{u \in G : \bar{y}_u^h = 1\}$ 
       $\bar{\mathbb{S}} \leftarrow \bar{\mathbb{S}} \cup \{s\}$ 
      {repair solution}
       $(\bar{y}^h, \bar{x}^h) \leftarrow \text{Repair}(\bar{y}^h, \bar{x}^h)$ 
      {solve DSP}
       $v(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h) \leftarrow \text{DSP}(\bar{y}^h, \bar{x}^h)$ 
       $\mathcal{D} \leftarrow \mathcal{D} \cup \{\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h\}$ 
    else
      {solve DSP}
       $v(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h) \leftarrow \text{DSP}(\bar{y}^h, \bar{x}^h)$ 
      {bounded DSP}
      {add optimality Benders cuts}
       $\mathcal{D} \leftarrow \mathcal{D} \cup \{\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h\}$ 
    end if
     $(y^c, x^c) \leftarrow \lambda(y^c, x^c) + (1 - \lambda)(\bar{y}^h, \bar{x}^h)$ 
    UB = min(UB, LB -  $\bar{\eta}^h$  +  $v(\bar{\pi}^h, \bar{\beta}^h, \bar{\rho}^h)$ )
  end if
   $h \leftarrow h + 1$ 
end while

```

Table 2.8: Results for bigger instances - Setting I

Versions	global-43			global-48			global-59			global-74			global-100			global-141		
	# Iters	GAP(%)	Time[s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]
Monolithic	-	0.00	1009	-	0.00	1879	-	0.00	3801	-	0.00	9882	-	0.00	45298	-	-	-
Alg-1-v2-I	24	0.00	166	44	0.00	941	37	0.00	3815	55	0.02	27149	21	-	-	37	6.42	-
Alg-3-v1	11	0.00	74	13	0.00	204	16	0.00	889	14	0.00	2523	13	0.07	-	8	1.48	-
Alg-2-v2-I	14	0.00	395	13	0.00	640	23	0.00	1590	56	0.00	23935	27	0.00	23031	34	0.00	59391
Alg-3-v2	8	0.00	388	9	0.00	709	8	0.00	1472	9	0.00	4545	8	0.00	14124	9	0.00	55291

Table 2.9: Results for bigger instances - Setting II

Versions	global-43			global-48			global-59			global-74			global-100			global-141		
	# Iters	GAP(%)	Time[s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]
Monolithic	-	0.00	1230	-	0.00	2137	-	0.00	4463	-	0.00	10023	-	0.00	39442	-	-	-
Alg-1-v2-I	26	0.00	325	46	0.00	1753	42	0.00	9196	53	0.01	-	22	-	-	33	12.49	-
Alg-3-v1	14	0.00	280	17	0.00	567	22	0.00	7062	19	0.00	162188	11	3.54	-	6	8.06	-
Alg-2-v2-I	13	0.00	395	14	0.00	732	25	0.00	1814	30	0.00	5272	35	0.00	35198	32	0.00	65951
Alg-3-v2	8	0.00	391	9	0.00	725	9	0.00	1686	9	0.00	4418	10	0.00	17995	10	0.00	62404

Table 2.10: Results for bigger instances - Setting III

Versions	global-43			global-48			global-59			global-74			global-100			global-141		
	# Iters	GAP(%)	Time[s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]
Monolithic	-	0.00	1859	-	0.00	3928	-	0.00	10301	-	0.00	37840	-	-	-	-	-	-
Alg-1-v2-I	46	0.00	63480	62	0.00	111016	46	2.97	-	62	4.63	-	22	-	-	26	90.60	-
Alg-3-v1	28	0.00	43694	28	0.00	70738	22	1.10	-	15	3.15	-	9	13.44	-	6	20.43	-
Alg-2-v2-I	31	0.00	631	27	0.00	1019	29	0.00	2896	36	0.00	1489	44	0.00	200848	32	0.27	-
Alg-3-v2	12	0.00	668	11	0.00	1178	16	0.00	3540	13	0.00	9154	14	0.00	83255	10	1.64	-

Table 2.11: Results for bigger instances - Setting IV

Versions	global-43			global-48			global-59			global-74			global-100			global-141		
	# Iters	GAP(%)	Time[s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]
Monolithic	-	0.00	1968	-	0.00	4386	-	0.00	13407	-	0.00	39828	-	-	-	-	-	-
Alg-1-v2-I	44	17.95	-	54	18.64	-	41	25.68	-	53	28.39	-	24	-	-	29	88.01	-
Alg-3-v1	23	13.71	-	18	17.81	-	14	24.36	-	9	31.04	-	6	41.90	-	5	39.75	-
Alg-2-v2-I	23	0.00	640	32	0.00	1410	20	0.00	2669	40	0.00	24736	39	0.00	114895	26	2.55	-
Alg-3-v2	13	0.00	755	14	0.00	1520	12	0.00	2887	16	0.00	10778	15	0.00	56425	6	2.90	-

Table 2.12: Results for bigger instances - Setting V

Versions	global-43			global-48			global-59			global-74			global-100			global-141		
	# Iters	GAP(%)	Time[s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]	# Iters	GAP(%)	Time [s]
Monolithic	-	0.00	45241	-	0.00	52210	-	0.00	152610	-	0.01	-	-	-	-	-	-	-
Alg-1-v2-I	51	2.59	-	59	5.48	-	49	15.98	-	44	14.53	-	22	-	-	29	66.40	-
Alg-3-v1	34	2.38	-	28	4.47	-	17	9.25	-	10	15.75	-	7	28.00	-	4	40.96	-
Alg-2-v2-I	21	0.00	712	28	0.00	1940	32	0.00	11554	42	0.00	35208	28	6.37	-	29	9.97	-
Alg-3-v2	13	0.00	923	15	0.00	2128	16	0.00	10078	16	0.00	19112	15	0.00	126468	7	7.30	-

As the best versions in setting I and II are Alg-1-v2-I and Alg-3-v1 and in settings III, IV and V, are Alg-2-v2-I and Alg-3-v2, the second experiment compares these versions with the Monolithic version. As can be seen on Tables 2.8, 2.9, 2.10, 2.11 and 2.12, the larger the instance, the more the performance of Alg-2-v2-I and Alg-3-v2 versions stand out. In setting I, which has the lowest fixed costs, Alg-3-v1 converges faster until global-74 instance. Alg-3-v2 outperforms the Alg-3-v1 on the global-59 instance in setting II. In the other settings, either Alg-2-v2-I or Alg-3-v2 present the best behavior. Again, it shows up that Pareto Optimal Benders cuts are effective on harder instances.

The fact that instances on setting V are harder than those on setting I, once more, stands out. For global-141 instance, while an optimal solution is determined in 55291 and in 62404 seconds in settings I and II, respectively, none of the versions can find an optimal solution within the maximum computational time allowed in the other settings.

Comparing versions Alg-2-v2-I and Alg-3-v2, we can observe that Alg-2-v2-I always requires more iterations to converge to an optimal solution. In setting III, for example, for global-100 instance, Alg-2-v2-I takes 3 times more iterations and almost 2.5 times more computational time to converge. However, not always Alg-3-v2 requires less time to find an optimal solution. For global-74 instance, in setting III, Alg-3-v2 takes 6 times more computational time to converge. This behavior can be explained by observing that in Alg-3-v2, Pareto Optimal Benders cuts are separated at each iteration, even though an infeasible (\bar{y}^h, \bar{x}^h) solution to formulation (2.1)-(2.10) is generated. As these cuts are stronger, the number of iterations required to the convergence of the method decreases, but the time required in each iteration increases (PM becomes harder to solve and also we have to consider that the PDSP is solved at each iteration). Therefore, there is a trade-off between the time spent in each iteration and the necessary number of iterations to the convergence of the algorithm.

It is remarkable the better performance of Alg-2-v2-I and Alg-3-v2 compared to Monolithic version. For global-100 instance, for example, while Alg-2-v2-I and Alg-3-v3 converges to an optimal solution using less than 50% of the maximum time set in five settings, the monolithic version is able to find an optimal solution only for the first two settings (I and II).

2.5 Insights from this study

The problem proposed in this paper may act as a tool for providing insights into the current air network. First, empirical results for intermediate instances are presented for a variety of cost parameter values. Next, in order to exemplify some practical benefits

of the work developed here, the largest instance (global-141 instance) is chosen to be analyzed.

2.5.1 Measuring the Influence of Economies of Scale and Transportation Costs on the Optimal Topologies

To illustrate the influence of the transportation costs in an optimal network, our formulation is solved to optimality by using different values for scale economies (α^H , α^G) and unitary operational costs (b^H , b^G). Our first analyses employ global-37, global-43 and global-48 instances in a series of numerical examples. Whereas it is not possible to make sweeping generalizations by deploying only three test problems, it is possible though to compare and contrast the features and characteristics of the attained network designs regarding the variations in the transportation costs.

We set the hub and gateway installation fixed costs (a^H and a^G), and the hub and gateway arc installation fixed costs (q^H and q^G) to 10,000.00, 100,000.00, 10.00, 1,000.00, respectively, to assess how the optimal networks are affected by varying the scale economies or by varying unitary operational costs. In the first experiment, the domestic and international unitary operational costs (b^H and b^G) are both assumed as 1.00. Six different scenarios were constructed in which local (α^H) and global (α^G) scale economies were chosen within the set $\{0.2, 0.5, 0.8\}$, but having $\alpha^H \geq \alpha^G$ since it is expected that larger and more fuel efficient carriers are employed on gateway arcs. In the second experiment, to analyze the influence of unitary operational costs in the design of the network, six different scenarios were constructed in which local (b^H) and global (b^G) unitary operational costs were chosen within the set $\{0.5, 1.0, 1.5\}$, but having $b^H \leq b^G$, since it is expected that international stopovers may involve performing customs, immigration and security checks. We adopted $\alpha^H = \alpha^G = 0.2$.

Tables 2.13 and 2.14 provide the results for variations in scale economies and unitary operational costs, respectively. The computational running times, the number of installed hubs and gateways, and hub and gateway arcs in the optimal solutions for the selected costs are reported. Table 2.13 and 2.14 also indicate in percentage for each example three other performance measures, the Passenger Mile per Direct Arc (PMDA), Passenger Mile per Hub Arc (PMHA) and Passenger Mile per Gateway Arc (PMGA). Passenger Miles per arc type are calculated by summing the multiplication of the number of passengers that traveled through an arc by the arc length, as demonstrated in equations (2.48), (2.49) and (2.50). By direct arc, we consider arc between

two non-hub nodes or between a hub and a non-hub node.

$$PMDA = \sum_{(u,v) \in A^L} d_{uv} \sum_{(i,j) \in W} f_{uv}^{ij} \quad (2.48)$$

$$PMHA = \sum_{(u,v) \in A^H} d_{uv} \sum_{(i,j) \in W} f_{uv}^{ij} \quad (2.49)$$

$$PMGA = \sum_{(u,v) \in A^G} d_{uv} \sum_{(i,j) \in W} f_{uv}^{ij} \quad (2.50)$$

Table 2.13: Optimal network under different economies of scale values

Instance	α^H	α^G	Time(s)	# Hubs	# Gateways	# Hub Arcs	# Gateway Arcs	PMDA	PMHA	PMGA
global-37	0.2	0.2	150	28	16	19	27	0.17	0.34	0.49
	0.5	0.2	171	28	19	10	35	0.28	0.20	0.52
	0.8	0.2	135	26	25	2	50	0.42	0.01	0.57
	0.5	0.5	163	28	20	16	39	0.28	0.22	0.50
	0.8	0.5	135	27	24	3	52	0.42	0.03	0.55
	0.8	0.8	147	27	23	15	46	0.43	0.06	0.51
global-43	0.2	0.2	694	32	17	25	27	0.19	0.37	0.44
	0.5	0.2	692	31	20	19	38	0.28	0.22	0.50
	0.8	0.2	676	28	25	4	51	0.44	0.02	0.54
	0.5	0.5	640	31	20	20	40	0.29	0.24	0.47
	0.8	0.5	646	28	24	6	54	0.45	0.03	0.52
	0.8	0.8	748	29	24	18	50	0.45	0.06	0.49
global-48	0.2	0.2	1447	35	17	34	28	0.22	0.41	0.37
	0.5	0.2	1137	34	22	23	44	0.32	0.25	0.43
	0.8	0.2	890	30	26	5	52	0.53	0.01	0.46
	0.5	0.5	1128	34	21	31	42	0.33	0.29	0.39
	0.8	0.5	873	31	24	11	54	0.53	0.04	0.43
	0.8	0.8	888	31	23	23	49	0.53	0.07	0.39

A detailed view of the results displayed in Table 2.13 shows a direct correlation between the optimal network and the relative difference in effectiveness for hubs and gateways. As the hub efficiency is lost, the investment priorities are re-routed through the international gateways. The same effect is observed on installation of hub and gateway arcs, when comparing to direct connections. In general, it is more attractive to select best global infrastructure and bring the international passengers close together using direct flights. It is important to recall that international demands can not be directly served, and therefore, as the gateways have their efficiency reduced, new attempts to improve the network cost structure are implemented, turning to hubs again as a cost-relieve device.

The important conclusion drawn here is that the intermediate network layer has its importance demonstrated specially for scenarios where a company is operating with old, not very effective international gateways like the ones available in on-development countries. Furthermore, ensuring the operational health and efficiency of both elements, hubs and gateways, is a task of capital importance to avoid unnecessary infrastructure mobilization. A careful planning of the installed capacity of these devices is advised,

and an up-to-date tracking of good operational metrics to select the proper moment of capacity expansion as well.

As displayed in Table (2.14), it seems that the sensitivity to reasonable changes in unitary costs is not as high, regarding the structure of the optimal network. Only slight variations on the network topologies could be found, typically on the number of hub and gateway arcs. This is expected as the unitary costs were augmented using a flat profile to avoid introducing any bias towards any special location. As such, more subtle effects might be expected.

