Optimization Models and Algorithms for the Design of Global Transportation Networks

Modèles et Algorithmes pour la Conception de Réseaux de Transport Mondiaux

Auteur : Fábio Francisco da Costa Fontes

Directeur : Prof. Gilles Goncalves

Thèse de doctorat de l'Université d'Artois

Spécialité Génie Informatique et Automatique

Composition du jury :

Andréa Cynthia Santos	Maître de conférences avec HDR à l'Université de Technologie de Troyes	Rapporteur
Olivier Péton	Professeur à l'IMT Atlantique	Rapporteur
Clarisse Dhaenens	Professeur à l'Université Lille 1	Présidente
Hifi M'Hand	Professeur à l'Université de Picardie Jules Verne	Examinateur
Hamid Allaoui	Professeur à l'Université d'Artois	Examinateur
Gilles Goncalves	Professeur à l'Université d'Artois	Directeur de thèse

École Doctorale SPI Lille Nord de France

Laboratoire de Génie Informatique et d'Automatique de l'Artois (LGI2A - EA3926)

Discipline: Sciences pour l'ingénieur

Thèse soutenue, le 27/10/2017

Abstract

The development of efficient network structures for freight transport is a major concern for the current global market. Demands need to be quickly transported and should also meet the customer needs in a short period of time. Traffic congestions and delays must be minimized, since CO_2 emissions must be controlled and affordable transport costs have to be offered to customers. Hub-and-spoke structure is a current network model used by both regional and intercontinental transportation, which offers an economy of scale for aggregated demands inside hub nodes. However, delays, traffic congestions and long delivery time are drawbacks from this kind of network. In this thesis, a new concept, which is called “sub-hub”, is proposed to the classic hub-and-spoke network structure. In the proposed network models, economy of scale and shorter alternative paths are implemented, thus minimizing the transport cost and delivery time. The sub-hub proposal can be viewed as a connection point between two routes from distinct and close regions. Transshipments without the need to pass through hub nodes are possible inside sub-hubs. This way, congestions can be avoided and, consequently, delays are minimized. Four binary integer linear programming models for hub location and routing problem were developed in this thesis. Networks with sub-hub and networks without sub-hub taking into account circular hub routes or direct connections between hubs are compared. These models are composed of four sub-problems (location, allocation, service design and routing), which hinders the solution. A cutting plane approach was used to solve small instances of problem, while a Variable Neighborhood Decomposition Search (VNDS) composed of exact methods (matheuristic) was developed to solve large instances. The VNDS was used to explore each sub-problem by different operators. Major benefits are provided by models with sub-hub, thus promoting the development of more competitive networks.

Keywords: Hub-and-spoke, sub-hub, binary integer linear programming, cutting plane, VNDS, matheuristic.

Résumé

Le développement de structures de réseau efficaces pour le transport de marchandises est fondamental sur le marché mondial actuel. Les demandes doivent être traitées rapidement, répondre aux besoins des clients dans les meilleurs délais, les congestions et les retards doivent être minimisés, les émissions de $CO2$ doivent être contrôlés et des coûts de transport moins élevés doivent être proposés aux clients. La structure hub-and-spoke est un modèle de réseau courant utilisé à la fois dans le transport régional comme dans le transport intercontinental, permettant une économie d'échelle grâce aux consolidations opérées au niveau des nœuds hub. Mais, les retards, les congestions et les longs délais de livraison sont des inconvénients de ce type de réseau. Dans cette thèse, un nouveau concept, "sub-hub", est ajouté à la structure du réseau classique hub-and-spoke. Dans les modèles de réseau proposés, une économie d'échelle et des chemins alternatifs plus courts sont mis en œuvre, en minimisant ainsi le coût de transport et le délai de livraison. Le sub-hub est vu comme un point de connexion entre deux routes distinctes de régions voisines. Des transbordements sans passer par les nœuds hub sont possibles au niveau des sub-hubs. Des congestions peuvent ainsi être évitées et, par conséquent, les retards associés sont ainsi minimisés. Quatre modèles de programmation linéaire en nombres entiers binaires du problème de la localisation de hubs et de routage sont développés dans cette thèse. Des réseaux avec sub-hub et des réseaux sans sub-hub prenant en compte des routes circulaires entre hubs ou des connexions directes entre hubs sont ainsi comparés. Ces modèles sont composés de quatre sous-problèmes (localisation, allocation, conception de service et routage) qui rendent complexe la recherche de solutions. Une approche cutting plane est testée pour résoudre de petites instances de problème tandis qu'une recherche à voisinage variable avec décomposition (VNDS) composée de méthodes exactes (matheuristic) a été développée pour résoudre de grandes instances. Le VNDS mis en œuvre, explore chaque sous-problème avec différents opérateurs. Des gains importants dans la fonction objective sont observés par les modèles avec sub-hub confirmant ainsi le développement de réseaux plus compétitifs.

Mots clés: Hub-and-spoke, sub-hub, programmation linéaire en nombres entiers binaires, cutting plane, VNDS, matheuristic.

Acknowledgments

First, I would like to thank God for the health and the knowledge he has given to me, which allowed this new achievement in my life. I also thank Our Blessed Lady for the providence in each achievement.

I also thank professor Gilles Goncalves, my advisor in these years of Ph.D. studies. For me, each meeting was a moment of learning, both at academic and professional levels. A friend and professional partner that I will keep for life.

This Ph.D. degree had contributions from important institutions.

Therefore, I would like to thank the Coordination for the Improvement of Higher Education Personnel (CAPES), an institution from the Ministry of Education in Brazil, and the Federal Rural University of Semi-Arid (UFERSA), institution where I teach. Both institutions allowed for the financial support necessary to this study.

I also thank the Artois University and the Doctoral School SPI Lille Nord de France for the opportunity to be part of an excellent education.

The achievement of a Ph.D. degree is the result of the joint of positive efforts over a lifetime. For this reason, I thank:

My father, Francisco Fontes (in memoriam) for the life examples that guide me, as well as my mother, Terezinha Fontes, which strength represents my safe harbor;

My siblings, Denes and Fernanda, for the friendship and partnership over my life and my aunts, Geralda and Irene, my angels on Earth. I also thank my grandparents (in memoriam), specially my grandfather, Antônio, for his lessons;

My wife, Adriana, for her trust, fellowship and love. I also thank her relatives (parents and siblings) for their support and friendship;

Once again, to Adriana, for our daughter Gabriela, our greatest achievement. I thank Gabriela for filling our hearts with happiness, love and hope;

My relatives (uncles and aunts, cousins and godparents) and my friends because, without them, life would not be full;

The LGI2A family for the reception in these years of Ph.D. studies. Being part of this laboratory and having the friendship of each one of you was a key for my accomplishment;

And all professors I have ever had for their contributions in my educational background.

Finally, I would like to thank professors Andréa Cynthia Santos and Olivier Péton for the review of my thesis and professors Clarisse Dhaenens, Hamid Allaoui and Hifi M'Hand for accepting to attend the defense of this thesis.

I dedicate this thesis to my family (grandparents, parents, aunts and siblings), who contributed to the completion of my best dreams, and also to my wife and my daughter for providing new dreams to me.

Por mais terras que eu percorra,
Não permita Deus que eu morra
Sem que volte para lá;
Sem que leve por divisa
Esse "V" que simboliza
A vitória que virá:
Nossa vitória final,
Que é a mira do meu fuzil,
A ração do meu bornal,
A água do meu cantil,
As asas do meu ideal,
A glória do meu Brasil.
(A section from the song Expedicionário)

Contents

List of Figures

List of Tables

List of Algorithms

Introduction

The development of efficient network structures for transportation and communication is a fundamental factor in the current global market.

Millions of information are transmitted by optic fiber networks. People are transported by planes, trains, cars and vessels both at regional and continental level using service networks. Different types of goods, in increasing quantities each year, are distributed in all parts of the planet by pre-established service networks.

Different factors must be taken into account for demands to be fluid in a particular distribution/transportation network. Mainly, it will depend on the application field for which network is designed.

In telecommunications, information needs to be fluid readily, disruptions must be avoided and a large volume of demands needs to be backed by the network.

In passenger transport and freight transport, demands need to be fluid quickly and meet customer needs in the shortest possible time. Congestions and delays need to be minimized. Moreover, CO_2 emissions have to be controlled because increasingly stringent environmental rules must be respected.

However, for all application fields, a statement that is still unanimous is that direct connections between source-destination are not the most economical way to transport goods from one origin to one destination. Therefore, demands must be transported by sharing intermediate nodes (hubs) and/or intermediate links (arcs connecting hubs) in a network structure named "hub-and-spoke".

In a hub-and-spoke network, demands from different sources are sent to an intermediate point (hub). Inside this hub, demands headed to the same region/destination are consolidated and after that they are transported to their destination, or they are transported to another hub (intermediate destination) where they are disaggregated according to their destination and, only after, delivered to this one.

In this study, hierarchical hub-and-spoke networks are studied. Though they are also used in telecommunications (see Resende and Pardalos (2008)), we focus on the characteristics of freight networks.

Hub network structures are mainly found in air passenger and freight services, offering structure at global, national and regional levels (Rodrigue et al., 2017).

Hub nodes are also known as transshipment points because inside them demands are downloaded from one service or transport mode and uploaded to other service or transport mode. Multimodal transport can be represented in a hub-and-spoke network.

In unimodal hub-and-spoke networks, the transport times of each demand are increased if compared to a network with direct connections between one source node and one destination node. But, larger vehicles can be used in the transport of aggregated demands between hub nodes and, consequently, an economy of scale is offered for this larger volume of demands.

However, congestions are favored by these demands concentrated at hub nodes. These congestions together with long hub routes (characteristic of hub network) increase transport lead time.

Normally, vehicles are sped or omit visits at some nodes on the route trying to minimize delays in a transportation network. But, the speed-up of a vehicle is related to the increase of fuel consumption and, consequently, to the increase of CO_2 emissions. Whereas, delays of just some demands can be eliminated by the omission/postponement of service in specific points of a network.

CO_2 emissions in transportation network can be reduced by changing transportation mode, vehicle type, vehicle route or its loading (Martínez and Fransoo, 2017).

In Martínez and Fransoo (2017), it is also said that a good transportation performance in terms of costs and emissions is strictly determined by the design of network, more specifically, its hub location.

For this reason, economies of scale offered by hub-and-spoke structure are fundamental to achieve low costs in a transport network. But, currently, the level of customer expectations is not only driven by low transport costs, but also by reduced travel times, reduced CO_2 emissions, low incidence of delays and of disruptions (omitted nodes). Therefore, shortest paths need to be offered by a network.

The general objective of this thesis is to develop a new concept called "sub-hub" inside the classic hierarchical hub-and-spoke network. In this network, transport costs are minimized both by economy of scale provided by hub networks as by shorter alternative paths offered by sub-hub locations.

In a hub-and-spoke network with additional sub-hubs, transshipments are performed both at hub nodes as at sub-hub nodes. In a hierarchical structure with two levels, where the first level is hub level and the second one is cluster level (one hub and its associated spoke nodes), sub-hub nodes are represented by nodes allowing the connection between two close clusters.

In this research, the developed models we propose are composed of circular routes at cluster level. Regarding hub level, models with circular route and models with direct connections between hubs have been investigated.

Then, sub-hub node is more than a single node allocated to two hubs sending and receiving demands from them. Sub-hub is a node where flows from two close regions can be transported without the need to transit via the hub network, so a shorter alternative route can be achieved.

In order to show the efficiency of the sub-hub add in, integer linear programming models of hub-and-spoke with sub-hub and without sub-hub have been developed for a comparative analysis.

The proposed models integrate four sub-problems: hub location problem, spoke allocation problem, interconnection between nodes (named in this work "service design") and routing problem, together.

A cutting plane approach has been implemented with Cplex solver version 12.6.3[1] and the help of lazy constraint callbacks.

However, because of the complexity of the developed mathematical models, only results for small instances are achieved by exact methods.

Then, aiming to validate the efficiency of hub-and-spoke with sub-hub over the model of hub-and-spoke without sub-hub, a metaheuristic is proposed for solving large instances.

The metaheuristic chosen is the Variable Neighborhood Decomposition Search (VNDS), a variant of the Variable Neighborhood Search (VNS) proposed by Mladenović and Hansen (1997). In VNDS, the original problem is decomposed in small problems (a part of the original problem) and each part is improved by dedicated local search operators. The improved partial solution is inserted in the original problem and then, the corresponding solution is verified.

But, partial solutions will be only accepted if an improved solution is observed in the original problem.

This metaheuristic was chosen in this work because of the characteristics of the studied problem. As said before, a hierarchical hub-and-spoke problem can be decomposed in four sub-problems (location, allocation, service design and routing).

In hub location-routing problems, where network is not composed of direct connections hub-hub and/or hub-spoke, but connected by an incomplete graph, location and routing must be solved together because the outcome, which occurs from one problem influences/is influenced by the results of the other one.

So, in the implemented metaheuristic, each sub-problem is explored by different operators. Moving between operators is guided by neighborhood change sequential procedure (see Hansen et al. (2016)), allowing to be build a solution of global problem by solving four sub-problems at the same time.

Therefore, this thesis is organized as follows:

In chapter 1, a state of the art about hub-and-spoke network structure is presented.

[1] Cplex is an optimization software package of IBM

In this chapter, different forms of hub location, spoke nodes allocation, service designs and routing are presented. Finally, the hub location-routing problem is highlighted using networks with different levels and connections.

In chapter 2, liner shipping network problem is described through a published paper. Based on the characteristics of deep sea and short sea services presented in liner shipping problems, a hub-and-spoke network with sub-hub is applied. Mathematical model of hub-and-spoke with sub-hub is created to develop networks with circular routes both at hub level and at cluster level. Results of computational tests are compared with a developed version of network without sub-hub.

In chapter 3, integer linear programming models for traveling salesman problem, shortest path problem and p-median problem are presented. Characteristics of these problems are found in binary integer linear programming models developed for hub location-routing problem. In this chapter, new variants of previous developed models of hub-and-spoke with sub-hub and models without sub-hubs are presented. Now, clusters can be composed of isolated nodes (one hub), two nodes (one hub and one spoke) or more than two nodes. In this chapter, models with/without sub-hub are also presented with direct connections between hubs (complete graph).

In chapter 4, a constructive heuristic and a dedicated VNDS are described to provide solutions for larger networks. Exact methods are used in both algorithms. Therefore, the implemented VNDS is also classified as a matheuristic.

In chapter 5, results of computational tests for all four models introduced before (chapter 3) are presented. These results are obtained by using exact method (cutting plane approach) for small instances and by using matheuristic for small and large instances (Australian Post benchmark). Testing metaheuristic with small instances allows us to validate the quality of our implemented metaheuristic for the studied problem. Moreover, tests with large instances have contributed to strengthen the statement that the transport cost in a hub-and-spoke network is reduced by the use of sub-hub nodes.

In chapter 6, the main conclusions and perspectives for future works are finally presented.

List of Publications

Journals

- da Costa Fontes, F.F., Goncalves, G.: A new hub network design integrating deep sea and short sea services at liner shipping operations. Int. J. Shipping and Transport Logistics 9(5), 580-600 (2017).
- da Costa Fontes, F.F., Goncalves, G.: A Variable Neighborhood Decomposition Search Based Approach to Solve Two-Level p-Hub Location and Routing Problem. Journal of Heuristics. Under Review.

Conferences

- da Costa Fontes F.F., Goncalves G.: Hub Location and Routing Problem with Alternative Paths. 4th IEEE International Conferene on Advanced Logistic and Transport, ICALT'2015, Valenciennes, France, May, 2015.
- da Costa Fontes F.F, Goncalves G.: Routing Problem with Pendular and Cyclic Service in a Hierarchical Structure of Hub and Spoke with Multiple Allocation of Sub-Hubs. International Conference on Industrial Engineering and Systems Management, IESM'2015, Serville, Spain, October, 2015.
- da Costa Fontes, F.F., Goncalves, G.: A Variable Neighborhood Decomposition Search Applied to the Hierarchical Hub Location and Routing Problem. Sixth International Workshop on Model-based Metaheuristics, Matheuristics 2016, Brussels, Belgium, September, 2016.
- Fontes F.F.C., Goncalves G.: Hub Location and Routing Problem: a Variable Neighborhood Decomposition Search based solving approach. 6th International Conference on Metaheuristics and Nature Inspired Computing, META'2016, Marrakech, Morocco, October, 2016.

Chapter 1

Hub-and-Spoke networks: variants and components

An introduction about Hub-and-Spoke network structure is described in this chapter. Different problems of this network structure are presented.

In this context of hub-and-spoke network, the hub location problem, the hub network design problem and hub location-routing problem are highlighted.

In this study, problems of hub-and-spoke are classified by taking into account characteristics these problems above, allowing the identification of existing ideas and innovative ideas.

Methods used to solve different problems of hub-and-spoke are also described. The solving methods are divided in exact methods and heuristic methods. In this study, a combination of the two methods has been used to solve the identified problems proposed. Then, combinations of exact method with heuristic method existing in the hub-and-spoke literature are also highlighted.

Applications of hub-and-spoke network are pointed out.

1.1 Introduction

Initially presented by O'Kelly (1986), the mathematical model for design of a hub-and-spoke structure has been incremented by different characteristics, providing different variants of this problem.

Advances in Operations Research (location models) and increasing computing power allowed for studies about more complex hub location models and new algorithms approaches (Campbell and O'Kelly, 2012).

In hub-and-spoke like structures, requests from spokes belonging to the same hub are sent to this hub. Then, these demands are aggregated according to their destination, and are finally delivered along the route taken. This route can be composed of links between hubs (origin and destination nodes allocated to different hubs) or only of regional links (origin and destination nodes allocated to the same hub).

So hub nodes act as transshipment and switching points (Campbell, 1994) but they are also viewed as consolidation points for regrouping demands.

Economies of scale[1] are obtained in inter-hubs transport because of these consolidated flows. Scattered demand points costly to operate by a point-to-point service, are also favored by a hub-and-spoke network. Therefore, an economy of scope[2] can be offered because scattered demand points can be met (Lin, 2010).

Characteristics of hub-and-spoke are mainly based on location and allocation aspects. However, some hub-and-spoke problems are also composed of characteristics concerning the service provided by the network.

In hub location problem, the network in hub level is generally composed by direct links (Alumur et al., 2012a), (Nickel et al., 2001) and direct connections between spoke nodes are not allowed (Nickel et al., 2001). While in hub network design problem, a variant of hub location problem, connections between hubs and/or between hub and spokes can be like a ring or a tree because these latter constraints are relaxed (O'Kelly and Miller, 1994), (Alumur, 2009).

In these new network configurations, a tactical planning operation which includes the selection and scheduling of the services (routes) to operate, the specification of terminal operations, and the routing of freight, characteristics of Service Network Design Problem (SNDP) (Crainic, 2000), was included in hub-and-spoke problems.

In freight transport characterized by a network with terminals where cargo and vehicles are consolidated, grouped or simply moved from one service to another, the service design is an important factor for a company that supplies transportation services or controls routing of goods (Crainic and Laporte, 1997).

In Thomadsen (2005), a hierarchical network design problem is defined. This problem is composed of four sub-problems (hub location, allocation of nodes, interconnection of nodes at hub level and at cluster level and the routing) which are solved together.

Hub-and-spoke system was defined in O'Kelly (1998) as a hierarchical system with two levels, at least. One level is made by hub-hub connections and a second one is made by a spoke-hub connections.

Four attributes are used in Şahin and Süral (2007) to classify the hierarchical facility location problem. The same attributes are used in Torkestani et al. (2016) for the hierarchical hub network problem. These attributes are:

- Flow pattern:

[1] The economy of scale means that increases in load volume transported result in lower unit operating costs

[2] The economy of scope means that increases in load variety transported result in a lower average operation costs

  - single-flow – flows are sent from the first (last) level until the last (first) level, crossing all intermediate levels;
  - multi-flow – flows can be started at any level (lower or higher) and can be ended at any level (lower or higher) without necessarily crossing all intermediate levels.

- Service availability:
  - non nested hierarchy - different services are offered by each level;
  - nested hierarchy - in each level, it is offered a different service plus services provided by lower levels.

- Spatial configuration:
  - coherent system - each demand is met by services of a single hierarchy (for example, from lower level or from intermediate level);
  - non coherent system - each demand can be met by services of different hierarchies (for example, from lower level and from intermediate level).

- Objectives:
  - median models - in these models, facilities are located in such a way that minimizes transport cost of network;
  - covering models - the main characteristic is that they are composed of two different models. In the first model, a minimized number of facilities are located in such a way that all customers are covered (set covering) by one facility, at least. In the second one, the maximum number of customers are covered by a specific number of facilities (maximum covering);
  - fixed charge location models - costs to establish facilities and costs of transport are minimized.

Therefore, features like service design, routing and scheduling are likewise present in hub-and-spoke problems.

In the next sections, characteristics of hub location problem, hub network design problem and hub location-routing problem used for constructing different models of hub-and-spoke will be described. We try to point out differences and similarities at the hub-and-spoke problems from the literature and a classification is proposed. Finally, the solution methods used to solve different problems of hub-and-spoke structure will be presented.

1.2 Hub Location Problem

The hub location problem is concerned with locating hub facilities and allocating non-hub nodes (demand centers) to these located hubs in order to route the flow between origin–destination pairs (Alumur, 2009).

Main problem is to select hub nodes, to allocate spokes to hub and then routing the demands.

In O'Kelly (1987), the number of spoke nodes allocated to a hub is not limited. All spoke nodes can be allocated to just one hub. Each spoke node must be assigned to a single hub and the number of hub nodes is established. This problem was described as Hub Location Problem (HLP).

Hub location problem is defined in Alumur and Kara (2008) as a problem of location of specific facilities named hubs and of assignment of spoke nodes to these hubs allowing flow of demands. Hub location problem can be classified as a topic of location problem (Hekmatfar and Pishvaee, 2009).

In Hekmatfar and Pishvaee (2009) and Farahani et al. (2013), a classification for hub location problems is based on the characteristics below:

- Solution domain:
  - Network – the domain of nodes, where hubs are chosen, is composed by all nodes;
  - Discrete – the domain of nodes, where hubs are chosen, is composed by a set of particular nodes;
  - Continuous – the domain of nodes, where hubs are chosen, is composed by a plane or a sphere.
- Objective function criterion:
  - Mini-max – the maximum transport cost of demands is minimized;
  - Mini-sum – the cost of hub location plus the cost of spoke allocation is minimized.
- How to determine the number of hubs to locate:
  - Exogenous – the number of hub nodes is defined as a parameter of the problem;
  - Endogenous – the number of hub nodes is defined during the execution. It is a part of the solution.
- Number of hub nodes:
  - Single hub – the network is composed of just one single hub;

  - Multiple hubs – the network is composed of p hubs. Where p can be defined by an exogenous format or by an endogenous format.

- Hub capacity:
  - Capacitated – each hub node has a limited capacity;
  - Uncapacitated – each hub node has an unlimited capacity.
- Cost of hub nodes:
  - No cost – the cost to open a hub node is not applied;
  - Fixed cost – the cost to open a hub node is defined as a parameter of the problem;
  - Variable cost – the cost to open a hub node is defined during execution.
- Allocation of a spoke node to hub nodes:
  - Single allocation – each node can be allocated to just one hub;
  - Multiple allocation – each node can be allocated to more than one hub.
- Cost of spoke allocation:
  - No cost – the cost to link a spoke node to a hub node is not applied;
  - Fixed cost – the cost to link a spoke node to a hub node is defined as a parameter of the problem;
  - variable cost – the cost to link a spoke node to a hub node is defined during the execution.

These characteristics have defined different problems in the literature.

Location in Hub location problem

exogenous location

In O'Kelly (1987), a hub location problem was presented with an exogenous characteristic. Besides that, all nodes are candidates to hub node. Currently, these characteristics are seen in several works (see for example Yaman (2011), EghbaliZarch et al. (2013), and Mahmutoğulları and Kara (2015)).

Exogenous characteristic of hub location problem is represented by a constraint (constraint 1.1).

Conventional notation used in this chapter: let a network represented by a complete graph $G = (V, E)$ where V is the set of nodes with n nodes and E is a set of edges with m edges.

$$\sum_{j=1}^{n} x_{jj} = p, \tag{1.1}$$

where x_{jj} is a binary variable that receives 1 if node $j \in V$ is allocated to itself (it is a hub node) or 0 if j is not a hub node. Parameter p is a parameter (integer) of the problem which sets the number of hub nodes to open.

In Campbell (1994), p-hub center problem is created as an analogous to the p-center problem. In that model, a mini-max objective function is presented for a discrete set of hubs.

endogenous location

Note that constraint (1.1) or its variants are removed in mathematical models where an endogenous characteristic is presented.

In O'Kelly (1992), constraint (1.1) is removed. But, the cost for installation of a hub node (a parameter of the model) is inserted in the objective function trying to minimize this cost and, consequently, to limit the number of hub nodes. This problem was described by Campbell (1994) as analogous to the uncapacitated facility location problem.

In hub covering problem proposed by Campbell (1994), cost for hub installation is minimized. In this problem, each demand point (each origin and destination) needs to be covered by a hub pair.

In Marianov et al. (1999), an endogenous characteristic is presented by an objective function in which the sum of the expected annual revenue from flow minus the sum annual amortized opening and continuing costs for each potential hub are maximized in a hub location competitive model.

Also in Marianov et al. (1999), an endogenous characteristic is presented with a variation of constraint (1.1) because its right-hand-side is modified. In their model, hubs are relocated and new hubs are added (total of hubs is increased). Then, right-hand-side is composed by the number of current hubs (hubs will be relocated or not) plus the number of new hubs.

In Yaman and Carello (2005), the quantity of traffic transiting through the hub is limited by the capacity of hub. But, the number of hubs is also limited by objective function because the cost of hub installation is inserted.

In Horner and O'Kelly (2001), an endogenous location of hubs and an endogenous discount factor level in links of network are both applied. Hub location is established by the minimization of transport cost and the discount factor applied in all edges of network is a consequence of the quantity of flow that circulates along the edge.

In problems where the number of hub nodes is determined by an endogenous form, the number of hub nodes are defined by different factors, as for example: capacity of hubs, hub installation costs and transport cost.

An other important characteristic presented in hub location problems is allocation. Single and multiple allocation of spoke nodes to hubs are used in different works in literature.

Allocation in Hub location problem

In all figures from this chapter, regional links (links between spokes or links spoke-hub) are represented by full lines, hub links (links hub-hub) are represented by dashed lines, hub nodes are represented by squares, spoke nodes are represented by circles and each region (or cluster) is represented by a different color.

single allocation

The single allocation of spoke nodes is represented at Figure 1.1. Constraints (1.2) and (1.3) are inserted in mathematical models to allow this characteristic.

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad \forall i \in V \tag{1.2}$$

$$x_{ij} \leq x_{jj}, \qquad \forall i, j \in V \tag{1.3}$$

where x_{ij} is a binary variable that receives 1 if the node $i \in V$ is allocated to the hub node $j \in V$ or 0 otherwise. In the constraint (1.2), each node i is allocated to just one hub j. And in the constraint (1.3), one node i is only allocated to a node j if j is a hub node ($x_{jj} = 1$).

Single allocation can be seen, for example, in Ernst and Krishnamoorthy (1996) for uncapacitated single allocation p-hub median problem and in Alumur et al. (2012b) for single allocation uncapacitated hub location problem with uncertainty in set-up costs or uncertainty in demands.

In the single allocation problem presented by Skorin-Kapov et al. (1996), constraint (1.3) is removed. Flow variable w_{ijkl} is used under the condition that i is allocated to hub k and j is allocated to hub l (constraints (1.4) and (1.5)).

$$\sum_{l=1}^{n} w_{ijkl} = x_{ik}, \qquad \forall i, j, k \in V \tag{1.4}$$

$$\sum_{k=1}^{n} w_{ijkl} = x_{jl}, \qquad \forall i, j, l \in V \tag{1.5}$$

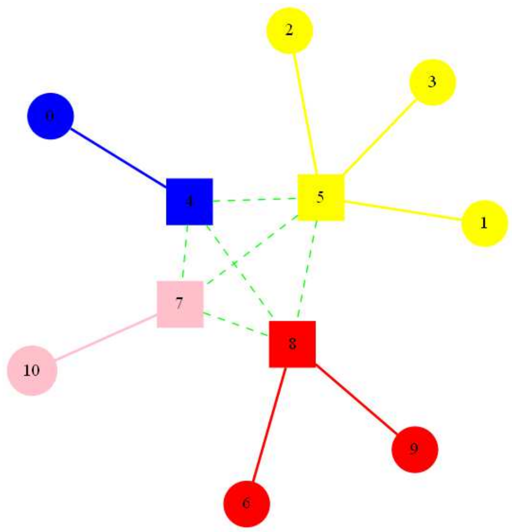


Figure 1.1: Hub-and-spoke network with single allocation

where w_{ijkl} is the percentage of flow from i to j routed through hub nodes k and $l \in V$.

In Yaman and Carello (2005), a capacitated single assignment hub location problem with modular link capacities is showed. In this model, allocation cost and cost of arc installation are minimized.

In Elhedhli and Hu (2005), uncapacitated single assignment hub location problem with congestion is presented. This model is obtained by the incorporation of a non-linear cost term, representing traffic congestion, in the objective function of uncapacitated single assignment p-hub location problem.

multiple allocation

In O'Kelly et al. (1996), models with single allocation and with multiple allocation are compared. In the second one, the flow between every origin-destination is routed between some hub pair (constraint (1.6)).

$$\sum_{k,l=1}^{n} w_{ijkl} = 1, \qquad \forall i,j \in V \tag{1.6}$$

Example of multiple allocation can be seen on Figure 1.2.

In Figure 1.2, multiple allocation is represented by circle nodes with two colors.

In Parvaresh et al. (2014), a multiple allocation p-hub median problem under intentional disruptions is presented as an integer bi-objective bi-level programming problem. In this mathematical model, hubs are chosen in the first level of problem to minimize two objective function (minimize transport cost in a normal situation and minimize transport

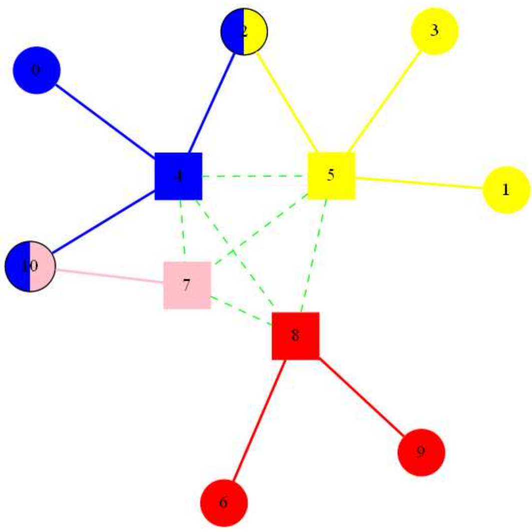


Figure 1.2: Hub-and-spoke network with multiple allocation

cost after r-interdiction of hubs). While in the second level, hubs are selected for interdiction to maximize the damage to the system. Constraint (1.6) is represented at first level of the problem (for a situation without disruption) and a similar constraint, using a similar variable, is represented at the second level.

However, in Parvaresh et al. (2014), the variable w_{ijkl} used in the first level and a similar variable used in the second level of the problem are binary variables. Then, despite multiple allocation, one demand can not be divided in fraction parts that cross different hub pairs. Each demand is assigned to a single hub pair.

In the uncapacitated r-allocation p-hub median problem presented by Yaman (2011), constraints (1.3) and (1.6) are used in the model and constraint (1.2) is replaced by constraint (1.7).

$$\sum_{j=1}^{n} x_{ij} \leq r, \qquad \forall i \in V \tag{1.7}$$

where r, in constraint (1.7), is a constant of problem representing number of j hub nodes that a spoke node $i \in V$ can be assigned to.

In this model, although the number of hubs to allocate each spoke node is defined during execution, a maximum number allowed can be defined before, allowing a single allocation ($r = 1$), a full allocation ($r = p$, where p is the number of hubs) or an intermediate allocation value ($1 < r < p$).

Routing in Hub location problem

Mathematically speaking, after location and allocation are defined, flows can be routed. Direct connections hub-hub and hub-spoke and the non connection spoke-spoke, two characteristics presented in hub location problems, simplify the demand routing.

Campbell (1994) used the constraints (1.8) and (1.9) for discrete hub location problems (p-hub median problem, uncapacitated hub location problem, p-hub center problem and hub covering problem).

$$w_{ijkl} \leq x_{kk}, \qquad \forall i, j, k, l \in V \tag{1.8}$$

$$w_{ijkl} \leq x_{ll}, \qquad \forall i, j, k, l \in V \tag{1.9}$$

Fraction of flow w_{ijkl} from node i to node j is routed through k and l if k and l are selected hub nodes.

In Skorin-Kapov et al. (1996), constraints (1.8) and (1.9) are replaced by constraints (1.10) and (1.11), offering a model with a tighter linear relaxation.

$$\sum_{l=1}^{n} w_{ijkl} \leq x_{kk}, \qquad \forall i, j, k \in V \tag{1.10}$$

$$\sum_{k=1}^{n} w_{ijkl} \leq x_{ll}, \qquad \forall i, j, l \in V \tag{1.11}$$

However, constraints (1.10) and (1.11) are used when the inter-hub network is composed of a full interconnected graph.

New constraints and variables were incorporated in the mathematical model of hub location problem and different network structures have been developed. In next section, different network structures format are presented.

1.3 Hub Network Design Problem

O'Kelly and Miller (1994) defined hub network design problem as a problem that involves finding optimal hubs location, assigning spoke nodes to hub and determining linkages between nodes and flows routing.

Hub network design problem are hub location problem because location and allocation are present in the models. But, the problems are composed of hierarchy of hubs, choice of inter-hub links and connection between spokes.

In Alumur et al. (2012a), a multimodal hub location and hub network design problem is built with a variation of the constraint (1.1). In this variation, nodes j are selected from a set of candidates hub nodes (discrete solution domain).