Table 2.14: Optimal network under different unitary operational costs

Instance	b^H	b^G	Time(s)	# Hubs	# Gateways	# Hub Arcs	# Gateway Arcs	PMDA	PMHA	PMGA
global-37	0.5	0.5	325	28	16	19	27	0.05	0.47	0.49
	0.5	1.0	304	28	16	19	27	0.05	0.47	0.49
	0.5	1.5	243	28	16	19	27	0.05	0.47	0.49
	1.0	1.0	348	28	16	19	27	0.17	0.34	0.49
	1.0	1.5	242	28	16	19	27	0.17	0.34	0.49
	1.5	1.5	323	28	16	15	27	0.26	0.26	0.49
global-43	0.5	0.5	819	32	17	27	28	0.08	0.48	0.44
	0.5	1.0	691	32	17	27	28	0.08	0.48	0.44
	0.5	1.5	606	32	17	27	28	0.08	0.48	0.44
	1.0	1.0	661	32	17	25	27	0.19	0.37	0.44
	1.0	1.5	928	32	17	25	28	0.19	0.37	0.44
	1.5	1.5	838	32	17	21	28	0.27	0.29	0.44
global-48	0.5	0.5	1005	35	17	39	29	0.11	0.52	0.37
	0.5	1.0	744	35	17	38	28	0.11	0.52	0.37
	0.5	1.5	879	35	17	38	28	0.11	0.52	0.37
	1.0	1.0	1458	35	17	34	28	0.22	0.41	0.37
	1.0	1.5	925	35	17	34	28	0.22	0.41	0.37
	1.5	1.5	1162	35	17	30	28	0.30	0.33	0.37

However, a detailed inspection of the metrics PMDA and PMHA show, as in the first experiment, that the local infrastructure or intermediate network layer has its importance reduced as the transportation costs in hubs and gateways get closer. Please, recall once more that no impact is expected in PMGA as there is no other way to cope with the global demand components except by going through the gateways. Once again, the direct flights become more and more attractive as the hubs display poor performance, in a effort to fast connect the international passengers. The hub layer infrastructure, to be important for relieving transportation costs, must be kept in good health, operating at good and safe overhead levels.

After the discussion of aforementioned results, the fundamental lesson is: a three layer network displaying specific and well defined roles for hubs and gateways may be of strategical value, provided that the transportation costs and economies of scale are kept under severe control. The differentiation between local and global infra-structure may be the key to relieve transportation costs when old, overloaded gateways are used, but to properly take advantage of the expected savings, it is required to keep the local hubs in good operational health. Otherwise, its is probably better dodge the costly

stopovers at the hubs and feed the gateways using direct connections, returning to two layer network protocol.

2.5.2 An illustrative example

The problem proposed in this paper may act as a tool for providing insights into the current air network. In order to exemplify some practical benefits of the work developed here, the global-141 instance in setting I is chosen to be analyzed. Figure 2.2 illustrates the network obtained for this instance. The top 20 busiest airports in the world considering total passenger traffic and total international passenger traffic in 2015 are displayed in Tables 2.15 and 2.16, respectively (ACI 2016). The situation of the airports in the solution is highlighted in third column.

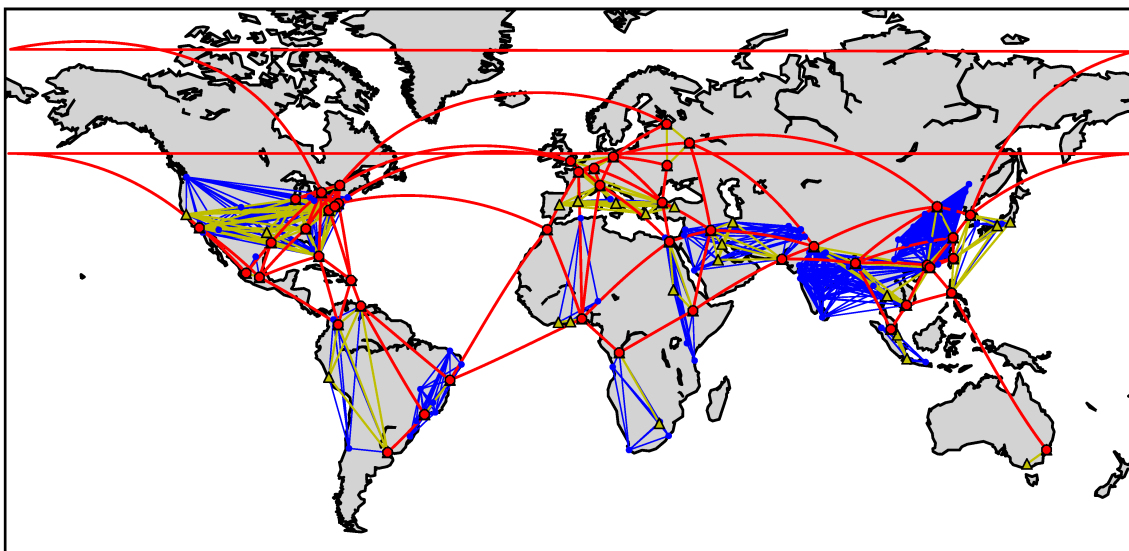


Figure 2.2: Optimal air network for instance global-141 setting I

Comparing the real scenario with the network proposed, some questionings may arise. Although Atlanta airport was considered the busiest airport in 2015, should it really be an active gateway in the solution? This airport has a strategic location from a domestic point of view, being within a two-hour flight of 80% of the United States population (ACI 2016). Nevertheless, ATL does not appear in international passenger traffic rank. In the same way, Beijing, Chicago, Los Angeles, Shanghai and Guangzhou airports appear as the top 20 busiest airports in Table 2.15 and also are active gateways in Figure 2.2, but they are not present in Table 2.16. Should these airports be considered as gateways or as hubs? On the other hand, London, Hong Kong, Paris, Istanbul and New York airports are active gateways in our solution, being present in both ranks.

Table 2.15: Total passenger traffic 2015

Rank	Airport City / Country / Code	Solution
1	Atlanta GA, US (ATL)	Gateway
2	Beijing, CN (PEK)	Gateway
3	Dubai, AE (DXB)	Not in data
4	Chicago IL, US (ORD)	Gateway
5	Tokyo, JP (HND)	Hub
6	London, GB (LHR)	Gateway
7	Los Angeles CA, US (LAX)	Gateway
8	Hong Kong , HK (HKG)	Gateway
9	Paris, FR (CDG)	Gateway
10	Dallas/Fort Worth TX, US (DFW)	Hub
11	Istanbul, TR (IST)	Gateway
12	Frankfurt, DE (FRA)	Not in data
13	Shanghai, CN (PVG)	Gateway
14	Amsterdam, NL (AMS)	Not in data
15	New York NW, US (JFK)	Gateway
16	Singapore, SG (SIN)	Hub
17	Guangzhou, CN (CAN)	Gateway
18	Jakarta, ID (CGK)	Hub
19	Denver CO, US (DEN)	Not a candidate
20	Bangkok, TH (BKK)	Hub

As the instances elaborated here contemplate the most populous cities, despite being well classified in both ranks, Dubai, Frankfurt and Amsterdam airports are not included in the instances. A similar situation is perceived with airports in Munich and Doha. They are in top 20 busiest international passenger traffic but they are not in the instance. In spite of being in data, Denver and Roma airports are not considered as gateway and hub candidates. This implies that instance creation is a point to be improved.

Dallas and Jakarta airports appear on the most-travelled airports in 2015 but they are not present in the rank when international passengers are considered. In the solution obtained for our model, they are active as hubs. On the contrary, although Tokyo, Singapore and Bangkok airports were considered as the most busiest airports of both total and only international passenger traffic, they are selected only as hubs in Figure 2.2. Of course the already existing infrastructure of these airports must be considered, but considering those cities the best transshipment localities in Asia from an economic point of view is a matter of discussion. The optimal solution obtained for our model activated Ho Chi Minh City (Vietnam) airport as gateways in this region. Also in Asia, Incheon, Taipei and Kuala Lumpur airports are the world busiest in terms of international passengers and are selected as gateways in Figure 2.2.

Table 2.16: Total international passenger traffic 2015

Rank	Airport City / Country / Code	Solution
1	Dubai, AE (DXB)	Not in data
2	London, GB (LHR)	Gateway
3	Hong Kong , HK (HKG)	Gateway
4	Paris, FR (CDG)	Gateway
5	Amsterdam, NL (AMS)	Not in data
6	Singapore, SG (SIN)	Hub
7	Frankfurt, DE (FRA)	Not in data
8	Incheon, KR (ICN)	Not in data
9	Bangkok, TH (BKK)	Hub
10	Istanbul, TR (IST)	Gateway
11	Taipei, TW (TPE)	Gateway
12	London, GB (LGW)	Gateway
13	Kuala Lumpur, MY (KUL)	Gateway
14	Madrid, ES (MAD)	Hub
15	Munich, DE (MUC)	Not in data
16	Doha, QA (DOH)	Not in data
17	Tokyo, JP (HND)	Hub
18	New York NW, US (JFK)	Gateway
19	Barcelona, ES (BCN)	Hub
20	Rome, IT (FCO)	Not a candidate

Despite being the fourteenth and the nineteenth world's busiest international passenger traffic airports, Madrid and Barcelona airports are proposed to be hubs in our solution. A similar question arises: Is Spain in a strategic place to concentrate international flows? The network drawn in Figure 2.2 proposed Berlin and The Ruhr airports, in Germany, as gateways. Would Germany not be a more strategic country to concentrate international flights?

In our solution, Lagos, in Nigeria, and Casablanca, in Morocco, are proposed as gateways in Africa. Considering that Africa occupies a strategic position on the globe, these two countries would be strategic points for concentration of international flow?

In Brazil, Sao Paulo and Salvador airports are activated as gateways. We know that Rio de Janeiro airport concentrates much more flights than Salvador airport. From a geographic point of view, should be better to concentrate international flights in Salvador?

The costs considered in this study are not real, the demand utilized is based on the population of the city (we ignore the nearby cities) and the algorithms developed are only able to solve instances up to 141 nodes until optimality. Despite these limitations, the solution drawn here could provoke insights and questionings in the way of air companies prioritize airports.

2.6 Conclusion and future research

In this paper, we studied a gateway hub location problem, which is a three-level hub location problem, considering domestic and international flows for design a global air network. We have proposed exact methods to solve the problem, based on a generalized Benders decomposition. Two features that greatly speed up the method have been proposed: a separation of SECs when the BMP solution results in a infeasible network and a repair procedure which allows to generate Benders optimality cuts from unbounded dual subproblems. Computational experiments showed the effectiveness of our algorithms, which significantly improve the solution time of the general purpose solver when solving large instance sizes. As future research, the difference in performance between algorithms Alg-2-v2-I and Alg-3-v2 needs to be investigated further.

The formulation can tackle instances containing up to 141 airports, which is still far from real-life problem sizes. Moreover, the costs and the demand matrix considered here are not real data. Nevertheless, our approach offers important questions about the way current global air network is designed. Some efforts should be directed to improve the elaboration of the instances and also, other techniques can be studied to enable determining optimal solutions for larger instances.

Chapter 3

Hub Network Design with Flexible Routes

Abstract

This paper introduces a hub network design problem with flexible routes. The routes are flexible in the sense that each route may contain a mix of hub and non-hub nodes. We assume that commodity transfers can only be performed at hubs and that transportation costs are flow-dependent, which implies that instead of imposing fixed discount factors to represent economies of scale, these economies of scale stem from the transport technology chosen to operate the routes. This makes the problem more difficult to formulate and solve than classical hub network design problems. We propose a mixed integer programming formulation and two metaheuristics based on the adaptive large neighborhood search paradigm to solve the problem. We report the results of computational experiments to assess the performance of the formulation and algorithms on a set of benchmark instances.

Key words: Scale economies; Hub-and-spoke network; Flexibility; Routing.

3.1 Introduction

Hubs are special facilities that act as transshipment or flow consolidation points in many-to-many transport applications. *Hub location problems* (HLPs) are a class of strategic logistics planning problems in which hub facilities must be located while demand nodes must be allocated to hubs in order to route the traffic between multiple origin-destination pairs. In hub-and-spoke systems, instead of directly connecting each origin-destination pair, flows from the same origin but with different destinations are consolidated at the hubs with other flows that have different origins but the same destination. Consolidating demands at hub terminals allows transportation carriers to

use larger vehicles to exploit economies of scale and achieve lower transportation costs. Given their wide presence in logistics systems, HLPs have been the focus of many researchers. Readers may refer to Alumur and Kara (2008), Campbell and O’Kelly (2012), Farahani et al. (2013) and Contreras (2015) for recent surveys on hub location.

Classic HLPs relies on three main assumptions. First, demand flows have to be routed through one or at most two hubs, implying that a non-hub node is directly connected to at least one hub facility. Second, the hub network is considered to be fully interconnected. Third, a constant discount factor representing economies of scale is applied to the unit transportation cost of inter-hub connections. In several applications, these assumptions are reasonable and provide a good approximation of reality; in others, they may lead to suboptimal solutions.

Considering a fixed discount factor to represent economies of scale on inter-hub links may be more suitable for applications in which links between hubs are associated with faster transportation modes. Nonetheless, it may be an oversimplification for applications in which economies of scale are a consequence of the bundling of flows on hub arcs. This simplification may lead to solutions that grant a discount factor to inter-hub arcs only, even though these connections may carry considerably less flow than non-hub links. Consequently, in many cases, this may lead to miscalculations of the total network cost as well as to erroneous decisions of hub locations and non-hub allocations.

Several modeling approaches have been introduced to overcome this simplification. For instance, O’Kelly and Bryan (1998) and de Camargo et al. (2009) approach this issue by proposing formulations in which the economies of scale are modeled as flow dependent functions on the inter-hub arcs in a fully interconnected hub network. Nonetheless, they have neither consider the selection of vehicles nor how they operate. Kimms (2006) argue that economies of scale stem from the transportation technology selected to be used in the system. He propose three multiple allocation p -hub median problems with direct service and with fixed and variable costs. The goal is to determine the optimal number of vehicles to be used on each arc of a fully interconnected hub network. Considering flow dependent transportation costs based on modular arc costs, Tanash et al. (2017) study a modular hub location problem, which bears similarity to the work of Kimms (2006). The total transportation cost is calculated not in terms of the per unit flow cost, but with respect to the number of vehicles used on each arc. Although these models design the hub-and-spoke network and assign resources to arcs aiming to minimize the total cost of transporting flows, the routing of the resources is not taken into account.

The need to jointly consider location and routing decisions arises naturally in applications that have nodes with insufficient demand to justify direct connections with

the hubs such as in many-to-many flow transportation systems. In these cases, besides the location of hubs and routing of flows through the network, local tours to serve and connect the non-hub nodes to the installed hubs need to be generated so that flows from many origins to many destinations can be transported at a minimal cost. This gives rise to the so-called *many-to-many hub location-routing problem* (MMHLRP) introduced by Nagy and Salhi (1998). The authors study a MMHLRP that impose capacity and maximum distance constraints to the local tours of a homogeneous fleet, and fixed costs to establishing hubs.

Wasner and Zäpfel (2004) consider a MMHLRP arising in a postal service application in which non-hub nodes are directly linked by local tours while all inter-hub flows is transferred through a central hub. Another variant of a MMHLRP is presented in Çetiner et al. (2010) in which an iterative two-stage solution procedure is described. In the first stage, hub location and non-hub node (multiple) assignments decisions are fixed whereas in the second stage a traveling salesman problem is solved for each installed hub. Camargo et al. (2013) and Rodríguez-Martín et al. (2014) study single allocation variants of the problem with bounded tour length to ensure service quality and present exact solution algorithms for solving them. Kartal et al. (2017) propose three different mixed integer programs (MIP) and two heuristic approaches to solve a single allocation p -hub median location and routing problem with simultaneous pickup and delivery. Karimi (2018) study a version of a single allocation hub location routing problem that considers hub and vehicle capacities. Kartal et al. (2019) present a p -hub center and routing network design problem which locates p -hubs, allocates non-hub nodes to the installed hubs, and generates vehicle routes for each installed hub in a way that the maximum travel time between all origin-destination pairs is minimized.

Most of the work on MMHLRPs assume that routing decisions happen only at the access level, and that the hub network is fully interconnected with one vehicle serving each pair of hubs. Nonetheless these assumptions may not lead to the most cost efficient topologies, since they may result in unnecessary routes. To prevent that, Lopes et al. (2016) investigate a MMHLRP with an incomplete hub network. Nodes are partitioned into tours with exactly one hub each, while creating an extra tour interconnecting all hubs. Hubs are restricted to serve a predetermined number of non-hub nodes, whereas the many-to-many flow decisions are implicitly made. Neither flow capacity on the tours or hubs nor maximal length on the routes are imposed.

In this paper we introduce a new general class of MMHLRPs denoted as *hub network design problems with flexible routes* (HNDPs). Besides considering hub location decisions and flow dependent transportation costs, HNDPs determine itineraries for the selected vehicles. One of the major advantages of HNDPs over classical models is that no particular topological structure, such as cycles (Contreras et al. 2017), stars (Labbé

and Yaman 2008), trees (de Sá et al. 2013), lines (Martins de Sá et al. 2015), incomplete (de Camargo et al. 2017) or fully interconnected hub networks (O’Kelly 1986, Skorin-Kapov et al. 1996, Hamacher et al. 2004), is assumed a priori. The network topology is endogenously determined, being induced from the involved costs. This ensures that vehicles will be used more efficiently and economically, and that economies of scale will be obtained according to the transport technology chosen to operate the routes and their actual utilization.

HNDPs consist of opening a set of hub nodes and covering a set of nodes with a set of heterogeneous vehicle routes. Vehicles are assumed to start their route on any node of the graph, serve some nodes, and then return to the same initial node. Routes are not forced to visit hub nodes. Load capacity constraints and a maximum time for completing the routes must be respected. There are installation costs for establishing a hub and using a vehicle, while fixed operational costs are incurred every time a vehicle moves through an arc. We assume that all demand flows have to be routed with the selected vehicle routes. Instead of applying the traditional fixed discount factor to inter-hub links, it is assumed that economies of scale are a direct consequence of the vehicles assigned to the routes and not of the arc type being used. No restriction of any kind is imposed on the path of a demand flow. That is, demand flow can be routed from its origin to its destination by using a single vehicle route or by using a set of vehicle routes. However, flow transfers between vehicles are allowed only at hub nodes. Here, a hub works exclusively as a transshipment facility to transfer flows from one vehicle to another. There is no limit on the number of intermediate nodes (hub and non-hub) a path of a demand flow can have.

Figure 3.1 illustrates a solution of a HNDP with four hub nodes, 13 non-hub nodes and seven vehicle routes. Note that there are routes that start and end at non-hub nodes, and that, even though there are multiple routes passing through the same non-hub node, there is no flow exchange on non-hub nodes. It is also worth observing that demand flows follow the direction of the routes, i.e. an origin-destination demand (i, j) may have a longer path than its counterpart, demand (j, i) .

HNDPs greatly differs from previously studied MMHLRPs. They assume a given exogenous network topology and a homogeneous fleet with routes that start and end at the hubs. Normally vehicles can only pass a limited number of times through each demand node, while the path of a demand flow is restricted by the prescribed network topology. Most important of all, MMHLRPs completely disregard that the economies of scale derive from the technology chosen to operate the routes and its utilization. To better highlight the similarities and dissimilarities between the HNDP and other MMHLRP, Table 3.1 summarizes their main characteristics.

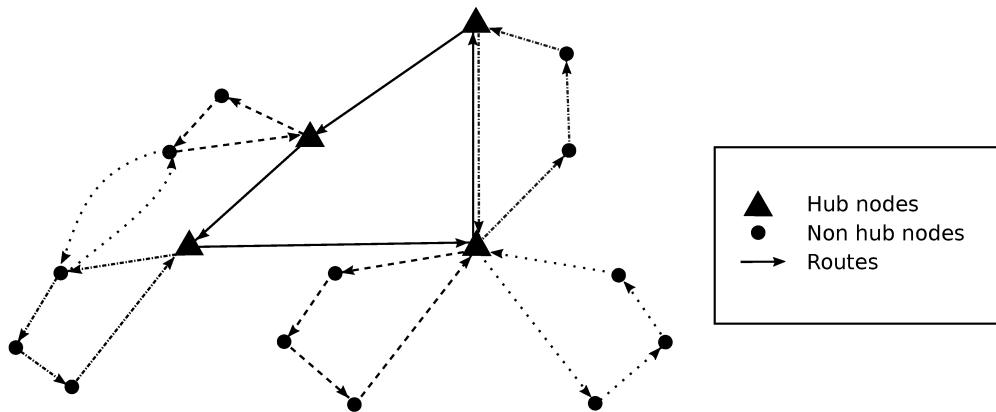


Figure 3.1: Example of a hub network with flexible routes

Table 3.1: Related literature to the HNDP

Research	Components						
	Vehicle Capacity	Route Length	Hub Network	Fleet Type	Allocation	Vehicle Cost	Hub Cost
Nagy and Salhi (1998)	✓	✓	FI	HO	SA		✓
Wasner and Zäpfel (2004)	✓	✓	FI	HO	MA		
Çetiner et al. (2010)		✓	FI	HO	MA		
Camargo et al. (2013)		✓	FI	HO	SA	✓	✓
Rodríguez-Martín et al. (2014)		✓	FI	HO	SA		
Lopes et al. (2016)		✓	PI	HO	SA		
Kartal et al. (2017)			FI	HO	SA		
Karimi (2018)	✓	✓	FI	HO	SA	✓	✓
Kartal et al. (2019)			FI	HO	SA		
Current research	✓	✓	PI	HE	MA	✓	✓

FI - Full Interconnected; PI - Partial Interconnected; SA - Single Allocation; MA - Multiple Allocation. HO - Homogeneous fleet; HE - Heterogeneous fleet.

To the best of our knowledge, no work in the literature considers simultaneously hub location and vehicle routing decisions with capacity constraints, without assuming a constant discount factor to represent economies of scale, and without imposing any restriction on the network topology and on the paths used to route demand flows. Our goal is to propose a more general framework capable of designing hub networks that employ several vehicle routes to transport the demands via a heterogeneous fleet. These assumptions render the HNDP more convoluted having a much higher complexity than similar works in the literature.

The HNDP have numerous applications in ground, air and maritime freight transportation networks. An example of a concrete application arises in liner shipping in which companies usually operate a set of cyclic routes to provide shipping services. In this case, hub facilities correspond to ports where transshipment of cargo can occur and non-hub nodes can be seen as ports where transshipment is not allowed. Demand flow represents requests from customers for shipment of containers. Vessel routes may

or may not contain hubs on them and there is no restriction in the path chosen to serve the requests. The goal of liner shipping companies is to design vessel routes at a minimal cost to satisfy the demand. Although shipping service routes cannot be reshuffled overnight, several circumstances, e.g. changes in demand, increase in fuel price and modification in ship capacity, drive a liner shipping company to adjust its service routes and ship deployment on a small scale from time to time (see, for instance, Gelareh and Pisinger 2011, Reinhardt and Pisinger 2012, Song and Dong 2013).

The main contribution of this paper is threefold. First, we introduce a new class of HLPs that encompass both strategic and tactical planning decisions. The strategic decisions consist of locating hub facilities and defining how many vehicles of each type should be used. Given that transforming a facility into a hub and owning a vehicle involve a significant capital investment, the decisions taken at this level are very important. The tactical decisions assign vehicles to arcs so that vehicle routes can define paths to route all demand flows. While these decisions can be taken separately, once strategic decisions are made, they may negatively impact the decisions of the tactical level, resulting in suboptimal solutions. Second, we propose a MIP formulation to solve the problem. Third, because the proposed MIP is too large to be solved in a timely manner by general purpose solver, we also develop two metaheuristics based on an adaptive large neighborhood search (ALNS). To evaluate the efficiency and limitations of our algorithms, extensive computational experiments are performed on instances with up to 50 nodes.

The remainder of this document is organized as follows. Section 3.2 provides a formal definition of the problem and presents the mathematical formulation. Section 3.3 describes in detail the two ALNS metaheuristics. Computational results are reported in Section 3.4, followed by conclusions in the last section.

3.2 Problem definition and formulation

Let $G = (N^O, A^O)$ be a directed graph, with the node set $N^O = \{0, \dots, n\}$ and the arc set $A^O = \{(i, j) : i, j \in N^O; i \neq j\}$. Node 0 is a fictitious depot, where vehicles must start and end their routes. The sets $N = \{1, \dots, n\}$ and $A = \{(i, j) : i, j \in N; i \neq j\}$ are the node and arc sets, respectively, without considering the depot. We define $H \subseteq N$ as the hub candidate set and h_k as the fixed cost incurred for installing a hub in node $k \in H$. There is a demand flow $w_{ij} \geq 0$ that needs to be routed from $i \in N$ to $j \in N$. We assume that w_{ij} can be split, i.e., can be routed using more than one path. We also consider that more than one vehicle can be used to serve a demand in a specific path. However, demand flows can change vehicles only at hub nodes. For example,

consider the path $1 - 2 - 3 - 4$ to serve demand w_{14} . If vehicle 1 operates arc $(1, 2)$ and vehicle 2 operates arcs $(2, 3)$ and $(3, 4)$, a hub must be installed at node 2. These features substantially increase the complexity of the design of the model and of the solution procedures.

We consider an heterogeneous fleet of vehicles in which each vehicle type has a different capacity, speed, fuel consumption, and other specific characteristics. We denote by \mathcal{R} the set of all vehicle types and use the index r to represent a particular vehicle type. For each $r \in \mathcal{R}$, τ^r denotes the capacity of a vehicle of type r and for each arc $(k, m) \in A$, t_{km}^r denotes the time it takes a vehicle of type r to traverse arc (k, m) . The fixed cost related to the acquisition of a vehicle type $r \in \mathcal{R}$ is denoted by a^r . We associate the fixed cost c_{km}^r with vehicle type $r \in \mathcal{R}$ traversing arc $(k, m) \in A$. Finally, let T be an upper bound on the length of the routes.

Let $\rho(r)$ be a function that returns the number of available vehicles of type r , and R be the total number of available vehicle types. Let $P = \{1, \dots, \rho(1), \rho(1) + 1, \dots, \rho(1) + \rho(2), \dots, \rho(1) + \dots + \rho(R)\}$ be the set of vehicles, and index p be used to represent a specific vehicle. Let $\phi(p)$ be a function that returns the type of vehicle p . For example, if we have three types of vehicles ($R = 3$), having three vehicles for type 1 ($\rho(1) = 3$), four vehicles for type 2 ($\rho(2) = 4$), and two vehicles for type 3 ($\rho(3) = 2$), then set $P = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. For this case, vehicles 1, 2, 3 are of type 1 ($\phi(1) = \phi(2) = \phi(3) = 1$), vehicles 4, 5, 6, 7 are of type 2 ($\phi(4) = \phi(5) = \phi(6) = \phi(7) = 2$), and vehicles 8 and 9 are of type 3 ($\phi(8) = \phi(9) = 3$).