There are problems where hub nodes are divided in different categories of hub, consequently, the location of hub is made at different levels. In Alumur et al. (2012c), the set of hubs is divided in hub, airport hub and central airport hub. A single central airport hub is firstly located. A group of airport hubs are located and assigned to the central airport hub and a group of located hub nodes are allocated to each airport hub. Therefore, a single location is realized at the first level (one central airport hub) and multiple locations are performed at the other levels (p-airport hub and p-hubs).

Currently, models of hub-and-spoke network are not only composed of characteristics, such as direct connections between hubs and no link between spoke nodes. Now, these structures are also modeled with different layers, without full connections at hub level, with links between spoke nodes at regional level allowing the creation a circular route (as in the traveling salesman problem solution) or to create routes (as in the vehicle routing problem) inside each cluster (region), etc.

These characteristics have generated different kinds of network structures, offering the routing of demands not only directly or with one hub stop or two hub stops, but with different configurations.

Service design in Hub network design problem

In Wieberneit (2008), the transportation of freight between two spokes, between one spoke and one hub or between two hubs is defined as a service. Service route was defined as a sequence of services using a specific fleet of vehicles. A service network is made by the union of all service routes.

In what follows, we describe this different configurations used to implement services in hub-and-spoke network.

In hierarchical structures like star shape network (Figure 1.3), hubs from a layer are linked to nodes of a lower layer by direct connections. Nodes from the same layer can be connected by direct links, but this characteristic does not need to appear at each layer.

Hierarchical star network structures can be seen in Yaman (2009) and Alumur et al. (2012c).

In Figure 1.3, a second class (type) of hub nodes are represented by triangles.

In Alumur et al. (2012c), a multimodal service is presented with three layers. At lower layer, demands between spoke node and hub node are transported by highway services. At intermediate layer, demands between hub node and airport hub are also transported by highway services. In this second layer, demands between hub nodes connected to the same airport hub can have direct transport by highway services without crossing airport hub. At top layer, demands between airport hubs and a central airport hub are transported by airline services. In this problem, the top level is composed of a single node.

In Yaman (2009), star structure is composed of a top hub layer fully connected hubs (similar to Figure 1.3).

The scheduling of demands is verified in Yaman (2009) and also in Alumur et al.

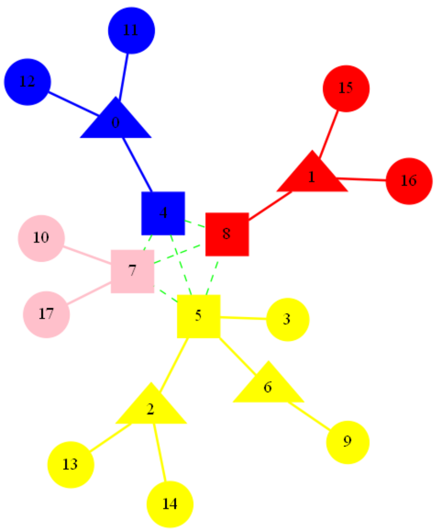


Figure 1.3: Hierarchical star shape network structure

(2012c). Departure time variables and arrival time variables are used in constraints to create a transport connection between nodes from different layers.

In Arshadi Khamseh and Doost Mohamadi (2014), the idea of incomplete hub network was used at top hub level of a star network structure. The model was named incomplete hierarchical hub center network problem with single assignment.

A tree structure at hub level is used in Contreras et al. (2009), Contreras et al. (2010) and de Sá et al. (2013). The model named tree of hubs location problem is composed of a small tree, which is made by hub nodes and a large tree. This large tree is composed by a small tree with an extension made by single allocation of spoke nodes to hubs (Figure 1.4).

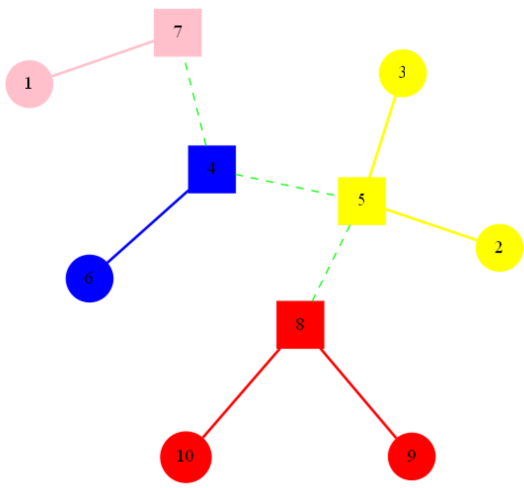


Figure 1.4: Tree of hubs network structure

The same tree network structure has been also used by Kim and Tcha (1992) and Lee et al. (1996). But, in these works, the objective function focused on allocation and installation costs, whereas in previous work, the objective function focused in allocation and routing costs.

In Alumur et al. (2012a), hubs are established with specific transportation modes. Then, two hubs using the same transportation mode can be linked. Links between all hub nodes for a specific transport mode, named ground transport mode, can also be presented. Then, different services are established by using different links.

Distinct facilities use hierarchical network structures for transport or for distribution of demands. A classical problem in literature is the n-Echelon Vehicle Routing Problem (nE-VRP). In this problem, customers are served by a central depot via n-stage distribution network (Drexl and Schneider, 2015).

In two-echelon Vehicle Routing Problem (2E-VRP), for example, the network is composed of a single depot connected to one or more intermediate depot, named satellites, by circular routes. These satellites are also connected to customers by circular routes (see Crainic et al. (2008) and Perboli and Tadei (2010)).

Different from network structures using hub facilities, in 2E-VRP, demands flow only in one direction (from central depot to end customers). Satellites are responsible for splitting the demand received from depot, allowing to the use of different routes (satellite-customers) to deliver the demand of different customers.

Although the depot-satellite level allows the connection between satellites by using vehicles, all received demand in each satellite is originated from depot. Unlike the hub-and-spoke problem that defines a general transport network (many-to-many), a 2E-VRP is used as a goods distribution network (one-to-many).

A literature review about 2E-VRP can be seen in Drexl and Schneider (2015).

Therefore, in the hub-and-spoke network literature, services are represented by different link connections, different levels and different modes of transport. But, demands are routed in all directions, because hub nodes are not only source nodes as in a factory or a warehouse, but are also transshipment points for the routing of flows. Links between hub facilities allow for transport demands from different levels of network.

Routing in Hub network design problem

Characteristics used for network design are linked with the ones from the routing of demands in a similar way as characteristics of spoke allocation are attached with the ones from hub location.

In different problems from hub-and-spoke structure literature, routing can be performed by visiting one hub, two hubs, or more. But in freight transportation, the more hub nodes are used in the path, the more transshipments, sorting operations and coordination between demand points and hubs are required (Yaman, 2009).

Mathematical models were also modified with the insertion of new variables and constraints, allowing for the representation of the network design services and, consequently, the routing in the different parts of the network structures.

Variable w (see constraints (1.8), (1.9), (1.10) and (1.11)) is not enough in networks with direct connections between spoke nodes and/or without fully interconnected hub network (Nickel et al., 2001).

In Nickel et al. (2001), a mathematical model using a network design formulation is presented and, besides variable w, variables s_{ijkl}, y_{kl} and z_{kl} are introduced.

s_{ijkl} is the fraction of flow between nodes i and j routed via spoke link $(k, l) \in E$, y_{kl} is a binary variable equal to 1 if hub edge $(k, l) \in E$ is used in the network or 0, otherwise. And z_{kl} is a binary variable equal to 1 if non hub edge $(k, l) \in E$ is used in the network or 0, otherwise.

New constraints were presented by Nickel et al. (2001) allowing the flow of demands.

Flow conservation constraints are the following:

$$\sum_{l=1}^{n}(w_{ijkl} + s_{ijkl} - w_{ijlk} - s_{ijlk}) = 1, \qquad \forall i, j, k \in V : k = i, i \neq j \tag{1.12}$$

$$\sum_{l=1}^{n}(w_{ijkl} + s_{ijkl} - w_{ijlk} - s_{ijlk}) = -1, \qquad \forall i, j, k \in V : k = j, i \neq j \tag{1.13}$$

$$\sum_{l=1}^{n}(w_{ijkl} + s_{ijkl} - w_{ijlk} - s_{ijlk}) = 0, \qquad \forall i, j, k \in V : k \neq i, k \neq j, i \neq j \tag{1.14}$$

Constraints (1.12), the flow of every vertex i towards every vertex j will leave the source node using the arc (i, l), where the flow can be a flow represented by variable w (flow in a arc-hub) or a flow represented by variable s (flow in a arc-non-hub).

Constraints (1.13), the flow of every vertex i towards every vertex j will be received in the destination node using the arc (l, j), where the flow will can be a flow represented by variable w (flow in a arc-hub) or a flow repesented by variable s (flow in a arc-non-hub).

Constraints (1.14), all flows that leave an intermediate vertex k (which is not the source or the destination), will be equal to the flow that has arrived at the intermediate vertex.

Flow at hub level is routed only via hub edges:

$$w_{ijkl} + w_{ijlk} \leq y_{kl}, \qquad \forall i, j, k, l \in V \tag{1.15}$$

Flow at regional level is routed only via spoke edges:

$$s_{ijkl} + s_{ijlk} \leq z_{kl}, \qquad \forall i, j, k, l \in V \tag{1.16}$$

Hub edges link only hub nodes:

$$y_{kl} \leq x_{kk}, \qquad \forall k, l \in V \tag{1.17}$$

$$y_{kl} \leq x_{ll}, \qquad \forall k, l \in V \tag{1.18}$$

The hub network structure presented in Nickel et al. (2001) was used to represent a public transport problem. Demands are routed using the shortest paths due to constraints (1.15) and (1.16). This characteristic makes sense in some network modes, for example, public transport network or in communication network, but not for all modes, such as liner shipping transport network.

In public transport network, generally, a regular service is performed by vehicles in both directions of an edge. In communications network, a signal is routed in the edges in both directions. Therefore, in a circular network, different demands can be routed, both in a clockwise as counterclockwise direction(Figure 1.5).

Generally, in a liner shipping transport network (see Christiansen et al. (2013) and Brouer et al. (2013)), goods are transported by vehicles performing a circular route in just one direction. One node is visited again after all other nodes had been visited. Then, in this situation, directed edges need to be created.

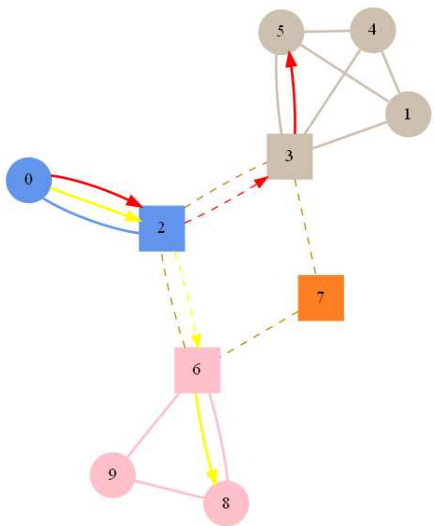


Figure 1.5: Network with demands between the nodes 0-5 and 0-8

In Figure 1.5, demand between nodes 0 and 5 are represented by directed red lines and demand between nodes 0 and 8 are represented by directed yellow lines.

In mathematical model of hub network design problem with profits, presented by Alibeyg et al. (2016), only demands offering a profit are routed (profit-oriented models). Service-oriented models are also showed. In this second case, even a fraction of demands not offering profits are routed, trying a penetration in the market by the company.

The concept of graph partitioning is used in the partitioning-hub location-routing problem (Catanzaro et al., 2011). In this problem, a given network is partitioned in sub-networks and one hub node is selected in each sub-network. The flow between two nodes can be directly routed by spokes from the same sub-network, but it needs to use hub nodes if the demand is between two different sub-networks. Demands between two non neighbor sub-networks are routed by several hub nodes.

In the next section are presented problem in structure hierarchical, but with two important characteristics: location problem and vehicle routing problem.

1.4 Hub Location-Routing Problem

Hub location-routing problem involves hub location planning with tour planning. This tour can occur in the various layers.

In Nagy and Salhi (2007), some hub problems are grouped in a special class of Location and Routing Problem (LRP). In these location routing problems, facilities (hubs) are connected to each other by routes. This special class is divided in four categories of problems:

- Transportation-location problem: connections inside clusters and connections between hubs are composed of direct links;
- Many-to-many location-routing problem: hub-spoke connections are composed of tours and hub-hub connections are composed of direct links. Tour planning is defined by Nagy and Salhi (2007) as a route with multiple stops;
- Vehicle routing-allocation problem: connections inside the clusters are composed of direct links and connection hub-hub are composed of tours. It is a particular variant of VRP problem in which not all customers need to be served by vehicles;
- Multi-level location-routing problem: connections inside the clusters and hub-hub are composed of tours.

In this section, Hub location-routing problem are based in Nagy and Salhi (2007). The combination of different configurations between networks in hub level and in regional level allows the different categories of hub location-routing problems.

Transportation-location problem

The Fully Interconnected Network Design Problem (FINDP) is composed of a totally interconnected network at hub level. Each hub node is inserted in a different cluster and each cluster is a disjoint set of nodes (single allocation). Nodes belonging to the same hub are fully interconnected (Figure 1.6).

This problem is showed by Thomadsen and Larsen (2007), where the network at hub level is named backbone network. The authors included a maximal and minimal number of clusters (consequently, the number of hub nodes). Each cluster is also composed of a minimal and maximal number of nodes. The fully interconnected network inside the clusters are named access network. Access network and backbone network are usually used in the literature about telecommunication network.

Still in Thomadsen and Larsen (2007), two variables are used to represent links. One variable representing links in backbone network and the other one in access network of the presented problem. The problem is described by the authors as a hierarchical problem with two layers, however, no flow variables are registered in the corresponding model.

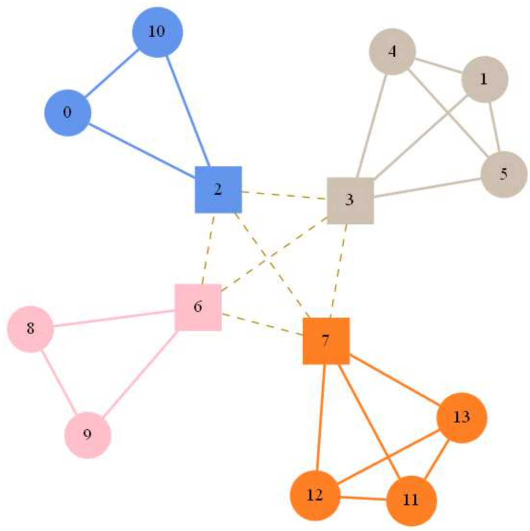


Figure 1.6: Hierarchical network fully connected at hub and cluster levels

In Saboury et al. (2013), it is presented a new mathematical model for the fully interconnected network design problem proposed by Thomadsen and Larsen (2007), in which the number of variables and constraints are minimized.

A hub location and routing problem is presented by Aykin (1995), but both direct connection hub-spoke and hub-hub as some direct connections source-destination (from different clusters) are used by the author. The service is offered by a direct connection source-destination just for a set of predetermined demands.

In Mahmutoğulları and Kara (2015), a network similar to the one presented by Aykin (1995) is used. Alternative paths are offered by source-destination direct links between spoke nodes. In these problems, spoke links are not parts (intermediate links) of a path for demand deliveries, only a direct connection from a spoke source to a spoke destination.

Many-to-many location-routing problem

Similar to one of the network structures presented in our study, Rodríguez-Martín et al. (2014) showed a hub network with direct connection between hubs and one circular route inside disjoint clusters. Other similarities are clusters composed of a single hub node, clusters with two nodes (one hub and a single spoke node) and clusters with more than two nodes making a circular route (Figure 1.7).

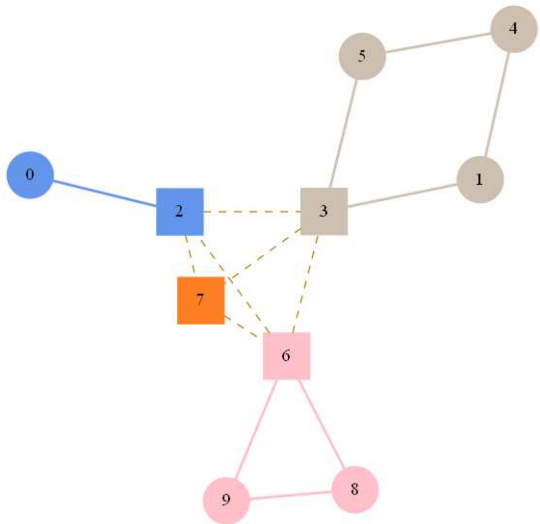


Figure 1.7: Network with direct hub links and circular routes in clusters

In Rodríguez-Martín et al. (2014), a variations of constraint (1.1) is presented in the mathematical model of a hub location and routing problem (HLRP). The left-hand-side of constraint (1.1) is divided in hub with no allocated spoke nodes, hub with only one spoke node allocated and hub node with two or more spoke nodes allocated.

However, the model developed by Rodríguez-Martín et al. (2014) is composed of a single allocation of spoke nodes to a hub and of capacity constraint limiting the number of arcs in a route, consequently, the number of nodes, which compose the cluster.

Nagy and Salhi (1998) proposed the many-to-many hub location-routing problem (MMHLRP). This problem is a two-level hub-and-spoke with direct connections at hub level. In each cluster, routes can be created as in a vehicle routing problem (Figure 1.8).

In the model presented by Nagy and Salhi (1998), demands are firstly collected, then, they are aggregated in the hub node and only then delivered. Therefore, demands with source and destination in the same cluster (same route of a cluster) are also initially transported to hub and, only after, delivered to their respective destinations.

In de Camargo et al. (2013), a new formulation for many-to-many location-routing problem was proposed. Pickup and delivered services are made by the same vehicle route. Different from (Nagy and Salhi, 1998), in this model version presented by de Camargo et al. (2013), a discount factor is used at hub level.

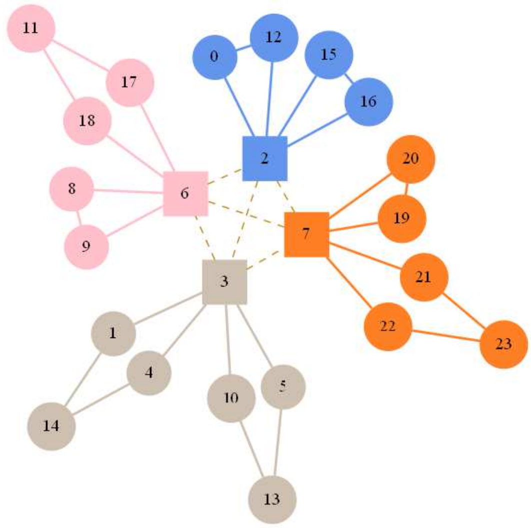


Figure 1.8: Network structure in many-to-many hub location-routing problem

A network model similar to the one presented in de Camargo et al. (2013) is used by Bostel et al. (2015). Hub nodes are chosen from a set of candidates nodes. The problem was classified as Hub Location Routing Problem (HLRP).

In many-to-many location-routing problem, nodes visited by vehicle routes can only be a pickup service, a delivery service or it can be a pickup and delivery service at the same time (Nagy and Salhi, 1998), (Rieck et al., 2014).

In Sun (2016), a network similar to the one presented in Figure 1.8 is composed of pickup service routes and delivery service routes. Services are done in different routes because the set of nodes used in pickup service is different from the set of nodes used in delivery service. The author defined the problem as Hub Location-Routing Problem (HLRP).

A hierarchical structure with multiple allocation is showed in Çetiner et al. (2010). The corresponding mathematical model is not presented, but an iterative two-stage procedure is described. In the first stage, hub location and multiple allocation of spoke nodes are performed. In the second stage, vehicle routing problem is solved inside each cluster. Different clusters are connected by direct hub arcs, but they can also be connected by one common node or one common arc (two nodes allocated to the same hub) (see Figure 1.9).

Multi-level location-routing problem

In Lopes et al. (2016), it is also presented a network structure similar to the one presented in our study. Hub level and clusters level are composed of circular routes. Each cluster is a disjoint set (single allocation of nodes) that can be composed of one node (only the hub), two nodes (hub and one spoke) or more (Figure 1.10).

However, as well as the model presented by Rodríguez-Martín et al. (2014), the num-

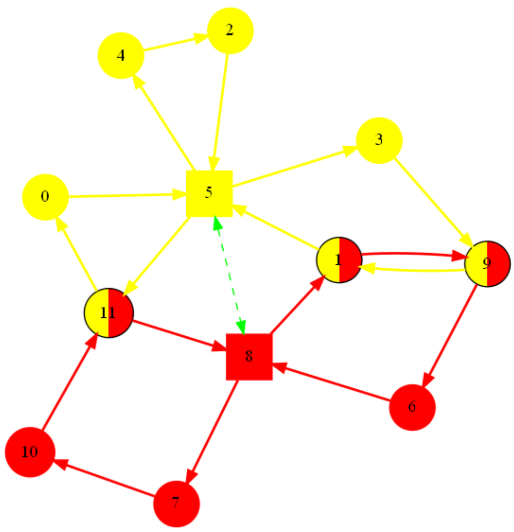


Figure 1.9: Hub network with circular regional tours and multiple allocation

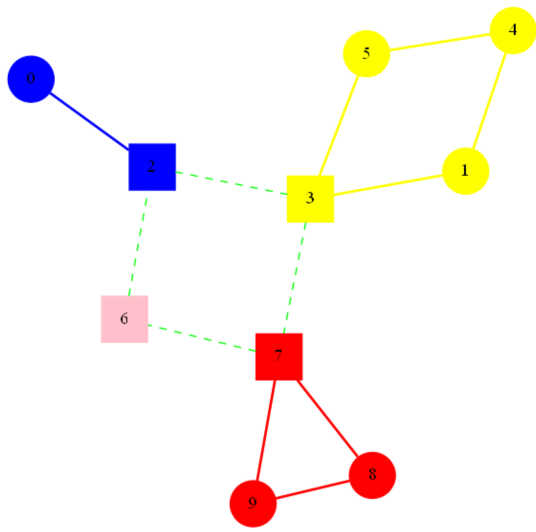


Figure 1.10: Circular routes both at hub and cluster levels

ber of nodes inside each cluster is also limited in Lopes et al. (2016).

In Lopes et al. (2016), the presented model was called Many-to-Many p-Location-Hamiltonian Cycle Problem (MMpLHP).

In telecommunication literature, networks with circular route at hub level and circular routes at cluster level are named Hierarchical Self-Healing Ring Networks (see Shi and Fonseka (1994)).

In ring networks, demands can be rerouting when a failure occurs in a link or node. Detection of failure and traffic rerouting at ring network can be achieved in a fraction of a second (Carpenter and Luss, 2006), when using the ring architecture named Synchronous Digital Hierarchy (SDH), or Synchronous Optical Network (SONET) in United States (Henningsson et al., 2006).

In Altinkemer (1994), a study concerning an access network (regional network) is presented. The mathematical model optimizes the flow inside the clusters by making several rings for each cluster. The authors classified the problem as similar to the multi-depot vehicle routing problem.

A study regarding access network is also done in Kang et al. (2000). The mathematical model allows the cluster to be composed of one or two hubs.

In Rodríguez-Martín et al. (2016a), Hierarchical Ring Network Problem (HRNP) is studied. Each cluster can be composed of q nodes (maximum) and only network design is approached (flow variables are not present).

While in Rodríguez-Martín et al. (2016b), a second model of hierarchical ring network problem is presented. In this second paper, the problem is described as ring/k-rings because one ring is made between hubs and k-rings can be made for each cluster. In each cluster, at least one ring is created. Similar to Rodríguez-Martín et al. (2016a), each cluster is composed of q nodes (maximum) and flow routing is not studied.

Hierarchical ring network with one circular route at hub level and one circular route for each cluster is presented in Thomadsen (2005). Location, allocation, service design and routing are solved by two mathematical models (ring partitioning and the ring-generation problem.). The first problem is a set-partitioning problem to choose the lowest cost subset of rings. While in the second problem, tour and routing are made. Clusters are composed of a limited number of nodes.

Vehicle routing-allocation problem

Following with different network structures, note that in Nagy and Salhi (2007) vehicle routing-allocation problem was also defined as a category of hub location-routing problem. In this category, a network structure is composed of a circular route only at hub level (Figure 1.11).

In Contreras et al. (2016) a network with direct hub-spoke connections and circular route at hub level is used. The network is described by the authors as a cycle-star network. A cycle hub location problem (CHLP) is presented in their works.

Routing in Hub location-routing problem

Now, flows in hub-and-spoke structure are routed by one-stop service (one hub is visited in the route), two-stop service (two hubs are used in the route), multiple stops in tours or no-stop service in Aykin (1994), Aykin (1995). In the fourth case, a set of routes allowing direct transporting are given as parameters of the hub location and routing problem.

In Sung and Jin (2001) and in Wagner (2007), a network divided in clusters is given. Inside each cluster, a hub node is selected and flows are routed by hub nodes or by direct links (source-destination). In this Cluster Hub Location Problem (CHLP), transport cost

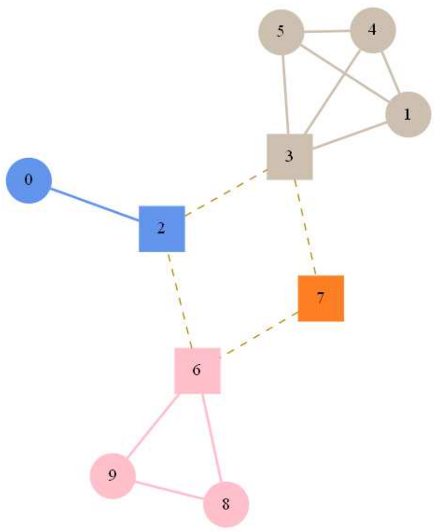


Figure 1.11: Network with circular hub route and direct links in clusters

using hub links, transport cost using direct links, as well as the fixed cost to install a hub in the selected node are minimized.

In this chapter, several hub and spoke problems have been presented. In the next section, a classification for the presented problems is showed (see Table 1.1). This classification is based on the identified problem (hub location problem, hub network design problem, hub location-routing problem).

1.5 Hub-and-spoke problems classification

Based on the presented studies, different characteristics of hub-and-spoke problem have been developed. In this section, these works and main characteristics are grouped in three different problem classes.

In Table (1.1), the first column is related to the categories of problems, second column concerns the main characteristics presented and third column is composed by different works presented before. Each line from Table (1.1) represents a different classification group.

Characteristics presented in column two are not all used in the same problem. Each problem uses a combination of characteristics.

However, hierarchical structures of hub location-routing problem combining direct connection spoke-spoke and multiple allocation were neglected by researchers (see Table 1.2).

Table 1.2 is based on the four categories of problems of hub location and routing problem presented in Nagy and Salhi (2007).

The models developed in our study are included in two different categories of Table 1.2:

many-to-many location-routing problem and multi-level location-routing problem.

Table 1.1: Proposed Hub-and-spoke classification

Categories of problem	Main characteristics	Works of literature
Hub Location Problem	- associated sub-problems: location, allocation, routing; - direct hub link; - no spoke-spoke link; - single allocation, multiple allocation.	- single allocation uncapacitated hub location problem (Alumur et al., 2012b) - multiple allocation p-hub median problem (Parvaresh et al., 2014) - single allocation p-hub median problem (O'Kelly, 1987) - capacitated single allocation hub location problem with modular link capacities (Yaman and Carello, 2005) - uncapacitated hub location problem with single allocation (Cunha and Silva, 2007) - uncapacitated r-allocation p-hub median problem (Yaman, 2011) - robust bi-objective uncapacitated single allocation p-hub median problem (Amin-Naseri et al., 2016) - reliable single allocation hub location problem (An et al., 2015) - reliable multi allocation hub location problem (An et al., 2015) - hub covering location problem (EghbaliZarch et al., 2013) - p-hub center problem (Campbell, 1994) - uncapacitated single assignment hub location problem with congestion (Elhedhli and Hu, 2005)
Hub Network Design Problem	- two or more hierarchical levels; - direct hub connection is relaxed; - spoke-spoke link is allowed; - associated sub-problems: location, allocation, service design, routing; - single allocation.	- multimodal hub location and hub network design problem (Alumur et al., 2012a) - tree of hubs location problem (Contreras et al., 2009), (Contreras et al., 2010), (de Sá et al., 2013) - partitioning-hub location-routing problem (Catanzaro et al., 2011) - incomplete hierarchical hub center network problem with single assignment (Arshadi Khamseh and Doost Mohamadi, 2014) - hierarchical multimodal hub location problem with time-definite deliveries (Alumur et al., 2012c) - hierarchical hub median problem with single assignment (Yaman, 2009)
Hub Location-Routing Problem	- two hierarchical levels; - associated sub-problems: location, allocation, tour service design, routing; - single allocation, multiple allocation; - direct hub connection is relaxed; - spoke-spoke link is allowed.	These problems are subdivided in Table 1.2 based on Nagy and Salhi (2007) classification

Our developed models are named: Many-to-Many p-Hub Location-Routing Problem with Sub-Hub (MMpHLRPSH), Many-to-Many p-Hub Location-Routing Problem (MM-pHLRP), Two-Level p-Hub Location-Routing Problem with Sub-Hub (tLpHLRPSH) and Two-Level p-Hub Location-Routing Problem (tLpHLRP).

Table 1.2: Hub location and routing problems based on Nagy and Salhi (2007) categories

Classification	single allocation	multiple allocation
Transportation-location problem	(Direct connection) - FINDP (Thomadsen and Larsen, 2007) (Saboury et al., 2013) * no flow routing - CHLP (Sung and Jin, 2001) (Wagner, 2007)	- HLRP (Aykin, 1995)
Many-to-many location-routing problem	(TSP inside clusters) - MMpHLRP Our fourth proposed problem - HLRP (Rodríguez-Martín et al., 2014) (VRP inside clusters) - MMHLRP (Nagy and Salhi, 1998) (de Camargo et al., 2013) - HLRP (Bostel et al., 2015) (Sun, 2016)	(TSP inside clusters) - MMpHLRPSH Our third proposed problem (VRP inside clusters) - HLRP (Çetiner et al., 2010) * no mathematical model
Vehicle routing-allocation problem	(Direct connection) - CHLP (Contreras et al., 2016)	-
Multi-level location-routing problem	(TSP inside the clusters) - tLpHLRP Our second proposed problem - MMpLHP (Lopes et al., 2016) * no flow routing - HRNP (Rodríguez-Martín et al., 2016a) * no flow routing (Thomadsen, 2005) (VRP inside clusters) - HRNP (Rodríguez-Martín et al., 2016b) * no flow routing	- tLpHLRPSH Our first proposed problem

Table 1.3: Main characteristics of hub location and routing problem identified in literature

PROBLEM	SOLUTION APPROACH		HUB LOCATION		ALLOCATION		NUMBER OF NODES IN A CLUSTER		SERVICE DESIGN IN REGIONAL NETWORK			SERVICE DESIGN IN HUB NETWORK		SERVICE		ROUTING
—	EXACT METHOD	APPROXIMATE METHOD	EXOGENOUS	ENDOGENOUS	SINGLE	MULTIPLE	LIMITED	UNLIMITED	SINGLE ROUTE	MULTIPLE ROUTES	DIRECT CONNECTION	CIRCULAR	DIRECT CONNECTION	UNIDIRECTIONAL	BIDIRECTIONAL	—
tLpHLRPSH	*	*	*			*		*	*			*		*		*
tLpHLRP	*	*	*		*			*	*			*		*		*
MMpHLRPSH	*	*	*			*		*	*				*	*		*
MMpHLRP	*	*	*		*			*	*				*	*		*
FINDP (Thomadsen and Larsen, 2007)	*		*		*		*				*		*		*	
FINDP (Saboury et al., 2013)		*	*		*		*				*		*		*	
HLRP (Aykin, 1995)		*		*		*		*			*		*		*	*
CHLP (Sung and Jin, 2001)	*			*	*			*			*		*		*	*
CHLP (Wagner, 2007)	*			*	*			*			*		*		*	*
HLRP (Rodríguez-Martín et al., 2014)	*		*		*		*		*				*		*	*
MMHLRP (Nagy and Salhi, 1998)		*		*	*			*		*			*		*	*
MMHLRP (de Camargo et al., 2013)	*			*	*			*		*			*		*	*
HLRP (Bostel et al., 2015)		*		*	*			*		*			*		*	*
HLRP (Sun, 2016)		*	*		*			*		*			*	*		*
HLRP (Çetiner et al., 2010)		*	*			*		*		*			*	*		*
CHLP (Contreras et al., 2016)	*	*	*		*			*			*	*			*	*
MMpLHP (Lopes et al., 2016)		*	*		*		*		*			*		*		
HRNP (Rodríguez-Martín et al., 2016a)	*			*	*		*		*			*			*	
HRNP (Thomadsen, 2005)	*			*	*		*		*			*		*		*
HRNP (Rodríguez-Martín et al., 2016b)	*			*	*		*			*		*			*	

In Table 1.3, the main characteristics of our developed models are compared with characteristics from works inserted in the Table 1.2.

Table 1.3 is started with characteristics of models developed in this thesis. For more details concerning our developed models, see chapter 3. Each asterisk represents a characteristic presented by the respective model.

Although hub-and-spoke network is composed of location, allocation, service design and routing, links between the characteristics of these parts have generated an integrated system.

The solving methods have been enhanced, which try to solve all parts of a hub-and-spoke system at the same time.

1.6 Solving methods for hub-and-spoke structure

Greater is the quantity of characteristics (see Table 1.1) used in a hub and spoke problem, higher is the practical difficulty of the problem solving.

Due to this difficulty, only small and medium instances of the problems presented in this chapter can be solved by exact methods. Big instances must be solved by heuristics, metaheuristics, or hybrid methods.

Based on the problem classification presented in Table 1.1, some problems and their solving methods are presented below.

Group 1 - Hub Location Problem

In this group is the greatest quantity of problems about hub and spoke. Consequently, the largest variety of solving methods is also presented. Below, some examples are mentioned:

Two heuristics were developed for single allocation p-hub median problem proposed by O'Kelly (1987). In the first one, all existing possibilities of p hub location are enumerated and each spoke node is allocated to the nearest hub. In the second one, the allocation is performed in a different way because the spoke node can be allocated to the first or the second nearest hub.