We now present an MIP formulation for the HNDDP. Our formulation combines decisions of two problems: 1) variables \mathbf{y} and \mathbf{f} related to the hub location problem, and 2) variables \mathbf{q} , \mathbf{x} and \mathbf{z} associated to the vehicle routing problem. Let $y_k \in \{0, 1\}$ be equal to one if and only if a hub is located at node $k \in H$. Let $f_{ijkm}^p \geq 0$ denote the percentage of demand w_{ij} passing through arc $(k, m) \in A$ in vehicle $p \in P$. Further, let $q^p \in \{0, 1\}$ be equal to one if and only if vehicle $p \in P$ is utilized. Also, let $x_{km}^p \in \{0, 1\}$ be equal to one if and only if vehicle $p \in P$ passes through arc $(k, m) \in A^0$, and $z_k^p \in \{0, 1\}$ be equal to one if and only if vehicle $p \in P$ passes through node $k \in N$. Using these variables, the HNDDP can be formulated as follows:

$$\min \sum_{k \in H} h_k y_k + \sum_{p \in P} a^{\phi(p)} q^p + \sum_{p \in P} \sum_{(k, m) \in A} c_{km}^{\phi(p)} x_{km}^p \quad (3.1)$$

$$\text{s.t.} : \sum_{p \in P} \sum_{(i, m) \in A} f_{ijim}^p = 1 \quad \forall (i, j) \in W \quad (3.2)$$

$$\sum_{p \in P} \sum_{(k, m) \in A} f_{ijkm}^p = \sum_{p \in P} \sum_{(m, k) \in A} f_{ijmk}^p \quad \forall (i, j) \in W, k \in N : k \neq i, k \neq j \quad (3.3)$$

$$\sum_{p \in P} \sum_{(k, j) \in A} f_{ijkj}^p = 1 \quad \forall (i, j) \in W \quad (3.4)$$

$$f_{ijkm}^p \leq x_{km}^p \quad \forall (i, j) \in W, (k, m) \in A, p \in P : k \neq j, m \neq i \quad (3.5)$$

$$\sum_{\substack{(m,k) \in A \\ m \neq j}} f_{ijmk}^p - \sum_{\substack{(k,m) \in A \\ m \neq i}} f_{ijkm}^p \leq y_k \quad \forall (i, j) \in W, k \in N, p \in P : k \neq i, k \neq j \quad (3.6)$$

$$\sum_{(k,m) \in A^0} x_{km}^p = \sum_{(m,k) \in A^0} x_{mk}^p \quad \forall k \in N, p \in P \quad (3.7)$$

$$\sum_{(0,m) \in A^0} x_{0m}^p = q^p \quad \forall p \in P \quad (3.8)$$

$$\sum_{(k,0) \in A^0} x_{k0}^p = q^p \quad \forall p \in P \quad (3.9)$$

$$\sum_{(k,m) \in A} t_{km}^{\phi(p)} x_{km}^p \leq q^p T \quad \forall p \in P \quad (3.10)$$

$$\sum_{(k,m) \in A} x_{km}^p = z_k^p \quad \forall k \in N, p \in P \quad (3.11)$$

$$\sum_{(k,m) \in A} x_{km}^p = z_m^p \quad \forall m \in N, p \in P \quad (3.12)$$

$$\sum_{u \in S} \sum_{v \in N^0 \setminus S} x_{uv}^p + \sum_{u \in S} \sum_{v \in N^0 \setminus S} x_{vu}^p \geq z_k^p \quad \forall k \in S, S \subset N, |S| \geq 2, p \in P \quad (3.13)$$

$$x_{km}^p \leq q^p \quad \forall p \in P, (k, m) \in A \quad (3.14)$$

$$\sum_{(i,j) \in W} w_{ij} f_{ijkm}^p \leq \tau^{\phi(p)} x_{km}^p \quad \forall (k, m) \in A, p \in P \quad (3.15)$$

$$f_{ijkm}^p \geq 0 \quad \forall (i, j) \in W, (k, m) \in A, p \in P : k \neq j, m \neq i \quad (3.16)$$

$$y_k \in \{0, 1\} \quad \forall k \in H \quad (3.17)$$

$$x_{km}^p \in \{0, 1\} \quad \forall (k, m) \in A^0, p \in P \quad (3.18)$$

$$q^p \in \{0, 1\} \quad \forall p \in P \quad (3.19)$$

$$z_k^p \in \{0, 1\} \quad \forall k \in N, p \in P. \quad (3.20)$$

The objective function (3.1) minimizes the total cost of designing the network and routing the vehicles. The first term of the objective function denotes the cost of installing hubs. The second term represents the cost of acquiring the vehicles, while the third term calculates the vehicle routing cost of traversing the arcs. Constraints (3.2)-(3.4) are flow balancing equations. Constraints (3.5) ensure that if any flow passes through an arc in a vehicle, then, this vehicle passes through such arc. Constraints (3.6) guarantee that if a demand flow changes vehicles in a node, then, such node must be a hub. Constraints (3.7) balance the vehicle routes, i.e. if a vehicle arrives at a node, it must leave that node. Constraints (3.8) and (3.9) guarantee that the acquired vehicles start and end their routes at the fictitious depot. Constraints (3.14)

ensure that a vehicle passes through an arc only if the vehicle is acquired. Constraints (3.15) are the capacity constraints, i.e. the total flow that passes through an arc in a vehicle must respect the capacity of that arc. Constraints (3.10) impose a travel time limit on each vehicle route. Constraints (3.11) and (3.12) activate variables z , i.e. if a vehicle traverses an arc, the nodes of that arc are in the vehicle route. Constraints (3.13) are the well-known sub-tour elimination constraints (SECs) which ensure that for each route sub-cycles will not be formed. Constraints (3.16)-(3.20) are the standard non-negativity and integrality conditions.

Let $\rho^s(r)$ and $\rho^e(r)$ be the first and the last indices of vehicle of type r , respectively. In order to reduce the symmetry of our formulation (3.1)-(3.20), we add the following constraints:

$$\sum_{(k,m) \in A} t_{km}^r x_{km}^p \leq \sum_{(k,m) \in A} t_{km}^r x_{km}^{p-1} \quad \forall r \in \mathcal{R}, p \in P : p > \rho^s(r), p \leq \rho^e(r) \quad (3.21)$$

$$q^p \leq q^{p-1} \quad \forall r \in \mathcal{R}, p \in P : p > \rho^s(r), p \leq \rho^e(r). \quad (3.22)$$

Constraints (3.21) ensure that longer routes will have vehicles indexed with smaller values for each type of vehicle. Constraints (3.22) guarantee that vehicles with a smaller index will be activated before vehicles with higher indices for each type of vehicle.

3.3 Solution algorithms

We next present two metaheuristics based on a ALNS framework to find feasible solutions for the HNDP in reasonable computation times. The ALNS framework was originally devised by Ropke and Pisinger (2006) for the pickup and delivery problem with time windows. It can be seen as an extension of the large neighborhood search framework of Shaw (1998) but with an adaptive layer. Our first algorithm, denoted as TDALNS, follows a top down approach, in which the hub configuration is defined first followed by the construction of the routes. The second algorithm, denoted as BUALNS, follows a bottom up approach, in which the routes are built before the hub configuration is established.

Both algorithms are iterative procedures in which at every iteration, two main phases are performed. The first phase selects a *destroy* operator to remove n^d nodes from the routes, while the second phase chooses a *repair* operator to reconstruct the solution. Operators are selected according to an adaptive probabilistic mechanism derived from the score of each operator. Operators that have successfully found new improving solutions have a higher score and, therefore, a higher probability to be chosen again.

A general overview of TDALNS and BUALNS heuristics are depicted in Algorithms 4 and 5, respectively. An initial feasible solution s is first constructed. A new solution s^{new} is then obtained by sequentially applying on the current solution s destroy and repair operators. In the TDALNS, whenever ω^{ih} iterations have been performed, a hub configuration operator is chosen to modify the current network before applying the destroy and the repair operators. In the BUALNS, the hub configuration is the last decision to be fixed, which is done at the same time the feasibility of the solution is analyzed. If the new solution is disconnected, a connection procedure is called to change the network.

In the TDALNS, we verify if the network is connected with respect to the hub configuration. In our case, the network is sometimes strongly connected, but there may be paths in which the transshipment occurs at non-hub nodes, thus violating our assumption that flow transfers between vehicles can only occur at hub nodes. Recall that a flow can pass from arc (k, m) to arc (m, l) only if the same vehicle passes through both arcs or if node m is a hub. Therefore, if the current set of open hubs does not guarantee the feasibility of the network, the solution is rejected. Otherwise, we verify if there is enough capacity in the selected vehicle routes to serve all demand pairs. For the BUALNS, once the network is connected we call a procedure that defines the hub configuration and analyzes the capacity feasibility at the same time. In both algorithms, if there is not enough allocated capacity on the vehicle route to serve all demand pairs, the solution is dynamically penalized by taking into account its capacity violation. The algorithms terminate when a limit on the maximum number of iterations is reached.

The algorithm always accepts a better solution, whereas a worse solution can be accepted depending on a simulating annealing criterion. Infeasible solutions can also be accepted. We note that there is no guarantee that the objective value of an infeasible solution will always be better than the objective value of an optimal solution. For this reason, at each iteration, we keep the best solution in general (s^*) and the best feasible solution (s^{f*}).

After destroying and repairing a solution, the resulting network may be infeasible for three different reasons. First, the network can be disconnected. In that case, a procedure is called to make it connected. Second, the network can be connected but to serve all the demand pairs some non-hub nodes may be used as transshipment points, which violates our assumption that flows can be transferred from one vehicle to another only at hub nodes. This situation may arise only in the TDALNS (and when it does, the solution is discarded). In the BUALNS, the hub configuration is made in a way that the feasibility from this point of view is always guaranteed. Third, the allocated capacity may not be enough to serve all the demand pairs. When this happens, the

solution is penalized and it can be accepted or not according to the simulated annealing criterion. The next subsections describe in detail all the procedures used in Algorithms 4 and 5. The two metaheuristics contain a high level of sophistication and complication. We could not do less than this because of the complexity of the problem.

Algorithm 4 Basic steps of TDALNS

```

s ← InitialSolution
s* ← s
sf* ← s
InitializeScores( $\pi^D, \pi^R, \pi^H$ )
ih ← 0, is ← 0
repeat
  if ih =  $\omega^{ih}$  then
    Oh ← ChooseHubConfigurationOperator(H,  $\pi$ )
    snew ← HubConfigurationOperators(Oh, s)
    ih ← 0
  else
    ih ← ih + 1
  end if
  Od ← ChooseDestroyOperator(D,  $\pi$ )
  Or ← ChooseRepairOperator(R,  $\pi$ )
  snew ← DestroyAndRepairOperators(Od, Or, snew)
  if snew is disconnected then
    snew ← ConnectNetwork(snew)
  end if
  if snew is connected from a hub configuration point of view then
    (snew, feasibleSolution, hubFeasibility) ← AnalyzeCapacityFeasibility(snew)
    if feasibleSolution = false then
      if hubFeasibility = true then
        Penalize(OF(snew))
      end if
    end if
    if feasibleSolution = true or hubFeasibility = true then
      if snew satisfies the acceptance criterion then
        s ← snew
      end if
      if OF(snew) < OF(s*) then
        s* ← snew
      end if
      if feasibleSolution = true then
        if OF(snew) < OF(sf*) then
          sf* ← snew
        end if
      end if
    end if
  end if
  if is =  $\omega^{is}$  then
    UpdateScores( $\pi^D, \pi^R, \pi^H$ )
    is ← 0
  else
    is ← is + 1
  end if
until {the stopping condition is met}

```

3.3.1 Initial solution

To construct an initial feasible solution, we divide the nodes into clusters by solving a p -median problem. In this problem, central nodes correspond to facilities and non-central nodes correspond to non-facility points. The objective is to minimize the sum of the distances between the chosen central nodes and the nodes located to each of them. This problem is formulated as an MIP and solved with CPLEX. Each cluster has then its central node with its respective assigned non-central nodes. For each cluster, we find a Hamiltonian cycle of minimal cost. To construct Hamiltonian cycles, we used the Concorde solver (Applegate et al. 2012). We also create a simple route for every pair of clusters that connects their central nodes. If the duration of any tour is greater than the maximum duration allowed, the tour is split into feasible tours. Even though we assign the largest vehicle type to perform all the tours, there is no guarantee that the network will have enough capacity to serve all the demand pairs. To define the hub configuration and to verify if any route needs to be duplicated, we solve an auxiliary mixed integer problem that we denote as the *routing flow problem* (RFP).

The RFP determines which hubs should be opened and whether the initial solution is feasible or if it would be necessary to add capacity to the network by duplicating some vehicle routes. Let \mathcal{P} be the set of all routes in a solution. We redefine variables f considering only the arcs that exist in the solution network. In other words, arc $(k, m) \in A^p$ is considered if there is any $p \in \mathcal{P}$ with $x_{km}^p = 1$. Let q^p be a decision variable to determine the number of times route p will be used, and let \mathcal{C}^p be a parameter that represents the total cost of route p . Let $c_{km}^{\phi(p)}$ be the estimated cost of transporting a unit flow through the arc (k, m) of vehicle p , i.e. $c_{km}^{\phi(p)} = \mathcal{C}^p / (d_{km} \times \tau^{\phi(p)})$. Using these variables, we formulate the RFP as the following MIP:

$$\min \sum_{k \in N} h_k y_k + \sum_{p \in \mathcal{P}} \mathcal{C}^p q^p + \sum_{p \in \mathcal{P}} \sum_{(i,j) \in W} \sum_{(k,m) \in A^p} c_{km}^{\phi(p)} f_{ijkm}^p \quad (3.23)$$

$$\text{s.t.} \sum_{p \in \mathcal{P}} \sum_{(i,m) \in A^p} f_{ijim}^p = 1 \quad \forall (i, j) \in W \quad (3.24)$$

$$\sum_{p \in \mathcal{P}} \sum_{(k,m) \in A^p} f_{ijkm}^p = \sum_{p \in \mathcal{P}} \sum_{(m,k) \in A^p} f_{ijmk}^p \quad \forall (i, j) \in W, k \in N : k \neq i, k \neq j \quad (3.25)$$

$$\sum_{p \in \mathcal{P}} \sum_{(k,j) \in A^p} f_{ijkj}^p = 1 \quad \forall (i, j) \in W \quad (3.26)$$

$$\sum_{\substack{(u,k) \in A^p \\ u \neq j}} f_{ijuk}^p - \sum_{\substack{(k,v) \in A^p \\ v \neq i}} f_{ijkv}^p \leq y_k \quad \forall (i, j) \in W, k \in N, p \in \mathcal{P} : k \neq i, k \neq j \quad (3.27)$$

$$\sum_{(i,j) \in W} w_{ij} f_{ijkm}^p \leq \tau^{\phi(p)} q^p \quad \forall (k, m) \in A^p, p \in \mathcal{P} \quad (3.28)$$

$$f_{ijkm}^p \geq 0 \quad \forall (i, j) \in W, (k, m) \in A^p, p \in \mathcal{P} : k \neq j, m \neq i \quad (3.29)$$

$$q^p \geq 0 \quad \forall p \in \mathcal{P} \quad (3.30)$$

$$y_k \in \{0, 1\} \quad \forall k \in N. \quad (3.31)$$

The objective function minimizes the cost of opening hubs, activating routes and routing demand flows. Constraints (3.24)-(3.28) are equivalent to constraints (3.2)-(3.4), (3.6), and (3.15), respectively. Constraints (3.29)-(3.31) show the variable domains. After solving the RFP, if a route is used more than once, i.e., $q^p > 1$, we create a copy of this route and add it to the set \mathcal{P} . If no flow passes through a route, then this route is removed from the set \mathcal{P} .

3.3.2 Destroy operators

The destroy phase consists of either removing n^d nodes from their respective routes or deleting r^d routes from the current solution. Each operator uses a different metric to choose the nodes/routes to be destroyed. The nodes/routes are always ordered from the best to the worst according to the operator metric. To introduce some randomization, instead of picking the best element (the one in the first position), we randomly choose a number $\sigma \in [0, 1)$, and then select the closest position to the σ^p value. The parameter p controls the desired level of diversification. With $p = 1$, the metric is ignored and the choice is completely random. With $p = \infty$, the best element is always chosen. As proposed by Shaw (1997), we use $p = 4$. The six destroy operators are:

- **Choose more expensive routes** (D^1): For each route r , we calculate an average cost per unit and per distance: $\mathcal{C}^r / (d^r \times \tau^r)$, in which \mathcal{C}^r , d^r and τ^r are the total cost, total distance and capacity of the route r , respectively.
- **Choose shortest routes** (D^2): The routes are ordered in a non-decreasing order with respect to their length.
- **Randomly choose nodes** (D^3): There is no metric in this operator. A node is randomly selected and removed from all the routes it belongs to.
- **Unit cost greedy savings node selection** (D^4): For each candidate node i , we define $\beta(i, r)$ as the difference between the total cost of route r with and without node i . The nodes are always removed from all the routes they belong to. For this operator, the metric of each node is calculated as $\sum_{r \in \mathcal{P}: i \in r} \beta(i, r)$, where the nodes are sorted in non-increasing order.

- **Shaw removal distance** (D^5): This operator and the following one are based on the Shaw removal heuristic used by Shaw (1997) and Ropke and Pisinger (2006). They remove nodes that are similar in some aspect. A route is first randomly chosen and then, a node k of such route is randomly chosen to be removed from this and other routes it may belong to. Then, the nodes are sorted, in relation to node k , from the closest to the furthest one. To introduce some randomization, distances are perturbed by a factor within the range $[0.80, 1.20]$.
- **Shaw removal demand** (D^6): A route is first randomly chosen and then, a node k of this route is randomly chosen to be removed from this and other routes it may belong to. Then, for each node $i \neq k$, we calculate the total demand between such nodes as $w_{ik} + w_{ki}$. The nodes with the largest values have a larger probability to be removed. To introduce some randomization, demands are perturbed by a factor within the range $[0.80, 1.20]$.

3.3.3 Repair operators

The repair operators insert the removed nodes into existing routes or generate new routes when the insertion is not possible. The operators are also selected according to the values of σ^p and are inspired by the basic greedy heuristic proposed by Shaw (1997).

- **Greedy distance route insertion** (R^1): We define Δd_{ir} as the change in the distance of route r incurred by inserting node i into a position that increases the distance the least. If it is not possible to insert node i into route r , we set $\Delta d_{ir} = \infty$. We randomly select one node i between the nodes that have been removed in the previous phase. For this node i , the routes are ordered from the lowest to the highest Δd_{ir} values. The route that is at the position closest to σ^p is chosen to be the new route of i . Once the route is chosen, node i is inserted in the position that increases the duration of the route the least. During the insertion of a node, we recalculate Δd_{ir} for all nodes that have not been reinserted yet.
- **Greedy demand insertion** (R^2): For each node removed i and for each existing route r , we calculate the change in the distance incurred by inserting i into position p of route r , Δd_{irp} . For each node i and route r , we calculate the demand between i and all the nodes belonging to r , D_{ir} . We randomly select node i among the nodes that have been removed in the previous phase. Then, for this node i , the pair route-position (r, p) is sorted in non-increasing order of D_{ir} and non-decreasing order of Δd_{irp} . The idea is to try to insert node i into

a route that contains nodes having the largest demand flow to and from node i . The route-position (r, p) that is the closest to the σ^p position value is chosen. During the insertion of a node, we recalculate Δd_{irp} and D_{ir} for all nodes that have not been reinserted yet.

- **Greedy node selection (R^3):** We define Δd_{ir} as the change in the distance of route r incurred by inserting node i in a position that increases the distance the least. If it is not possible to insert i into route r , we set $\Delta d_{ir} = \infty$. We define c_i as the "cost" of inserting node i at its overall best position, such as $c_i = \min_{r \in R} \{\Delta d_{ir}\}$. Finally, we sort the nodes that need to be reinserted from the smallest to the highest value of c_i . We first choose a node to be reinserted according to σ^p value, and insert it at its minimum cost position. During the insertion of a node, we recalculate c_i for all nodes that have not been reinserted yet.

After executing the repair operator, we verify if the capacity incident to any node is less than the incoming/outgoing flow of that node. If so, we try to insert the node in a route according to repair operator R^1 . If it is not possible, we store such node in a list L . After verifying the capacity incident to each node, if there is at least one node in L , we generate a Hamiltonian cycle of minimal cost containing the nodes in L , splitting it if necessary. Every time we have just one node in L , we choose another node k to form a route between them. This node is randomly chosen between the three nodes from and to which node i has the greatest demand.

3.3.4 Verifying the connectivity of the solution

There is no guarantee that the solution network obtained after applying the destroy and repair operators is connected. If this network is disconnected, we try to make it connected. We first identify the set V of partitions of the solution. While it is possible, we try to take a node from a partition s and put it into a route of another partition s^s , such that $s \neq s^s$. If this movement is feasible, partitions s and s^s correspond now to just one partition, allowing us to eliminate partition s^s from set V . If we cannot connect two partitions by putting a node from one partition into a route of another partition, we choose the closest two nodes between the two partitions and we construct a new route connecting these two nodes.

3.3.5 Hub configuration and capacity-feasibility check

The destroy and repair operators as well as the procedure applied to verify if the solution is connected are the same for both TDALNS and BUALNS. However, the way

we define the hub configuration of a solution and the mechanism to check whether a solution is feasible with respect to the vehicle capacities differ from one version to the other. We next describe how each of the algorithms perform these two steps.

3.3.5.1 TDALNS Algorithm

In the TDALNS algorithm, the initial hub configuration is obtained by solving a p -median problem. Whenever ω^{ih} iterations have been performed, we modify the hub configuration before applying the destroy operator. We start by closing the hubs incident to exactly one route in the current solution. A hub configuration operator is next chosen to add one more hub to the current solution. We consider three hub configuration operators. Operators H^1 and H^3 choose the new hub according to the σ^p value.

- **Choose node with the highest incoming-outgoing flow (H^1):** Choose among the nodes with the highest incoming-outgoing flow.
- **Choose node randomly (H^2):** Randomly choose a node to become a hub.
- **Choose node according to geographic position (H^3):** Choose among the nodes that are closest to the central point of the graph.

We verify if the network is connected with respect to the hub configuration. If the hub configuration is not feasible, the solution is discarded. Otherwise, we check if the current network has enough capacity to serve all the demand flows. This part of the metaheuristic is responsible for most of the total computational time. Even though we have tried different approaches, preliminary experiments showed that a path based formulation was the most successful approach to do this verification. Thus, we solve a linear problem that we denote as the *capacity verification problem* (CVP). Let \mathcal{P}^{ij} be the set containing all feasible paths to serve demand pair $(i, j) \in W$ over the solution network defined by the arcs in the active vehicle routes, i.e., $A(s) = \{(k, m) \in A : \bar{x}_{km}^p = 1\}$, where $x_{km}^p = 1$ indicates that vehicle p passes through arc (k, m) on the current solution s . Let parameter $b_{kmp}^l \in \{0, 1\}$ be equal to one if path l contains vehicle $p \in P$ passing through arc $(k, m) \in A$. We also define variables $e_{km}^p \geq 0$ to represent the overflow on arc $(k, m) \in A$ for vehicle $p \in P$, and $f_l^{ij} \geq 0$ to be the percentage of demand w_{ij} that goes via path $l \in \mathcal{P}^{ij}$. The CVP can then be

written as:

$$\min \sum_{p \in P} \sum_{\substack{(k,m) \in A: \\ \bar{x}_{km}^p = 1}} e_{km}^p \quad (3.32)$$

$$\text{s.t.: } \sum_{l \in \mathcal{P}^{ij}} f_l^{ij} = 1 \quad \forall (i, j) \in W \quad (3.33)$$

$$\sum_{(i,j) \in W} \sum_{\substack{l \in \mathcal{P}^{ij}: \\ (k,m,p) \in l}} w_{ij} f_l^{ij} - e_{km}^p \leq \tau^{\phi(p)} \bar{x}_{km}^p \quad \forall p \in P, (k, m) \in A : \bar{x}_{km}^p = 1 \quad (3.34)$$

$$f_l^{ij} \geq 0 \quad \forall (i, j) \in W, l \in \mathcal{P}^{ij} \quad (3.35)$$

$$e_{km}^p \geq 0 \quad \forall (k, m) \in A, p \in P : \bar{x}_{km}^p = 1, \quad (3.36)$$

where $\bar{\mathbf{x}}$ represents the current solution that forms the connected network. The objective function minimizes the total overflow of the network (3.32). Constraints (3.33) guarantee that all the demand pairs will be served by a path, while constraints (3.34) calculate the overflow on each arc.

The current solution can either have a feasible network or an infeasible one due to either a lack of capacity or an insufficient number of installation hubs. To solve the CVP, instead of enumerating all paths, we generate them on the fly. First, an auxiliary graph $\tilde{G} = (N, \tilde{A}(s))$ is created with the active arcs and vehicles of the current solution s . For each pair $(i, j) \in W$, we initialize \mathcal{P}^{ij} with its shortest path in the current network. Then, the CVP is solved.

If the CVP objective function is equal to zero, then the procedure terminates. If transshipment occurs only at hub nodes, solution s is feasible. Otherwise, paths already added to the \mathcal{P}^{ij} set were not enough to serve all the demand pairs. However, as set \mathcal{P} does not necessarily contain all possible paths to serve the demand, we can not yet assure that the solution is infeasible. Thus, we verify if there is any new path to be considered when solving the CVP in order to prevent overflow to occur. Let W' be the set of demand pairs $(i, j) \in W$, that need to pass through any arc with overflow to be served. So, after constructing set W' and updating the auxiliary graph \tilde{G} , forbidding arcs with overflow, we look for new paths. On one hand, if we cannot find any new path for demand pairs with overflowing arcs, the solution is considered infeasible and penalized according to the overflown quantity found. On the other hand, if the value of the objective function (3.32) is zero and if transshipment occurs only at hub nodes, the solution is feasible. Otherwise, if the value of the objective function (3.32) is zero while any transshipment occurs at non-hub nodes, then the hub configuration is considered infeasible and the solution is discarded.

Procedure FindPath is used to identify the shortest path from node i to node j in

the current auxiliary graph \tilde{G} . We use Dijkstra's algorithm, in which the cost of the arcs corresponds to their length. If in the path, it is necessary to change vehicles in a node m , and if node m is not an active hub, we add to the shortest path cost the cost of activating node m as a hub.

Our approach shares similarities with column generation algorithms because we create paths on demand. The CVP corresponds to the Master Problem. The pricing subproblem is equivalent to identifying new paths for pairs $(i, j) \in W$. The CVP is solved heuristically because instead of using the dual variables to price new columns, we forbid arcs with overflow. We solve the CVP until the value of the objective function is zero or until it is not possible to add more paths to the problem according to our criteria.

3.3.5.2 BUALNS Algorithm

In the BUALNS algorithm, after ensuring a connected network, we first verify if the current solution has enough arc capacity and if it is the case, the hub network is defined. The procedure to check arc capacity feasibility consists of solving the CVP, as described before for the TDALNS algorithm. The nodes in which transshipment occurs are stored in set H . If the solution is detected to be infeasible, it is penalized according to the overflow quantity found and can be accepted or not according to a simulated annealing criterion. Otherwise, we need to define a feasible hub network. To define the cheapest hub configuration, we solve the Hub Activation Problem (HAP) stated as follows:

$$\min \sum_{k \in H} h_k y_k \quad (3.37)$$

$$\sum_{l \in \bar{\mathcal{P}}^{ij}} f_l^{ij} = 1 \quad \forall (i, j) \in W \quad (3.38)$$

$$\sum_{(i,j) \in W} \sum_{\substack{l \in \bar{\mathcal{P}}^{ij} \\ (k,m,p) \in l}} w_{ij} f_l^{ij} \leq \tau^{\phi(p)} \bar{x}_{km}^p \quad \forall p \in P, (k, m) \in A \quad (3.39)$$

$$y_k \geq f_l^{ij} \quad \forall k \in H, (i, j) \in W, l \in \bar{\mathcal{P}}_k^{ij} \quad (3.40)$$

$$f_l^{ij} \geq 0 \quad \forall (i, j) \in W, l \in \bar{\mathcal{P}}^{ij} \quad (3.41)$$

$$y_k \in \{0, 1\} \quad \forall k \in H. \quad (3.42)$$

Let $\bar{\mathcal{P}}^{ij}$ be the set of paths for demand pair $(i, j) \in W$ that were added in the CVP phase, and $\bar{\mathcal{P}}_k^{ij}$ be the set of paths for demand pair $(i, j) \in W$ containing node k , such as node k needs to become a hub so that the path becomes feasible. In the HAP, the objective function minimizes the cost of activating hubs. Constraints (3.38) guarantee that all demand pairs are served, while constraints (3.39) make sure that the allocated capacity is respected. Constraints (3.40) ensure that flow can pass through a path that

needs a node to become a hub just if this node is activated as a hub. Constraints (3.41) and (3.42) show the variable domains. It is important to note that if $|H| = 1$, we do not have to solve the HAP, because in that case there is no decision to be made, we just activate the only hub that is in set H .

3.3.6 Selection of operators

Let π_i be the weight associated with operator i . The probability of choosing operator i is given by $\pi_i / \sum_j \pi_j$. The update of the weight of each operator is done according to the performance of each operator in a time interval (segment) by means of scores. The score of each operator is initially set to 10. At each iteration, each score can be increased by a factor as follows: (a) θ^1 if the operator results in a new best solution, (b) θ^2 if the operator results in a new solution worst than the minimum, but better than the current one, and (c) θ^3 if the operator results in a solution that is worse than the current one, but satisfies the acceptance criterion. At the end of each segment we calculate new weights for each operator, as follows:

$$\pi_i = (1 - \gamma)\pi_i + \gamma \frac{\eta_i}{\zeta_i},$$

where η_i is the score of operator i obtained during the last segment, ζ_i is the number of times operator i was used in the last segment, and γ is the reaction factor that controls how quickly the weight adjustment algorithm reacts to the last segment performance.