Classical metaheuristics like GRASP, Tabu Search, Variable Neighborhood Search (VNS) and genetic algorithm are used to solve different versions of hub location problem. In Peiró et al. (2014), Grasp was used to solve uncapacitated r-allocation p-hub median problem. In Martí et al. (2015), the same problem was solved by the Scatter Search methodology. In Cunha and Silva (2007), a memetic algorithm (genetic algorithm with a local search) is applied on Uncapacitated Hub Location Problem with Single Allocation and with variable discount factors on the linkages between hubs. In Yaman and Carello (2005), a capacitated single allocation hub location problem with modular link capacities is solved by a local search approach. This problem is decomposed in two sub-problems (location and assignment). A tabu search is applied on location sub-problem to find the best set of hubs. In assignment sub-problem, a basic local search is applied for the allocation of spoke nodes.

Combinations of metaheuristics are also used in the literature in order to improve results. A hybrid metaheuristic is used to solve a robust bi-objective uncapacitated single allocation p-hub median problem in Amin-Naseri et al. (2016). In this metaheuristic, the diversification phase is performed by a scatter search and high-quality solutions are explored by a Variable Neighborhood Search.

Exact methods are also represented in hub location problem literature. In An et al. (2015), a Lagrangian relaxation method is used in a branch and bound framework trying to solve a reliable single allocation hub location problem and a reliable multi allocation hub location problem.

Group 2 - Hub Network Design Problem

In this group, the observed problems are mainly solved by exact methods and decomposition methods:

In Alumur et al. (2012a), the multimodal hub location and hub network design problem is solved in two phases: initially a covering problem is solved by Gurobi solver and, after that, the allocation is optimized together with transportation.

Tree of hubs is solved with Lagrangean relaxation by Contreras et al. (2009). In de Sá et al. (2013), it is solved with Benders decomposition.

A branch and cut is used by Catanzaro et al. (2011) for the partitioning-hub location-routing problem.

In Arshadi Khamseh and Doost Mohamadi (2014), incomplete hierarchical hub center network problem with single assignment is solved with optimization software (GAMS and Cplex).

Hierarchical multimodal hub location problem with time-definite deliveries is solved with Cplex solver by Alumur et al. (2012c). The problem is solved in two different forms: without valid inequalities insertion and with valid inequalities.

A proposed mixed integer programming model for the hierarchical hub median problem with single assignment is solved by Cplex by Yaman (2009).

Group 3 - Hub Location-Routing Problem

In this group, problems are mainly solved by exact methods or by heuristic methods using decomposition.

Due to the fact that hub location and routing problems are composed of different sub-problems, then, decomposition methods seem to be a natural way to solve them.

In Thomadsen and Larsen (2007), fully interconnected network design problem was solved by a column generation in a branch and bound framework, which is named branch and price. In Sung and Jin (2001), a dual-based approach is used for the cluster hub location problem. In Wagner (2007), the cluster hub location problem is solved by a constraint programming approach. A branch-and-cut algorithm was used to solve a hub location and routing problem presented by Rodríguez-Martín et al. (2014). Cycle hub location problem is also solved with an branch and cut in Contreras et al. (2016). Hierarchical ring network design problem is solved in (Rodríguez-Martín et al., 2016a) with branch-and-cut and in Thomadsen (2005) with branch-and-price algorithm. Branch-and-cut is also used to solve the ring/k-rings network design problem in (Rodríguez-Martín et al., 2016b). While in de Camargo et al. (2013), the many-to-many hub location routing problem is solved with Benders decomposition.

In Aykin (1995), the hub location and routing problem is solved in an iterative manner after the decomposition of problem in hub location problem and routing problem. A decomposition method is also presented by Nagy and Salhi (1998) for the many-to-many hub location routing problem. In the decomposition presented by Nagy and Salhi (1998), location is solved before routing. But an inter-relation between location and routing is described as a third part of decomposition method. In Çetiner et al. (2010), an iterative hub location and routing heuristic is used for the hub-location routing problem.

However, works using metaheuristic are also found:

In Saboury et al. (2013), two hybrid metaheuristics were proposed to solve the fully interconnected network design problem. In the first hybrid metaheuristic, a VNS is used in the improvement phase of a Simulated Annealing framework. In the second one, a VNS is used in the improvement phase of a Tabu Search. Cycle hub location problem is solved with GRASP in Contreras et al. (2016). In Lopes et al. (2016), three metaheuristics are used to solve the many-to-many p-location-hamiltonian cycle problem: a biased random-key genetic algorithm, a multi-start with VND improvement algorithm and a simple local search with multistart.

Our proposed models are classified in this third group of problems (see Table 1.2 and chapter 3). A decomposition VNS is used to solve the proposed models of hub location-routing problems.

1.7 Conclusions

Hub-and-spoke structures were developed with the idea of routing demands using intermediate points (hubs) instead of direct connections.

Although delivery time is increased by these intermediate points and a larger congestion is verified in these area as consequence of the concentrated flows, this network structure is used because an economy of scale is offered due to a larger volume of flow in consequence of the aggregate demands.

Works have been developed allowing demands to flow between spoke nodes without using hub nodes. But, this characteristic is only offered between two nodes inside the same cluster (set composed of one hub and its allocated spoke nodes) (Rodríguez-Martín et al., 2014) or for a specific set of demands through of alternative paths (Aykin, 1995). Consequently, a small reduction will be verified in the transport cost if compared to a pure hub-and-spoke.

When a multiple allocation is verified in problems from the literature, spoke-spoke connection as parts of a delivery path is not used. Only few punctual and direct (source-destination) demands are met by this characteristic and the transport cost reduction is not so impacted.

Mathematical models for hub-and-spoke problem with routing of flow in hierarchical network structure composed of spoke nodes connections inside the clusters plus connections between clusters (sub-hub idea), which means multiple allocation of spoke nodes, were not identified in literature.

Mathematical models with routing of flow in hierarchical network allowing spoke connections (direct connection between any pair of nodes belonging to the same cluster, circular routes in cluster or star structure with links between spokes from the same cluster, etc), besides being composed of disjoint cluster, have been mainly designed with fully connected hubs. Circular routes at hub and cluster level with routing is only identified in telecommunication literature for ring network problem.

In the two next chapters, mathematical models for routing of flow in hierarchical hub-and-spoke networks are presented. Models are created to allow network structures with circular routes at cluster level and circular route at hub level and network structures with circular routes at cluster level and direct connections at hub level. In both network structures, two new models are proposed: in the first one, only disjoint clusters are allowed and, in the second one, clusters with an intersection point in a spoke node are allowed.

Models were developed for freight transport network. Therefore, two groups of directed edges (at cluster and hub levels) were inserted allowing demands to flow using unidirectional circular services.

In this study, importance/reality of economy of scale in hub network was not discussed because it is a characteristic used almost by all researchers of hub-and-spoke literature. Economy of scale is also presented in our models.

However, developed models are compared trying to show that, in a hub-and-spoke network, it can be offered a larger gain in transport cost when both economy of scale in hub arcs and shorter alternative paths are mixed in a single model.

Chapter 2

A new hub network design at liner shipping operations

A paper published in the International Journal of Shipping and Transport Logistics: A new hub network design integrating deep sea and short sea services at liner shipping operations (da Costa Fontes and Goncalves, 2017)

Abstract: Liner shipping is an increasingly attractive maritime transport, not only by the growing volume of goods transported in the world, but also by the diversity of products that can be carried and its service stability. At liner shipping operation, a hub and spoke structure is often used in which deep sea services take place at the hub network and short sea services can be done in the regional network between the hub and its respective spoke nodes. In this study, a new hierarchical structure of hub and spoke with sub-hub is presented. Deep sea services and short sea services are depicted in circular form and the sub-hubs are intersection points of neighbors regional networks. Both cargo-routing with transshipment in hub ports or in sub-hub ports are allowed. The binary integer linear programming model proposed in this study offers, together, a hub location with multiple spoke allocation problem, a service design problem and the cargo routing problem. Experimental results show good performances compared to the classic hub and spoke network structure.

Keywords: Maritime transport; liner shipping; deep sea; short sea; network design; service design; hub and spoke; hierarchical structure; sub-hub; routing cargo; transshipment; hub location; multiple allocation; binary integer linear programming.

2.1 Introduction

Increasing globalization of trade is currently experienced by the world, and liner shipping has been largely responsible for this change of behavior in the international trade scenario.

Maritime transport has been classified, (Christiansen et al., 2007) and (Andersen,

Figure 2.1: Container of 1 TEU. Source: www.dimensionsinfo.com/20ft-container-size/

2010), on tramp shipping, industrial shipping and liner shipping.

Regarding tramp shipping, the ship route is defined by cargo destination, which is, usually comprised of a single product. A tramper is a ship that has no fixed routing (as well as itinerary or schedule) and is available at short notice to load any cargo from and to any port.

Concerning industrial shipping, the transportation is provided by the owner of the goods. This type of operation has been used by oil companies, where only one product is transported, and the route is set according to the demand.

For a liner shipping, the goods are carried in containers. Routes are pre-defined and the service occurs in a regular way. A liner service generally fulfills the schedule, unless in cases where a call at one of the ports has been unduly delayed due to natural causes.

During the 60s, with the growth of world trade, the operations of liner shipping were no longer being met efficiently by the break bulk system (Stopford, 2009). The cargo handling procedure was simplified by the load unitization with palletization and containerization, thus accelerating the transport operations.

Through this load unitization, goods from different owners are transported, at the same time, by a single vessel. Different types of cargos also became transportable, because they are containerized according to their characteristics, as in the case of products using refrigerated containers.

According to Andersen (2010), containers are produced in a large number of variants, but the 20-foot-long (1 TEU - Twenty-foot Equivalent Unit) and 40-foot-long (2 TEU) are the most commons (Figure 2.1).

Trying to accompany this growth on maritime transport, companies have increasingly invested in more efficient ships. Greater carrying capacity combined with lower fuel consumption is showed by these new ship generations. Recently, the French company CMA CGM has launched Kerguelen, which is one of the largest container ship in the world with a capacity for 17.722 TEU.

The vessel capacity is measured in TEU and the classification of vessel size is based on the largest possible canal size that can be transited by the vessel. For example, Panamax

(Panama Canal), Suezmax (Suez Canal), and Malaccamax (Strait of Malacca), (Andersen, 2010).

Ports and channels are being upgraded to receive these increasingly larger vessels, such as the post-panamax vessels, and to carry out charging and discharging more quickly.

Liner shipping has been an increasingly attractive goods transport system, not only by the growing volume of goods transported in the world, but also by the diversity of products that can be carried and by the service stability. In these operations, routes are modified only six months after notification and the rates are published and remain stable for long periods.

According to Plum (2013), the business of liner shipping is often compared to public transit systems, such as bus lines, where each station/port has a frequency of visits. In the case of liner shipping, the frequency is usually weekly.

The sequence of ports visited by a vessel, operating a given service, usually is in circular form. Soon, if it is necessary four weeks in order to a ship to visit all the ports of a service, then, to make a weekly visit to all ports belonging to this service, four ships are necessary to be used at the operations along the same line.

Liner shipping operations are divided into deep sea and short sea service. In short sea service, regional offerings are collected by small ships, then they are transported to large capacity ports.

Large capacity ports are used as hub, collecting the offer from different regional ports and consolidating those with similar destinations. After consolidated, goods are sent to their regions of destinations through a large vessel operating a deep sea service.

These large vessels are used to connect hub ports, traveling continental distances, distributing demands and collecting offers from different regions.

Once the destination region of goods is achieved through the port hub, a second short sea service takes place for delivering to the final destination.

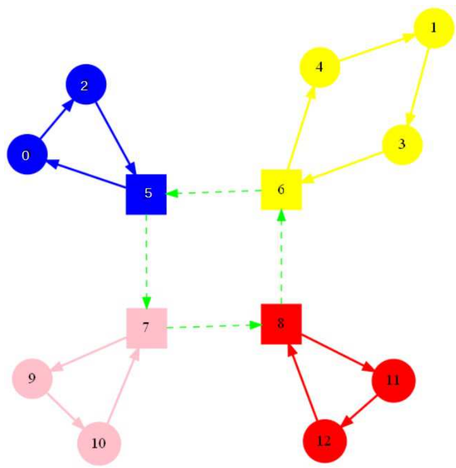
Hub ports are transshipment points for commodities. The end point of a commodity, with origin and destination in different regions, is achieved after the transshipment in two hub ports, being the first in the hub from the source region and the second in the hub from the destination region.

However, in liner shipping operations, it is highlighted that ports hub also has its own offer and demand, both at regional and continental levels.

Deep sea and short sea operations are connected via hub ports, making a large network of services. In Andersen (2010), it is described that a series of major ports referred to the hubs will provide the interface between the service networks of short sea and deep sea carriers through the transshipment of freight.

A network with 13 ports is represented by a directed graph on Figure 2.2, where the hub ports vertices are represented by squares and the spokes allocated to each hub nodes

Figure 2.2: Hub and spoke network for deep sea and short sea service



are represented by circular vertices. Deep sea services are represented by the dashed arcs and short sea services are represented by continuous arcs.

The services are presented in Figure 2.2 in cyclic form, both as deep sea level and short sea level. Using this network, goods can be sent from one port to any other port in the network. The route that the goods have to travel is unique in this way.

At the planning of a supply service in maritime transport, several factors are considered. Wieberneit (2008) relates some of them, which are highlighted:

- which is the route of a service?
- which is the frequency?
- which vehicle do we use?
- how is each commodity routed?

With the decision of the routes of each service, the integration between deep sea and short sea, seeking to minimize the time that a cargo is stored in a port, can be better planed by maritime companies.

While that, with visit frequency at each port and type of vehicle that will be used in the service, an efficient berth scheduling can be planned by the ports for each ship and an estimated flow can be developed, on which a better scheduling crane can be offered by the ports.

When the routing sequence of each commodity is set, the transit time is possible to be estimated.

All these factors involved at the decision of the implementation of a service network are still difficult because the supplier needs to keep a customer satisfaction level and meet the environmental legislations.

Due to hub and spoke structure being used by liner shipping network, routes between hubs at deep sea service are highlighted with a larger capacity and, consequently, with greater economies of scale.

Given the fact that short sea service can be done in cyclic form using a ship to collect/deliver the goods of all ports from a region, a regional commodity does not need to perform a transshipment, because the goods are just routed from the source port to the destination port, which is located in the same region.

Ports with small volume of goods are also favored by this type of service in cyclic form, because cargos from different points are collected by a single ship, where the ship capacity is completed with small demands.

The main drawbacks of a hub and spoke network are the longer routes and the time spent in terminals, which are consequences of this structure (Crainic and Kim, 2007).

Due to the time windows that ships must respect to visit ports, delays are facilitated with the use of long routes. Also, the long time used in a terminal by a ship ends up allowing the congestion training by the ships waiting to dock.

In the case of maritime transport, these delays are also observed in consequence of bad weather, mechanical problems, strike at ports or crew and congestion in passageway.

When the whole demand is serviced by a ship, a revenue is received by it, but if the demand is serviced by the ship with late, failed to be met or it is answered in part, a penalty is generally paid by the carrier.

Therefore, trying to reduce these delays, the speed of navigation can be increased or visits at some ports on the route are omitted by ships.

With the increase in the speed of navigation, the delay can be reduced by the ship, but this alternative ends up bringing other consequences. When a ship is speeded up, spending on fuel consumption and CO_2 emissions are proportionally increased. The cost of bunker fuel is a cubic function of speed (Brouer et al., 2013).

The CO_2 produced by maritime transport is estimated at 2.7% of CO_2 emission worldwide, of which 25% corresponds to container vessels (Plum, 2013).

About the omission of a port by a ship trying to reduce the delay on the route, several consequences are generated both for the ship as for the omitted port.

When a port is omitted by a ship operating in deep sea service, the offer and demand of an entire region will no longer be served by the ship. The offer from the omitted port will be answered only by the next ship in the route, requiring the payment of a penalty by the ship.

A demand that should be delivered by a ship at the port omitted can only be delivered in the next visit of the ship at the port, because the route is in cyclic form and has the same direction for all ships allocated to the service.

Observing Figure 2.2, if hub ports 5 and 6 are in two close regions and hub ports 7 and 8 are in other two neighbor regions, a commodity originating from spoke node 12 and with destination in spoke 10 needs to perform a journey across the network, visiting far regions and only then reaching the final port.

For a maritime company to get a competitive advantage between its competitors, it is necessary a service network that meets its customers in the shortest possible time, maintaining customer satisfaction and performing a green logistics, avoiding to violate environmental laws of different countries.

Reaching and sustaining a competitive advantage is only achieved by long-term actions. In Christiansen et al. (2007), network design is classified as a strategic planning problem in maritime transport and routing as tactical planning problem.

For maritime companies, an efficient and sustainable network structure is achieved more easily with the deployment of a supply service.

With the effective design of a network, the amount of vessels operating in each service can be reduced and, consequently, the fuel expenses, the amount of CO_2 emitted and the cost for the crews are also reduced.

The number of transshipments can be further reduced with the choice of hub ports, and, as a consequence, the travel time of the goods can be reduced too.

With an efficient implementation of short sea service, the main haul of regional journey undertaken by trucks and trains can be done by ships, achieving greater economies of scale and leaving just the door-to-door service to trucks. It can help reduce road network congestions, as well as CO_2 emissions (Andersen, 2010).

This chapter is organized as follows: In section 2, the main researches in maritime transport are highlighted with a focus on the hub location, service network design and routing problem. Section 3 describes our proposed mathematical model for designing network with alternative paths. Section 4 shows the results we obtained and, finally, as the conclusion part, the main observations and directions of future researches are presented.

2.2 Literature Review

Liner shipping has increasingly attracted the interest of researchers within a perspective of operational research, with works addressing different contexts of the problem.

Characteristics of transshipment operations are highlighted by Meng and Wang (2011b), Zheng et al. (2014) and Mulder and Dekker (2014).

In Meng and Wang (2011b), a mathematical programming model with equilibrium constraints for intermodal hub and spoke network design (IHSND) was developed, where it is defined one physical network and an operational network. Physical network involves nodes, links and transshipment lines, and the transfer process of containers in a hub is reflected by operational network. The cost of transportation on the network and the cost of operations in the hubs are minimized by the model.

A two-phase mathematical programming model for liner hub and spoke shipping network design is presented in Zheng et al. (2014). Phase 1 presents the hub location and the spoke allocation, where direct links from spoke nodes to hubs node are observed together with a simple allocation, or in other words, each spoke node is allocated to only one hub and each hub is visited by the demand of their spokes. Maritime cabotage constraints are also presented at the phase 1. Meanwhile in phase 2, routing with the deployment of a fleet of ships subject to transit time restriction is carried out. In this phase, it is used a hub expansion technique, where a hub is expanded in n hubs, n being the number of commodities that were transshipped in this hub.

In Mulder and Dekker (2014) fleet design and ship scheduling are combined by considering them as necessary points in the building of the service network, along with a cargo routing, to the proposed problem. A set of routes with transshipment cost and capacity of the route of ships are considered by the authors.

Different formulations of service scheduling or of cargo routing are addressed in Zheng et al. (2014), Mulder and Dekker (2014), Wang and Meng (2014), Meng and Wang (2011a), Plum et al. (2014a) and Plum et al. (2014b).

A mixed-integer non-linear and non-convex programming model for liner shipping network design problem with deadlines is presented in Wang and Meng (2014). At the model, it is maximized the profit for a single route on a loop form, respecting the constraints of container shipment demand split delivery, transit deadlines, weekly frequency service and port team depending on the handled container volume.

In Meng and Wang (2011a), a set of feeder routes connected by a hub with empty container repositioning is presented. A sequence of port calling is presented by each feeder route, it is formed by a cyclic route. Laden and empty containers are related by the mathematical model.

An arc-flow and a path-flow model for a single liner shipping service design problem is showed by Plum et al. (2014a). At the model, the demand among the various points of the route, the arc capacities and the limited time of travel for commodities are presented.

Capacity constraints were also used in Mulder and Dekker (2014) and Plum et al. (2014a), at the building of the mathematical model.

In Plum et al. (2014b), a service flow model for the liner shipping network design problem is proposed, where the service is represented with butterfly ports. Butterfly ports are ports where a vessel is received twice in the same route. At the proposed model, ports and services are represented by nodes and the arcs are used to represent service-port. Demand flows from port node to a service node and from service node to a port node using service-ports arcs.

In Zheng et al. (2014), Mulder and Dekker (2014) and Wang and Meng (2012), the authors worked with maritime problem using characteristics of vessel fleet.

In Wang and Meng (2012), a mixer-integer nonlinear stochastic mathematical model is presented for the liner shipping scheduling problem in order to handle uncertainties related to time of arrival at a port and start time of cargo handling at the port. In this model, the objective function includes shipping costs, bunker costs plus late start handling costs.

Time limits are considered by Zheng et al. (2014), Wang and Meng (2014), Plum et al. (2014a), Wang and Meng (2012) and Brouer et al. (2013) to represent characteristics of travel time, delay and cargo-handling.

in Brouer et al. (2013), a mathematical model for the vessel schedule recovery problem is presented. The model aims to minimize sailing costs, misconnecting cargo costs and delay costs of a group of containers. In the model, it is assumed that there will be one misconnecting cargo when a container group is late and there will be a port omission when there is misconnecting cargo.

According to Christiansen et al. (2013), network design models, for liner shipping problem, published during the last decade, were grouped in 4 categories: models with a single route or sets of routes without transhipment; hub and feeder route models where each feeder port is connected to a single hub port, models where some ports are classified as hub ports without any constraints on the number of hubs and non-hub ports a route may visit; and multi route models without any separation of hub and non-hub ports.

In these groups, the models are distinguished by having a capacitated network, by the use of time-limited, by manner of service schedule, by single or multiple allocation of spokes nodes to hubs, by the use of a fleet of ships, or transshipment costs, etc. But the representation of a network with the format used by regular maritime transport, i.e. a route representing regional short sea service and one hub route representing deep sea service, all in a circular form, is still neglected even in the literature on hub and spoke problem.

2.3 Problem Definition

This study is bringing together the problems of hub location, spoke allocation with single and multiple allocations, service design problem and cargo routing problem.

Deep sea and short sea services are integrated in the new hub and spoke structure. The hierarchical structure of the model is provided by the multiple allocation of some spoke nodes.

Deep sea service is represented by a set of arcs connecting the hub nodes. Unlike most existing studies in the literature (Alumur and Kara, 2008), (Campbell, 1994), where the network of hubs is represented by a complete graph (i.e. direct links between the hubs), we consider in this study that, for every hub node, there are one incoming arc and one outgoing arc, allowing the cyclical visit of nodes by one ship (hub route).

Each hub node has a set of spoke nodes allocated to it. Generally, in the hub and spoke literature, direct links are observed between the spoke nodes and the hub to which spoke nodes are allocated (Zheng et al., 2014), (Hamacher et al., 2004), (Yaman and Carello, 2005), not allowing a demand to travel between two spokes without first visiting a hub. In this research, for each hub and its respective group of spoke nodes, the arcs are linked to allow a cyclical visit between the hub and spoke nodes (sub route), representing the short sea service.

Consequently, in the current version of the mathematical model, we consider only services (i.e. one-way cyclic routes) which are composed with 3 nodes at least.

In de Camargo et al. (2013), due to a postal service problem to be solved, a cyclic visit between the hub and its respective spoke nodes is observed. But the delivery of a demand between two spoke nodes belonging to the same cluster needs to accomplish transshipment at the hub, where the current demand is aggregated with demands from different sources, and then proceed to the destination spoke. In short sea service, demands are collected and delivered by ships traveling between spoke nodes, transshipment is just made to satisfy demands of different regions (clusters).

Note that in this study, transshipments are not only performed in hubs but they can also be done in sub-hubs. Sub-hubs are spoke nodes allocated to two different hubs because they are situated with an equivalent distance from both two hubs. These spoke nodes are used as intersection points between the two corresponding regional routes.

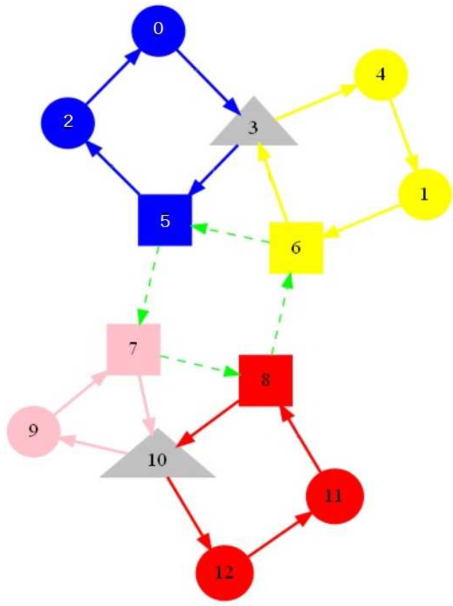
Due to sub-hubs are included in sub-routes (regional routes), they have no arc-hub links, but only regional links, so the transshipment held in sub-hub corresponds to cargo of ships operating a short sea service.

Because hubs and sub-hubs are also considered as ordinary ports (i.e. spokes), they are verified to have offers and demands to satisfy (da Costa Fontes and Goncalves, 2015a). So, hub and sub-hub are not only points of transshipment and aggregation of demands, but are also regarded as start and end points of flow of goods.

The number of hub ports is limited, but the amount of spokes allocated to each hub is resulted of the proximity between the hub and each spoke. The number of sub-hubs is also not limited. The only restrictions are that, between two hubs, there is at most a single sub-hub, and a sub-hub can only exist if two nearby regions constraint are verified.

Generally, in the literature on hub and spoke networks, only two hubs are visited by the demand between two regions, the hub of origin region and the hub of destination region (Hamacher et al., 2004), (Yaman, 2011), (O'Kelly et al., 1996), because the hub network is a complete graph. As in this research, the hub network is in circular form, demands using the deep sea service, after been uploaded in the hub from source region, will travel some hubs of network without performing any transshipment, then they will

Figure 2.3: Hub and spoke network with sub-hub for deep sea and short sea service



be downloaded at the hub of destination region.

The goods that are transported between two neighboring regions have the option of using a deep sea service, performing the transshipment in two hub ports, or just use a short sea service, accomplishing the transshipment in only one sub-hub. The path that will be used by the goods is defined by the option with the biggest economy or, in other words, it is defined by the path using an economy of scale, or the path using a route that offers a lower cost because it is shorter.

Note that this alternative path makes the solution more robust against uncertainties of strikes, bad weather, mechanical problems or long times spent in a hub terminal.

On Figure 2.3, the result of a problem for a network with 13 nodes is showed, where the hubs are represented by squares, sub-hubs are represented by triangles and spokes are represented by circles.

2.3.1 Omitted Regions

In a hub and spoke network without sub-hubs, where both the deep sea service as short sea service are represented in circular form and in just one direction, the omission of ports by a vessel conducting a route between hubs, generates significant delay to the demands of the omitted ports. Deep sea service is performed between continents with a sequence of ports that will be visited. The omitted demand will only be delivery after the ship complete the route and return at the omitted port. In the case of network with sub-hubs, the ship may choose to omit a port where it is permitted to deliver the cargo at the next port of the network, and from this the goods travel using a regional route and perform a transshipment at sub-hub, until reaching the omitted region.

Therefore, in the network with sub-hub, the minimization of delay in cargo delivery from an omitted port is obtained, because it prevents goods to travel continental distances, crossing the entire network. The presence of sub-hubs allows that an alternative route is used by the cargo, using short sea service to reach the region omitted by the ship operating in deep sea.

As a hub also distributes goods from the region for the entire network, then, the omission of a hub port will impair not only the region omitted, but also all ports that would receive demands from the omitted region.

The offer of the omitted region can be distributed to the other regions of the network by the next ship operating the network between hubs, with, of course, a delay of one week. However, the cargo of a neighboring region may not suffer with this kind of delay because is possible to use an alternative route with transshipment in a sub-hub.

Thus, the alternative paths offered by sub-hub transshipments, allow for reduced delays in delivery time of some demands without the need to increase the speed of the ship. As consequence, fuel expenses are reduced in this network and, thus, it reduces CO_2 emissions.

In a network that has a hub and spoke structure, the use of alternative paths through regional routes provides a minimized time of routing cargo, enabling greater customer satisfaction, a reduction in spending on transshipment and, consequently, a reduction in the overall cost of network.

2.3.2 Mathematical Model

This study presents a binary integer linear programming model for the hub location and spoke allocation problem together with the service design problem and the routing problem.

Let V be the set of nodes with n nodes, c_{ij} is the cost per unit of flow from node i to node j where i, $j \in V$, p is the number of hubs, fl_{ij} is the flow from node i to node j and α is the discount factor on the unit of flow between hubs.

Hub location, sub-hub location and spoke allocation are defined by variables of model: $x_{ij} = 1$, if the vertex i is allocated to hub j, or 0 otherwise, and $t_i = 1$, if i is sub-hub, or 0 otherwise.

The variable $y_{ij} = 1$, if there exists the arc-non-hub $(i; j)$, or 0 otherwise, and $b_{ij} = 1$, if there exists the arc-hub $(i; j)$, or 0 otherwise, are responsible by the service design.

While the variables $w_{ijkl} = 1$, if there exists a flow from i to j routed via arc-hub $(k; l)$, or 0 otherwise and $s_{ijkl} = 1$, if there exists a flow from i to j routed via arc-non-hub $(k; l)$, or 0 otherwise, correspond to the cargo routing.

The model can be stated as follows:

$$min \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} c_{ij}\ x_{ij} + \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \sum_{k=1, k\neq j}^{n} \sum_{l=1, l\neq k, l\neq i}^{n} fl_{ij}\ c_{kl}\ s_{ijkl} +$$
$$+ \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \sum_{k=1, k\neq j}^{n} \sum_{l=1, l\neq k, l\neq i}^{n} fl_{ij}\ \alpha\ c_{kl}\ w_{ijkl}$$

$s.t.:$

$$\sum_{j=1}^{n} x_{jj} = \ p \tag{2.1}$$

$$x_{ij} \leq x_{jj}, \qquad \forall i, j \epsilon V \tag{2.2}$$

$$x_{ij} + x_{ii} \leq 1, \qquad \forall i, j \epsilon V,\ \ i \neq j \tag{2.3}$$

$$t_i \leq \sum_{j=1, j\neq i}^{n} x_{ij}/2, \qquad \forall i \epsilon V \tag{2.4}$$

$$t_i + x_{ii} \leq 1, \qquad \forall i \epsilon V \tag{2.5}$$

$$\sum_{j=1}^{n} x_{ij} = \ 1 + t_i, \qquad \forall i \epsilon V \tag{2.6}$$

$$x_{ij} + x_{ik} + x_{lj} + x_{lk} \leq 3, \qquad \forall i, j, k, l \epsilon V, \quad i \neq j, k \neq j, k \neq i, l \neq j, l \neq k, l \neq i \tag{2.7}$$

$$y_{ij} + x_{ii} + x_{jj} \leq 2, \qquad \forall i, j \epsilon V,\ \ i \neq j \tag{2.8}$$

$$y_{ij} + x_{ik} + x_{jl} \leq 2 + x_{jk} + x_{il}, \qquad \forall i, j, k, l \epsilon V,\ \ i \neq j, l \neq k \tag{2.9}$$

$$x_{ij} + x_{kj} + x_{lj} + y_{ik} + y_{il} \leq 4, \qquad \forall i, j, k, l \epsilon V, \quad i \neq j, k \neq i, l \neq i, l \neq k \tag{2.10}$$

$$y_{ij} + y_{ji} \leq 1, \qquad \forall i, j \epsilon V,\ i \neq j \tag{2.11}$$

$$y_{ij} + t_i + t_j \leq 2, \qquad \forall i, j \epsilon V,\ i \neq j \tag{2.12}$$

$$\sum_{j=1, j \neq i}^{n} y_{ij} = 1 + t_i, \qquad \forall i \epsilon V \tag{2.13}$$

$$\sum_{j=1, j \neq i}^{n} y_{ji} = 1 + t_i, \qquad \forall i \epsilon V \tag{2.14}$$

$$\sum_{j=1, j \neq i}^{n} b_{ij} = x_{ii}, \qquad \forall i \epsilon V \tag{2.15}$$

$$\sum_{j=1, j \neq i}^{n} b_{ji} = x_{ii}, \qquad \forall i \epsilon V \tag{2.16}$$

$$b_{ij} + b_{ji} \leq 1, \qquad \forall i, j \epsilon V,\ i \neq j \tag{2.17}$$

$$\sum_{l=1, l \neq i}^{n} w_{ijkl} + \sum_{l=1, l \neq i}^{n} s_{ijkl} = 1, \qquad \forall i, j \epsilon V,\ k = i, j \neq i \tag{2.18}$$

$$\sum_{l=1, l \neq j}^{n} w_{ijlk} + \sum_{l=1, l \neq j}^{n} s_{ijlk} = 1, \qquad \forall i, j \epsilon V,\ k = j, j \neq i \tag{2.19}$$

$$\sum_{l=1, l \neq i, l \neq k}^{n} w_{ijkl} + \sum_{l=1, l \neq i, l \neq k}^{n} s_{ijkl} = \sum_{l=1, l \neq j, l \neq k}^{n} w_{ijlk} + \sum_{l=1, l \neq j, l \neq k}^{n} s_{ijlk},$$
$$\forall i, j, k \epsilon V,\ k \neq i, k \neq j, j \neq i \tag{2.20}$$

$$s_{ijkl} \leq y_{kl}, \qquad \forall i, j, k, l \epsilon V,\ l \neq k, l \neq i, k \neq j, j \neq i \tag{2.21}$$

$$w_{ijkl} \leq b_{kl}, \qquad \forall i, j, k, l \epsilon V,\ l \neq k, l \neq i, k \neq j, j \neq i \tag{2.22}$$

$$x_{ij},\ t_i,\ y_{ij},\ b_{ij},\ w_{ijkl},\ s_{ijkl} \in \{0, 1\} \qquad \forall i, j, k, l \epsilon V \tag{2.23}$$

The objective function minimizes the costs. The first part minimizes the location and allocation costs and the second part minimizes the overall transportation cost, where there are the cost of flow just in arc hub, utilizing the parameter α, and the cost of flow in arc non-hub.