3.3.7 Penalizing infeasible solutions

When we detect that the new solution does not contain enough capacity to serve all demand pairs, we add to the objective function the quantity $\alpha_{km} \sum_{p \in P} \sum_{(k,m) \in A} e_{km}^p$ which carries the penalization factor α_{km} . This factor is initially set to $10 \sum_{r \in \mathcal{R}} \max_{(k,m) \in A} c_{km}^r$. At every 25 iterations, we change this penalization factor. If the best solution s^* is feasible, the penalization factor is decreased to $\alpha_{km} = 0.90\alpha_{km}$. Otherwise, it is increased to $\alpha_{km} = 1.10\alpha_{km}$.

3.3.8 Acceptance criterion

To accept a new solution, we adopt a simulated annealing criterion. An improving solution is always accepted whereas a worse solution is accepted with probability $e^{-\hat{(\phi(s') - \phi(s))/t}}$, where $\phi(s)$ returns the objective function value of solution s . The temperature t is calculated by the formula $t_i = ct_{i-1}$ at iteration i . We set the initial temperature as a percentage of the objective function of the initial solution, such

as $t_0 = \phi(s^0)c^0$. We determine c according to Ribeiro et al. (2014). We suppose a final temperature as a percentage of the objective function of the initial solution, $t_f = \phi(s^0)c^f$. Let Ω be the maximum number of iterations performed, then c is calculated as $c = \sqrt[\Omega]{\frac{t_f}{t_0}}$.

3.4 Computational experiments

We next present the results of the computational experiments performed to assess the behavior of our exact and heuristic algorithms. In all experiments, we set a maximum time limit of 24 hours of CPU time (86,400 seconds). All experiments were performed on an Intel(R) Xeon(R) CPU E5-2687W v3 @ 3.10GHz computer with 750 GB of memory running on a Linux environment. The formulations were coded in C++ using the Concert Technology of CPLEX 12.9 to solve them. Up to four threads were allowed to run during the executions. After some preliminary testes, to improve performance, we set the following variable priority order for branching: y , q , z and x . We chose the barrier optimizer to solve the nodes of the branch-and-bound tree.

The separation and addition of SECs were implemented via lazy cut callbacks and user cut callbacks. To detect the violated SECs for a fractional or integer solution, for each vehicle $p \in P$, such that $q^p > 0$, we solved a series of minimum s - t cut problems in a support graph $G' = (N^{0'}, A^{0'})$ in which $N^{0'} = \{i \in N^0 | z_i^p > 0\}$ and $A^{0'} = \{(i, j) \in A^0 | x_{ij}^p > 0\}$. We set the fictitious depot as the source node s and the nodes $i \in N^{0'}$ as the sink node t . A violated SEC is identified every time the value of the resulting minimum cut is less than 2 units. We used the Concorde callable library by Applegate et al. (2012) to solve the associated mincut problems.

The metaheuristics were also coded in C++. We use the irace package (López-Ibáñez et al. 2011), an automatic configuration tool, to fine tune the values of the parameters used in both algorithms. We used the default package setting using a maximum number of 58320 experiments during the tuning. We decided to run the irace Package on 16 different instances. For the TDALNS algorithm, we change the hub configuration every 50 iterations, i.e. $\omega^{ih} = 50$. The number of nodes n^d and routes r^d to be destroyed are picked within the range $[1, \lambda \times n]$ and $[1, \lambda \times R]$, respectively, where the parameter R represents the total number of routes in the current solution. The parameter λ starts with value 0.1 and is increased by 0.1 every Ψ iterations. For the TDALNS and BUALNS algorithms, we set Ψ to 50 and 5, respectively. When λ reaches the maximum value Υ , we set it to 0.1 again. For the TDALNS and BUALNS algorithms, we use $\Upsilon = 0.6$ and $\Upsilon = 0.7$, respectively. The weights of the operators are reset every 50 iterations to $\omega^{is} = 50$. For the TDALNS algorithm, we use $(\theta^1, \theta^2, \theta^3) =$

$(0, 10, 5)$, while, for the BUALNS variant, we use $(\theta^1, \theta^2, \theta^3) = (10, 5, 0)$. The reaction factor γ is set to 0.1. For the acceptance criterion, we consider $c^0 = 1$ and $c^f = 0.01$. The heuristics were applied 10 times to each instance with each execution taking 25,000 iterations.

3.4.1 Test instances

We used two sets of instances for our experiments. The first data set was randomly generated, whereas the second one was extracted from the Australian Post (AP) standard data set Ernst and Krishnamoorthy (1996). Both data sets assume $T = 24$ and two types of vehicles which were set to have the same speed of 1 unit per hour.

For the first data set, the nodes were randomly scattered around clusters. To guarantee the feasibility of the instance, we consider that the central point of each cluster is at most 12 units far from each other central point. We considered a 16×16 square region with up to four possible clusters. The x and y coordinates of the central points of clusters 1, 2, 3 and 4 are $(4, 4)$, $(12, 12)$, $(12, 4)$ and $(4, 12)$, respectively.

We generated a set of instances by varying the number of clusters, and the total number of nodes in the network. For instance, if clusters 2 and 3 were selected, and an instance with 10 nodes is being created, then the other 8 nodes would be randomly generated. To scatter the non central cluster nodes, we first randomly assign them to a central node, and then we randomly generate their x and y coordinates such that they will lay within three units of distance to a central node of the cluster. The demands between the central points of the clusters were randomly chosen as $\lceil U(0.10, 0.30) \times 750 \rceil$, while between the other nodes they were randomly selected as $\lceil U(0.01, 0.10) \times 100 \rceil$, where the operator $\lceil \cdot \rceil$ returns the nearest integer.

We assumed that all nodes are hub candidates, while the fixed costs to install a hub in a node were randomly chosen within the range $[20000, 30000]$. In our experiments, we selected problems with $|N| = \{6, 7, 8, 9, 10, 20, 30, 40, 50\}$ nodes. These random instances are referred to as NR, where N represents the number of nodes considered in the instance. For example, instance 10R contains 10 nodes. The vehicle costs were set to $b^I = 3000$, $b^{II} = 4500$, $a^I = 20000$, $a^{II} = 35000$ for each instance. For each problem size, we generated 4 instances for the HNDP varying the value of τ^I to assess the effect of different economies of scale. Table 3.2 shows the capacity values considered. The potential economies of scale obtained when using a larger vehicle when it is fully utilized is calculated as $((a^{II} + b^{II})/\tau^{II})/((a^I + b^I)/\tau^I)$.

The AP data set consists of demand flows and Euclidean distances between 200 districts of Australia. We selected problems with $|N| = \{10, 15, 20, 25, 30, 35, 40, 45, 50\}$ nodes, but disregarding flows w_{ii} for each $i \in N$, i.e. we set $w_{ii} = 0$. We changed the

Table 3.2: Capacity of the vehicles and potential economies of scale for the random instances

Instances	economies of scale	τ^I	τ^{II}
6R-10R	0.21	90	750
	0.34	150	
	0.50	220	
	0.80	350	
20R	0.21	250	2040
	0.34	400	
	0.50	590	
	0.80	1070	
30R	0.21	290	2360
	0.34	470	
	0.50	690	
	0.80	1100	
40R	0.21	370	3000
	0.34	600	
	0.50	880	
	0.80	1400	
50R	0.21	510	4160
	0.34	830	
	0.50	1210	
	0.80	1940	

distances in a way that from each node it is possible to go to at least half of the total nodes of the graph within 12 hours. The vehicle costs were set to $b^I = 6000$, $b^{II} = 9000$, $a^I = 20000$, $a^{II} = 30000$ and $\tau^{II} = 3000$ for each instance. For each problem size, we generated four instances for the HNRP by varying the value of τ^I to assess the effect of different economies of scale. Table 3.3 shows the capacity values considered for the small vehicle and the potential economies of scale obtained.

Table 3.3: Capacity of the vehicles and potential economies of scale for the AP instances ($\tau^{II} = 3,000$)

economies of scale	τ^I
0.20	400
0.30	600
0.50	1,000
0.80	1,400

3.4.2 Solving small instances

We first focus on analyzing the performance of the proposed MIP formulation and algorithms to solve small instances. Table 3.4 reports the results obtained for instances 6R, 7R, 8R, 9R, 10R, and 10AP considering economies of scale of 0.21/0.20, 0.34/0.30,

0.50 and 0.80. We consider four vehicles of type I and two vehicles of type II. Column Dev(%) presents the deviation between the optimal solution and the solutions found by the MIP formulation, TDALNS and BUALNS. In particular, columns under the heading AVG(BKS) report the deviation between the optimal solution and the average(best) solution found by TDALNS and BUALNS. Column TRN (s) presents the time spent to solve the linear programming relaxation of the root node of the branch-and-bound tree. Column LP(%) reports the deviation between the optimal solution and the linear programming relaxation value.

Table 3.4: Results for small instances: 6R, 7R, 8R, 9R, 10R and 10AP.

economies of scale Instances	Time(s)			Dev(%)					MIPFV		
	MIP	TDALNS	BUALNS	MIP	TDALNS		BUALNS		TRN (s)	LP (%)	
					BKS	AVG	BKS	AVG			
0.21/0.20	6R	34	14	27	0.00	0.00	0.00	0.00	0.00	0	19.84%
	7R	71	15	25	0.00	0.00	0.00	0.00	0.00	1	20.36%
	8R	86	17	31	0.00	0.00	0.00	0.00	0.00	2	18.73%
	9R	264	22	33	0.00	0.00	0.00	0.00	0.00	4	16.90%
	10R	17454	26	55	0.00	0.00	0.00	0.00	0.00	11	24.07%
	10AP	84177	28	50	0.00	4.77	5.72	4.74	4.75	16	29.12%
Average		17014	20	37	0.00	0.80	0.95	0.79	0.79	6	21.50%
0.34/0.30	6R	76	13	26	0.00	0.00	0.00	0.00	0.00	0	32.91%
	7R	982	14	28	0.00	0.00	0.00	0.00	0.00	1	33.07%
	8R	317	17	32	0.00	0.00	0.00	0.00	0.00	2	18.73%
	9R	1691	22	36	0.00	0.00	0.00	0.00	0.00	6	17.20%
	10R	27077	26	57	0.00	0.00	0.00	0.00	0.00	11	23.88%
	10AP	30561	28	54	0.00	0.00	0.00	0.00	0.00	19	18.58%
Average		10117	20	39	0.00	0.00	0.00	0.00	0.00	7	24.06%
0.5	6R	43	17	30	0.00	0.00	4.25	0.00	1.58	0	36.63%
	7R	204	22	29	0.00	1.38	1.56	1.38	1.38	1	36.03%
	8R	2394	24	37	0.00	0.00	0.00	0.00	0.00	2	30.78%
	9R	32403	24	36	0.00	0.00	0.00	0.00	0.00	5	32.96%
	10R	86400	26	57	5.20	8.67	9.33	0.00	8.45	13	29.84%
	10AP	86400	26	60	16.03	0.83	0.83	0.83	0.83	20	33.59%
Average		34641	23	42	3.54	1.81	2.66	0.37	2.04	7	33.31%
0.8	6R	20	17	30	0.00	0.00	0.00	0.00	0.00	0	39.27%
	7R	17	22	30	0.00	0.00	0.00	0.00	0.00	1	23.93%
	8R	86	25	37	0.00	0.00	0.00	0.00	0.00	3	20.93%
	9R	431	23	44	0.00	0.00	0.00	0.00	0.00	182	20.22%
	10R	10068	26	54	0.00	0.00	1.23	0.00	0.58	12	29.94%
	10AP	37758	27	67	0.00	0.00	0.00	0.00	0.00	19	33.35%
Average		8063	23	44	0.00	0.00	0.21	0.00	0.10	36	27.94%

As can be seen in Table 3.4, the devised metaheuristics take much less time than

the exact algorithm to solve the instances, as expected. The MIPFV version requires 804 and 434 times more computational effort on average than the TDALNS and the BUALNS versions, respectively. For instance 9R-0.80, the computational effort needed to solve just the linear relaxation of the formulation is about eight and four times larger than the time spent by the TDALNS and BUALNS algorithms, respectively. The MIPFV formulation was not able to reach optimality for two instances within the given time limit, 10R-0.50, and 10AP-0.50, due to the complexity of the problem. However, for instance 10R-0.50, the BUALNS was able to find the optimal solution within less than one minute. The TDALNS and BUALNS failed to find the optimal solution in four (10AP-0.20, 7R-0.50, 10R-0.50, and 10AP-0.50) and three (10AP-0.20, 7R-0.50, and 10AP-0.50) instances, respectively, of the 24 proposed test problems.

Table 3.5 shows the results obtained for instances 20R, 15AP and 20AP. We considered five vehicles for each type I and II. We took into account two additional different performance measures: Dev_BKS is the percentage deviation between the best solution found by the version and the best known solution; while GAP_CPLEX is the optimality gap found by CPLEX within 24 hours of CPU time.

Table 3.5: Results for medium instances: 20R, 15AP and 20AP

economies of scale	Instances	Time(s)			Desv_BKS(%)			MIPFV	
		MIPFV	TDALNS	BUALNS	MIPFV	TDALNS	BUALNS	TRN (s)	GAP_CPLEX(%)
0.21/0.20	20R	86400	143	329	0.00	0.00	0.69	52199	32.74%
	15AP	86400	62	156	0.00	0.00	0.00	2627	29.05%
	20AP	86400	124	314	0.00	0.26	0.00	54102	26.01%
0.34/0.30	20R	86400	141	332	0.00	2.08	0.00	86388	100.00%
	15AP	86400	63	159	0.00	0.00	1.42	2038	27.76%
	20AP	86400	122	325	0.00	0.40	0.00	57579	30.01%
0.5	20R	86400	140	337	0.00	0.00	1.84	86385	100.00%
	15AP	86400	69	176	0.00	2.20	0.00	1626	33.03%
	20AP	86400	149	391	0.00	0.00	0.00	38073	31.35%
0.8	20R	86400	127	353	0.00	0.00	3.22	83842	27.21%
	15AP	86400	61	181	0.00	0.00	0.00	1116	30.64%
	20AP	86400	121	378	0.00	4.44	0.00	29823	26.64%
Average		86400	110	286	0.35	1.14	0.95	41317	41.20%

When the instance size is greater than 15 nodes, it becomes much more difficult to solve by CPLEX regardless of the formulation being used. Before calling the CPLEX solver, the best solution found by the devised metaheuristics is supplied as an initial solution to it. The fact that the metaheuristics performed better than MIPFV stands out once again. While the metaheuristics found good solutions within 400 seconds, the MIPFV took 41316 seconds on average just to attain the linear programming relaxation of the formulation for these instances. For some instances (20R-0.30 and 20R-0.50),

the MIPFV was not able to find any lower bound within one day of computing time. The results only highlight how difficult the problem is.

3.4.3 Solving large instances

To better assess the devised metaheuristics, we also tested them on larger instances. Since the complexity of the problem does not allow the MIPFV exact approach to neither prove the optimality nor improve the best solution found by the proposed metaheuristics for instances with 15 nodes, we chose to solve larger instances with more than 25 nodes using only the metaheuristics. We considered five vehicles for each type I and II. Table 3.6 reports the average computational time in seconds, the deviation between the best solution found by the version and the best known solution (Dev_BKS), and the deviation between the average objective function value found and the best known solution (Dev_avg).

According to Table 3.6, the TDALNS version gets worse solutions than BUALNS though taking less time. Table 3.5 shows that BUALNS version provides slightly better solutions on average. For instances with 15 and 20 nodes, the BUALNS version found the best known solution eight times, whereas the TDALNS variant got the best known solution seven times. For the large instances, although the TDALNS is in average 3 times faster than the BUALNS version, the BUALNS is in average 2 times better to find the best known solutions, and 1.4 times better to find averages values. To obtain these values we used the ratio of the summed averages.

Despite taking more computing time, the BUALNS seems to present a better performance. The larger computational effort is due to the step in which the CPLEX solver is called in each iteration to find the cheapest hub configuration. In the TDALNS the hub configuration is randomly defined. In order to confirm this behavior, we performed a statistical test. Generally, there are two types of statistical tests, namely parametric and non-parametric. Three conditions must be met to use a parametric test, including normality, independence, and heteroscedasticity. A Kolmogorov-Smirnov test was carried out to check the first condition (i.e. normality). The results reported in Table 3.7 show that the normality condition is not fulfilled in our case (i.e., the p-value is less than .05).

For this reason, a non-parametric test, the Wilcoxon test, was used to compare the devised algorithms in terms of best known solutions found. To calculate the Wilcoxon signed-rank test, the Wilcoxon function of the scipy Python package was used. First, we used the default alternative hypothesis to test if the algorithms were statistically different and later, we use the alternative hypothesis to verify if the BUALNS was better than the TDALNS version. The p-values of the Wilcoxon tests are reported in

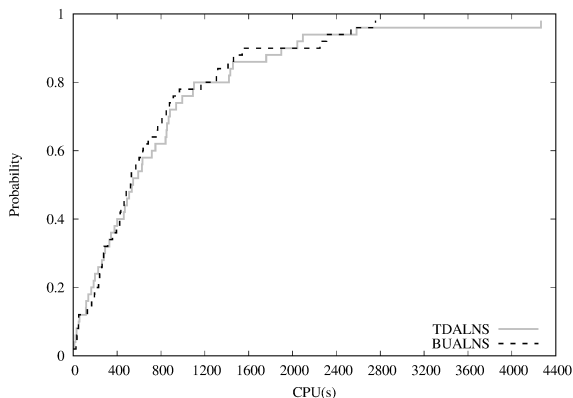


Figure 3.2: TTT plot for 25AP-030

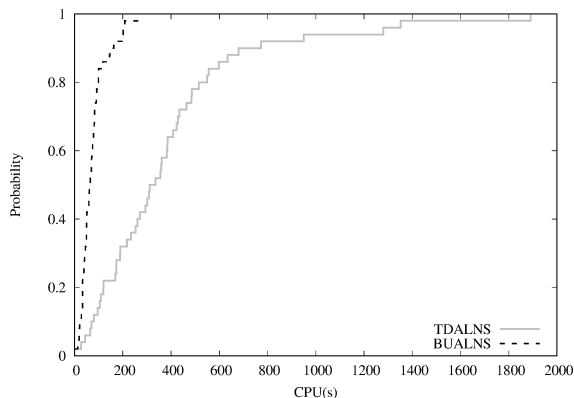


Figure 3.3: TTT plot for 25AP-080

Table 3.8. The results show that there is a significant difference between the algorithms, and that the BUALNS is indeed better than TDALNS to find BKSs.

To draw further insights from the algorithms, their performance to find a target solution is studied. The worst average objective function value found for two selected instances (25AP-030, 25AP-080) is given as the target solution. Each method is run for 50 executions, using different seeds each time, until the target solution is found. The computational time required to find it is recorded. The results are then displayed using time-to-target (TTT) plots introduced by Aiex et al. (2002, 2007). For a given instance, a TTT plot displays on the ordinate axis the probability of the algorithm to obtain a solution as good as a given target value within a running time in seconds, shown on the abscissa axis. Figures 3.2 and 3.3 show the attained TTT plots for the two instances. For instance 25AP-030, it can be seen that both versions have very similar behavior. The BUALNS finds the target solution 100% of the time within 2800 seconds, while the TDALNS has around 2% of probability to spend more seconds than that. On the other hand, for instance 25AP-080, the BUALNS has a much higher probability to find better solutions than the TDALNS in a shorter time. It took 10% of the time to assert the target solution.

Finally, the benchmark performance profiles of the algorithms were examined. Performance profiles, shown in Figure 3.4, are graphic tools which aid to evaluate and compare the performance of a set of algorithms on solving a problem set. They show the plot of a performance function over a chosen performance measure. Further details of performance profiles can be found in Dolan and Moré (2002). Here, the number of times the best known solution (BKS) is found by an algorithm is used as a performance measure. Hence the performance function $\rho(\tau)$ represents the proportion of instances solved by an algorithm, with performance within a factor τ of the best behavior obtained, considering all the analyzed algorithms. The TDALNS finds the BKS for only 40% of the instances whereas the BUALNS finds the BKS for 80% of the instances.

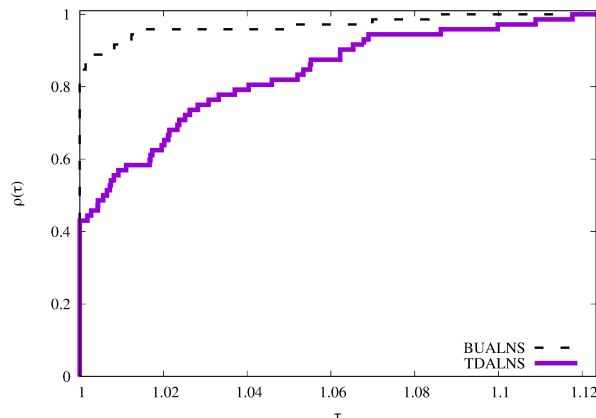


Figure 3.4: Performance profile of the devised algorithms.

After these analyses we can point out that, although the hub configuration is a more strategic decision than defining the vehicle routes, and that strategic decisions are usually taken before tactical decisions, the results here indicate that locating the hubs after knowing the vehicle routes is a better practice.

3.5 Conclusion

In this paper, we have studied a flexible hub network design problem. The flexibility stems from the fact that no fixed discount factors representing economies of scale are imposed. Here economies of scale directly derive from the transport technology chosen to operate the routes which ensures that vehicles can be used more efficiently. Moreover, no particular topological structure is assumed, which means that: (i) vehicle routes do not necessarily have hubs on them; (ii) there is no limitation on the number of vehicles that passes through each demand node, and; (iii) the demand can be served directly or passing through as many non-hub or hub nodes as it is cost convenient. These aspects make the problem more complexity to formulate and solve. The HNDF is specifically suitable for liner shipping network design, in which a hub network design with flexible periodic routes must satisfy demands for shipments of containers at minimal cost. A mathematical formulation and two metaheuristics based on the adaptive large neighborhood search heuristic were proposed. On the one hand, the MIPFV takes considerable computational time just to solve the linear programming relaxation of the formulation for small instances, highlighting how difficult the problem is. On the other hand, the devised metaheuristics find good solutions within a reasonable time. The computational results showed that, for the considered instances, the bottom up approach found better solutions than the top down approach.

Algorithm 5 Basic steps of BUALNS

```

 $s \leftarrow \text{InitialSolution}$ 
 $s^* \leftarrow s$ 
 $s^{f*} \leftarrow s$ 
InitializeScores( $\pi^D, \pi^R$ )
 $is \leftarrow 0$ 
repeat
   $O^d \leftarrow \text{ChooseDestroyOperator}(D, \pi)$ 
   $O^r \leftarrow \text{ChooseRepairOperator}(R, \pi)$ 
   $s^{new} \leftarrow \text{DestroyAndRepairOperators}(O^d, O^r, s^{new})$ 
  if  $s^{new}$  is disconnected then
     $s^{new} \leftarrow \text{ConnectNetwork}(s^{new})$ 
  end if
  ( $s^{new}, \text{feasibleSolution}$ )  $\leftarrow \text{AnalyzeCapacityFeasibilityAndDefineHubConfiguration}(s^{new})$ 
  if  $\text{feasibleSolution} = \text{false}$  then
    Penalize( $OF(s^{new})$ )
  end if
  if  $s^{new}$  satisfies the acceptance criterion then
     $s \leftarrow s^{new}$ 
  end if
  if  $OF(s^{new}) < OF(s^*)$  then
     $s^* \leftarrow s^{new}$ 
  end if
  if  $\text{feasibleSolution} = \text{true}$  then
    if  $OF(s^{new}) < OF(s^{f*})$  then
       $s^{f*} \leftarrow s^{new}$ 
    end if
  end if
  if  $is = \omega^{is}$  then
    UpdateScores( $\pi^D, \pi^R$ )
     $is \leftarrow 0$ 
  else
     $is \leftarrow is + 1$ 
  end if
until {the stopping condition is met}

```

Table 3.6: Results for large instances: 30R, 40R, 50R, 25AP, 30AP, 35AP, 40AP, 45AP and 50AP.

Scale economies	Instances	Time(s)		Dev_BKS(%)		Dev_Avg(%)	
		TDALNS	BUALNS	TDALNS	BUALNS	TDALNS	BUALNS
0.20	30R	465	1246	0.00	0.41	4.25	1.14
	40R	1257	3084	3.50	0.00	10.14	4.40
	50R	2886	7958	4.95	0.00	10.78	5.30
	25AP	217	598	2.62	0.00	7.04	2.35
	30AP	383	1130	2.08	0.00	5.12	2.24
	35AP	646	1918	0.00	0.59	3.70	4.04
	40AP	1074	3326	8.07	0.00	13.24	7.50
	45AP	1607	5384	0.00	7.20	10.78	10.31
	50AP	2517	8595	4.48	0.00	9.67	5.75
Average		1228	3693	2.86	0.91	8.30	4.78
0.30	30R	464	1256	0.65	0.00	3.35	1.30
	40R	1267	3454	6.69	0.00	12.27	5.70
	50R	2826	8153	4.92	0.00	8.46	6.38
	25AP	221	623	0.00	1.42	8.15	7.68
	30AP	387	1140	5.26	0.00	8.85	4.64
	35AP	643	2014	0.00	1.88	3.87	5.12
	40AP	1077	3488	0.00	7.21	14.97	14.15
	45AP	1621	5668	4.05	0.00	8.04	6.20
	50AP	2581	9021	0.00	3.01	10.81	8.52
Average		1232	3869	2.40	1.50	8.75	6.63
0.50	30R	443	1242	0.69	0.00	4.29	2.55
	40R	1187	3293	0.00	1.98	5.83	6.67
	50R	2818	8460	0.39	0.00	5.61	6.91
	25AP	217	756	0.00	2.54	8.45	8.48
	30AP	389	1208	1.12	0.00	9.23	6.70
	35AP	640	2134	5.85	0.00	12.02	6.17
	40AP	1164	3748	10.63	0.00	13.82	6.48
	45AP	1621	6121	0.00	1.81	8.29	4.81
	50AP	2530	9752	0.50	0.00	8.49	6.22
Average		1223	4079	2.13	0.70	8.45	6.11
0.80	30R	429	1242	2.30	0.00	4.27	5.73
	40R	1214	3561	0.00	6.96	6.81	12.15
	50R	2887	9124	0.00	8.61	6.87	14.34
	25AP	218	746	12.79	0.00	18.15	5.70
	30AP	379	1430	2.46	0.00	11.94	4.82
	35AP	629	2513	10.55	0.00	15.36	6.20
	40AP	1042	4600	0.00	2.46	10.01	8.81
	45AP	1597	7269	0.98	0.00	12.84	6.23
	50AP	2483	11621	7.15	0.00	19.21	8.39
Average		971	3317	4.01	2.57	10.48	8.25

Table 3.7: Results of the normality analysis.

Algorithm	Kolmogorov-Smirnov test	
	Statistic	P-value
TDALNS	0.445	0.000
BUALNS	0.235	0.001

Table 3.8: Results of the Wilcoxon tests.

Pairwise comparison	Alternative	Statistic	P-value
TDALNS-BUALNS	default	233.0	0.0
TDALNS-BUALNS	greater	1198.0	0.0

Chapter 4

Final Remarks

This thesis addresses two hub-and-spoke network design problems focuses on transportation systems. The first problem, presented in Chapter 2, proposes the design of a global air transportation system, relying on a three-tier hierarchical hub-and-spoke network. The problem mainly arises from the differentiation of domestic and international passengers. Having nodes located in different geographic regions, gateways, hubs, inter-hub, and inter-gateways connections are installed and so local and global flows are routed at minimal transportation and fixed costs. The GHLP is modeled as a multi-commodity flow-based hub and spoke system and given its large scale multi-commodity nature and its induced decomposable matrix structure, the mathematical formulation introduced here is solved by two specialized Benders decomposition algorithms. A new repair procedure allows to generate Benders optimality cuts from unbounded dual subproblems and a tailored dual subproblem solution algorithm calculates the optimal dual values to produce Benders optimality cuts without the need of a Simplex solver. Computational experiments show that while the monolithic version failed to solve medium-size instances, the exact algorithms solve larger ones in a reasonable time.