The transport cost calculated in the objective function does not record the cost of the vessel crossing an arc (aggregate flow), but the cost of each commodity going through an arc (disaggregated flow).

The constraints (2.1) - (2.7) are responsible for creating hub locations and allocation of spoke nodes to hub nodes, while the constraints (2.8) - (2.17) create the arcs linking the nodes, representing the service design, and the constraints (2.18) - (2.22) transport the flow on these arcs.

In the constraint (2.1), the number of hubs is equal to p. For the constraints (2.2), a vertex i will just be allocated to a vertex j if j is a hub. The constraints (2.3) assure that if i is a hub, it can not be allocated to a hub j or if i is allocated to j, i can not be a hub. In the constraints (2.4), a vertex i will just be a sub-hub if it is allocated to 2 hubs. The constraints (2.5) assure every vertex will be a hub or a spoke, and a vertex will just be a sub-hub if it is not a hub. The constraints (2.6) assure that every vertex i will be allocated to one hub j, or to two hubs j in the case of i to be a sub-hub. In the constraints (2.7), 2 sub-hubs can not be allocated to the same hubs.

For the second group of constraints, the constraints (2.8) assure that arc represented by variable y_{ij} will not link hubs, it will either link spokes or 1 hub with 1 spoke. The constraints (2.9) will not permit the existence of an arc linking the vertex i and j if the vertex i and j are allocated to different hubs. There will just exist an arc linking the vertex i and j if they are allocated to the same hub. The constraints (2.10) will not allow two arcs going out of a vertex i to two other vertices that belong to a same hub k. The constraints (2.11) assure that the network uses the arc $(i; j)$ or the arc $(j; i)$ to flow the demands. The constraints (2.12) do not permit an arc linking 2 sub-hubs. The constraints (2.13) assure that, from every vertex i, 1 arc go out, but if i is a sub-hub, 2 arcs go out. The constraints (2.14) assure that every vertex i will receive 1 arc, but if i is a sub-hub, it will receive 2 arcs. The constraints (2.15) assure that from every vertex hub i, 1 arc represented by variable b_{ij} go out. In the constraints (2.16), every vertex hub i will receive 1 arc represented by variable b_{ji}. For the constraints (2.17), the network uses the arc-hub $(i; j)$ or the arc-hub $(j; i)$, where, arc hub links 2 hubs and arc non-hub either links a hub with a spoke or links two spokes.

Network structure is built by these two groups of constraints above and the next group of constraints is responsible for creating the cargo routing in this network.

Constraints (2.18), (2.19) and (2.20) satisfy the flow conservation. Constraints (2.18) state that the flow of every vertex i towards every vertex j will leave the source using the arc $(i; l)$, where the flow can be a flow represented by variable w (flow in an arc hub) or a flow represented by variable s (flow in an arc non-hub). The constraints (2.19) assure that the flow of every vertex i towards every vertex j will arrive at the destination using the arc $(l; j)$, where the flow should be a flow in a arc hub or a flow in a arc non-hub. The constraints (2.20) ensure that all outgoing flows from an intermediate vertex k (that is not the source or the destination), will be equal to all ingoing flows from that intermediate vertex. The constraints (2.21) assure that the flow represented by variable s will just exist if there exists the arc non-hub and in the constraints (2.22), the flow represented by variable w will just exist if there exists the arc-hub.

Constraints (2.23) are the integrality constraints.

Instances solved by this mathematical model need to have, at least, 3 hubs. For each cluster, it is necessary, at the minimum, 3 nodes.

Because there are possibilities to form a subtour at hub network level when we consider instances with 6 or more hubs, constraints to eliminate subtours are added in the previous model.

For the same reasons at regional route, it was performed the same subtour elimination when a cluster has 6 nodes or more.

Then, for each viable solution found, H will be the set of selected hubs nodes with p nodes, C will be the set of selected clusters with $C = \{C_1, C_2, ..., C_m\}$, C_q will be the set of selected nodes, which belong to the same cluster q and G will be any subset of nodes from a set H or from a set C_q. The constraints are:

$$\sum_{i \in G, j \in H-G} b_{ij} \geqslant 1, \qquad 2 \leq \mid G \mid \leq \mid H \mid -2, \quad with \mid H \mid = p, \quad q \geq 6 \tag{2.24}$$

$$\sum_{i \in G, j \in C_q - G} y_{ij} \geqslant 1, \qquad 2 \leq \mid G \mid \leq \mid C_q \mid -2,$$
$$\forall\, C_q \quad with \mid C_q \mid \geq 6, \quad where\ q \in \{1, 2, ..., m\}, \quad with \quad m = p \tag{2.25}$$

Constraints (3.24) are responsible to eliminate subtours at hub route level and constraints (3.25) to eliminate subtours at regional route level.

2.4 Computational Results

The computational tests were executed with a processor Intel Core i7 with 3.0 GHz and 8 GB of memory and the mathematical model was implemented using Cplex Concert Technology C++.

In the literature, it was not found a mathematical model where there are together the characteristics: cyclic route between hubs (deep sea service) integrated with cyclic routes between each hub and its spoke nodes (short sea services) and unlimited number of spoke nodes for each hub. Therefore, the version without sub-hub, (Figure 2.2), was created in this study to compare this basic model with the one with sub-hub.

For this model version without sub-hub, it was removed the constraints (2.4), (2.7), (2.10) and (2.12) and the constraints (2.5), (2.6), (2.9), (2.13) and (2.14) of mathematical model were changed.

All instances were generated and can be found in our web site. The results are presented in the tables below:

Where OF is the objective function value and ET is the execution time. The execution time is in seconds. The execution time limited for Cplex was 9000.

Table 2.1: results for 9 nodes

Model	n	q	α	Objective Function	Gap	Execution Time
with sub-hub	9	3	0.4	487927	0	114.81
without sub-hub	9	3	0.4	532664	0	72.39
with sub-hub	9	3	0.7	592315	0	326.08
without sub-hub	9	3	0.7	680965	0	179.59
with sub-hub	9	3	0.9	653369	0	6085.91
without sub-hub	9	3	0.9	779832	0	531.52

Table 2.2: results for 10 nodes

Model	n	q	α	Objective Function	Gap	Execution Time
with sub-hub	10	3	0.2	629064	0	1398.58
without sub-hub	10	3	0.2	679935	0	650.36
with sub-hub	10	3	0.4	770670	0	1621.27
without sub-hub	10	3	0.4	841871	0	789.75
with sub-hub	10	3	0.7	1047110	22.4	9000.00
without sub-hub	10	3	0.7	1083930	0	1416.3

Tables 2.1 - 2.6 are results of instances with 9, 10, 11, 12, 13 and 15 nodes, respectively. Table 2.7 shows the gain of the objective function between the two models only for

Table 2.3: results for 11 nodes

Model	n	q	α	Objective Function	Gap	Execution Time
with sub-hub	11	3	0.4	1357950	45.45	9000.00
without sub-hub	11	3	0.4	1136500	17.61	9000.00

Table 2.4: results for 12 nodes

Model	n	q	α	Objective Function	Gap	Execution Time
with sub-hub	12	3	0.4	1848460	59,44	9000.00
without sub-hub	12	3	0.4	1199250	31.19	9000.00
with sub-hub	12	4	0.4	833150	0	2832.28
without sub-hub	12	4	0.4	958804	7.20	9000.00
with sub-hub	12	4	0.5	903520	0	5362.16
without sub-hub	12	4	0.5	1079580	14.69	9000.00
with sub-hub	12	4	0.6	972758	0	7128.80
without sub-hub	12	4	0.6	1139640	8.64	9000.00
with sub-hub	12	4	0.7	1055950	7.15	9000.00
without sub-hub	12	4	0.7	1288350	17.14	9000.00

Table 2.5: results for 13 nodes

Model	n	q	α	Objective Function	Gap	Execution Time
with sub-hub	13	4	0.2	987313	18.94	9000.00
without sub-hub	13	4	0.2	1057700	14.38	9000.00
with sub-hub	13	4	0.4	1158740	9.84	9000.00
without sub-hub	13	4	0.4	1385330	19.54	9000.00
with sub-hub	13	4	0.7	1475780	13.64	9000.00
without sub-hub	13	4	0.7	1714260	18.16	9000.00

instances where optimal solutions have been found.

Gap (calculated by the Cplex) is the ration between the solution found by Cplex after the relaxation of integrality of model (lower bound, for minimization case) and the best

Table 2.6: results for 15 nodes

Model	n	q	α	Objective Function	Gap	Execution Time
with sub-hub	15	4	0.3	1521400	28.78	9000.00
without sub-hub	15	4	0.3	1599630	26.75	9000.00
with sub-hub	15	4	0.5	1721850	24.00	9000.00
without sub-hub	15	4	0.5	2036050	31.03	9000.00

Table 2.7: objective function value comparison between the two models

n	α	gain in OF
9	0.4	8.39
9	0.7	13.01
9	0.9	16.21
10	0.2	7.48
10	0.4	8.45
10	0.7	3.39

solution found with the integrality constraints. The Gap is given in percentage and the execution time is in seconds. The execution time was limited to 9000 seconds.

At the results, it was observed that, if α parameter is small, for example 0.4, hub routes are more attractive, consequently, the alternatives paths by sub-hubs are less used. So the objective function value for the model without sub-hub is close to objective function value with sub-hub. Nevertheless, in the test with 9 nodes, the model with sub-hub still have a gain (8.39%) concerning the objective function and the test with 10 nodes, a gain (7.48%) was also found, (see Table 2.7). Note that the gain is given in absolute value.

When α is high, for example 0.9, hub routes are less attractive. In this case, alternative paths by sub-hubs are more used and the gain of the objective function is more significant (16,21%) between the model with sub-hub and the model without sub-hub, (Table 2.7). The test with 10 nodes and $\alpha = 0.7$, at the model with sub-hub stopped with a gap of 22.4% (Table 2.2), but it still presents a gain of 3.39% (Table 2.7).

Just for the tests with 9 and 10 nodes, optimal solutions have been found by both models (Tab 2.1) and (Table 2.2). In this case, it was observed that the model with sub-hub has a higher execution time. This increase of time execution occurs because the increasing number of viable solutions generated by sub-hub properties (alternative routes).

Another interesting point to mention, it is possible to observe for the tests with 12 nodes and α equals to 0.4, 0.5 and 0.6 (Table 2.4), that Cplex is able to find optimal solution when sub-hubs are allowed.

For the remainder tests, Cplex is not able to give an optimal solution within the time limit we have set (9000 s). However, the model with sub-hub still gives better objective function values than the model without sub-hub (see Tables 2.4, 2.5 and 2.6), except for tests with 11 nodes and 3 hubs, and 12 nodes and 3 hubs (see Tables 2.3 and 2.4), because the observed Gap is too large.

Concerning the tests with 11 nodes and 3 hubs, and 12 nodes and 3 hubs, for different α, the gap for both models was very high.

2.5 Conclusions

Competition between container terminals has increased due to the large growth rates on major seaborne container routes. Terminals face with more and more containers to be handled in a short time at low cost (Stahlbock and Voß, 2008).

Due the growing number of large ships, ports must be adapted to receive these vessels. At the port, it is necessary a quick transshipment operation and deposits with large capacity to hold goods leaving through the port and the goods that need to arrive through the port.

The competition between different ports is intensified in a network structure with hub ports, since a port being used as a hub port generates a greater trade flow in the chosen port.

In this new structure of hub and spoke with sub-hub, it is important that ports of small capacity are upgraded to obtain a competitive advantage ahead from the similar ports, trying to be a sub-hub.

Therefore, the service network quality will be favored by competition between ports, both at continental and regional levels.

In the network structure of hub and spoke, long routes are regarded as a characteristic. However, in this new proposed network, small routes are created by the use of alternative paths for routing of cargo between neighboring regions. The travel time for some commodities is minimized with these short paths.

This travel time is minimized, not only by the fact that the cargo does not need to travel long routes in a network between hubs, but also because the transshipment of cargo is not performed twice, one at source region hub and a second one at destination region hub, but only once at the sub-hub.

Better cargo distribution across the network will be obtained with the transshipment in sub-hubs, relieving the hub ports and decreasing the time spent by a vessel in a hub.

Therefore, another negative effect of a hub and spoke structure can be decreased with this new proposed network structure.

This network structure arises as an alternative for liner shipping operations, because it allows maritime companies to combat the delays of services, in order to keep the customer satisfaction, without losing the characteristic of a green logistic.

Also regarding other routing aspects, this new structure can be very interesting.

The tests were performed with small instances, because both the increase of p parameter, as well the increase of α parameter generate a large number of solutions, and they do not permit Cplex solver to find an optimal solution.

However, the tests showed that the proposed hubs network with sub-hubs generates a set of alternative routes that offer a significant gain on the transport economy.

For results with large-sized instances, it is interesting to think about the use of decomposition methods or of heuristics/ metaheuristics. We intend to validate this network structure for large instances and we plan to investigate those resolution methods in the forthcoming months.

Other aspect we are working on is the definition of a mathematical model with sub-hubs allowing hubs without allocation of spokes. Real situations where one isolated port, but with large supply and/or demand by consequence of intermodal freight transportation, could represent one cluster, will be portrayed by this new model.

Consequently, limited characteristic of the model presented in this study, where each cluster needs to have at least 3 nodes, will be removed with the investigation of models allowing isolated hubs.

Chapter 3

Integer Linear Programming Models for Hub-and-Spoke Network

Similar to different researches dealing with hub-and-spoke network structure, several methodologies from operations research are used in this study.

Mathematical programming and graph theory models, as well as algorithms, are described in this chapter, along with the mathematical models developed.

More specifically, classic models from combinatory optimization, such as the traveling salesman problem and p-median problem, and also the shortest path problem are described. Their main algorithms are addressed. The constraints that constitute these problems are used in the building of the models in order to solve the proposed problems.

These proposed problems are modeled by binary integer linear programming models. A cutting plane approach is implemented seeking to solve the addressed models and it is also described in this chapter.

Two mathematical models, both with circular hub route and circular routes inside clusters are described. In the first one, connections between clusters are allowed. While on the second one, clusters are disjoint sets.

After that, two other mathematical models are proposed. These are different from the first two, because the network at hub level is composed of a complete graph.

These models are different from those presented in chapter 2 because, now, a direct connection between the sub-hub nodes is allowed and clusters can be composed of only one node (isolated hub), two nodes (one hub and one spoke) or even more. These new models can also be composed of circular routes with only two nodes, while on the former model, three nodes were necessary to create one route.

These developed models fill a gap in the literature of hub-and-spoke, as it was mentioned on chapter 1, and they can represent real existing situations where goods are moved in a network, as it will be described here.

3.1 Introduction

A great deal of researches about hub-and-spoke network are explored through of optimization field. Concepts of mathematical programming and graph theory are used to model different situations where a hub-and-spoke network could be used.

Initially, the hub-and-spoke problem was modeled as an integer non-linear programming (O'Kelly, 1987). After that, in Campbell (1991), the p-hub median problem was formulated as an integer linear programming model and, in Campbell (1994), the uncapacitated hub location, the p-hub center and the hub covering problems were presented as integer linear programming models.

In fact, several problems are still modeled as integer non-linear programming. For example: a multiple allocation hub-and-spoke network design under hub congestion was formulated as a nonlinear mixed integer programming by de Camargo et al. (2009) and a robust bi-objective uncapacitated single allocation p-hub median problem was modeled as a mixed integer nonlinear programming in Amin-Naseri et al. (2016).

However, over a context of discrete linear programming models, different problems of hub-and-spoke network may also be seen in the literature.

Examples of a mixed-integer linear programming model can be seen in: Hamacher et al. (2004) for uncapacitated hub location problem with multiple allocation; Correia et al. (2011) for single-allocation hub location problem with capacity choices and balancing requirements; Peiró et al. (2014) for uncapacitated r-allocation p-hub median problem and Rieck et al. (2014) for many-to-many location-routing with inter-hub transport and multi-commodity pickup-and-delivery.

A 0-1 (or binary) integer linear programming model is formulated, for example, in: Lopes et al. (2016) for the proposed many-to-many p-location-hamiltonian cycle problem and in Thomadsen and Larsen (2007) for fully interconnected backbone and access Networks.

In hub-and-spoke problems, different networks are developed to properly flow demands. For this reason, several concepts of graph theory are also used for the following problems:

p-median problem, uncapacitated facility location problem, p-center problem and covering problems (Campbell, 1994). These problems from location theory are well studied in the literature, as seen in (Tamir, 2001), (Reese, 2005), (Li, 2013) and (Farahani et al., 2012).

Some characteristics of the vehicle routing problem are used to design the routes at cluster level, as well as a few characteristics of the traveling salesman problem are also used to make routes both at cluster and hub level.

Another category of problems from graph theory that is included on the literature of hub-and-spoke is the network flow problem. The shortest path algorithm may be implemented for all demands and the flow conservation is also considered essential.

In next sections, classic optimization problems used in transportation literature are described (section 3.2) and our proposed problems are presented (section 3.3).

3.2 Classic Problems Used in Network Structure

On the proposed models of this study, characteristics of shortest path problem, traveling salesman problem and p-median problem are used. p-median problem is also used on the constructive heuristic (see chapter 4). In this section, these problems are described.

3.2.1 Shortest Path Problem

Shortest path problem is a network problem modeled and solved by linear programming (Formulation SPP). Thus, it is classified as a network flow programming (Chinneck, 2006).

In the shortest path problem, a shortest path between a source node and a destination node is determined.

Let a network made by the graph $G = (V, A)$ where V is the set of nodes with n nodes and A is a set of arcs with m arcs. c_{ij} is the cost (for example: distance or time) per unit flow on arc $(i, j) \in A$ (arc linking the nodes $i \in V$ and $j \in V$). Variable x_{ij} is an integer variable that receives the value 1, if arc $(i, j) \in A$ is inserted in the solution of shortest path between source node s and destination node t, or 0, otherwise. The mathematical model is as follows:

Formulation of the Shortest Path Problem (SPP)

$$min \sum_{(i,j)\in A} c_{ij}\ x_{ij}$$

$s.t.:$

$$\sum_{j|(i,j)\in A} x_{ij} - \sum_{j|(j,i)\in A} x_{ji} = 1 \qquad if\ i = s, \tag{3.1}$$

$$\sum_{j|(j,i)\in A} x_{ji} - \sum_{j|(i,j)\in A} x_{ij} = 1 \qquad if\ i = t, \tag{3.2}$$

$$\sum_{j|(i,j)\in A} x_{ij} - \sum_{j|(j,i)\in A} x_{ji} = 0 \qquad if\ i \neq s,\ i \neq t \tag{3.3}$$

$$x_{ij} \in \{0,1\} \qquad (i,j) \in A \tag{3.4}$$

In formulation SPP, the cost of path is minimized in the objective function. Constraints 3.1, 3.2 and 3.3 are flow conservation constraints.

In constraints 3.1, the number of outcoming arcs allowed in the source node is equal to one. In constraints 3.2, the number of incoming arcs allowed in the destination node is equal to one. With constraints 3.3, the number of outgoing arcs is warranted to be equal to the number of incoming arcs for each intermediate node (different from source or destination).

Constraints 3.4 are integrality constraints.

Because the shortest path problem is recurrently used on transportation and telecommunication networks (Ahuja et al., 1989), several formats of problems have been studied in the literature. The main formats of shortest path problem are: finding the shortest path between a single pair (one source and one destination), finding the shortest path between a single source and all other nodes and finding the shortest path between all-pairs (from every node to every other node).

Problems can also be composed of all arcs with positive cost or of some arcs with negative cost.

Some algorithms for shortest path problem are also presented in the Graph Theory literature. Among existing algorithms, the most used ones are: Dijkstra algorithm ((Dijkstra, 1959), see details in the chapter 4), Bellman-Ford algorithm (see Ford Jr (1956)), Floyd-Warshall (Floyd, 1962) and Johnson's algorithm (Johnson, 1977).

3.2.2 Traveling Salesman Problem

Traveling Salesman problem was described by Dantzig et al. (1954) as:

"*Find the shortest route (tour) for a salesman starting from a given city, visiting each of a specified group of cities, and then returning to the original point of departure.*"

In other words, in the traveling salesman problem, the shortest Hamiltonian cycle is searched from all possible cycles formed by all cities.

Note that for a group of n cities, each city can only be visited once by the salesman and a pre-set order of visits is not established by the problem.

The mathematical formulation presented by Dantzig et al. (1954) is a binary integer linear programming formulation (Formulation TSP).

Let a set of nodes (cities) V with n nodes, c_{ij} is the cost between the nodes i and j, where i and $j \in V$. The variable x_{ij} receives the values 1, if the node j is visited in the cycle after the node i, or 0, otherwise. The mathematical model is as follows:

Formulation of the Traveling Salesman Problem (TSP)

$$min \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} c_{ij} \; x_{ij}$$

$$s.t. :$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \forall i \in V \tag{3.5}$$

$$\sum_{j=1}^{n} x_{ji} = 1 \qquad \forall i \in V \tag{3.6}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq \mid S \mid -1 \qquad S \subset V, \quad 2 \leq \mid S \mid \leq \mid V \mid -2 \tag{3.7}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in V \tag{3.8}$$

In Formulation TSP, the cost of tour is minimized in the objective function. The constraints (3.5) means that in each node i there is only one arc where the salesman is allowed to go out. Whereas constraints (3.6) means that in each node i there is only one arc from which the salesman is allowed to arrive.

Using only the constraints (3.5) and (3.6), subtours are not avoided (see Figure 3.1). So, constraints (3.7) are added as subtour elimination constraints.

In Figure 3.1, square node is start and end point of tour. Circle nodes are intermediate nodes of tour.

The solution presented in Figure 3.1 is not a valid solution for the traveling salesman problem because two tours were made with the set of the 7 nodes presented.

Using the subtour elimination constraint, as showed by Figure 3.1, the subset S can be composed by the nodes 1, 2 and 6. Then, the constraint 3.3.1.

$$x_{12} + x_{26} + x_{61} \leq 3 - 1 \tag{3.3.1}$$

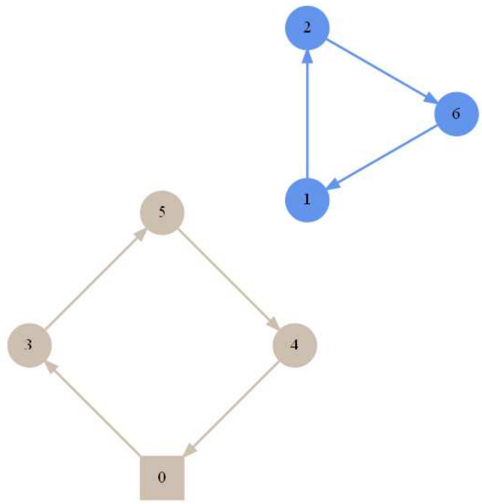


Figure 3.1: Graph with 7 nodes making subtour.

Therefore, only two arcs are allowed between the nodes of subset $S = \{1, 2, 6\}$. Consequently, the subtour identified in Figure 3.1 can be avoided by the constraint (3.3.1).

Finally, constraints (3.8) are integrality constraints.

The Presented mathematical model is composed of $(2^n - 2n - 2)$ subtour elimination constraints (Laporte, 1992). For this reason, finding a solution for the model is challenging when a large value of n (number of nodes) is used.

When that model was proposed by Dantzig et al. (1954), a cutting plane algorithm was also proposed by the authors in order to solve the problem. The formulation was relaxed (the subtour constraint was suppressed) and the existence of subtour was checked among the solutions provided by the relaxed problem. In the case of one subtour was identified, a constraint forbidding that specific subtour was inserted in the model and the solving continues.

An alternative, but equivalent form, to the subtour elimination constraint is the cut-set constraint (Laporte, 1992). Below, cut-set constraint is showed (see constraint (3.9). It is equivalent to sub-tour elimination constraint (Constraints (3.7))

$$\sum_{i \in S} \sum_{j \in V-S} x_{ij} \geq 1 \qquad S \subset V, \quad 2 \leq \mid S \mid \leq \mid V \mid -2 \tag{3.9}$$

Using the cut-set constraint for the example of Figure 3.1, and considering that the subset S can be composed by the nodes $1, 2$ and 6, then:

$$x_{10} + x_{13} + x_{14} + x_{15} + x_{20} + x_{23} + x_{24} + x_{25} + x_{60} + x_{63} + x_{64} + x_{65} \geq 1 \tag{3.5.1}$$

Therefore, at least one arc must be inserted between the sub-set $S = \{1, 2, 6\}$ and the sub-set $V - S = \{0, 3, 4, 5\}$. Consequently, the subtour identified in Figure 3.1 can be avoided by the constraint (3.5.1).

Because the traveling salesman problem is one of the most studied/researched problems in the combinatorial optimization field (Laporte, 1992), several variants of this problem have been proposed in literature. For example: the traveling salesman problem with time windows (see Dumas et al. (1995)), multiple traveling salesman problem (see Gavish and Srikanth (1986)) and prize collecting traveling salesman problem (see Balas (1989)).

Due to the complexity of traveling salesman problem, several heuristics have been proposed in the literature in order to solve large instances of this problem. Some examples are: Nearest neighbor algorithm (see Bellmore and Nemhauser (1968)) (details are given in the chapter 4), Christofides' heuristic (see Christofides (1976)) and GKS heuristic (see Glover et al. (2001)).

3.2.3 p-Median Problem

In the p-median problem, p facilities may be installed, which could be schools, hospitals, warehouses or ambulances, etc. The main idea is to decrease the demand weighted average distance (see Reese (2005))(see Daskin and Maass (2015)).

The capacity of the facilities are unlimited in this problem and, consequently, customers allocation are only defined by distance and demand.

A binary integer linear programming formulation of this problem is presented below (Formulation p-MP).

Let a complete graph $G = (V, E)$ where V is the set of vertices composed of n nodes and E is the set of edges. p is the number of facilities, d_i is the demand of customer i, c_{ij} is the cost of satisfying customer i from facility j and x_{ij} is the allocation variable that receive 1 if the customer i is allocated to facility j, or 0 otherwise. The mathematical model is as follows:

Formulation of the p-Median Problem (p-MP)

$$min \sum_{i=1}^{n} \sum_{j=1}^{n} d_i \; c_{ij} \; x_{ij}$$

$s.t.:$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \forall i \in V \tag{3.10}$$

$$\sum_{j=1}^{n} x_{jj} = p \tag{3.11}$$

$$x_{ij} \leq x_{jj} \qquad \forall i, j \in V \tag{3.12}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in V \tag{3.13}$$

In the Formulation p-MP, the demand-weighted total cost is minimized in the objective function. In the constraints 3.10, each customer i is assigned to only one facility. In the constraint 3.11, the number of facilities to locate is equal to p. In the constraints 3.12, a customer i could be allocated to j if j is a facility. Constraints 3.13 are integrality constraints.

The complexity of p-median problem is NP-hard on general graphs and networks for an arbitrary p. Although the p-median problem is solved in polynomial time when p is previously established, the problem is solved with expensive computational time (Reese, 2005).

Several heuristics have been proposed to solve the p-median problem. Some methods found in the literature are highlighted in Mladenović et al. (2007): Greedy, Stingy, Dual ascent, Alternate and Interchange.

3.3 Proposed Models

In this subsection, four mathematical models are proposed for the p-hub location and routing problem. In the two first models, the network created is composed of circular routes both at hub level and cluster level (set composed of hub and their allocated spoke nodes). Whereas in the last two models, the network is composed of circular routes in cluster and direct links between hubs (complete graph at hub level).

As it was previously said, some characteristics of the traveling salesman problem, the p-median problem and the shortest path problem are used in the proposed problems.

The number of hub nodes is previously defined and the hub selection is performed among all nodes.

The size of cluster is defined by its number of nodes. From now on, clusters will be composed by a standalone hub (hub without allocated spoke node), two nodes (one hub and one spoke) or more.

The number of spoke nodes allocated to a hub is not limited. Therefore, all spoke nodes can be allocated to the same hub.

Each circular route is composed of all nodes from a set of hubs or from each cluster. Therefore, a traveling salesman problem is solved for each set of nodes.

Demands are routed on the proposed network by searching the shortest path between the origin and the destination nodes. Routes are composed of directed arcs.

All developed models are composed of a location problem, an allocation problem, a service design problem and a routing problem.

Below, the mathematical models for all versions are presented. In the version with sub-hubs, all constraints are described. While on the version without sub-hub, only constraints different from the version with sub-hub are described.

3.3.1 Model with circular hub route and sub-hubs

This first network model (Figure 3.2) was developed trying to minimize the delivery time for some demands and to reduce transport cost in the network with an economy of scale at hub network level (da Costa Fontes and Goncalves, 2016b).

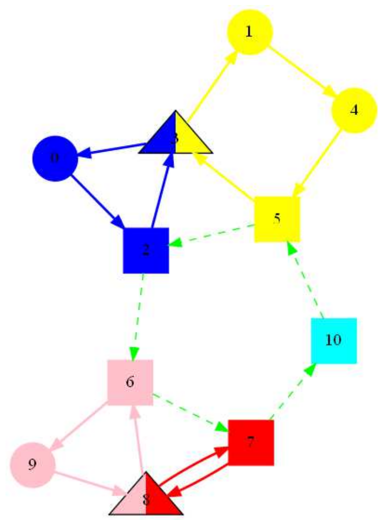


Figure 3.2: Network with circular hub network and with sub-hubs.

In all figures representing network presented in this section, the square nodes represent the hubs, circles nodes are the spokes and triangle nodes refer to sub-hubs. Each color represents a particular cluster, hub links are represented by green dashed lines and regional routes are represented by full lines with the same color as the cluster.

Sub-hub is a particular spoke node which is allocated to two different close clusters. In other words, sub-hub is an intersection point between two regional routes where transshipments can be performed.

Now, in the model of this version, two sub-hubs can be linked by a regional arc (see arc (3,8) in Figure 3.3). But, also in this model, one sub-hub can only be allocated to a pair of close hubs and a pair of close hubs can only be allocated to one sub-hub. Therefore, the examples presented in Figure 3.4 and 3.5 cannot be built by our developed model.

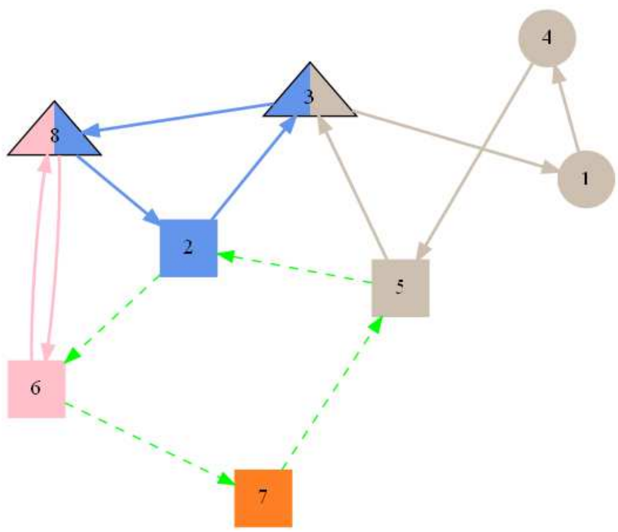


Figure 3.3: Network composed of a link between two sub-hubs.

In Figure (3.3), the cluster blue is made by nodes 2 (hub), 3 (sub-hub) and 8 (sub-hub). Then, a route is made between these nodes.

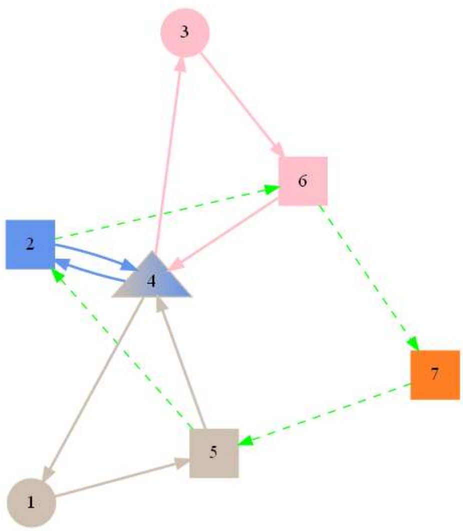


Figure 3.4: Forbidden situation: network with one node allocated to three hubs.

In network of Figure (3.4), one sub-hub (node 4) is allocated to hub pair 2 and 5 and to hub pair 2 and 6. This characteristic is not allowed by the constraints of the proposed model.

In the network presented on Figure (3.5), two sub-hubs, nodes 1 and 3, are allocated to the same pair of hubs, nodes 2 and 5. This characteristic is not allowed in our

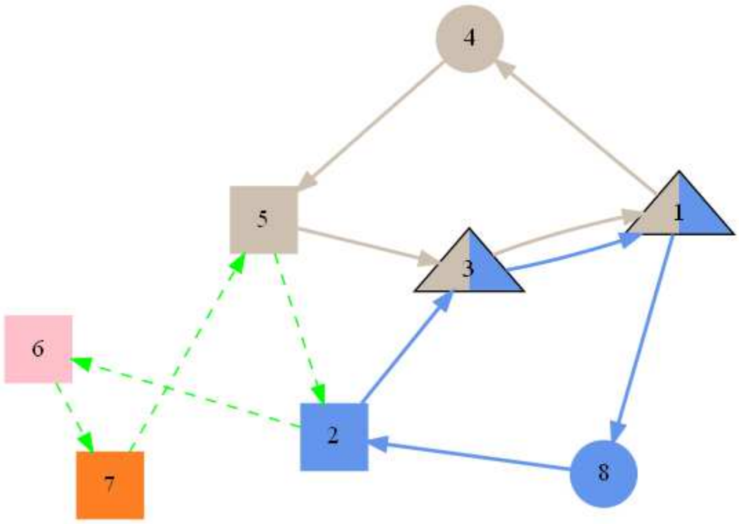


Figure 3.5: Forbidden situation: network with two nodes allocated to the same pair of hubs.

mathematical models.

This first model we propose can be classified as a Two-Level p-Hub Location-Routing Problem with Sub-Hub, because the network is composed of two different levels of routes. Routes in second level can be design with an intersection point and demands are routed from each node to all others nodes.

Before presenting the mathematical model, some information should first considered:

Let V be the set of nodes with n nodes, this mathematical model is composed of constants: cost matrix, where c_{ij} is the cost per unit of flow from node i to node j where i, $j \in V$, the value p being the number of hubs; flow matrix, where fl_{ij} is the flow from node i to node j and the value α representing the discount factor on the unit of flow between hubs.

Hub location, sub-hub location and spoke allocation are defined by the variables of model: $u_i = 1$ if the vertex i is a hub with no spoke allocated to it, or 0 otherwise, $x_{ij} = 1$, if the vertex i is allocated to hub j, or 0 otherwise, and $t_i = 1$, if i is sub-hub, or 0 otherwise. A vertex i is a hub with spoke locations if $x_{ii} = 1$.

Service design is composed of the variable $y_{ij} = 1$, if there exists the arc-non-hub $(i; j)$, or 0 otherwise, and $b_{ij} = 1$, if there exists the arc-hub $(i; j)$, or 0 otherwise.

Concerning cargo routing, it was created variables $w_{ijkl} = 1$, if there exists a flow from i to j routed via arc-hub $(k; l)$, or 0 otherwise and $s_{ijkl} = 1$, if there exists a flow from i to j routed via arc-non-hub $(k; l)$, or 0 otherwise.

The model can be described as follows:

Formulation of the Two-Level p-Hub Location-Routing Problem with Sub-Hub (tLpHLRPSH)

$$min \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} fl_{ij}\ c_{ij}\ x_{ij} + \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \sum_{k=1, k\neq j}^{n} \sum_{l=1, l\neq k, l\neq i}^{n} fl_{ij}\ c_{kl}\ s_{ijkl} + \\ + \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \sum_{k=1, k\neq j}^{n} \sum_{l=1, l\neq k, l\neq i}^{n} fl_{ij}\ \alpha\ c_{kl}\ w_{ijkl}$$

$s.t.:$

$$\sum_{j=1}^{n} x_{jj} + \sum_{j=1}^{n} u_j = p \tag{3.14}$$

$$x_{ij} \leq x_{jj}, \qquad \forall i, j \in V \tag{3.15}$$

$$t_i \leq \sum_{j=1, j\neq i}^{n} x_{ij}/2, \qquad \forall i \in V \tag{3.16}$$

$$u_i + t_i + x_{ii} \leq 1, \qquad \forall i \in V \tag{3.17}$$

$$\sum_{j=1}^{n} x_{ij} = 1 + t_i - u_i, \qquad \forall i \in V \tag{3.18}$$

$$x_{ij} + x_{ik} + x_{lj} + x_{lk} \leq 3, \qquad \forall i, j, k, l \in V, \\ i \neq j, k \neq j, k \neq i, l \neq j, l \neq k, l \neq i \tag{3.19}$$

$$y_{ij} + x_{ik} + x_{il} \leq 2 + x_{jk} + x_{jl}, \qquad \forall i, j, k, l \in V, \\ j \neq i, k \neq i, k \neq j, l \neq i, l \neq k, l \neq j \tag{3.20}$$

$$y_{ij} + x_{ik} + x_{jl} \leq 2 + x_{jk} + x_{il} + t_i/2 + t_j/2, \qquad \forall i, j, k, l \in V, \\ i \neq j, k \neq j, i \neq l, l \neq k \tag{3.21}$$

$$x_{kj} + x_{lj} + y_{ik} + y_{il} \leq 3, \qquad \forall i,j,k,l \in V, \quad j \neq i, k \neq i, l \neq i, l \neq k \tag{3.22}$$

$$x_{kj} + x_{lj} + y_{ki} + y_{li} \leq 3, \qquad \forall i,j,k,l \in V, \quad j \neq i, k \neq i, l \neq i, l \neq k \tag{3.23}$$

$$y_{ij} + y_{ji} \leq 1 + x_{ii} + x_{jj}, \qquad \forall i,j \in V,\ i \neq j \tag{3.24}$$

$$\sum_{j=1, j\neq i}^{n} y_{ij} = 1 + t_i - u_i, \qquad \forall i \in V \tag{3.25}$$

$$\sum_{j=1, j\neq i}^{n} y_{ji} = 1 + t_i - u_i, \qquad \forall i \in V \tag{3.26}$$

$$\sum_{j=1, j\neq i}^{n} b_{ij} = x_{ii} + u_i, \qquad \forall i \in V \tag{3.27}$$

$$\sum_{j=1, j\neq i}^{n} b_{ji} = x_{ii} + u_i, \qquad \forall i \in V \tag{3.28}$$

$$b_{ij} + b_{ji} \leq 1 + 2/p, \qquad \forall i,j \in V,\ i \neq j \tag{3.29}$$

$$\sum_{l=1, l\neq i}^{n} w_{ijkl} + \sum_{l=1, l\neq i}^{n} s_{ijkl} = 1, \qquad \forall i,j \in V,\ i = k, i \neq j \tag{3.30}$$

$$\sum_{l=1, l\neq j}^{n} w_{ijlk} + \sum_{l=1, l\neq j}^{n} s_{ijlk} = 1, \qquad \forall i,j \in V,\ j = k, j \neq i \tag{3.31}$$

$$\sum_{l=1, l\neq i, l\neq k}^{n} w_{ijkl} + \sum_{l=1, l\neq i, l\neq k}^{n} s_{ijkl} = \sum_{l=1, l\neq j, l\neq k}^{n} w_{ijlk} + \sum_{l=1, l\neq j, l\neq k}^{n} s_{ijlk} \qquad \forall i,j,k \in V, \quad k \neq i, k \neq j, j \neq i \tag{3.32}$$

$$s_{ijkl} \leq y_{kl} \qquad \forall i,j,k,l \in V, \qquad l \neq k, l \neq i, k \neq j, j \neq i \tag{3.33}$$

$$w_{ijkl} \leq b_{kl} \qquad \forall i,j,k,l \in V, \qquad l \neq k, l \neq i, k \neq j, j \neq i \tag{3.34}$$

$$x_{ij},\ u_i,\ t_i,\ y_{ij},\ b_{ij},\ w_{ijkl},\ s_{ijkl} \in \{0,1\} \qquad \forall i,j,k,l \in V \tag{3.35}$$

In the objective function, the location and allocation cost and the overall transportation cost are minimized. For location of hub nodes and allocation of a node to a hub, the quantity of flow is taken into account, while the transport cost is divided in two parts: cost of flow at arc non-hub and cost of flow at arc hub with a discount factor α.

The Location and allocation are composed by constraints (3.14) - (3.19):

In constraint (3.14), the number of hubs is equal to p. In this model, isolated points with large demand can also be selected as a hub. Then, two different types of hubs are allowed, hubs with spoke allocation x_{jj} and standalone hubs (hub with no spoke) u_j.

Constraints (3.15), a vertex i will be allocated to a vertex j if j is a hub with allocation.

In constraints (3.16), a vertex i will be chosen as sub-hub if it is allocated to at least two hubs. Then, multiple allocation (double allocation) is allowed to some nodes, using these constraints.

In constraints (3.17), every vertex may be chosen as a hub with allocation, a hub without allocation or a spoke. A vertex can just be chosen as a sub-hub if it is not a hub.

In constraints (3.18), every vertex i will be allocated to one hub j, or to two hubs j in the case of i being a sub-hub. However, it can also be a hub with or without allocation.

Because of constraints (3.16) and (3.18), networks as the one presented in Figure (3.4) are not allowed.

In constraints (3.19), two sub-hubs cannot be allocated to the same pair of hubs. Networks, as the one presented in Figure (3.5), are forbidden by these constraints. The maximum number of sub-hubs is $(p * (p-1)/2)$, where p is the number of hubs. This limit is imposed by constraints (3.19).

Between the constraints (3.20) - (3.29), arcs linking the nodes are created. Service design is represented by these constraints:

In constraints (3.20), an arc y_{ij} linking one sub-hub i with an other one spoke node (sub-hub or not) is allowed, but only if they are allocated in a common hub.

In constraints (3.21), the existence of an arc linking the vertex i and j if the vertex i and j are allocated to different hubs will be not allowed. There will only exist an arc linking the vertex i and j if they are allocated to the same hub. Then, arc y_{ij} will also not link hubs. It may link spokes (sub-hub or not) or one hub with one spoke. In constraints (3.21), a link between two sub-hubs is allowed, but a common hub is needed due to constraints (3.20).

The characteristic presented by Figure (3.3) is allowed because of constraints (3.20) and (3.21).

In constraints (3.22), two arcs going out of a vertex i to two others vertex that belong to the same hub j will be not allowed.

In constraints (3.23), two arcs arriving in a vertex i from two other vertices that belong to a same hub j will not be allowed. In other words, with the constraints (3.22) and (3.23), two arcs are sent from a sub-hub i to two different clusters and two arcs are received by a sub-hub i from two different clusters.

In constraints (3.24), in network, arc-non-hub (i, j) or arc-non-hub (j, i) is used. But if the cluster just has one spoke (sub-hub or not), arcs-non-hub (i, j) and (j, i) can be used simultaneously.

In constraints (3.25), one arc-non-hub leaves every vertex i, but if i is a sub-hub, two arcs can leave. However, if the vertex i is a hub without allocation, arc-non-hub does not exist.

In constraints (3.26), one arc-non-hub will be received by every vertex i, but if i is a sub-hub, two arcs will be received. However, if the vertex i is a hub without allocation, arc-non-hub cannot be received.

In constraints (3.27), one arc-hub leaves every vertex hub i.

In constraints (3.28), one arc-hub will be received by every vertex hub i.

In constraints (3.29), both arc-hub (i, j) or arc-hub (j, i) can be used in the network. But, if the problem just has two hubs, both arcs (i, j) and (j, i) can be used.

Because of constraints (3.27), (3.28) and (3.29), a minimum number equal to two hubs is required to solve this model. Although, the idea of alternative path is used to minimize the transport cost, hub arcs with economy of scale cannot be suppressed from the model because they significantly reduce transport cost.

After the mentioned constraints, the network is now built. Then, the cargo routing constraints are done by:

Flow conservation constraints are represented by the constraints (3.30), (3.31), (3.32).

In constraints (3.30), the flow of every vertex i towards every vertex j will leave the source using the arc (i, l), where the flow can be a flow w (flow in a arc-hub) or a flow s (flow in a arc-non-hub). Two hubs are linked by arc-hub and a hub with a spoke (sub-hub or not) or two spokes (sub-hubs or not) are linked by an arc-non-hub.

In constraints (3.31), the flow of every vertex i towards every vertex j will be received in the destination using the arc (l, j), where the flow can be a flow w (flow in an arc-hub) or a flow s (flow in an arc-non-hub).

In constraints (3.32), all flows that leave an intermediate vertex k (which is not the source or the destination), will be equal to the flow that has arrived at the intermediate vertex.

In Constraints (3.33), the flow s will just exist if the arc y exists.

In constraints (3.34), the flow w will just exist if an arc-hub b exists.

Finally, constraints (3.35) are integrality constraints.

Regarding the problem instances of problems with six or more hubs, a subtour at the hub-route cannot be avoided by the constraints above. Constraints to eliminate subtours are added in the previous model.

For the same reasons, at regional route, constraints to eliminate subtours are added in the previous model for clusters with more than four nodes. Subtours are avoided by the constraints (3.24) in clusters with four or less nodes.

Then, for each supposedly viable solution found, H will be the set of selected hubs nodes with p nodes, C will be the set of selected clusters with allocation, where $C = \{C_1, C_2, ..., C_m\}$, C_q will be the set of selected nodes, which belong to the same cluster q, h_q will be the hub node from set C_q (cluster q) and S will be any subset of nodes from a set H or from a set C_q. The constraints are:

$$\sum_{i \in S, j \in H-S} b_{ij} \geq 1, \qquad 2 \leq \mid S \mid \leq \mid H \mid -2, \quad with \mid H \mid = p \tag{3.36}$$

$$\sum_{i,j \in S} y_{ij} \leq \mid S \mid -1 + \sum_{i \in S} x_{ii}, \quad 2 \leq \mid S \mid \leq \mid C_q \mid -2, \ and \ h_q \in (C_q - S)$$
$$\forall \ C_q, \quad where \ q \in \{1, 2, ..., \sum_{j=1}^{n} x_{jj}\} \tag{3.37}$$

Constraints (3.36) are cut-set constraints. These constraints are equivalent to subtour elimination constraints. Constraints (3.36) are generated and included at the model, whenever it is necessary to eliminate subtours at hub route.

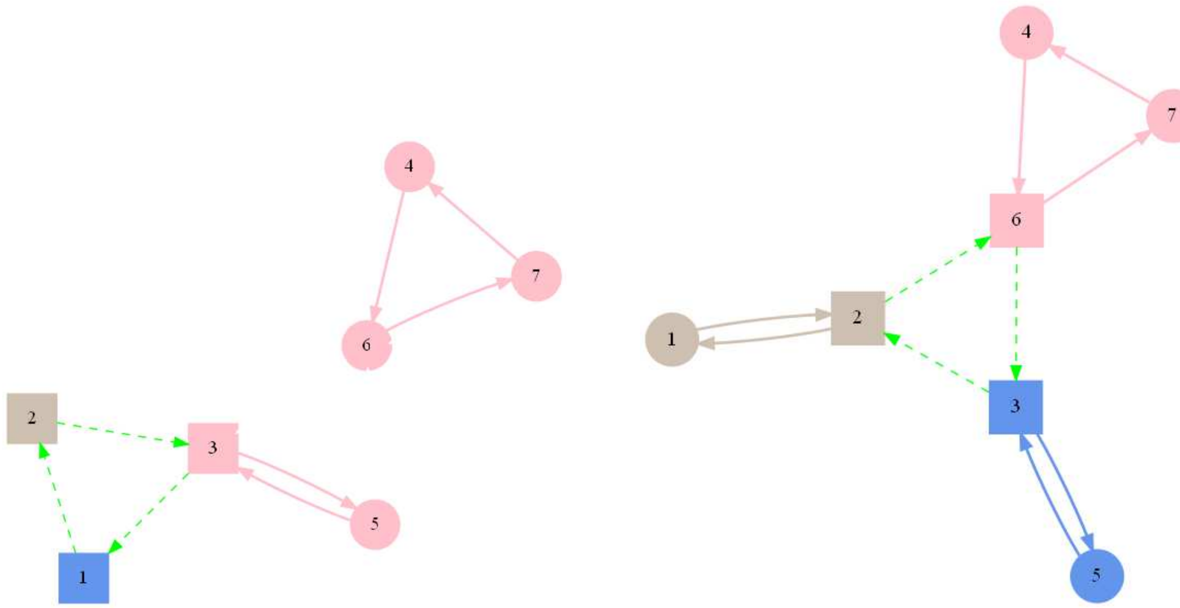
In this problem, a cluster can be constituted by a number of nodes varying between one and $n-p+1$. Cluster with one node is composed by hub without allocation and the cluster with the maximal number of nodes is composed by all spoke nodes and the hub.

A second important characteristic for each cluster is that each node can be selected as a hub node. Then, during the execution of the problem, solving the service design in each cluster corresponds to solving the traveling salesman problem without a fixed start point and without a predetermined number of nodes.

Constraints (3.37) are a variation of subtour elimination constraints. In these constraints, the subset S selected is not composed by the hub from an analyzed cluster. Then, sub-set S can be accepted as cluster of a next solution found if some node from S is a hub (see Figure 3.6a and 3.6b).

Then, constraints (3.37) are generated and included in the model, whenever necessary, to eliminate subtours at regional routes.

A cutting plane approach is used to solve the developed mathematical model. In algorithm, the formulation is relaxed because the subtour constraints are suppressed from the model. Each time an integer-feasible solution for the relaxed problem is found, the existence of a subtour is checked. When a subtour is identified, a constraint forbidding



(a) Network with subtour. (b) Network without subtour.

Figure 3.6: possible solution in phase i and in phase $i+k$ of exact method.

that specific subtour is inserted in the model. This dynamic iteration with the optimization process is made in the Cplex solver through callback procedure. More specifically, a lazy-constraint callback is used.

Lazy constraints are constraints not provided in the constraint matrix of the problem, but they cannot be violated by a solution of relaxed problem (Beltramin, 2015).

During the callback, the feasibility of each integer solution of relaxed problem is checked. Any violated lazy constraint is related, allowing the solver to discard the unfeasible solution and to add the violated lazy constraint in the relaxed problem (Bertsimas et al., 2016).

In the implemented cutting plane, one subtour elimination constraint is inserted in the model after the algorithm find the network solution presented in Figure 3.6a (phase i of resolution algorithm). However, when doing that, the network solution presented in Figure 3.6b can no more be found in a phase $i+k$ of resolution because it is eliminated of the search tree if the classical subtour elimination constraint or the classical cut-set constraint are used. Note that the network solution presented in Figure 3.6b is a valid solution because no subtour appears.

Seeking to avoid the problem describe above, a variant of subtour elimination constraint was developed in this study. It is used on the model when a subtour is found in a cluster (see constraint (3.37)). In relation to the case of Figure 3.6a, the sub-set S selected in the implemented algorithm is composed by the nodes 6, 4 and 7.

In the proposed model, setting $p = n$, all nodes are forced to be hub. Then, the proposed problem is reduced to the traveling salesman problem in this special case.

Considering that the traveling salesman problem is NP-Hard, then the proposed problem is also NP-Hard.

3.3.2 Model with circular hub route and without sub-hub

In this second network model, a single allocation of spoke nodes is presented (Figure 3.7). We introduce this model in order to compare its performances with the ones from previous model (da Costa Fontes and Goncalves, 2016b).

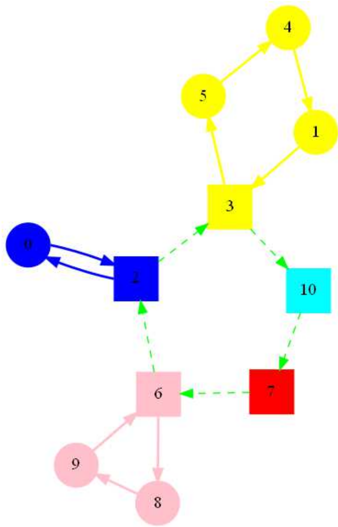


Figure 3.7: Network with circular hub network and without sub-hub.

In this version, transport cost is minimized only by economy of scale.

Transshipments are only realized in hub nodes.

This model was classified as a Two-Level p-Hub Location-Routing Problem because the network is composed of two different routes and demands are routed from each node to all other nodes.

This mathematical model was developed from the previous model. Some constraints corresponding to sub-hubs were removed and others were modified because the variable t_i is no longer presented in this model.

Constraints (3.16), (3.19), (3.20), (3.22) and (3.23) were removed.

Below, the whole model is presented. Although just the replaced constraints are described.

Let V be the set of nodes with n nodes, c_{ij} is the cost per unit of flow from node i to node j where i, $j \in V$, p is the number of hubs, fl_{ij} is the flow from node i to node j and α is the discount factor on the unit of flow between hubs.

Hub location and spoke allocation are defined by the variables of model: $u_i = 1$ if the vertex i is a hub with no spoke allocated to it, or 0 otherwise, and $x_{ij} = 1$, if the vertex i is allocated to hub j, or 0 otherwise. A vertex i is a hub with spoke locations if $x_{ii} = 1$.

Service design is composed of the variable $y_{ij} = 1$, if there exists the arc-non-hub $(i;j)$, or 0 otherwise, and $b_{ij} = 1$, if there exists the arc-hub $(i;j)$, or 0 otherwise.

With respect to the cargo routing, it was created the variables $w_{ijkl} = 1$, if there exists a flow from i to j routed via arc-hub $(k;l)$, or 0 otherwise and $s_{ijkl} = 1$, if there exists a flow from i to j routed via arc-non-hub $(k;l)$, or 0 otherwise.

The model can be defined as follows:

Formulation of the Two-Level p-Hub Location-Routing Problem (tLpHLRP)

$$min \sum_{i=1}^{n} \sum_{j=1,j\neq i}^{n} fl_{ij}\ c_{ij}\ x_{ij} + \sum_{i=1}^{n} \sum_{j=1,j\neq i}^{n} \sum_{k=1,k\neq j}^{n} \sum_{l=1,l\neq k,l\neq i}^{n} fl_{ij}\ c_{kl}\ s_{ijkl} + \\ + \sum_{i=1}^{n} \sum_{j=1,j\neq i}^{n} \sum_{k=1,k\neq j}^{n} \sum_{l=1,l\neq k,l\neq i}^{n} fl_{ij}\ \alpha\ c_{kl}\ w_{ijkl}$$

$s.t.:$

$$\sum_{j=1}^{n} x_{jj} + \sum_{j=1}^{n} u_j = p \tag{3.38}$$

$$x_{ij} \leq x_{jj}, \qquad \forall i,j \in V \tag{3.39}$$

$$u_i + x_{ii} \leq 1, \qquad \forall i \in V \tag{3.40}$$

$$\sum_{j=1}^{n} x_{ij} = 1 - u_i, \qquad \forall i \in V \tag{3.41}$$

$$y_{ij} + x_{ik} + x_{jl} \leq 2, \qquad \forall i,j,k,l \in V,\ i \neq j, k \neq j, i \neq l, l \neq k \tag{3.42}$$

$$y_{ij} + y_{ji} \leq 1 + x_{ii} + x_{jj}, \qquad \forall i,j \in V,\ i \neq j \tag{3.43}$$

$$\sum_{j=1, j\neq i}^{n} y_{ij} = 1 - u_i, \qquad \forall i \in V \tag{3.44}$$

$$\sum_{j=1, j\neq i}^{n} y_{ji} = 1 - u_i, \qquad \forall i \in V \tag{3.45}$$

$$\sum_{j=1, j\neq i}^{n} b_{ij} = x_{ii} + u_i, \qquad \forall i \in V, \tag{3.46}$$

$$\sum_{j=1, j\neq i}^{n} b_{ji} = x_{ii} + u_i, \qquad \forall i \in V, \tag{3.47}$$

$$b_{ij} + b_{ji} \leq 1 + 2/p, \qquad \forall i,j \in V,\ i \neq j \tag{3.48}$$

$$\sum_{l=1, l\neq i}^{n} w_{ijkl} + \sum_{l=1, l\neq i}^{n} s_{ijkl} = 1, \qquad \forall i,j \in V,\ i = k, i \neq j \tag{3.49}$$

$$\sum_{l=1, l\neq j}^{n} w_{ijlk} + \sum_{l=1, l\neq j}^{n} s_{ijlk} = 1, \qquad \forall i,j \in V,\ j = k, j \neq i \tag{3.50}$$

$$\sum_{l=1, l\neq i, l\neq k}^{n} w_{ijkl} + \sum_{l=1, l\neq i, l\neq k}^{n} s_{ijkl} = \sum_{l=1, l\neq j, l\neq k}^{n} w_{ijlk} + \sum_{l=1, l\neq j, l\neq k}^{n} s_{ijlk}$$
$$\forall i,j,k \in V, \qquad k \neq i, k \neq j, j \neq i \tag{3.51}$$

$$s_{ijkl} \leq y_{kl} \qquad \forall i,j,k,l \in V, \qquad l \neq k, l \neq i, k \neq j, j \neq i \tag{3.52}$$

$$w_{ijkl} \leq b_{kl} \qquad \forall i,j,k,l \in V, \qquad l \neq k, l \neq i, k \neq j, j \neq i \tag{3.53}$$

$$x_{ij},\ u_i,\ y_{ij},\ b_{ij},\ w_{ijkl},\ s_{ijkl} \in \{0,1\} \qquad \forall i,j,k,l \in V \tag{3.54}$$

Constraints (3.17) were replaced by constraints (3.40):

Every vertex may be a hub with allocation, a hub without allocation or a spoke.

Constraints (3.18) were replaced by constraints (3.41):

Every vertex i will be allocated to one hub j. But, it can also be a hub without allocation.

Constraints (3.21) were replaced by constraints (3.42):

In these constraints, the existence of an arc linking the vertex i and j, will not be allowed if they are allocated to different hubs. Arc linking the vertex i and j will just exist if they are allocated to the same hub. Then, arc y_{ij} will also not link hubs. It will either link spokes or one hub with one spoke.

Constraints (3.25) were replaced by constraints (3.44):

One arc-non-hub leaves every vertex i. However, if the vertex i is a hub without allocation, arc-non-hub does not exist.

Constraints (3.26) were replaced by constraints (3.45):

One arc-non-hub is received in every vertex i. However, if the vertex i is a hub without allocation, arc-non-hub is not received in the node.

Constraints (3.35) were replaced by constraints (3.54): these are integrality constraints.

A cutting plane approach focused on subtour elimination, similar to the one presented in the first version, was implemented in order to solve this second model.

This second model was developed because a network with the presented characteristics, along with routing mechanism in a same model, was not found in the literature about hub network.

Similar to the first one model, the proposed problem is also NP-Hard.

3.3.3 Model with direct hub connections and sub-hubs

Different of the two first models, now the network at hub level is composed of a complete graph. Between two hub nodes, the transport can be done in both directions (da Costa Fontes and Goncalves, 2015b).

In this new model, distinct regions are more quickly accessible, allowing for a shorter travel time associated with an economy of scale. However an increasing number of vehicles in the network can be registered.

In this third model, sub-hubs are also used to provide a network with lower transport costs along with shorter alternative paths.

Clusters can be composed of isolated nodes (one hub), two nodes (one hub and one spoke) or more. Connections between nodes from a same cluster are done in circular form by directed arcs and, between two close clusters, one intersection spoke node (sub-hub) can be offered (Figure 3.8).

Based on the categories of hub location-routing problem presented by Nagy and Salhi (2007), (see chapter 1), this model is classified as Many-to-Many p-Hub Location-Routing Problem with Sub-Hub.

The mathematical formulation is presented below and constraints are then described.

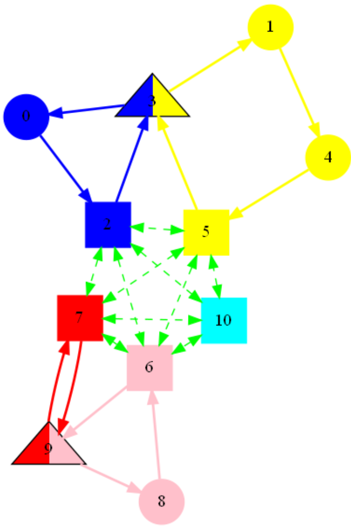


Figure 3.8: Network with direct hub connections and with sub-hubs.

Let V be the set of nodes with n nodes, c_{ij} is the cost per unit of flow from node i to node j where $i, j \in V$, p is number of hubs, fl_{ij} is the flow from node i to node j and α is the discount factor on the unit of flow between hubs.

Hub location, sub-hub location and spoke allocation are defined by variables of model: $u_i = 1$ if the vertex i is a hub with no spoke allocated to it, or 0 otherwise, $x_{ij} = 1$, if the vertex i is allocated to hub j, or 0 otherwise, and $t_i = 1$, if i is sub-hub, or 0 otherwise. A vertex i is a hub with spoke locations if $x_{ii} = 1$.

Variable $y_{ij} = 1$, if there exists an arc-non-hub $(i; j)$, or 0 otherwise, is responsible for the service design in cluster level.

For the cargo routing, the variables created were $w_{ijkl} = 1$, if there exists a flow from i to j routed between the hubs k and l, or 0 otherwise and $s_{ijkl} = 1$, if there exists a flow from i to j routed via arc-non-hub $(k; l)$, or 0 otherwise.

The model can be defined as follows:

Formulation of the Many-to-Many p-Hub Location-Routing Problem with Sub-Hub (MMpHLRPSH)

$$min \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} fl_{ij}\ c_{ij}\ x_{ij} + \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \sum_{k=1, k\neq j}^{n} \sum_{l=1, l\neq k, l\neq i}^{n} fl_{ij}\ c_{kl}\ s_{ijkl} + $$
$$+ \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \sum_{k=1, k\neq j}^{n} \sum_{l=1, l\neq k, l\neq i}^{n} fl_{ij}\ \alpha\ c_{kl}\ w_{ijkl}$$

$s.t.:$

$$\sum_{j=1}^{n} x_{jj} + \sum_{j=1}^{n} u_j = \ p \tag{3.55}$$

$$x_{ij} \leq x_{jj}, \qquad \forall i, j \in V \tag{3.56}$$

$$t_i \leq \sum_{j=1, j \neq i}^{n} x_{ij}/2, \qquad \forall i \in V \tag{3.57}$$

$$u_i + t_i + x_{ii} \leq 1, \qquad \forall i \in V \tag{3.58}$$

$$\sum_{j=1}^{n} x_{ij} = \ 1 + t_i - u_i, \qquad \forall i \in V \tag{3.59}$$

$$\begin{aligned} x_{ij} + x_{ik} + x_{lj} + x_{lk} \leq 3, \qquad & \forall i, j, k, l \in V, \\ & i \neq j, k \neq j, k \neq i, l \neq j, l \neq k, l \neq i \end{aligned} \tag{3.60}$$

$$\begin{aligned} y_{ij} + x_{ik} + x_{il} \leq 2 + x_{jk} + x_{jl}, \qquad & \forall i, j, k, l \in V, \\ & j \neq i, k \neq i, k \neq j, l \neq i, l \neq k, l \neq j \end{aligned} \tag{3.61}$$

$$\begin{aligned} & y_{ij} + x_{ik} + x_{jl} \leq 2 + x_{jk} + x_{il} + t_i/2 + t_j/2, \\ & \qquad \forall i, j, k, l \in V, \ i \neq j, k \neq j, i \neq l, l \neq k \end{aligned} \tag{3.62}$$

$$\begin{aligned} x_{kj} + x_{lj} + y_{ik} + y_{il} \leq 3, \qquad & \forall i, j, k, l \in V, \\ & j \neq i, k \neq i, l \neq i, l \neq k \end{aligned} \tag{3.63}$$

$$\begin{aligned} x_{kj} + x_{lj} + y_{ki} + y_{li} \leq 3, \qquad & \forall i, j, k, l \in V, \\ & j \neq i, k \neq i, l \neq i, l \neq k \end{aligned} \tag{3.64}$$

$$y_{ij} + y_{ji} \leq 1 + x_{ii} + x_{jj}, \qquad \forall i, j \in V, \ i \neq j \tag{3.65}$$

$$\sum_{j=1, j \neq i}^{n} y_{ij} = 1 + t_i - u_i, \qquad \forall i \in V \tag{3.66}$$

$$\sum_{j=1,j\neq i}^{n} y_{ji} = 1 + t_i - u_i, \qquad \forall i \in V \tag{3.67}$$

$$\sum_{l=1,l\neq i}^{n} w_{ijkl} + \sum_{l=1,l\neq i}^{n} s_{ijkl} = 1, \qquad \forall i,j \in V,\ i = k, i \neq j \tag{3.68}$$

$$\sum_{l=1,l\neq j}^{n} w_{ijlk} + \sum_{l=1,l\neq j}^{n} s_{ijlk} = 1, \qquad \forall i,j \in V,\ j = k, j \neq i \tag{3.69}$$

$$\sum_{l=1,l\neq i,l\neq k}^{n} w_{ijkl} + \sum_{l=1,l\neq i,l\neq k}^{n} s_{ijkl} = \sum_{l=1,l\neq j,l\neq k}^{n} w_{ijlk} + \sum_{l=1,l\neq j,l\neq k}^{n} s_{ijlk}$$
$$\forall i,j,k \in V, \qquad k \neq i, k \neq j, j \neq i \tag{3.70}$$

$$s_{ijkl} \leq y_{kl} \qquad \forall i,j,k,l \in V, \qquad l \neq k, l \neq i, k \neq j, j \neq i \tag{3.71}$$

$$\sum_{l=1,l\neq i,l\neq k}^{n} w_{ijkl} \leq x_{kk} + u_k \qquad \forall i,j,k \in V, \qquad k \neq j, j \neq i \tag{3.72}$$

$$\sum_{l=1,l\neq j,l\neq k}^{n} w_{ijlk} \leq x_{kk} + u_k \qquad \forall i,j,k \in V, \qquad k \neq i, j \neq i \tag{3.73}$$

$$x_{ij},\ u_i,\ t_i,\ y_{ij},\ w_{ijkl},\ s_{ijkl} \in \{0,1\} \qquad \forall i,j,k,l \in V \tag{3.74}$$

In the objective function, the location and allocation cost and the overall transportation cost are minimized.

Location and allocation are composed by constraints:

Constraints (3.55), where the number of hubs is equal to p. Two different types of hubs are allowed by the model, hubs with allocations x_{jj} and standalone hubs (hub with no spokes) u_j.

Constraints (3.56), where a vertex i will be allocated to a vertex j if j is a hub with allocation.

Constraints (3.57), where a vertex i will be chosen as sub-hub if it is allocated to at least two hubs.

Constraints (3.58), where every vertex may be chosen as a hub with allocation, or a hub without allocation, or a spoke. A vertex can just be chosen as a sub-hub if it is not a hub.

Constraints (3.59), where every vertex i will be allocated to one hub j, or to two hubs j in the case of i to be a sub-hub. But, it can also be a hub with or without allocation.

Constraints (3.60), where two sub-hubs cannot be allocated to the same pair of hubs.

In order to represent the service design, arcs linking the nodes are set by the following constraints:

Constraints (3.61), where an arc y_{ij} linking one sub-hub i with another spoke node (sub-hub or not) is allowed, but only if they are allocated in a common hub.

Constraints (3.62), where the existence of an arc linking the vertex i and j, if the vertex i and j are allocated to different hubs, will be not allowed. There will just exist an arc linking the vertex i and j if they are allocated to the same hub. Then, arc y_{ij} will also not link hubs. It will either link spokes (sub-hub or not) or one hub with one spoke.

In constraints (3.62), the link between two sub-hubs is allowed, but a common hub is needed because of constraints (3.61).

Constraints (3.63), where two arcs going out of a vertex i to two others vertex that belong to a same hub j will be not allowed.

Constraints (3.64), where two arcs arriving in a vertex i from two others vertex that belong to a same hub j will be not allowed.

Constraints (3.65), where in the network, the arc-non-hub (i, j) or the arc-non-hub (j, i) is used. However, if the cluster just has one spoke (sub-hub or not), the arcs-non-hub (i, j) and (j, i) can be used.