While the GHLP supports strategic decisions, the second problem, presented in Chapter 3, considers both strategic and tactical characteristics in the design of the transportation system. The HNDPs proposes the design of a more generic and flexible network, having a wide range of applications, including air, ground and liner shipping industries. The problem is flexible in the sense that no network topology is imposed, vehicle routes may or may not have hubs on them, demand path can consist of a single vehicle route or a set of them. Most important of all, economies of scale derive from the technology chosen to operate the routes and its utilization, instead of being represented by a fixed discount factor. It is important to highlight once more that this flexibility makes the problem more difficult to formulate and solve than classical hub

network design problems. We propose a MIP formulation to solve the problem and as the proposed MIP is too large to be solved in a timely manner by general purpose solver, two metaheuristics based on an adaptive large neighborhood search (ALNS) are developed. In the computational experiments, the metaheuristic that follows a bottom up approach performs better than the one with a top down approach.

This thesis presents two hub-and-spoke network design problems with the potential for practical applications by considering several characteristics of real transportation systems. Mathematical formulations were proposed to model these problems while exact and heuristic algorithms, based on decomposition methods and adaptive large neighborhood search metaheuristics are proposed to solve them. Although the proposed algorithms can tackle instances that still far from real-life size problems, our approaches can work as an important tool to offer questions and insights about the way current transportation networks are designed. Thus, some issues that remain open for future research consist of developing metaheuristics to solve instances larger than 141 nodes for GHLP and in developing exact efficient methods to optimally solve the HNDPs.

Bibliography

- ACI (2016). Aci media releases. Technical report, Airports Council International.
- Adler, N. (2005). Hub-spoke network choice under competition with an application to western europe. *Transportation Science*, 39(1):58–72.
- Adler, N. and Smilowitz, K. (2007). Hub-and-spoke network alliances and mergers: Price-location competition in the airline industry. *Transportation Research Part B: Methodological*, 41(4):394 – 409.
- Ahuja, R., Magnanti, T., and Orlin, J. (1993). *Network Flows*. Prentice-Hall, Upper Saddle River.
- Aiex, R. M., Resende, M. G., and Ribeiro, C. C. (2002). Probability distribution of solution time in grasp: An experimental investigation. *Journal of Heuristics*, 8(3):343–373.
- Aiex, R. M., Resende, M. G., and Ribeiro, C. C. (2007). Ttt plots: a perl program to create time-to-target plots. *Optimization Letters*, 1(4):355–366.
- Alumur, S. and Kara, B. Y. (2008). Network hub location problems: The state of the art. *European Journal of Operational Research*, 190(1):1 – 21.
- Applegate, D., Bixby, R., Chvátal, V., and Cook, W. (2012). The traveling salesman problem, concorde tsp solver.
- Arslan, S., Güreş, N., and Yılmaz, H. (2011). A comparison of airline service expectations between passengers of domestic and international flights. *International Journal of Social Sciences and Humanity Studies*, 3:377 – 386.
- Barnhart, C., Belobaba, P., and Odoni, A. R. (2003). Applications of operations research in the air transport industry. *Transportation Science*, 37(4):368–391.
- Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252.
- Birge, J. R. and Louveaux, F. V. (1988). A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34(3):384 – 392.
- Camargo, R. S. d., Miranda, G. d., and Løkketangen, A. (2013). A new formulation and an exact approach for the many-to-many hub location-routing problem. *Applied Mathematical Modelling*, 37(12-13):7465–7480.
- Campbell, J. F., Ernst, A. T., and Krishnamoorthy, M. (2002). Hub location problems. In

- Drezner, Z. and Hamacher, H., editors, *Facility location : application and theory*, pages 373–407. Springer Berlin.
- Campbell, J. F. and O’Kelly, M. E. (2012). Twenty-five years of hub location research. *Transportation Science*, 46(2):153–169.
- Catanzaro, D., Gourdin, E., Labbé, M., and Özsoy, F. A. (2011). A branch-and-cut algorithm for the partitioning-hub location-routing problem. *Computers & Operations Research*, 38(2):539 – 549.
- Çetiner, S., Sepil, C., and Süral, H. (2010). Hubbing and routing in postal delivery systems. *Annals of Operations research*, 181(1):109–124.
- Contreras, I. (2015). *Hub Location Problems*, pages 311–344. Springer International Publishing, Cham.
- Contreras, I., Cordeau, J.-F., and Laporte, G. (2011). Benders decomposition for large-scale uncapacitated hub location. *Operations Research*, 59(6):1477–1490.
- Contreras, I. and Fernández, E. (2014). Hub location as the minimization of a supermodular set function. *Operations Research*, 62(3):557–570.
- Contreras, I., Tanash, M., and Vidyarthi, N. (2017). Exact and heuristic approaches for the cycle hub location problem. *Annals of Operations Research*, 258(2):655–677.
- Cordeau, J.-F., Soumis, F., and Desrosiers, J. (2000). A benders decomposition approach for the locomotive and car assignment problem. *Transportation science*, 34(2):133–149.
- Costa, A. M. (2005). A survey on benders decomposition applied to fixed-charge network design problems. *Computers & operations research*, 32(6):1429–1450.
- de Camargo, R. S., de Miranda Jr, G., O’Kelly, M. E., and Campbell, J. F. (2017). Formulations and decomposition methods for the incomplete hub location network design problem with and without hop-constraints. *Applied Mathematical Modelling*, 51:274–301.
- de Camargo, R. S., Gilberto de Miranda, J., and Luna, H. P. L. (2009). Benders decomposition for hub location problems with economies of scale. *Transportation Science*, 43(1):86–97.
- de Carvalho, R., de Camargo, R. S., Martins, A. X., and Saldanha, R. R. (2017). A parallel heuristics for the single allocation hub location problem. *IEEE Latin America Transactions*, 15(7):1278–1285.
- de Sá, E. M., de Camargo, R. S., and de Miranda, G. (2013). An improved benders decomposition algorithm for the tree of hubs location problem. *European Journal of Operational Research*, 226(2):185–202.
- de Sá, E. M., Morabito, R., and de Camargo, R. S. (2018). Benders decomposition applied to a robust multiple allocation incomplete hub location problem. *Computers & Operations Research*, 89:31–50.
- Directorate-General for Mobility and Transport (European Commission) (2019). EU trans-

- port in figures - Statistical pocketbook 2019 . <https://op.europa.eu/en/publication-detail/-/publication>. Online; accessed 13 January 2020.
- Dolan, E. D. and Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213.
- Ernst, A. T. and Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, 4(3):139 – 154. Hub Location.
- Ernst, A. T. and Krishnamoorthy, M. (1998). Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. *European Journal of Operational Research*, 104(1):100–112.
- Farahani, R. Z., Hekmatfar, M., Arabani, A. B., and Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64(4):1096 – 1109.
- Gelareh, S., Monemi, R. N., and Nickel, S. (2015). Multi-period hub location problems in transportation. *Transportation Research Part E: Logistics and Transportation Review*, 75:67 – 94.
- Gelareh, S. and Nickel, S. (2011). Hub location problems in transportation networks. *Transportation Research Part E: Logistics and Transportation Review*, 47(6):1092 – 1111.
- Gelareh, S. and Pisinger, D. (2011). Fleet deployment, network design and hub location of liner shipping companies. *Transportation Research Part E: Logistics and Transportation Review*, 47(6):947–964.
- Geoffrion, A. M. and Graves, G. W. (1974). Multicommodity distribution system design by benders decomposition. *Management science*, 20(5):822–844.
- Hamacher, H. W., Labbé, M., Nickel, S., and Sonneborn, T. (2004). Adapting polyhedral properties from facility to hub location problems. *Discrete Applied Mathematics*, 145(1):104–116.
- Hansen, M. (1990). Airline competition in a hub-dominated environment: An application of noncooperative game theory. *Transportation Research Part B: Methodological*, 24(1):27 – 43.
- Hong, S. and Harker, P. T. (1992). Air traffic network equilibrium: Toward frequency, price and slot priority analysis. *Transportation Research Part B: Methodological*, 26(4):307 – 323.
- Hsu, C.-I. and Wen, Y.-H. (2003). Determining flight frequencies on an airline network with demand–supply interactions. *Transportation Research Part E: Logistics and Transportation Review*, 39(6):417 – 441.
- IATA (2015). 20 year passenger forecast. Technical report, The International Air Transport Association.
- International Air Transport Association (2019). IATA annual review 2019.

- <https://www.iata.org/publications/Pages/annual-review.aspx>. Online; accessed 13 January 2020.
- Jr., J. T. B. (2012). A spatial analysis of fedex and ups: hubs, spokes, and network structure. *Journal of Transport Geography*, 24:419 – 431. Special Section on Theoretical Perspectives on Climate Change Mitigation in Transport.
- Karimi, H. (2018). The capacitated hub covering location-routing problem for simultaneous pickup and delivery systems. *Computers & Industrial Engineering*, 116:47–58.
- Kartal, Z., Hasgul, S., and Ernst, A. T. (2017). Single allocation p-hub median location and routing problem with simultaneous pick-up and delivery. *Transportation Research Part E: Logistics and Transportation Review*, 108:141–159.
- Kartal, Z., Krishnamoorthy, M., and Ernst, A. T. (2019). Heuristic algorithms for the single allocation p-hub center problem with routing considerations. *OR Spectrum*, 41(1):99–145.
- Kimms, A. (2006). *Perspectives on Operations Research: Essays in Honor of Klaus Neumann*.
- Labbé, M. and Yaman, H. (2004). Projecting the flow variables for hub location problems. *Networks: An International Journal*, 44(2):84–93.
- Labbé, M. and Yaman, H. (2008). Solving the hub location problem in a star–star network. *Networks: An International Journal*, 51(1):19–33.
- Labbé, M., Yaman, H., and Gourdin, E. (2005). A branch and cut algorithm for hub location problems with single assignment. *Mathematical programming*, 102(2):371–405.
- Leung, J. M. Y., Magnanti, T. L., and Singhal, V. (1990). Routing in point-to-point delivery systems: Formulations and solution heuristics. *Transportation Science*, 24(4):245–260.
- Lopes, M. C., de Andrade, C. E., de Queiroz, T. A., Resende, M. G., and Miyazawa, F. K. (2016). Heuristics for a hub location-routing problem. *Networks*, 68(1):54–90.
- López-Ibáñez, M., Dubois-Lacoste, J., Stützle, T., and Birattari, M. (2011). The irace package, iterated race for automatic algorithm configuration. Technical report, Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles
- Magnanti, T. L. and Wong, R. T. (1981). Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484.
- Mao, L., Wu, X., Huang, Z., and Tatem, A. J. (2015). Modeling monthly flows of global air travel passengers: An open-access data resource. *Journal of Transport Geography*, 48:52 – 60.
- Martins de Sá, E., Contreras, I., Cordeau, J.-F., Saraiva de Camargo, R., and de Miranda, G. (2015). The hub line location problem. *Transportation Science*, 49(3):500–518.
- Mercier, A., Cordeau, J.-F., and Soumis, F. (2005). A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research*, 32(6):1451 – 1476.

- Nagy, G. and Salhi, S. (1998). The many-to-many location-routing problem. *Top*, 6(2):261–275.
- O’Kelly, M. and Bryan, D. (1998). Hub location with flow economies of scale. *Transportation Research Part B: Methodological*, 32(8):605 – 616.
- O’Kelly, M. E. (1986). The location of interacting hub facilities. *Transportation Science*, 20(2):92–106.
- O’Kelly, M. E. (1987). A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32(3):393 – 404.
- Papadakos, N. (2008). Practical enhancements to the magnanti–wong method. *Operations Research Letters*, 36(4):444 – 449.
- Preis, T., Huang, Z., Wu, X., Garcia, A. J., Fik, T. J., and Tatem, A. J. (2013). An open-access modeled passenger flow matrix for the global air network in 2010. *PLoS ONE*, 8(5):e64317.
- Reinhardt, L. B. and Pisinger, D. (2012). A branch and cut algorithm for the container shipping network design problem. *Flexible Services and Manufacturing Journal*, 24(3):349–374.
- Ribeiro, G. M., Desaulniers, G., Desrosiers, J., Vidal, T., and Vieira, B. S. (2014). Efficient heuristics for the workover rig routing problem with a heterogeneous fleet and a finite horizon. *Journal of Heuristics*, 20(6):677–708.
- Rodríguez-Martín, I., Salazar-González, J.-J., and Yaman, H. (2014). A branch-and-cut algorithm for the hub location and routing problem. *Computers & Operations Research*, 50:161–174.
- Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472.
- Sasaki, M., Furuta, T., and Suzuki, A. (2009). Gateway location models. In *The Eighth International Symposium on Operations Research and Its Applications (ISORA ’09)*, pages 356–363. ORSC & APORC.
- Schossler, M. and Wittmer, A. (2015). Cost and revenue synergies in airline mergers : Examining geographical differences. *Journal of Air Transport Management*, 47:142–153.
- Shaw, P. (1997). A new local search algorithm providing high quality solutions to vehicle routing problems. *APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK*.
- Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In Maher, M. and Puget, J.-F., editors, *Principles and Practice of Constraint Programming — CP98*, pages 417–431, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Skorin-Kapov, D., Skorin-Kapov, J., and O’Kelly, M. (1996). Tight linear programming

- relaxations of uncapacitated p-hub median problems. *European Journal of Operational Research*, 94(3):582 – 593.
- Song, D.-P. and Dong, J.-X. (2013). Long-haul liner service route design with ship deployment and empty container repositioning. *Transportation Research Part B: Methodological*, 55:188–211.
- Tanash, M., Contreras, I., and Vidyarthi, N. (2017). An exact algorithm for the modular hub location problem with single assignments. *Computers & Operations Research*, 85:32 – 44.
- Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160.
- United Nations Conference on Trade and Development (2018). UNCTD - Handbook of Statistics . <https://unctad.org/en/PublicationsLibrary>. Online; accessed 14 January 2020.
- Wasner, M. and Zäpfel, G. (2004). An integrated multi-depot hub-location vehicle routing model for network planning of parcel service. *International Journal of Production Economics*, 90(3):403–419.
- Yaman, H. (2009). The hierarchical hub median problem with single assignment. *Transportation Research Part B: Methodological*, 43(6):643 – 658.