Constraints (3.66), where one arc-non-hub leaves every vertex i, but if i is a sub-hub, two arcs can leave. However, if the vertex i is a hub without allocation, arc-non-hub does not exist.

Constraints (3.67), where one arc-non-hub will be received by every vertex i, but if i is a sub-hub, two arcs will be received. However, if the vertex i is a hub without allocation, arc-non-hub cannot be received.

Cargo routing constraints, in the built network, is done by:

Constraints (3.68), where the flow of every vertex i towards to every vertex j will leave the source using the arc (i, l), where the flow can be a flow w (flow between two hubs) or a flow s (flow in a arc-non-hub). A hub with a spoke (sub-hub or not) or two spokes (sub-hubs or not) are linked by arc-non-hub.

Mathematically speaking, arc hubs are not needed in the model for the routing of demands. That is because direct flows between each pair of hubs are allowed and triangle inequality is satisfied.

Constraints (3.69), where the flow of every vertex i towards to every vertex j will be received in the destination using the arc (l, j), where the flow can be a flow w or a flow s.

Constraints (3.70), where a flow that leaves an intermediate vertex k (which is not the source or the destination), will be equal to the flow that has arrived at the intermediate vertex.

Constraints (3.71), where flow s will exist if arc y exists.

Constraints (3.72) and (3.73), flow w will exist if the pair of hubs k and l exist, where hub can be with or without allocation nodes.

Constraints (3.74) are integrality constraints.

At regional route, constraints to eliminate subtours are added in the model to clusters with more than four nodes.

Then, for each supposedly viable solution found, C will be the set of selected clusters with allocation, where $C = \{C_1, C_2, ..., C_m\}$, C_q will be the set of selected nodes which belong to the same cluster q, h_q will be the hub node from set C_q (cluster q) and S will be any subset of nodes from a set C_q. Constraints are:

$$\sum_{i,j \in S} y_{ij} \leq \mid S \mid -1 + \sum_{i \in S} x_{ii}, \qquad 2 \leq \mid S \mid \leq \mid C_q \mid -2, \; and \; h_q \in (C_q - S)$$
$$\forall \; C_q, \quad where \; q \in \{1, 2, ..., \sum_{j=1}^{n} x_{jj}\} \tag{3.75}$$

Where constraints (3.75) are generated and included in the model, whenever necessary, to eliminate subtours at regional routes.

A cutting plane approach focused on the subtour elimination in cluster level was also implemented to solve this third model.

3.3.4 Model with direct hub connections and without sub-hub

In this last network model, a complete graph is presented at hub level, circular routes are made at cluster level and a single allocation of spoke nodes is used (Figure 3.9).

Similar to the second model, transport cost is only minimized by an economy of scale and transshipments are only accomplished in hub nodes (da Costa Fontes and Goncalves, 2015b).

This model is classified as Many-to-Many p-Hub Location-Routing Problem.

This mathematical model was developed from the previous model. Some constraints referring to sub-hubs were removed and others were modified because the variable t_i is not presented in this model.

The constraints (3.57), (3.60), (3.61), (3.63) and (3.64) were removed.

Below, the whole model is presented. But, just the replaced constraints are described.

Let V be the set of nodes with n nodes, c_{ij} is the cost per unit of flow from node i to node j where i, $j \in V$, p is the number of hubs, fl_{ij} is the flow from node i to node j and α is the discount factor on the unit of flow between hubs.

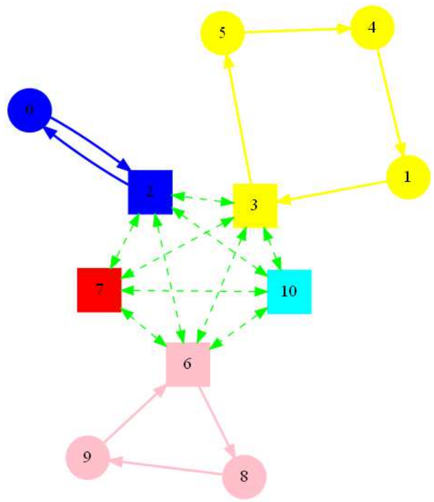


Figure 3.9: Network with direct hub connections and without sub-hub.

Hub location and spoke allocation are defined by variables of model: $u_i = 1$ if vertex i is a hub with no spoke allocated to it, or 0 otherwise and $x_{ij} = 1$, if vertex i is allocated to hub j, or 0 otherwise. A vertex i is a hub with spoke locations if $x_{ii} = 1$.

The variable $y_{ij} = 1$, if there exists arc-non-hub $(i; j)$, or 0 otherwise, is responsible by the service design in cluster level.

And for the cargo routing, it was created the variables $w_{ijkl} = 1$, if there exists a flow from i to j routed between the hubs k and l, or 0 otherwise and $s_{ijkl} = 1$, if there exists a flow from i to j routed via arc-non-hub $(k; l)$, or 0 otherwise.

The model can be defined as follows:

Formulation of the Many-to-Many p-Hub Location-Routing Problem (MMpHLRP)

$$min \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} fl_{ij}\ c_{ij}\ x_{ij} + \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \sum_{k=1, k\neq j}^{n} \sum_{l=1, l\neq k, l\neq i}^{n} fl_{ij}\ c_{kl}\ s_{ijkl} +$$
$$+ \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \sum_{k=1, k\neq j}^{n} \sum_{l=1, l\neq k, l\neq i}^{n} fl_{ij}\ \alpha\ c_{kl}\ w_{ijkl}$$

$s.t.:$

$$\sum_{j=1}^{n} x_{jj} + \sum_{j=1}^{n} u_j = \ p \tag{3.76}$$

$$x_{ij} \leq x_{jj}, \qquad \forall i, j \in V \tag{3.77}$$

$$u_i + x_{ii} \leq 1, \qquad \forall i \in V \tag{3.78}$$

$$\sum_{j=1}^{n} x_{ij} = \ 1 - u_i, \qquad \forall i \in V \tag{3.79}$$

$$y_{ij} + x_{ik} + x_{jl} \leq 2, \qquad \forall i, j, k, l \in V,\ i \neq j, k \neq j, i \neq l, l \neq k \tag{3.80}$$

$$y_{ij} + y_{ji} \leq 1 + x_{ii} + x_{jj}, \qquad \forall i, j \in V,\ i \neq j \tag{3.81}$$

$$\sum_{j=1, j\neq i}^{n} y_{ij} = 1 - u_i, \qquad \forall i \in V \tag{3.82}$$

$$\sum_{j=1, j\neq i}^{n} y_{ji} = 1 - u_i, \qquad \forall i \in V \tag{3.83}$$

$$\sum_{l=1, l\neq i}^{n} w_{ijkl} + \sum_{l=1, l\neq i}^{n} s_{ijkl} = 1, \qquad \forall i, j \in V,\ k = i, i \neq j \tag{3.84}$$

$$\sum_{l=1, l\neq j}^{n} w_{ijlk} + \sum_{l=1, l\neq j}^{n} s_{ijlk} = 1, \qquad \forall i, j \in V,\ j = k, j \neq i \tag{3.85}$$

$$\sum_{l=1, l\neq i, l\neq k}^{n} w_{ijkl} + \sum_{l=1, l\neq i, l\neq k}^{n} s_{ijkl} = \sum_{l=1, l\neq j, l\neq k}^{n} w_{ijlk} + \sum_{l=1, l\neq j, l\neq k}^{n} s_{ijlk}$$
$$\forall i, j, k \in V,\ k \neq i, k \neq j, j \neq i \tag{3.86}$$

$$s_{ijkl} \leq y_{kl} \qquad \forall i, j, k, l \in V, \qquad l \neq k, l \neq i, k \neq j, j \neq i \tag{3.87}$$

$$\sum_{l=1,l\neq i,l\neq k}^{n} w_{ijkl} \leq x_{kk} + u_k \qquad \forall i,j,k \in V, \qquad k \neq j, j \neq i \tag{3.88}$$

$$\sum_{l=1,l\neq j,l\neq k}^{n} w_{ijlk} \leq x_{kk} + u_k \qquad \forall i,j,k \in V, \qquad k \neq i, j \neq i \tag{3.89}$$

$$x_{ij},\ u_i,\ y_{ij},\ w_{ijkl},\ s_{ijkl} \in \{0,1\} \qquad \forall i,j,k,l \in V \tag{3.90}$$

The constraints (3.58) are replaced by the constraints (3.78): Every vertex may be a hub with allocation, or a hub without allocation, or a spoke.

Constraints (3.59) are replaced by the constraints (3.79): Every vertex i will be allocated to one hub j. But it can also be a hub with or without allocation.

Constraints (3.62) are replaced by the constraints (3.80): The existence of an arc linking the vertex i and j if vertex i and j are allocated to different hubs will not be allowed. There will only exist an arc linking vertex i and j if they are allocated to the same hub. Then, arc y_{ij} will also not link hubs. It will either link spokes or one hub with one spoke;

Constraints (3.66) are replaced by the constraints (3.82): one arc-non-hub leaves every vertex i. However, if the vertex i is a hub without allocation, arc-non-hub does not exist.

Constraints (3.67) are replaced by the constraints (3.83): one arc-non-hub will be received by every vertex i. However, if the vertex i is a hub without allocation, arc-non-hub cannot be received.

Constraints (3.74) are replaced by the constraints (3.90): they are integrality constraints.

A cutting plane approach focused on the subtour elimination in cluster level was also implemented to solve this last model.

This fourth model was developed because a network with the presented characteristics along with routing mechanism in a same model was not found in the literature about hub network.

In Rodríguez-Martín et al. (2014), hub location-routing problem was defined as a combination of hub location and multi-depot vehicle routing problems, presenting characteristics of a difficult problem.

Applications for the four network models presented in this chapter can be seen in different cargo delivery situations. More precisely, liner shipping network problem and postal delivery problem can be represented by developed networks.

3.4 Conclusions

In this chapter, four new mathematical models about hub-and-spoke network problems were presented.

The models are classified as hub location-routing problems. Demands are transported using two different levels of network.

At a cluster level, network is composed of circular routes with directed arcs. Then, demands are routed in a single direction inside the clusters.

At hub level, two different networks were presented. The first one is composed of directed arcs in circular form and the second one is composed of undirected arcs within a complete graph.

Though hub-and-spoke network with directed arcs is not a common characteristic in the literature, more realistic applications in cargo delivery problems can be very well represented by this characteristic.

However, the main characteristic in the proposed models is the utilization of shorter alternative paths alongside an economy of scale for a great economy in the transport cost when a hub-and-spoke network is built.

Some demands can be transported with transshipment inside hub nodes or sub-hub.

Shorter alternative paths are offered because of a multiple allocation of some spoke nodes. Normally, a multiple allocation with direct connection hub-spoke is presented in the literature about hub-and-spoke. In the proposed problems with sub-hubs, nodes in cluster level are connected by a circular route. Consequently, demands from whole region can be transferred through a sub-hub (multiple allocated node).

Some characteristics from the traveling salesman problem, the p-median problem and the shortest path problem are used in the developed models.

Because of the complexity of the problems, only small instances can be solved by the described cutting plane approach (see results in the chapter 5). Then, a metaheuristic is implemented seeking to solve large-sized instances of problem.

An implemented Variable Neighborhood Decomposition Search is described in the next chapter and the results are presented in chapter 5.

Chapter 4

A VNS Metaheuristic Based Approach

In this problem, large and medium instances can not be solved by Cplex solver because the problem is composed of sub-complex problems (location, allocation, service design and routing). So, we decided to use metaheuristic to solve large-size problem instances.

First an introduction about metaheuristics is presented in this chapter. Characteristics of the Variable Neighborhood Search (VNS) and of its variant, the Variable Neighborhood Decomposition Search (VNDS), are also shown. After, an implemented constructive heuristic and variants of a proposed VNDS to solve large instances of the network service design problem with two hierarchy levels are described.

Four variants of proposed VNDS were implemented to solve, respectively, the four mathematical models described in chapter 3: Two-Level p-Hub Location-Routing Problem with Sub-Hub, Two-Level p-Hub Location-Routing Problem, Many-to-Many p-Hub Location-Routing Problem with Sub-Hub and Many-to-Many p-Hub Location-Routing Problem.

4.1 Introduction to Metaheuristics

In the literature on metaheuristics, an exact definition of the term is still not presented. According to Sörensen et al. (2015), until recently, a clear definition of the word metaheuristic has been missing, and it could be argued that it is still disputed.

But, some similar settings are shown from the definitions of some authors:

"Metaheuristic is an algorithm designed to solve approximately a wide range of hard optimization problems without having to deeply adapt to each problem" (Boussaïd et al., 2013);

"Metaheuristics, in their original definition, are solution methods that orchestrate an interaction between local improvement procedures and higher level strategies to create a process capable of escaping from local optima and performing a robust search of a solution space" (Glover and Kochenberger, 2003);

"Metaheuristics are general high-level procedures that coordinate simple heuristics and rules to find good (often optimal) approximate solutions to computationally difficult combinatorial optimization problems" (Ribeiro et al., 2007);

"Metaheuristics are approximate algorithms which basically try to combine basic heuristic methods in higher level frameworks aimed at efficiently and effectively exploring a search space" (Blum and Roli, 2003);

"A metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization algorithms. The term is also used to refer to a problem-specific implementation of a heuristic optimization algorithm according to the guidelines expressed in such a framework" (Sörensen and Glover, 2013);

"Meta-heuristic is an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search spaces using learning strategies to structure information in order to find efficiently near-optimal solutions" (Osman and Kelly, 1996).

In the presented definitions what can be highlighted is that:

- Metaheuristics are approximate algorithms. Because, normally, optimal solutions are not offered by them, but, only, good solutions at a viable computational time;

- Metaheuristics are used to solve hard optimization problems. Because large-size problem instances can not be solved by exact methods;

- Metaheuristics are generic algorithms that could be used and adapted in general problems. Different from heuristics that are created and used to a specific problem;

- Metaheuristics are frameworks. Iterations between improvement heuristics (local search) and high level strategies are guided by them.

Strategic procedures used into metaheuristics seek to maintain a balance between the intensification phase and the diversification phase of the solution search process.

In the intensification phase, a specific region from solution space is explored because a good quality with respect to find results is presented. Whereas, in the diversification phase, distinct regions of the solutions space are explored, thus avoiding a metaheuristic to be trapped in a local optimum during the intensification phase.

Based on distinct characteristics of the metaheuristics, different classifications are presented in the literature. Many classification criteria may be used for metaheuristics (Boussaïd et al., 2013).

In a classic classification observed in the literature, metaheuristics are grouped in single-solution based metaheuristics and population-based metaheuristics (Boussaïd et al., 2013), (Martí and Reinelt, 2011), (Blum and Roli, 2003).

Single-solution based metaheuristics (or trajectory methods) group is composed of metaheuristics where the whole search procedure of optimal solution is accomplished with the use of a single solution. Through the processes, one current solution is replaced by a neighbor solution offering a better result. This new solution can also be replaced by another one, and so on, until the search procedure can be concluded.

Among metaheuristics classified into single-solution based on metaheuristics set, the classic ones are Simulated Annealing (Kirkpatrick et al., 1983), Tabu Search (Glover, 1986), Iterated Local Search (Lourenço et al., 2003), Grasp (Feo and Resende, 1989) and Variable Neighborhood Search (Mladenović and Hansen, 1997).

While a population-based metaheuristics group, a set of solutions is used during the whole search procedure. These solutions can be modified, combined, replaced by other ones and also discarded. But through the process, a set of solutions can always be manipulated in the search of optimal solution.

This set of metaheuristics is composed of classic examples such as Ant Colonies (Colorni et al., 1991), Particle Swarm Optimization (Kennedy and Eberhart, 1995) and Genetic Algorithm (Holland, 1992).

In another metaheuristic classification, they are divided in natural processes and non-natural processes (man-made processes) (Sörensen and Glover, 2013), (Talbi, 2009), (Blum and Roli, 2003).

Metaheuristics from natural processes group have shown a good characteristic of diversification. Among these metaheuristics there are, for example: analogy with the thermodynamic process is used by Simulated Annealing and analogy with the genetic process is done by the Genetic Algorithm.

On the other hand, intensification procedure has been more characteristic to non natural processes. Examples of metaheuristics are Tabu Search, where a memory to forbid or to select certain movements into solution space is used, and Variable Neighborhood Search (VNS), in which different neighborhood structures during the process of solution search are used.

Metaheuristics can also be classified as deterministic and stochastic methods. In deterministic metaheuristics, an optimization problem is solved by deterministic decisions (e.g. classic tabu search), so same results are provided by each run. Whereas, in stochastic metaheuristics, some random decisions are applied during the search process (e.g. simulated annealing) (Talbi, 2009) which give different results at each run.

But, independently of classification, a set of subordinate heuristics is used in each metaheuristic (Parejo et al., 2011). Solved problem-type will define which subordinate heuristics are used. The customization (or instantiation) of one metaheuristic to a given problem yields a heuristic to the latter (Ribeiro et al., 2007).

Initial solution to start a metaheuristic can be chosen at random way or it can be built by a constructive heuristic. Constructive heuristics are algorithms where the solution is built step-by-step from scratch. Greedy algorithms (Algorithm 1), are examples of constructive heuristic.

In algorithms from Ribeiro et al. (2007), the notation is: consider a combinatorial optimization problem *minimize* $f(S)$ *subject to* $S \in X$, defined by a ground set $E = \{e_1, ..., e_n\}$, a set of feasible solutions $X \subseteq 2^E$, and an objective function $f : 2^E \rightarrow \mathbb{R}$. $S^* \in X$ is an optimal solution such that $f(S^*) \leq f(S), \forall S \in X$.

In Algorithm (1), at each iteration, an element from $E = \{e_1, ..., e_n\}$ (ground set) is selected to be integrated in the partial solution S. Algorithm stops once the solution to be fully completed.

Algorithm 1 Greedy algorithm for minimization (Ribeiro et al., 2007).

```
GreedyAlgorithm()
S ← ∅;
Evaluate the incremental cost of each element e ∈ E;
while S is not a complete feasible solution do
    Select the element s ∈ E with the smallest incremental cost;
    S ← S ∪ {s};
    Update the incremental cost of each element e ∈ E \ S;
end while
return S;
```

After an initial solution has been defined, it can be enhanced into improvement or intensification phase of the metaheuristics. Examples of improvement phase algorithms used to refine a current solution are:

- local search (Algorithm 2), where a new solution is found in a neighborhood of the current solution;

- exact method. In this case, metaheuristic is known in the literature as matheuristic (Boschetti et al., 2009);

- Variable Neighborhood Descent (VND) (Algorithm 3) (used in VNS algorithm), where two or more neighborhoods of current solution are explored to get a new solution;

- path-relinking (Glover, 1986) (Algorithm 5), where the combination of two or more solutions from a neighborhood space generates a path of points (i.e. solutions) between and beyond these selected points. These points on the path can be sources for new paths.

There are also metaheuristics in which aspects from different metaheuristics are combined. These metaheuristics are known as hybrid metaheuristics (Sörensen and Glover, 2013), (Blum and Roli, 2003).

A neighborhood of a solution S is a set $N(S) \subseteq X$. S^* is a local optimum in $N(S)$ if $f(S^*) \leq f(S), \forall S \in N(S^*)$ (Ribeiro et al., 2007).

In Algorithm 2, the neighborhood structure ($N(S)$) is explored from a solution S, searching an improvement. If a new solution S' is found, neighborhood structure is explored from S', searching again a new improvement. This algorithm is executed until a solution does not present an improvement in its neighborhood.

Algorithm 2 Local Search for minimization (Ribeiro et al., 2007).

```
LocalSearch(S_0)
S ← S_0;
while S is not locally optimal do
    Find S' ∈ N(S) with f(S') < f(S);
    S ← S';
end while
return S;
```

In Algorithm (3), consider:

- An optimization problem *minimize* $f(x)$ *subject to* $x \in X$, $X \subseteq S$;
- $f : S \to \mathbb{R}$: Objective function;
- X: set of feasible solutions;
- S: solution space;
- $N = N_1, ..., N_{\ell_{max}}$: set of neighborhoods;
- x' : initial solution;
- x : current solution;
- ℓ : list of neighborhoods. Where $1 \leq \ell \leq \ell_{max}$;
- y: solution in the neighborhood $N_\ell(x)$;

In Basic VND (Algorithm 3), after change is completed with a neighborhood structure, the process can be restarted from the first neighborhood (if the solution is improved) or can be started from the next neighborhood when solution is not improved (see Algorithm 4).

Algorithm 3 Sequential VND using the best improvement search strategy (Hansen et al., 2016).

1: B-VND (x, ℓ_{max}, N)
2: **repeat**
3:     $stop \leftarrow false$;
4:     $\ell \leftarrow 1$;
5:     $x^{'} \leftarrow x$;
6:     **repeat**
7:         $x^{''} \leftarrow argmin_{y \in N_{\ell}(x)} f(y)$; //$y$ is a solution of $N_{\ell}(x)$, where $f(y)$ has minimum value.
8:         $Neighborhood\_change\_sequential(x, x^{''}, \ell)$;
9:     **until** $\ell = \ell_{max}$;
10:     **if** $f(x^{'}) \leq f(x)$ **then**
11:         $stop \leftarrow true$;
12:     **end if**
13: **until** $stop = true$;
14: return $x^{'}$;

After the last neighborhood is explored, if the current solution is not improved the whole process is stopped.

In Sequential neighborhood change step (Algorithm 4) solution found in the neighborhood structure (line 7, Algorithm 3) is verified in order to accept (when solution is improved) or not this one as current solution. Algorithm 4 also decide which neighborhood will be explored in next iteration.

Classics neighborhood change step procedures are described in Hansen et al. (2016).

Algorithm 4 Sequential neighborhood change step (Hansen et al., 2016)

Neighborhood_change_sequential(x, x', k)
if $f(x') < f(x)$ **then**
    $x \leftarrow x'$; //a better solution is found
    $k \leftarrow 1$; //return to first neighborhood
else
    $k \leftarrow k + 1$; //try to improve the next neighborhood
end if

For Algorithm 5 consider:

- m: operation move (each solution $S' \in N(S)$ is reached from S by this operation);
- S_s: starting solution;
- S_t: target solution (S_s and S_t are two good solutions);

- $\triangle(S_s, S_t)$: set of moves needed to reach St from S_s. Symmetric distance between S_s and S_t;
- $\bar{S}$: best solution;
- m^*: best move;
- $S \oplus m$: solution resulting from applying move m to solution S.

At each step of Algorithm 5 all moves from the current solution S are examined and the one with the lower cost is selected. The procedure is finished when S_t is reached ($\Delta(S, S_t) = \emptyset$). A path of solution is generated between S_s to S_t (Ribeiro et al., 2007).

Algorithm 5 Path-relinking for minimization (Ribeiro et al., 2007).

PathRelinking(S_s, S_t)
Compute the symmetric difference $\Delta(S_s, S_t)$;
$\bar{f} \leftarrow min\{f(S_s), f(S_t)\}$;
$\bar{S} \leftarrow argmin\{f(S_s), f(S_t)\}$;
$S \leftarrow S_s$;
while $\Delta(S, S_t) \neq \emptyset$ **do**
    Let m^* such that $f(S \oplus m^*) \doteq min\{f(S \oplus m) : m \in \Delta(S, S_t)\}$;
    $\Delta(S \oplus m^*, S_t) \leftarrow \Delta(S, S_t) \setminus \{m^*\}$;
    $S \leftarrow S \oplus m^*$;
    if $f(S) \leq \bar{f}$ **then**
        $\bar{f} \leftarrow f(S)$;
        $\bar{S} \leftarrow S$;
    end if
end while
return $\bar{S}$;

Looking for a new solution in the improvement phase is made in the neighborhood of the current solution. This search can use a best improvement strategy or it can use first improvement strategy, where process is finalized after finding the first neighbor the cost of which is better than the actual solution.

Representation (encoding) of a solution used by a metaheuristic is also defined by problem-type solved. According to Talbi (2009), the representation plays a major role in the efficiency and effectiveness of any metaheuristic and constitutes an essential step in designing a metaheuristic.

In Talbi (2009) some classical representations of solutions that are used by metaheuristics to solve a large variety of optimization problems are presented (Figure 4.1).

In Figure 4.1, different problem are presented with the appropriate encoding.

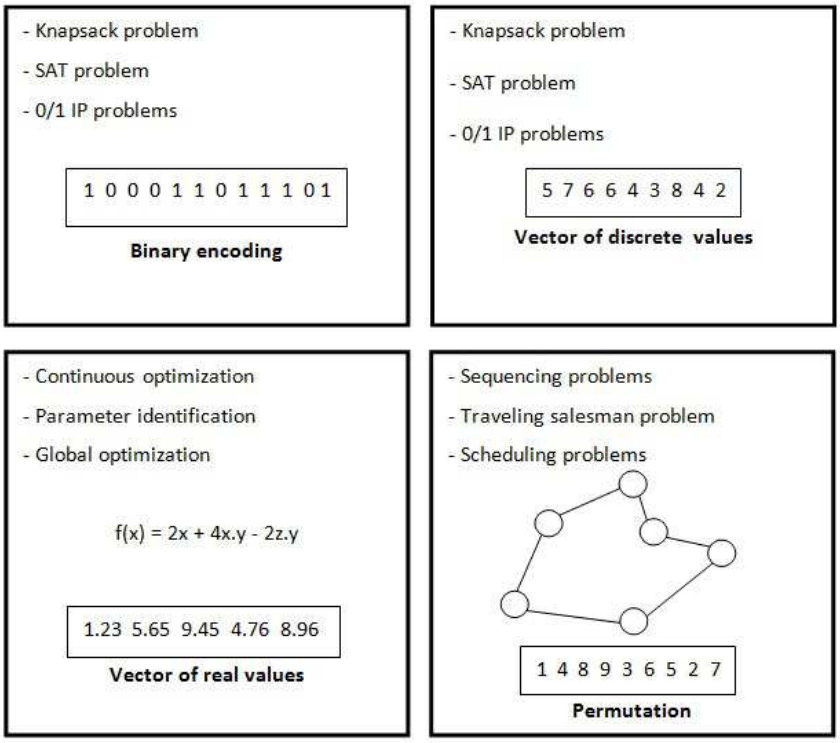


Figure 4.1: Some classic encodings (Talbi, 2009).

Mathematical models studied in this work are composed of location, allocation, service design and routing problems. This characteristic was considered as a fundamental point when we decided to choose a metaheuristic based-approach to solve large-size problem instances. A VNS variant was selected because each problem could be explored by a set of different neighborhood structures.

These problems will be called sub-problems in the following presentation because each one is part of the main problem studied.

In the next section, details of VNS are given. More precisely implemented constructive heuristic and implemented Variable Neighborhood Decomposition Search (a variant of VNS method) are presented.

4.2 Variable Neighborhood Search (VNS)

Variable neighborhood search (VNS), (Algorithm 6), is a metaheuristic whose basic idea is systematic change of neighborhood within a local search (Hansen et al., 2010).

Systematic change of neighborhood is based on the fact that a local optimum for one neighborhood structure can not be the local optimum for another neighborhood; global optimum found, will be a local optimum to all neighborhoods; and for many problems local optimum with respect to one or several neighborhoods are relatively close to each other structure (Hansen et al., 2010), (Hansen and Mladenovic, 2003).

Variable Neighborhood Search has been applied to different optimization problems. Rabelo and Jones (2013) described that VNS shows an excellent capability to solve scheduling problems to optimal or near-optimal schedules.

Algorithm 6 VNS (Boussaïd et al., 2013).

1: VNS(s, k_{max}, N);
2: **while** the stopping criterion is not satisfied **do**
3:     $k \leftarrow 1$;
4:     **while** $k < k_{max}$ **do**
5:         $s' \leftarrow$ shaking $(s, k, N(s))$; //select a random s' in the n^{th} neighborhood $N_s(s)$ of s
6:         $s'' \leftarrow$ Local Search (s');
7:         **if** s'' is better than s **then** //if objective function is improved
8:             $s \leftarrow s''$;
9:             $k \leftarrow 1$;
10:         **else**
11:             $k \leftarrow k + 1$;
12:         **end if**
13:     **end while**
14: **end while**
15: return s; //return the best solution met

In Hansen et al. (2016), ingredients to build different variants of VNS are presented. In Algorithm (6) the Basic VNS is presented, composed of:

- A current solution (s);
- initial solution of a local search procedure (s');
- number of neighborhood structures k_{max};
- A set of neighborhood structures $N_k, k = 1, ..., k_{max}$;
- Shaking procedure: in this step, a perturbation mechanism is applied to the current solution s. A random solution is selected. It is achieved by trying to explore different parts of search space;
- Improvement procedure (or intensification phase): in this step (line 6), a local search is used to exploit the neighborhood by a descent method. Note that, a VND method (Algorithm 3) can be used by some variants of VNS;
- Neighborhood change step (line 8 to line 13): in this step, a sequential neighborhood change step (Algorithm 4) is applied.

Shaking is a procedure used in the VNS to escape from local minima while exploring the solution space (see Hansen et al. (2016), Boussaïd et al. (2013)).

Variants of VNS are described in Hansen et al. (2016). Between the variants of VNS, there is Variable Neighborhood Decomposition Search (VNDS) in which improvement phase ingredient is just applied on a reduced part of the problem (Algorithm 7).

In the VNDS, the problem is decomposed. The subproblem generated represents a part from the original one with a reduced solution space.

Algorithm 7 Classic VNDS (Hansen et al., 2016).

VNDS(x, k_{max}, N)
repeat
    $k \leftarrow 1$;
    repeat
        //step(1): shaking
            $x' \leftarrow Shake(x, k, N)$;
        //step(2): getting sub-problem
            $y \leftarrow x' \setminus x$;
        //step(3): solving sub-problem
            $y' \leftarrow$ Improvement procedure(y);
        //step(4): substituting the solution
            $x'' = (x' \setminus y) \cup y'$;
        //step(5): moving between operators
            Neighborhood change sequential (x, x'', k);
    until $k = k_{max}$;
until stopping condition is fulfilled;
return x;

Classic VNDS (Algorithm 7) is the classic VNS (Algorithm 6) with two new steps included: step 2, where a part of solution (sub-problem solution) given for the global problem is selected to be improved in step 3; step 4, where improved solution of sub-problem is added to solution of the global problem.

VNDS is not yet widely used by researchers. But there are some works of literature, where VNDS is used to solve, for example, supply chain management planning problem (Ahuja et al., 1993), the minimum weighted k-cardinality tree problem (Urošević et al., 2004), large-scale continuous location allocation problems (Lejeune, 2006), 0-1 multidimensional knapsack problem (Brimberg et al., 2006) and p-median problem (Hanafi et al., 2010).

4.3 Implemented Constructive Heuristic

Before we saw that we need to begin a metaheuristic by producing one or a set of solutions (i.e. population). A single-solution based method (or trajectory method) is used in the VNS.

This single-solution may be randomly generated or with the help of a constructive heuristic. We decided to implement a constructive heuristic because parts of problem (sub-problems) can be explored by exact methods.

Providing an Initial Solution by a Constructive Heuristic

Constructive heuristic developed to produce initial solution (x) for the studied problem is composed of 4 steps (Algorithm 8). Each step is used to solve a different sub-problem. Step are described below.

Algorithm 8 Constructive Heuristic

```
Constructive_heuristic()
To solve hub location and spoke allocation problem;
To solve sub-hub location problem;
To solve service design problem;
To produce shortest path tree from each node to all others nodes;
return x; //initial solution
```

4.3.1 Step1: To solve hub location and spoke allocation problem

In this step, an integer programming formulation of the p-median problem (see chapter 3), (Mladenović et al., 2007), (Reese, 2005) is solved by Cplex solver (version 12.6.3).

The main idea is to get the desired number of clusters with one hub associated to each cluster. Objective function 4.1 (f_{p_med}) minimizes distance between spoke and its associated hub:

Let V be the set of demand nodes with n nodes, c_{ij} is the distance between the customer $i \in V$ and the facility $j \in V$. Variable $x_{ij} = 1$ if the customer i is allocated to facility j, or 0 otherwise.

$$min \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} c_{ij}\ x_{ij} \tag{4.1}$$

Objective function 4.1 is used in p-median problem presented by Mladenović et al. (2007). The objective is minimize the sum of the distances.

Nodes are divided into p (number of hubs) clusters, where each cluster is composed of one hub with or without allocated spoke nodes. In other words, one cluster could be constituted by only one single hub node, but regarding the number of spoke nodes, it can be formed with a quantity between 0 and $(n-p)$ spokes, where n is the total number of nodes in network. Note that we set number p to desired number of hubs before solving p-median problem. In this way, the problem can be solved in polynomial time (Garey and Johnson, 1979). Thus different cluster configurations can be provided depending on strategic decisions (i.e. the number of hubs to open).

4.3.2 Step2: To solve sub-hub location problem

In this step, the main idea is to identify close clusters and create a relationship between two close clusters. This relationship will permit a multiple allocation for some spoke nodes. These nodes will be called sub-hub nodes.

With previous step, each node is included into one single cluster. Concerning sub-hub node, it can be allocated to two different close clusters. To be attractive, we need to know which clusters are close together. To do that, a distance metric is defined below. Then we have to select the best candidate node to be the sub-hub node of two close clusters (see the Algorithm 9).

Distance between two clusters is defined as distance between their hubs.

In Algorithm (9), C is the set of clusters with its hubs, C_A is distance matrix (or allocation cost matrix) and f_A is the allocation cost function of our objective function (4.2):

$$min \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} fl_{ij}\; c_{ij}\; x_{ij} \tag{4.2}$$

Algorithm 9 Sub-hubs location

1: Sub_hubs_location(C, C_A, f_A)
2: Evaluate the distance metric; //(i.e. proximity level)
3: Identify close clusters;
4: choose sub-hub($C, C_A, CC\_relationship, f_A$); //Select candidates to be sub-hubs
5: return(C);

Still in Algorithm (9), *CC_relationship* is set by close clusters identified in line 3 (Algorithm 9). Sub-hub node to each relationship can be chosen by Algorithm 10.

This step 2 is constituted by:

i) - evaluate distance metric:

In this point, average distance between hubs (average d_H) and corresponding deviation are calculated.

$$d_{CC} = (average\ d_H) - (one\ deviation) \tag{4.3}$$

d_{CC} is the metric distance used to choose sub-hubs.
d_H is the distance between 2 hubs.

ii) - identify close clusters:

If distance between two hubs is less than the previous metric d_{CC}, then these clusters are declared as close ones and can share a spoke node chosen as sub-hub node.

iii) - Select candidates to be sub-hubs:

At most one sub-hub for each pair of close clusters can be chosen (see Algorithm 10).

In fact, sub-hub is a spoke node of one cluster, that will be shared by a close cluster (Figure 4.2). So, allocation cost (Function 4.2) of the second cluster is increased by this process.

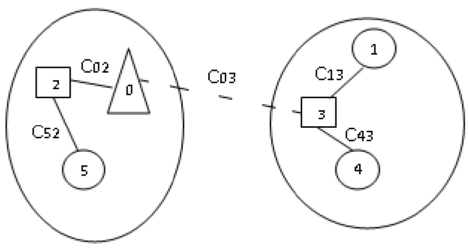


Figure 4.2: Close clusters with sub-hub and allocation cost.

In Figure 4.2, squares are hubs, small circles are spokes, triangle is sub-hub, big circles are clusters, full lines and dashed line represent distance cost (allocation cost) (C_{ij}).

Still in Figure 4.2, initial clusters are hub 2 with spoke nodes 0 and 5 and hub 3 with spoke nodes 1 and 4. Then f_{p_med} was:

$$f_{p\_med} = (C_{02} + C_{52}) + (C_{13} + C_{43}) \tag{4.1.1}$$

After node 0 has been chosen as sub-hub of these clusters, the allocation cost is:

$$f_{p\_med} = (C_{02} + C_{52}) + (C_{13} + C_{43} + C_{03}) \tag{4.1.2}$$

Algorithm 10 choose sub-hub

choose_sub-hub($C, C_A, CC\_relationship, f_A$)
for each CC_relationship[i] **do** //i is a pair of close cluster
  $minimum\_iac[i] \leftarrow big\_number$;
  for each node j from CC_relationship[i] **do**
    $iac[j] \leftarrow$ distance of node j to hub from close cluster i
    if ($iac < minimum\_iac$) **then**
      $minimum\_iac[i] \leftarrow iac[j]$;
      $CC\_relationship[i] \leftarrow j$; //j is the sub-hub
    end if
  end for
  if ($minimum\_iac[i] \neq big\_number$) **then**
    node from $CC\_relationship[i]$ is shared between close clusters;
    f_A is updated; //equation (4.2)
  end if
end for
return(f_A);

In Figure 4.2, incremental allocation cost of node 0 ($iac[0]$) (see Algorithm 10) is the cost (C_{03}).

In Algorithm 9 there is a possibility for two clusters to be identified as close clusters (line 3), but sometimes no sub-hub could be selected in line 4. This is true when clusters are only composed of isolated hub or when a candidate node already plays as a sub-hub of an other pair of close clusters (Figure 4.3)

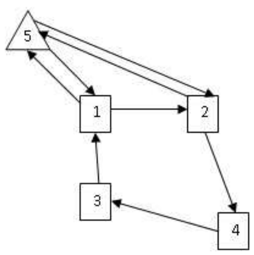


Figure 4.3: Two relationships of close clusters and only one sub-hub.

In Figure 4.3, the network is composed of four hubs (1, 2, 3 and 4) and one sub-hub (5). Each hub number is used to represent its respective cluster. The pairs of close clusters are identified as follows: 1 and 2; 1 and 3. Cluster 1 was initially composed of nodes 1 and 5. Cluster 2, 3 and 4 were initially composed of one node (their hub). Node 5 was selected sub-hub from close clusters 1 and 2. From this moment, no sub-hub can be chosen for pair of close clusters 1 and 3 because cluster 1 has just one spoke (sub-hub

5) and cluster 3 is just composed of one node (hub 3).

When one pair of close clusters (relationship between two clusters) can not receive one sub-hub in constructive heuristic, the relationship is nevertheless kept and maybe one sub-hub will be chosen in improvement heuristic of our VNDS algorithm.

4.3.3 Step3: To solve service design problem

Crainic and Kim (2007) describe service network design problem which integrates two types of major decisions. The first one is to determine the service network, that is, to select the routes, or in other words, to create a scheduling of nodes visitings. The second decision is to determine the routing of a request considering the scheduling of visits created before.

In this step, the main idea is to create a network of services, i.e. a route in each cluster and a route at hub-level for the model with circular hub route. When the model solved is composed of direct hub connection, service design is only created inside each cluster.

In this work, routing is done inside each cluster in a circular form and between clusters (hub routes) in circular form (model with circular hub route) or with direct links between hubs (model with direct hub connection). Each sequence of visited nodes represents one service provided by the network. For example, in Figure (3.2), the network is composed of 5 circular services.

In Figure 3.2, for example route 5-3-1-4-5 is a regional service where hub 5 is the origin and the final destination of route and all other nodes are intermediate stops. Each node can be both a point of delivery and collect, and hub 5 and spoke 3 are also transhipment points. Hub 5 is a transhipment point because it connects that service with hub service 5-10-7-6-2-5, and spoke node 3 is also a transhipment point because it connects two regional services (sub-hub).

Each circular route consists in solving a travelling salesman problem. The number of nodes in each cluster is not limited. Then, a regional routes can have all spoke nodes. To save time, a nearest neighbor constructive heuristic algorithm can be used (see the following Algorithm 11).

The nearest neighbor Algorithm described in Bellmore and Nemhauser (1968) - (Algorithm 12), is always used for a set of nodes (cluster or set of hubs) to create route.

Algorithm 11 is used by models with circular hub route to built hub routes and regional route. But in models with direct hub connection, it is just used to build regional routes.

The first node, in regional route (line 2 and line 5 - Algorithm 12), is the hub node. In hub route, first node is also the first hub from a list of hubs (i.e lexical order).

The results obtained with the Nearest Neighbor Algorithm, allow to build a sparse matrix of cost between each node of circular route.

Note that a discount factor α is applied to the cost between two nodes of hub route.

Algorithm 11 Creating service design

```
Service_design(S) //S is the set made by each cluster and by set of hubs
for each set_of_nodes do //cluster or set of hubs
    if (number of nodes > 3) then
        Nearest Neighbor Algorithm(set_of_nodes);
    end if
end for
return S_r; //S_r is the set of routes
```

Algorithm 12 Nearest Neighbor Algorithm - based on Bellmore and Nemhauser (1968) description

```
1: Nearest_Neighbor_Algorithm(set_of_nodes)
2: Select the first node;
3: Find the nearest unvisited node and go there;
4: Are there any unvisited node left? If yes, repeat step 2;
5: Return to the first node;
6: return r; //r is the constructed route
```

The network with economy of scale is represented by a dedicated sparse matrix.

4.3.4 Step4: To produce the shortest path trees

In this step, the main idea is to create the routing of all demands based on service network we created in previous step.

Routing cargo procedure consists in satisfying all customer requests. That is, a delivery of demands from each node to all others nodes. For a particular request, which services must be taken to satisfy the corresponding customer? Because our objective function is to minimize total distance traveled by all cargos, the shortest paths between two nodes need to be calculated. The shortest path from one source node s to all other nodes is represented by the shortest path tree (Figure 4.4).

Because the network created during the previous step (line 4 from Algorithm 8) does not have negative arcs, then the shortest path tree can be built by Dijkstra's Algorithm, described in Dijkstra (1959) and presented below (Algorithm 13).

Dijkstra Algorithm is executed n (number of nodes) times, where a different source node s is used to produce a different shortest path tree. Then, the best routing cargo is offered by a set of shortest path trees associated to the analyzed network.

where:

- source node: s;

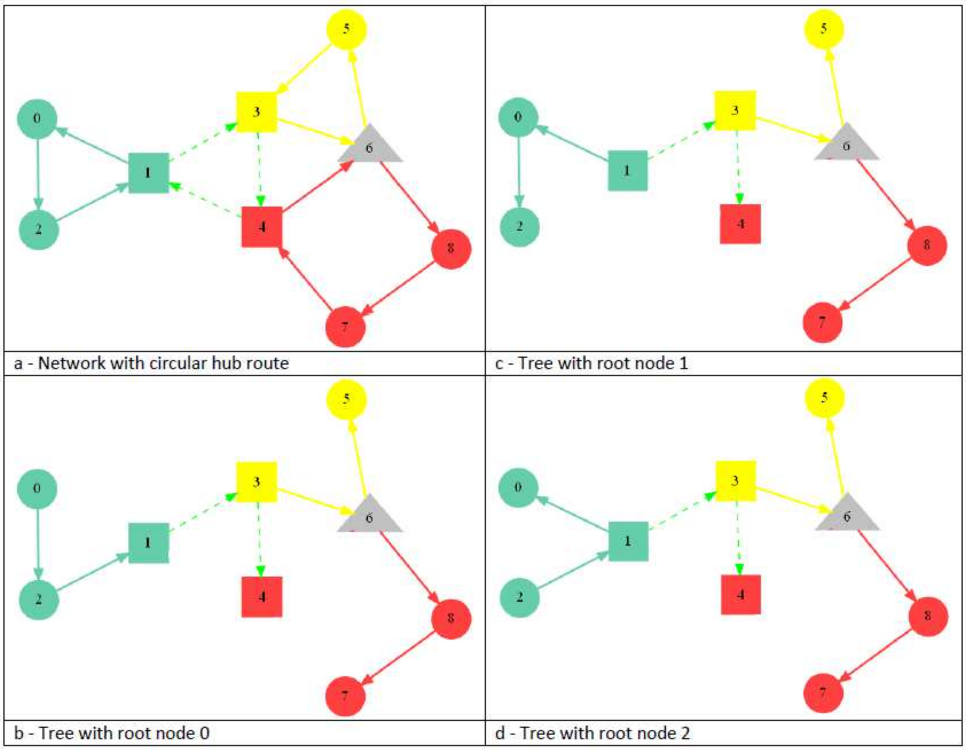


Figure 4.4: Network and possible solution of Dijkstra's Algorithm for different sources.

- sparse matrix of routes built in Algorithm (11): network;
- set of nodes temporarily labeled: $\bar{S}$;
- set of nodes permanently labeled: S (at each iteration the algorithm transfers a node i in the set $\bar{S}$ with smallest distance to the set S) (Ahuja et al., 1993);
- set of arcs going out from a node $i : A(i)$;
- predecessor index to maintain the tree: $pred[s]$;
- distance label from source s to node $i : d[i]$;
- cost of a arc from node i to node $j : cij$.

After Dijkstra's Algorithm to be solved for a source i, the associated transport cost is calculated (Algorithm 14):

Where $d[j]$ is distance from source node i to any node j calculated by Dijkstra's Algorithm.

After each shortest path tree, $total\_cost$ (Algorithm 14) is calculated and the global transport cost is updated:

In Algorithm (15), network is represented by a sparse matrix of routes built in Algorithm (11).

Algorithm 13 Dijkstra Algorithm - based on Ahuja et al. (1993).

Dijkstra(s, network)
    $S := \emptyset$;
    $\bar{S} := N$;
    $d[i] := \infty$ for each node $i \in N$; //initialize the distance vector
    $d[s] := 0$; //set the source node
    $pred(s) := 0$;
while $\mid S \mid < n$ **do**
    let $i \in \bar{S}$ be a node for which $d[i] = min\{d[j] : j \in \bar{S}\}$;
    $S := S \cup \{i\}$;
    $\bar{S} := \bar{S} - \{i\}$;
    for each $(i, j) \in A(i)$ **do**
        if $d[j] > d[i] + c_{ij}$ **then** //a new shortest path is found
            $d[j] := d[i] + c_{ij}$ //distance from s to j = distance s to i+ cost arc (i, j)
            $pred[j] := i$;
        end if
    end for
end while
return $d[]$;

Algorithm 14 Transport cost for node "i" requests

Transport_cost(i)
for each node $(j \in n)$ **do**
    $total\_cost(i) = fl[i][j] * d[j]$;
end for
return $total\_cost(i)$;

Algorithm 15 Global transport cost for all requests

Global_transport_cost($network$)
$f_T(network) \leftarrow 0$;
for each node $(i \in n)$ **do**
    $f_T(network) = f_T(network) + Transport\_cost(i)$;
end for
return $f_T(network)$;

Remember that discount factor for using the hub route, has already been integrated in step3 of constructive heuristic(to solve service design problem).

4.3.5 Evaluate Objective Function Cost

After the four steps of constructive heuristic have been solved, a viable solution of the problem is obtained. The objective function cost of the analyzed problem is constituted by allocation cost calculated in Algorithm (10) and by global transport cost calculated in Algorithm (15).

$$Objective\ function = allocation\ function(f_A) + global\ transport\ cost(f_T) \tag{4.4}$$

This feasible solution and its evaluated cost are used as initial solution by Variable Neighborhood Decomposition Search (VNDS), a variant of VNS (Variable Neighborhood Search), to improve the result.

4.4 Implemented Variable Neighborhood Decomposition Search

In this study, the problem is composed of 4 sub-problems: hub and sub-hub location or only hub location (models without sub-hub), spoke allocation, service design and network routing.

Operators of an implemented Variable Neighborhood Decomposition Search (VNDS), (Algorithm 16), were built by considering the characteristics of those sub-problems. Moving between local search operators is guided by the neighborhood change sequential procedure (Algorithm 4) (da Costa Fontes and Goncalves, 2016a).

Objective function f_O is evaluated in neighborhood change sequential procedure using equation (4.4) defined earlier.

A reduced part of the solution space of the problem is selected when a getting the sub-problem (step(1) in the Algorithm 16).

A different improvement algorithm is done for each sub-problem by using specific operators.

In models with sub-hub, two Variable Neighborhood Descent (VND), one Local Search (LS) and shake ($N_{k_{special}}$ – line 21) are used. But in version for models without sub-hub, one shake ($N_{k_{special}}$), two LS and only one VND are used.

For hub and sub-hub locations sub-problem:

- Version for model with sub-hub, 1 operator ($improve\_hub\_and\_improve\_sub\_hub$), responsible for searching a local optimum for each hub and sub-hub location, was

Algorithm 16 Implemented VNDS.

VNDS(x, k_{max}, N)
$test\_shake \leftarrow 0$;
$x'' \leftarrow x$;
repeat
    $k \leftarrow 1$;
    while $k \leq k_{max}$ **do**
        //step(1): getting sub-problem(x, N_k)
            select the operator N_k with exactly p solution attributes from solution x;
            denote this subset by y;
        //step(2): solving sub-problem
            $y' \leftarrow$ Improvement procedure(y); //y' is the improved solution
        //step(3): substituting the solution (it creates original problem solution)
            $x' = (x \setminus y) \cup y'$;
        //step(4): moving between operators
            Neighborhood change sequential (x, x', k);
    end while;
    if $((f_O(x) < f_O(x''))$ **then** //f_O is the objective function
        $x'' \leftarrow x$;
    end if
    if (number of $test\_shake$ is not met) **then** //parameters values are presented in the chapter 5
        $x \leftarrow$ Shake($x'', N_{k_{special}}$);
        $test\_shake \leftarrow test\_shake + 1$;
    end if
until number of $test\_shake$ is met;
return x'';

developed. In this operator, improvement phase algorithm used is a VND composed of three neighborhood structures (*improve_hub*, *improve_sub_hub* and shake *remove_sub_hub* ($N_{\ell_{special}}$ in the Algorithm 17));

- Version for model without sub-hub, the operator is the *improve_hub*. In this operator, the neighborhood structure is explored by a LS.

Concerning sub-problem of spoke node allocations one operator and one shake procedure were defined:

- *spoke_nodes_allocations* operator: responsible for getting the local optimum for spoke allocation. In this operator, a local search is used;

- *shift_spoke_node* shake-operator: the function of shaking current solution is attributed to this procedure. The neighborhood structure ($N_{k_{special}}$) in Algorithm (16), allows the allocation of a random spoke node to a different cluster.

One operator (*improve_two_routes_and_improve_one_route*) was developed for service design sub-problem. In this operator, the improvement phase algorithm used is a VND composed of two neighborhood structures:

- two_routes_improve;
- one_route_improve.

About routing sub-problem Dijkstra's Algorithm is always used.

Difference between classic VNDS (Algorithm 7) and our implemented version (Algorithm 16) is that shake procedure is executed after all neighborhood structures have been explored. A specific number of shake procedures is realized.

After each shake procedure, algorithm is restarted in the first step with the first neighborhood.

Details of each operator, for each version of problem studied, are described below.

4.4.1 Routing Sub-problem - Dijkstra's Algorithm

Dijkstra's Algorithm (Algorithm 13) is used at the routing sub-problem because it offers the smallest path to follow in network in order to satisfy demands from each node to all other nodes.

Objective function used in all problem variants consists in minimizing two parts: allocation cost part and transport cost part (see equations(4.4)). In the implemented VNDS (Algorithm 16), transport cost is always evaluated after the shortest path problem has been solved for all nodes (routing sub-problem) using Dijkstra's Algorithm (Algorithm 13).

Similar to constructive heuristic, after execution of Dijkstra's Algorithm for each source node, the total cost (Algorithm 14) and global transport cost (Algorithm 15) are calculated.

Although objective function is composed of transport cost plus allocation cost, most of this value is given by transport cost. Then, an exact method is used to assure good solution at routing and, consequently, good objective function value.

4.4.2 Operators for tLpHLRPSH model

In the proposed model, we supposed that it is allowed to have between 1 route (composed of only standalone hub nodes) and $p+1$ routes (p is the number of hubs).

Sequence of operators used for this model is:

- a) *Improve_hub_and_improve_sub_hub*;
- b) *improve_two_routes_and_improve_one_route*;
- c) *spoke_nodes_allocations*.

The best results were presented with the sequence of operators above.

shift_spoke_node shake-operator is used after all improvement operators have been executed (Algorithm 16).

a) *improve_hub_and_improve_sub_hub* operator

About the *improve_hub_and_improve_sub_hub* operator, the step 2 of Algorithm (16), several neighborhood structures are explored through of a Cyclic Variable Neighborhood Descent (Cyclic VND) procedure (Algorithm 17).

In the Algorithm 17, in each neighborhood operator is explored a different problem ℓ. Then, N_ℓ is the neighborhood of the problem ℓ, $N_{\ell_{special}}$ is the neighborhood used in the shake *remove_sub_hub* and f_ℓ is the cost function of the problem ℓ.

Algorithm 17 Cyclic VND used for location sub-problem.

Cyclic VND(y, ℓ_{max}, N)
quantity_of_shake $\leftarrow 0$;
$y' \leftarrow y$;
repeat
    $\ell \leftarrow 1$;
    while $\ell \leq \ell_{max}$ **do**
        $y'' \leftarrow argmin_{z \in N_\ell(y)} f_\ell(z)$ //z is a solution of $N_\ell(y)$, where $f_\ell(z)$ has minimum value.
        Neighborhood_change_cyclic(y, y'', ℓ, f_ℓ);
    end while
    if $((f_1(y) = f_1(y'))||(f_2(y) = f_2(y')))$ **then** //no improvement in hub route cost (see equation 4.5) or not modification in allocation cost (see equation 4.6)
        quantity_of_shake $\leftarrow$ *quantity_of_shake* $+ 1$;
        if (*quantity_of_shake* is not met) **then**
            $y \leftarrow$ shake $(y', N_{\ell_{special}})$
        end if
    else
        $y' \leftarrow y$;
    end if
until *quantity_of_shake* is met;
return y';

Cyclic VND is composed of 3 neighborhood structures: *improve_hub*, *improve_sub_hub* and shake *remove_sub_hub*($N_{\ell_{special}}$).

Procedure *Neighborhood_change_cyclic* used in Cyclic VND (Algorithm 17) is presented in Algorithm (18).

Algorithm 18 Cyclic Neighborhood Change step for Cyclic VND (Hansen et. al., 2016)

Neighborhood_change_cyclic(y, y'', ℓ, f_ℓ)
if $f_\ell(y'') < f_\ell(y)$ **then**
    $y \leftarrow y''$; //a better solution is found
end if
$\ell \leftarrow \ell + 1$; //try to improve the next neighborhood

A Sequence of neighborhood structures trying to improve the solution is:

- a.1) *improve_hub*;
- a.2) *improve_sub_hub*;
- a.3) shake *remove_sub_hub*.

After finishing first *improve_hub* neighborhood structure, search is started with second neighborhood structure (*improve_sub_hub*) whether there is improvement or no improvement of current solution (Algorithm 18).

After *improve_hub* and *improve_sub_hub* neighborhood structures have been executed, without improvement, shake *remove_sub_hub* is executed (Algorithm 17, line 13).

a.1) *improve_hub* neighborhood structure

For each cluster with spoke node allocations, we try to find a better hub node. Switches between each spoke node (except for sub-hub nodes because they are allocated to two hubs) with its original hub are tested. The new hub route is built again by the Nearest Neighbor Algorithm (Algorithm 12) in order to evaluate the sub-problem cost (i.e. cost of the new hub route - equation (4.5)):

Let H be the set of hub nodes with p nodes, c_{ij} is the distance between the hub $i \in H$ and the hub $j \in H$. Variable $b_{ij} = 1$ if the arc between the hubs i and j is included in the hub route, or 0 otherwise.

$$f_{hr} = \sum_{i \in H} \sum_{j \in H, j \neq i} c_{ij}\ b_{ij} \tag{4.5}$$

If an improvement is found, the switch/move is made and then we do it again with the next cluster.

A best improvement search strategy is used for each cluster.

Note that the relationships between close clusters built in the constructive heuristic are not modified by this neighborhood structure. The number of relationships and clusters included in each one remains the same.

a.2) *improve_sub_hub* neighborhood structure

In $Improve\_sub\_hub$ structure, a new sub-hub is checked for each pair of close clusters. The distance cost (allocation cost) between each candidate sub-hub spoke node and the two corresponding hub nodes is re-evaluated (equation 4.6), trying to find a new sub-hub with lower distance cost than actual sub-hub:

Let C_C be a set composed of two close clusters (CC_relashionship[i]), H_c is a set composed of two hubs from C_C ($H_c \subseteq C_C$, c_{ij} is the distance between the spoke node $i \in C_C$ and the hub $j \in H_C$. Variable $x_{ij} = 1$ if the spoke i is allocated to hub j, or 0 otherwise.

$$f_2 = \sum_{i \in C_c} \sum_{j \in H_c, j \neq i} c_{ij} \; x_{ij} \tag{4.6}$$

Again, relationships between close clusters are not modified by this neighborhood structure, only hubs are updated. This process is an exhaustive method in which the sub-hub making the smallest allocation cost is selected.

Best improvement search strategy is again used in this neighborhood structure.

When one pair of close clusters does not currently have a common sub-hub, and there is at least one candidate, a sub-hub can be selected in this neighborhood structure.

The allocation cost is updated after including sub-hub in this neighborhood structure. With that cost, sub-problem cost can be evaluated.

a.3) shake *remove_sub_hub* neighborhood structure

In Cyclic VND (Algorithm 17), a shake is implemented. With this shake operator, one close cluster relationship is randomly selected, and the associated sub-hub node is then removed. Then this node will be reallocated to its original cluster (see Figure (C.7) in Appendix C.

Remember that a number of relationships between close clusters were created in the constructive heuristic (Algorithm 9, line 3). This number is fixed and the clusters included in each relationship (close clusters) are also fixed.

If a sub-hub node is selected in the constructive heuristic it can be modified in the improvement phase (*improve_sub_hub* neighborhood structure), but the $CC\_relationship$ will always have a sub-hub.

Therefore, the fixed number of sub-hubs can be relaxed with a shake in which sub-hubs can be removed from close clusters relationship. After this shake, the relationship between two close clusters still exists, but without any sub-hub node selected.

After removing a sub-hub, the corresponding node can not be selected again during the next execution of *improve_sub_hub* neighborhood structure in order to avoid inefficient cycles.

Also, if a new sub-hub is selected in the *improve_sub_hub* neighborhood structure, this sub-hub can not be removed in the next shake procedure. This procedure prevents that a *CC_relationshipi* without sub-hub receive a sub-hub in *improve_sub_hub* neighborhood structure and after, the same sub-hub is removed in next shake *remove_sub _hub*.

This shake inside a VND algorithm, was developed trying to make a diversification search in a specific part of the problem, the location of sub-hub nodes.

b) *improve_two_routes_and_improve_one_route* operator

A Decomposition Variable Neighborhood Descent (Decomposition VND) procedure (Algorithm 19), composed of 2 neighborhood structures (*two_routes_improve* and *one_ route_improve*), is developed for the service design sub-problem.

This VND was designated as Decomposition VND, because the search strategy (Algorithm 19, line 7) is just used in a reduced part (one or two routes in the network) of the sub-problem (see Algorithm 21).

In the Algorithm 19, y represents the set of the routes, r represents the selected routes, Nr represents the neighborhood of the selected routes r and f_R is the route cost. f_T is the transport cost (see Algorithm 15).

In this VND, a neighborhood structure will be explored while improvements are found in it.

But in each neighborhood structure, clusters routes are selected randomly. Then, the non improvement of a neighborhood is considered after several unsuccessful improvement tests.

Therefore, in change neighborhood procedure, neighborhood changing is only made after a fixed number of iterations without any improvement (Algorithm 20).

b.1) *two_routes_improve* neighborhood structure

Cost of 2 routes, randomly chosen, can be modified by first neighborhood structure (*two_route_improve*) of that sub-problem (Algorithm 21). This neighborhood structure was implemented because the routing cost of demands is minimized by synchronization of services provided by each cluster at hub level.

Algorithm 19 Decomposition VND used to service design sub-problem.

Decomposition VND(y, ℓ_{max}, N)
repeat
  $\ell \leftarrow 1$;
  $y' \leftarrow y$;
  $quantity\_test \leftarrow 0$;
  while $\ell \leq \ell_{max}$ **do**
    $y'' \leftarrow (y \setminus r) \cup argmin_{z \in N_r, r \subset y} f_R(z)$ $//y''$ is composed of (part of y) + z. z is a solution of N_r, where r is a sub-set of y and $f_R(z)$ has minimum value. f_R is the route cost (see equations (4.5) and (4.7)).
    Neighborhood_Change($y, y'', \ell, quantity\_test$);
  end while
until $(f_T(y') = f_T(y))$; //no improvement in transport cost
return y';

Algorithm 20 Neighborhood Change step for Decomposition VND

1: Neighborhood_change($y, y'', \ell, quantity\_test$)
2: **if** $f_T(y'') < f_T(y)$ **then** //transport cost
3:   $y \leftarrow y''$; //a better solution is found
4:   $quantity\_test \leftarrow 0$;
5: **else**
6:   $quantity\_test \leftarrow quantity\_test + 1$;
7:   **if** $(quantity\_test = fixed\_number)$ **then**
8:     $\ell \leftarrow \ell + 1$; //try to improve the next neighborhood
9:     $quantity\_test = 0$;
10:   **end if**
11: **end if**

In *two_routes_improve*, (Algorithm 21), 2 random and different routes are selected (line 2) and for each one, 2-opt Algorithm (Croes, 1958) is applied (line 4).

2-Opt algorithm consists in replacing two non-adjacent edges of a tour by two others that create an other more interesting tour (Figure 4.5).

In Figure 4.5, arcs a and b were replaced by arcs c and d.

2-Opt algorithm is described in Algorithm 22 and Figure 4.6.

This algorithm 2-opt is implemented to select the neighbor solution of route with the smallest cost (line 7 - line 9), independently if this cost is better than the current solution. The idea is to modify the existing route (Algorithm 21, Line 4). This characteristic permits a diversification of solution, because a route can not be good in cluster, but it can help minimize the cost of sub-problem solution (Algorithm 20, Line 2).

Algorithm 21 Two-routes improve neighborhood structure

$two\_routes\_improve(y)$
$route \leftarrow$ Select two random routes(y)
for each select route i **do**
    $test\_route[i] \leftarrow 2\_Opt\_algorithm(route[i])$; //Modify each one selected route
end for
$y'' \leftarrow (y \setminus route) \cup test\_route[1] \cup test\_route[2]$;
return y'';

Algorithm 22 2-Opt Algorithm

2_opt_algorithm$(route)$
$f_R(test\_route) \leftarrow big\_number$; //it allows to select the first route
for $i \leftarrow$ second node until the next-to-last node of the current route **do**
    for $j \leftarrow i + 1$ until the last node of the current route **do**
        $actual\_route \leftarrow$ nodes from current route before the position $i+$ reverse sequence of nodes from current route with initial cutoff point i and end cutoff point $j+$ sequence of nodes from current route after the position j until the last node from current route; //see Figure (4.6)
        calculate $f_R(actual\_route)$; //route cost (see equations (4.5) and (4.7))
        if $(f_R(actual\_route) < f_R(test\_route))$ **then** //the best generated route is selected
            $teste\_route \leftarrow actual\_route$;
            $f_R(test\_route) \leftarrow f_R(actual\_route)$;
        end if
    end for
end for
return $test\_route$;

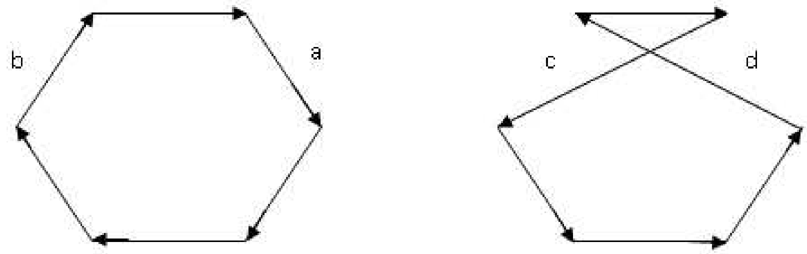


Figure 4.5: a 2-opt swap.

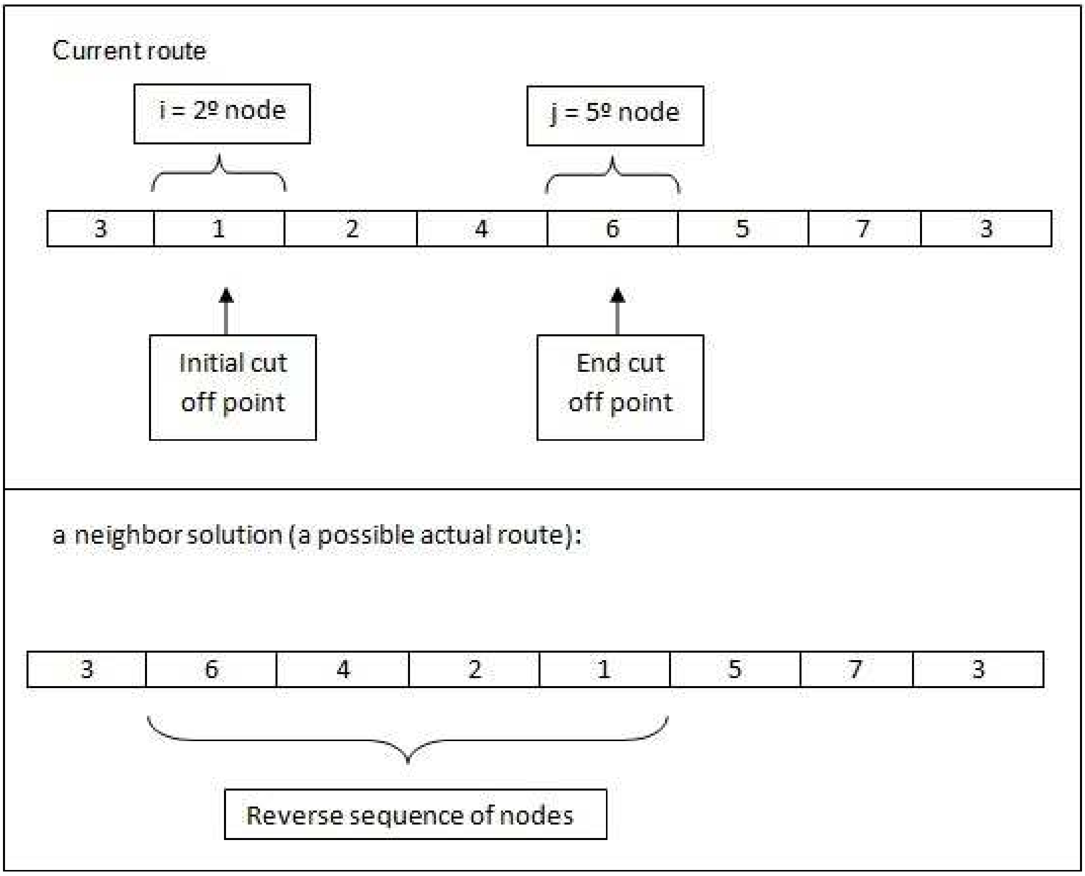


Figure 4.6: Example of a route built in 2-opt algorithm.

In 2-opt algorithm, the evaluated cost is the cost of modified route (route cost - f_R). This cost can be represented by the equation (4.5) if a hub route is selected or by the equation (4.7) if a regional route is selected (cluster p (c_p)).

In the equation (4.7), c_{ij} represents the distance between the nodes $i \in c_p$ and $j \in c_p$. Variable $x_{ij} = 1$ if the arc linking the nodes i and j is used in the route, or 0 otherwise.

$$f_R = \sum_{i \in C_p} \sum_{j \in C_p, j \neq i} c_{ij}\ y_{ij} \tag{4.7}$$

Within this neighborhood structure, after modifying two routes, that is, after modifying part of the solution, a new solution is built.

Part of the sub-problem solution is selected to be improved (Algorithm 21, line 3 - line 4), and after that it is added in the global solution of the sub-problem (Algorithm 21, line 6).

Cost function evaluated by sub-problem is global transport cost (see equation 4.4) of network (Algorithm 20, line 2).

b.2) *one_route_improve* neighborhood structure

Cost of only 1 route, randomly chosen, is modified by a second neighborhood structure (*one_route_improve*). Using this neighborhood structure allows to escape possible local optimum achieved by previous *two_routes_improve* neighborhood structure (Figure 4.7).

one_route_improve neighborhood structure was created with the same steps as before (Algorithm 21). The only difference is that just one route is randomly selected and modified.

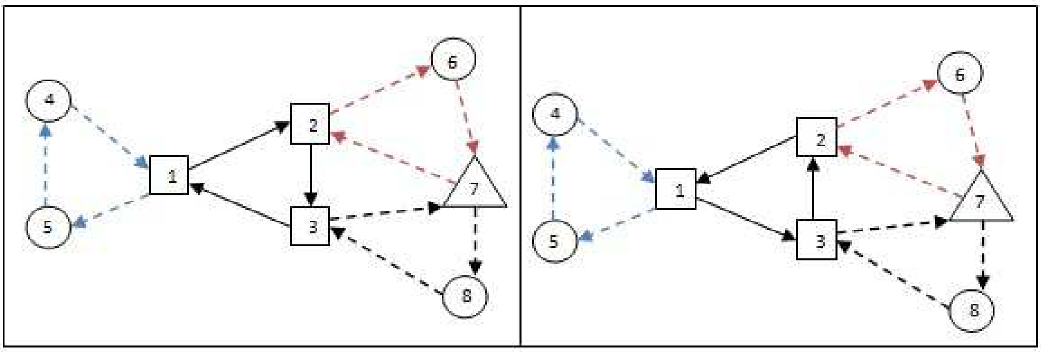


Figure 4.7: a - local optimum b - global optimum.

Note that for problems where all nodes are hubs, they can not be improved by *two_routes_improve* neighborhood structure, but they can be improved by *one_route_improve* neighborhood structure.

An important point to highlight is that *one_route_improve* applied twice is different of *two_routes_improve* applied once. Because in the second one, the solution part of sub-problem is inserted in the global solution of sub-problem after modifying two routes, while for the first one, the solution part of sub-problem is inserted each time a route is modified.

c) *spoke_nodes_allocations* operator

Remember that the allocation of spoke nodes has been achieved by an exact method (p-median problem) using Cplex solver during the constructive heuristic phase. But, for the problem studied, allocation will be dependent not only on the distance between hub and spoke, but also of the flow demand. Then, a *spoke_nodes_allocations* operator has been developed to improve initial allocation.

In this operator, a Local Search is used to explore the neighborhood (Algorithm 23). The main idea is to remove a random spoke node from its original cluster and to reinsert it in an other one, improving global transport cost (equation 4.4). The operation of insertion is repeated for each cluster and the best improvement will be accepted only if the new solution is better than the current solution.

In the Algorithm 23) p is the quantity of hubs (or clusters).

Algorithm 23 Local Search used to allocation sub-problem

1: LS_Allocation_Node(y, N)
2: $number\_iterations \leftarrow 0$;
3: $y' \leftarrow y$;
4: **repeat**
5:     $test \leftarrow 0$;
6:     Select a random spoke node i, and after, identify the cluster of this random node;
7:     **while** $(test < p)$ **do**
8:         **if** $(cluster[test] \neq$ origin cluster of node $i)$ **then**
9:             $y'' \leftarrow$ insertion$(N(y), i, cluster[test])$; //the solution is made after the node i to be inserted in the $cluster[test]$. In each test, a different solution from the same neighborhood of y is generated.
10:             **if** $(f_T(y'') < f_T(y'))$ **then** //f_T is transport cost
11:                 $y' \leftarrow y$";
12:             **end if**
13:         **end if**
14:         $test \leftarrow test + 1$;
15:     **end while**
16:     $number\_iterations \leftarrow number\_iterations + 1$;
17: **until** $((f_T(y') < f_T(y)) || number\_iterations$ is met)
18: return y';

In *spoke_nodes_allocations* operator, we use global transport cost instead of allocation cost (see equation 4.4) to evaluate solution of allocation sub-problem because the sub-hub was developed to improve the travelled distance by goods through the whole network.

Spoke_nodes_allocations operator is finished after the first improvement has been accepted or after a specific number of attempts without any improvement (Algorithm 23, line 17).

d) *shift_spoke_node* **shake-operator**

The shake procedure is used as classic VNS with the goal to diversify the search, getting away from a local optimum, and maybe, allowing to find global optimum.

The shake procedure used in this work (Algorithm 24) has the same characteristic of *spoke_nodes_allocations* operator, except that, shake procedure is finished after the random spoke node is included in the best cluster, i.e., in the cluster with the smallest global transport cost, even though this transport cost is not better than the smallest global transport cost of the current solution. Remember that all clusters except the original one ($p-1$ clusters) are tested in the reinsertion process (line 6, Algorithm 24).

Algorithm 24 Shake used in allocation sub-problem

```
1:  Shake Allocation(y, N)
2:  number_iterations ← 0;
3:  y' ← y;
4:  test ← 0;
5:  Select a random spoke node i, and after, identify the cluster of this random node;
6:  while (test < p) do
7:      if (cluster[test] ≠ origin cluster of node i) then
8:          y'' ← insertion(N(y), i, cluster[test]);
9:          if (number_iterations = 0) then
10:             y' ← y'';
11:             number_iterations ← 1;
12:         else
13:             if (f_T(y'') < f_T(y')) then
14:                 y' ← y";
15:             end if
16:         end if
17:     end if
18:     test ← test + 1;
19: end while
20: return y';
```

After *spoke_node_allocations* operator and *shift_spoke_node* shake-operator are executed, close clusters without sub-hubs (clusters previously composed of isolated hubs or clusters without any free spoke nodes to be sub-hubs) can receive spoke nodes and these nodes can be now chosen as sub-hubs of these close clusters.

One spoke node modified of cluster can also be chosen as a hub in its corresponding cluster. Then, these operators allow to diversify the search.

4.4.3 Operators for tLpHLRP model

in this subsection, in order to compare our proposed model with sub-hubs with the classic model (with no sub-hub), we are presenting the changes made in the implemented VNDS. This implementation has the same characteristics as the previous one, except that no sub-hubs are presented by the model. First, in constructive heuristic (Algorithm 8) step 2 is removed. Secondly, in location sub-problem, during the improvement phase composed by Algorithm (17), it is only necessary to improve the hub location.

Sequence of operators used for this model is the following one:

- e) *improve_hub*;
- f) *improve_two_routes_and_improve_one_route*;
- g) *spoke_nodes_allocations*.

Also here, the *shift_spoke_node* shake-operator is used after all operators have been executed (Algorithm 16).

Below, the *improve_hub* operator is described for this model. All other operators used in this model are similar to operators described for model tLpHLRPSH.

e) *improve_hub* operator

Improve_hub_and_improve_sub_hub operator now becomes only the *improve_hub* operator.

Consequently, for location sub-problem, in step 2 of Algorithm (16), just one neighborhood structure is used (the *improve_hub* structure). Several neighborhood structures are usually explored by a VND procedure, but when only one neighborhood structure is explored, the improvement phase algorithm corresponds to a basic local search heuristic (see Algorithm 25).

Algorithm 25 Local Search used to location sub-problem

```
LS_Location(y, N)
repeat
    y' ← y;
    y'' ← argmin_{z∈N(y)} f(z) //f is hub route cost.
    if (f_hr(y'') < f_hr(y)) then //improvement in hub route cost.
        y ← y''; //go along that direction
    end if
until (f_hr(y) = f_hr(y')); //no improvement in hub route cost.
return y';
```

The search strategy used in local search is the best improvement. Each spoke node for each cluster is verified if there is a new hub minimizing the hub route cost f_{hr} (equation 4.5). After all clusters have been analyzed by the algorithm, if a new solution, better than the last one, is found, the process restarts.

The search process of a new hub made by this operator is similar to the one made by *improve_ hub* neighborhood structure used in Algorithm 17. Then, for each switch hub-spoke, a new hub route is built by Nearest Neighbor Algorithm (Algorithm 12) allowing to evaluate the sub-problem cost (cost of hub route).

4.4.4 Operators for MMpHLRPSH model

This implemented VNDS presents the same operators described for model tLpHLRPSH.

In this problem, a solution can be represented by zero route when all n nodes are hubs (n number of desired hubs), because the hub nodes are connected by direct links and only regional routes are formed by circular routes.

So there is no need to build hub routes during the constructive heuristic.

The sequence of operators used for this model is:

- h) *improve_hub_and_improve_sub_hub*;
- i) *improve_two_routes_and_improve_one_route*;
- j) *spoke_nodes_allocations*.

shift_spoke_node shake-operator is again used after all operators have been executed (Algorithm 16).

Below, only the characteristics of *improve_hub_and_improve_sub_hub* operator and of *improve_two_routes_and_improve_one_route* operator are presented. All other operators are similar to the ones used for model tLpHLRPSH described before.

h) *improve_hub_and_improve_sub_hub* operator

In this operator, a hub route is artificially made to make the evaluation of improve_hub neighborhood structure possible. In fact, the route between hubs is composed of direct links.

Initial hub route is built by the Nearest Neighbor Algorithm (Algorithm 12) and then improved by 2-opt Algorithm (Algorithm 22).

The sequence of neighborhood structures explored in this Cyclic VND is:

- h.1) *improve_hub*;
- h.2) *improve_sub_hub*.

- h.3) shake *remove_sub_hub*.

After *improve_hub* and *improve_sub_hub* neighborhood structures are executed, if no improvement is noticed, the shake *remove_sub_hub* is then executed.

h.1) ***improve_hub*** **neighborhood structure**

In this neighborhood structure, a random cluster with allocation is selected and a new hub is searched between available spoke nodes (remember that sub-hub can not be a candidate). Switches are made between each spoke nodes with its previous (original) hub and a new artificial hub route is built again by the Nearest Neighbor Algorithm (Algorithm 12) for each switch hub-spoke.

This neighborhood structure uses best improvement search strategy for each selected cluster.

This algorithm is finished, with improvement or not, after all nodes from selected cluster have been analyzed.

All other processes are similar to operators used in the implementation of model tLpHLRPSH.

i) ***improve_two_routes_and_improve_one_route*** **operator**

two_routes_improve neighborhood structure and *one_route_improve* neighborhood structure are used together when at least two regional routes exist.

Only one *route_improve* is used for the solution in which there is just 1 route.

Note that solutions composed of only hub nodes and direct arcs linking the nodes, have already defined service design. Therefore, this operator is no longer used.

All other processes are similar to the ones used for the implemented version for circular hub routes (see Algorithm 19, Algorithm 20, Algorithm 21 and Algorithm 22).

4.4.5 Operators for MMpHLRP model

A variant of the previous VNDS is adapted for model MMpHLRP.

In this model, step 2 (to solve the sub-hub location problem) of the constructive heuristic (Algorithm 8) is not executed and the neighborhood structure *improve_hub* is just executed by a local search because sub-hub nodes are not used in this version model.

Circular hub route is not built in the constructive heuristic because direct links between hubs are existing in this model.

Sequence of operators used for this model is:

- k) *improve_hub*;
- l) *improve_two_routes_and_improve_one_route*;

- m) *spoke_nodes_allocations*.

shift_spoke_node shake-operator is always used after all operators be executed (Algorithm 16).

Below, only the characteristics of *improve_hub* operator and of *improve_two_routes_and_improve_one_route* operator are presented. All other operators are similar to the ones described before.

k) *improve_hub* operator

In this operator, the neighborhood structure is simply explored by Local Search (Algorithm 25).

The cost evaluated in the Local Search (Algorithm 25) is the hub route cost (f_{hr}) (see equation 4.5). Similarly to the version used in model MMpHLRPSH, an initial hub route is artificially built by the Nearest Neighbor Algorithm (Algorithm 12) and improved by 2-opt Algorithm (Algorithm 22).

All other processes executed in this variant are similar to the ones used for model MMpHLRPSH. Namely, a random cluster with allocation is selected and a new hub node is searched between the spoke nodes. The process is finished after all spoke nodes have been tested.

l) *improve_two_routes_and_improve_one_route* operator

Because direct links exist between hubs, i.e., the hub network service design is already defined. Then, *two_routes_improve* and *one_route_improve* neighborhood structure can only be executed if there are regional routes.

In this implemented version of VNDS, the same characteristics as in version for model MMpHLRPSH can be seen.

4.5 Data Structure to Represent Solution

In this work, a solution representing the clusters in the network is recorded in a dynamic two dimensions array with p (number of hubs) lines. Each line (i.e. cluster) is composed by node numbers included in the corresponding cluster (Figure 4.8).

Nodes 2, 3, 8 and 9 are hub nodes. Nodes 0, 7 and 4 are sub-hub nodes because they belong to two different clusters.

More details about the representation of the solution used in this work can be seen in the Appendix C.

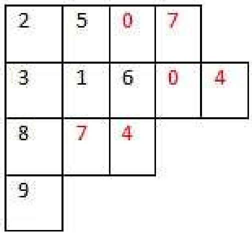


Figure 4.8: Array representing the clusters of a network with 10 nodes, 4 clusters and 3 sub-hubs.

4.6 Conclusion

In this chapter an introduction about Metaheuristic is presented in which some definitions, characteristics and classifications are described.

Large size instances of proposed models can not be solved by Cplex solver. Therefore, we chose to use a VNDS (variant of VNS) to solve these instances. This chapter is also concentrated on VNS structure and the description of implemented versions of VNDS. Constructive heuristics and various improvement operators are presented.

In VNDS, the different parts of the problem are explored by each operator because in each neighborhood structure a dedicated sub-problem is improved (location, allocation, service design and routing).

Then, different improvement algorithms are used for each operator.

Two VND (Algorithm (17) and Algorithm (19)) were implemented, each one with a new characteristic. In the first one, used for location sub-problem, a shake procedure was created. Whereas in the second one, used for service design sub-problem, a decomposition of the sub-problem was made to improve a part of the solution.

The proposed problem variants are divided in circular hub route and direct hub connections. Inside each one, problems are, again, divided in model with sub-hubs and model without sub-hubs.

The implemented versions of VNDS depend on the proposed variants of problems. Then, the main differences between implemented versions are:

1 - between model with sub-hub and model without sub-hub:

In the first one, a VND is used to explore the neighborhood structures *improve_hub*, *improve_sub_hub* and shake *remove_sub_hub*. While in the second one, a Local Search is used to explore the neighborhood structure *improve_hub*. Because in the second one (without sub-hub) the location sub-problem is just proposed for hubs.

2 - between circular hub route and direct hub connections:

In the direct hub connections problems, the results of the objective function between model with sub-hubs and model without sub-hubs are close (small gap) (see chapter 5).

About *improve_hub* operator, we can see a difference between the circular hub route model and the direct hub links model. *improve_hub* neighborhood structure, in circular version, each cluster with allocation is tested trying to improve the hub location. While, in the direct hub links version, only one random cluster with allocation is selected and tested. In this second version, a slow descent movement towards the local optima is achieved.

Lastly, note that this implemented metaheuristic can also be classified as a matheuristic because exact methods as the Dijkstra's Algorithm are used inside some procedures of the proposed algorithm.

Chapter 5

Experimental results

In this chapter, computational tests and statistical analysis of the experiments we performed are presented.

Initially, results for circular model with sub-hub and for circular model without sub-hub for small and large instances are presented and then compared. These results were achieved by the execution of an exact method and by the VNDS implemented.

After that, results for the direct hub links model with sub-hub and for direct hub links model without sub-hub for the same small and large instances are showed and compared. These results were also achieved by the performing of an exact method and the VNDS implemented.

Finally, the main points are highlighted in the conclusions.

5.1 Introduction

Two groups of instances were used to assess our VNDS metaheuristic based approach and to compare the network models with and without sub-hub.

In the first group, 36 small instances were generated (see appendix A) and solved by the exact method. These instances were also solved by our implemented VNDS algorithms.

In the second group, 45 large instances ranging between 100 and 200 nodes were solved by our VNDS. These instances are the AP (Australian Post) instances and they are available on the web site `http://www.optsicom.es/uraphmp/`. In this web site, each AP instance is composed of a number of nodes, a demand matrix (n x n) where flows from one node to itself are allowed and a distance matrix (n x n). In our work, a distance matrix and a demand matrix with no flow from one node to itself are used. Therefore, AP instances were adjusted for our problem.

The exact method was implemented with Cplex Concert Technology C++, where the constraints for subtour elimination are inserted by lazy constraint callback. VNDS algorithms were implemented in C++. The 12.6.3 version Cplex was used to solve instances.

The results obtained with the small instances are used to compare the gain in objective function from the model with sub-hub against the model without sub-hub. However, the quality of results obtained by VNDS is also checked because these results can be compared with the ones given by exact method when small instances are considered.

The present tests were executed with two different machine configurations:

- machine configuration 1: a cluster made of 5 nodes, each having the following configuration: 2 processors Intel R Xeon R E5-2630 v3 with 8 cores per processor. Cores with Base frequency 2.4 GHz and Max frequency 3.2 GHz; memory: 48GB, 2133MT/s, shared between the cores of a node. Each instance was run on one core, with limited memory (2.8GB).

- machine configuration 2: processor Intel Core i7 with 3.0 GHz and 8 GB of memory.

Small instances were solved by exact method using machine configuration 1 and solved by VNDS using machine configuration 2. Although machine configuration 1 is superior to machine configuration 2, the execution time of VNDS for small instances was considered inexpressive (less than 10 seconds).

Large instances were solved using machine configuration 1.

Each instance (small and large) was executed 30 times by VNDS in order to smooth the stochastic behavior of these approaches. So, Objective Function (Ob.Func.) and Execution Time (E.T.) of VNDS execution reported are the averages of results.

Parameters used in the algorithms were fixed in: $test\_shake = 20$ (algorithm 16), $quantity\_of\_shake = 1$ (algorithm 17), $quantity\_test = 30$ (algorithm 19) and $number\_iterations = 30$ (algorithm 22).

Time limit used in exact method was set to 172800 seconds (48 hours).

Name of each instance is coded with number of nodes (n), number of hubs (p) and discount factor using hub route (α).

In the next section, results for circular models are presented. After that, the results for direct hub links models are reported in a second section.

5.2 Results for models with circular hub route

In this section, results for the model tLpHLRPSH (with sub-hub) are compared with results for the model tLpHLRP (without sub-hub).

5.2.1 Small Instances

The results for the first group of instances are presented in Tables (5.1) and (5.2).

In Tables (5.1) and (5.2), Ob.Func. is objective function value, E.T. is the execution time in seconds, N.S-H. is the number of sub-hubs to open.

Table 5.1: Results with Exact Method and VNDS for circular model with small instances (9 and 10 nodes)

Instance	Exact Method with sub-hub			Exact Method without sub-hub		VNDS with sub-hub				VNDS without sub-hub			
n_p_α	**Ob.Func.**	**E.T.**	**N.S-H.**	**Obj.Func.**	**E.T.**	**Obj.Func.**	**E.T.**	**P.D.(%)**	**Std.Dev.**	**Ob.Func.**	**E.T.**	**P.D.(%)**	**Std.Dev.**
9_3_0,3	**463198**	52787	1	**478477**	773	**463198**	7	0	0	**478477**	7	0	0
9_3_0,6	**567586**	101904	1	**609196**	1213	**567586**	7	0	0	**609196**	7	0	0
9_3_0,9	669564 gap 9,3	172800	1	**739915**	5949	**663436**	7	-	0	**739915**	7	0	0
9_4_0,3	**388768**	15978	1	**395103**	725	**388768**	7	0	0	**395103**	7	0	0
9_4_0,6	**516839**	82409	3	**560864**	3090	532602	7	3	0	**560864**	7	0	0
9_4_0,9	618141 gap 4,71	172800	5	**726624**	19744	675416	7	9	192	**726624**	7	0	0
9_5_0,3	**328122**	4804	3	**339502**	384	337895	7,7	3	0	347636	7,21	2,4	0
9_5_0,6	**474321**	10402	3	**541639**	8446	516214	7,3	8,8	977,5	556471	7,23	2,73	0
9_5_0,9	596556 gap 2,9	172800	4	**742869**	90300,08	692463	7,32	-	0	765302	7,05	3,02	0
10_3_0,3	703189 gap 5,60	172800	0	**699028**	1393	*699028*	8	-	0	**699028**	9	0	0
10_3_0,6	906439 gap 8,2	172800	2	**960086**	3567	960086	9	-	0	**960086**	9	0	0
10_3_0,9	1147930 gap 16,01	172800	1	**1221140**	21408	1160520	9	-	0	**1221140**	9	0	0
10_4_0,3	**582824**	163628	1	**597835**	1113	**582824**	9	0	0	**597835**	9	0	0
10_4_0,6	**796933**	90474	2	**923419**	18918	867354	9	9	0	**923419**	7	0	0
10_4_0,9	959884 gap 1,6	172800	2	1246300 gap 7	172800	1146970	9	-	0	1248190	7	0,15	1258
10_5_0,3	**509372**	70308	3	**568169**	14696	561723	8,56	10,27	0	579341	8,3	1,97	0
10_5_0,6	**746842**	93087	3	**917916**	108034	872906	8,58	16,88	0	**917916**	7,05	0	0
10_5_0,9	936404 gap 1,55	172800	3	1256490 gap 12,14	172800	1172910	8,57	-	0	*1256490*	7,21	-	0

Table 5.2: Results with Exact Method and VNDS for circular model with small instances (11 and 12 nodes)

Instance	Exact Method with sub-hub			Exact Method without sub-hub		VNDS with sub-hub				VNDS without sub-hub			
n_p_α	**Ob.Func.**	**E.T.**	**N.S-H.**	**Ob.Func.**	**E.T.**	**Ob.Func.**	**E.T.**	**P.D.(%)**	**Std.Dev.**	**Ob.Func.**	**E.T.**	**P.D.(%)**	**Std.Dev.**
11_3_0,3	1055150 gap 32,04	172800	2	**967850**	160676	*967850*	9	-	0	**967850**	8	0	0
11_3_0,6	1256240 gap 26,80	172800	2	**1223470**	139804	**1196550**	7	-	0	**1223470**	7	0	0
11_3_0,9	1401520 gap 23,04	172800	3	1488330 gap 10,57	172800	1411430	7	-	0	*1479100*	7	-	0
11_4_0,3	726546 gap 9,09	172800	1	**756180**	23980	**716266**	7	-	0	**756180**	7	0	0
11_4_0,6	1001360 gap 12,72	172800	2	**1100710**	169275	**999164**	8	-	0	**1100710**	7	0	0
11_4_0,9	1208980 gap 12,99	172800	6	1468050 gap 15,85	172800	1274730	9	-	0	*1445240*	7	-	0
11_5_0,3	630079 gap 6,01	172800	2	**646408**	11824	658970	7,38	-	32302	655692	7,18	1,44	24077,15
11_5_0,6	925571 gap 9,7	172800	4	1085730 gap 10,75	172800	1086241	7,27	-	52067	*1023940*	7,19	-	0
11_5_0,9	1138700 gap 9,34	172800	4	1525580 gap 23,12	172800	1377810	7,36	-	77101	*1401470*	7,33	-	0
12_3_0,3	1385100 gap 46,6	172800	2	1083010 gap 19,27	172800	*1101019*	8	-	30861	*1055146*	8	-	3325
12_3_0,6	1365020 gap 30,13	172800	2	1286540 gap 13,65	172800	*1208600*	7	-	0	*1273012*	8	-	3319
12_3_0,9	1411580 gap 22,11	172800	3	1512480 gap 12,34	172800	*1384500*	7	-	0	*1492316*	7	-	2148
12_4_0,3	924162 gap 27,61	172800	2	841066 gap 5,78	172800	*807955*	7	-	0	841680	7	-	5434
12_4_0,6	1085480 gap 17,41	172800	5	1117240 gap 6,12	172800	*1051320*	7	-	0	1123081	7	-	6794
12_4_0,9	1320820 gap 18,82	172800	6	1448830 gap 15,25	172800	*1281150*	7	-	0	*1405021*	7	-	8210
12_5_0,3	730227 gap 14,42	172800	4	765177 gap 9,72	172800	*719672*	7,42	-	0	*752057*	7,84	-	0
12_5_0,6	981661 gap 11,76	172800	3	1140280 gap 17,26	172800	1007760	8,49	-	0	*1081340*	8,57	-	0
12_5_0,9	1216930 gap 13,29	172800	5	1419020 gap 17,95	172800	1289027	7,44	-	34488	*1410610*	8,57	-	0

Table 5.3: Average of E.T. and P.D. for results of tables (5.1) and (5.2)

-	Exact Method with sub-hub	Exact Method without sub-hub	VNDS with sub-hub		VNDS without sub-hub	
-	**E.T.**	**E.T.**	**E.T.**	**P.D.(%)**	**E.T.**	**P.D.(%)**
average	143849	94369	7,7	1,66	7,5	0,65

Note that for VNDS results, Ob.Func. and Std.Dev. are respectively the average and the standard deviation of the objective function over 30 runs. P.D. is the percentage deviation from optimal solution (see equation (5.1)). E.T. is the average value of execution time in seconds.

$$P.D. = 100 \times \frac{Ob.Func._{\ VNDS} - Ob.Func._{\ exact\ method}}{Ob.Func._{\ exact\ method}} \tag{5.1}$$

(see Martí and Reinelt (2011)).

In tables (5.1) and (5.2), results in bold are optimal solutions and results in italic represent an improved gap by VNDS.

i) Exact method

1. Version without sub-hub

For the model without sub-hub, optimal results were found by the exact method in 21 instances (see tables (5.1) and (5.2)).

Among the instances for which the optimal value has not been found, it is possible to find those with 12 nodes or those with a high α value (0,9) (with some exceptions).

2. Version with sub-hub

For the model with sub-hub, optimal results were found by the exact method in 10 instances (see Tables (5.1) and (5.2)). Model with sub-hub showed more complexity to be solved.

In this version, optimal value was not found in instances with 11 and 12 nodes. It was also not found in all instances with α value equals to 0,9.

In the Appendix B, a report with the whole result of the instance 9_3_03 is presented.

3. Comparison between the two exact versions

Solutions space in the model with sub-hubs is larger than in the model without sub-hub. Consequently, execution time is higher.

Though the number of optimal solutions found by the model with sub-hub is less than the number of optimal solutions found by the model without sub-hub, the gain (see equation 5.2) obtained by the model with sub-hub is showed in table (5.4).

Table 5.4: Observed gain for tLpHLRPSH over tLpHLRP (exact method for small instances)

Instance	Gain in Ob.Func. (%)	Instance	Gain in Ob.Func. (%)	Instance	Gain in Ob.Func. (%)
9_3_0,3	**3,2**	10_4_0,3	**2,5**	11_5_0,3	*2,5*
9_3_0,6	**6,8**	10_4_0,6	**13,7**	11_5_0,6	14,7
9_3_0,9	*9,5*	10_4_0,9	22,9	11_5_0,9	25,3
9_4_0,3	**1,6**	10_5_0,3	**10,3**	12_3_0,3	-27,8
9_4_0,6	**7,8**	10_5_0,6	**18,6**	12_3_0,6	-6,1
9_4_0,9	*14,9*	10_5_0,9	25,5	12_3_0,9	6,6
9_5_0,3	**3,3**	11_3_0,3	*-9,01*	12_4_0,3	-9,8
9_5_0,6	**12,4**	11_3_0,6	*-2,7*	12_4_0,6	2,8
9_5_0,9	*19,7*	11_3_0,9	5,8	12_4_0,9	8,8
10_3_0,3	*-0,6*	11_4_0,3	*3,9*	12_5_0,3	4,5
10_3_0,6	*5,6*	11_4_0,6	*9,02*	12_5_0,6	13,9
10_3_0,9	*6*	11_4_0,9	17,6	12_5_0,9	14,24

$$gain = 100 \times \frac{Ob.Func._{\ without\ sub-hub} - Ob.Func._{\ with\ sub-hub}}{Ob.Func._{\ without\ sub-hub}} \tag{5.2}$$

In table (5.4), gain values are given in bold when both models give optimal solution on the other hand, they are given in italic when only the model without sub-hub gives the optimal solution.

Note that negative gain values are only achieved for instances when model with sub-hub did not find the optimal solution and the objective function value is lower than the one provided by the model without sub-hub.

4. Scale factor analysis (α parameter)

Table 5.5: increase of E.T.: circular_small instances, $\alpha = 0,6$ and $\alpha = 0,9$ compared with $\alpha = 0,3$

-	increase of execution time (%)			
-	α = 0,6		α = 0,9	
n_p	model with sub-hub	model without sub-hub	model with sub-hub	model without sub-hub
9_3	93	57	227	669
9_4	415	326	981	2623
9_5	116	2097	3497	23396
10_3	-	156	-	1508
10_4	-44	1599	5	15425
10_5	32	6351	145	1075
11_3	-	-13	-	7
11_4	-	606	-	620
11_5	-	1361	-	1361
average	67	758	539	5187

In table (5.5), each line represents an increase (or a decrease) of execution time (see tables (5.1) and (5.2)) when alpha value is increased.

In the first line of table (5.5), execution time of instance 9_3_0,6 for the model with sub-hub increased 93% compared to instance 9_3_0,3. Whereas instance 9_3_0,9 increased 227% compared to instance 9_3_0,3.

Note that when a model reaches execution time limit, the symbol "-" is added in the corresponding case. For example, in instances 11_5_0,6 and 11_5_0,9 the execution time limit is reached by the model without sub-hub.

Instances 10_4_0,6 (model with sub-hub) and 11_3_06 (model without sub-hub) did not present an increase of execution time. Execution time increases when α value is increased for all other instances.

Concluding, a high alpha parameter value means a smaller discount factor, so hub routes are less attractive. Searching optimal solution is more difficult and, consequently, the execution time is increased when alpha value is increased (see average values of table (5.5)).

ii) VNDS approach

1. Version without sub-hub

For model without sub-hub, optimal results were found by VNDS in 17 instances:

- In 16 instances, optimal value was achieved in 30 executions of VNDS (see tables (5.1) and (5.2));
- In 1 instance (11-5-0,3), optimal value was achieved in 26 executions (see table (5.6)).

In 4 instances (9-5-0,3; 9-5-0,6; 9-5-0,9 and 10-5-0,3), the optimal value achieved by the exact method was not found by VNDS. But, a small percentage deviation is presented: 2,4%, 2,73%, 3,02% and 1,97%, respectively.

For the 15 remaining instances where the exact method does not present the optimal, it is possible to notice:

- The gap was decreased in 12 instances: 11-3-0,9; 11-4-0,9; 11-5-0,6; 11-5-0,9; 12-3-0,3; 12-3-0,6; 12-3-0,9; 12-4-0,3; 12-4-0,9; 12-5-0,3; 12-5-0,6 and 12-5-0,9 (see tables (5.1) and (5.2)). Gap was calculated by the Cplex;
- An equal result was found in 3 instances: 10-4-0,9; 10-5-0,9 and 12-4-0,6 (see tables (5.1) and (5.2)).

Instances for which a standard deviation of objective function is observed, the best results found over 30 runs are showed in table (5.6). The number of times over 30 runs that best objective value has been got is indicated in last column (nb. of times).

Table 5.6: Best results with VNDS for instances presenting a standard deviation (tLpHLRP)

instance	best objective function value	nb. of times
10_4_0,9	1246300	9
11_5_0,3	**646408**	26
12_3_0,3	1049220	7
12_3_0,6	1267600	8
12_3_0,9	1485980	3
12_4_0,3	837008	17
12_4_0,6	1117240	17
12_4_0,9	1397470	15

In tables (5.6), result in bold is optimal solution.

Therefore, good results were presented by VNDS between the 36 instances tested because average P.D. value is 0,65% over the 36 instances.

2. Version with sub-hub

VNDS method found optimal solution for 8 instances. Of the 10 instances where the exact method found the optimum, the VNS method found only 4. (see tables (5.1) and (5.2)).

Compared to exact method, the optimal value was not found in 6 instances:

- 9-4-0,6 and 9-5-0,3 - present a percentage deviation of 3%;
- 9-5-0,6 - presents a percentage deviation of 8,8%;
- 10-4-0,6 - presents a percentage deviation of 9%;
- 10-5-0,3 - presents a percentage deviation of 10,27%;
- 10-5-0,6 - presents a percentage deviation of 16,88%.

Between the 26 instances where the optimal was not found by exact method, the gap was decreased by VNDS in 13 instances: (9-3-0,9; 10-3-0,3; 11-3-0,3; 11-3-0,6; 11-4-0,3; 11-4-0,6; 12-3-0,3; 12-3-0,6; 12-3-0,9; 12-4-0,3; 12-4-0,6; 12-4-0,9 and 12-5-0,3) (see tables (5.1) and (5.2)). Four new optimums were found by VNDS method for instances 9-3-0,9; 11-3-0,6; 11-4-0,3; 11-4-0,6.

About the 13 instances where VNDS did not improve the gap:

- in 11 instances, results for VNDS are better than those achieved by tLpHLRP (9-4-0,9; 9-5-0,9; 10-3-0,9; 10-4-0,9; 10-5-0,9; 11-3-0,9; 11-4-0,9; 11-5-06; 11-5-0,9; 12-5-0,6 and 12-5-0,9);
- in 1 instance (10-3-0,6), the result is equal to the optimal of tLpHLRP;
- in only one instance (11-5-0,3), VNDS did not present a result better or equal to model tLpHLRP. But, the best result found among 30 executions of VNDS was 648384 (see table (5.7)), near to optimal result of tLpHLRP (646408)(see table (5.2)).

For instances which a standard deviation of objective function is observed, the best results found over 30 runs are showed in table (5.7). The number of times over 30 runs that the best objective value has been achieved is indicated in last column (nb. of times).

Table 5.7: Best results with VNDS for instances presenting a standard deviation (tLpHLRPSH)

instance	best objective function	nb. of times
9_4_0,9	675381	29
9_5_0,6	516036	29
11_5_0,3	648384	27
11_5_0,6	990556	3
11_5_0,9	1284560	9
12_3_0,3	1023660	4
12_5_0,9	1282730	29

3. Comparison between the two VNDS versions

The solutions presented by VNDS for model with sub-hub are better than the results for the model without sub-hub (average value of gain is 5,15% over the 36 instances) (see gain in table (5.8)).

The greatest number of optimal values was found by the model without sub-hub, but values found by the model with sub-hub are more promising.

Gains of objective function (see equation (5.2)) obtained by VNDS for the model with sub-hub against the VNDS for model with no sub-hub are given in table (5.8).

The model without sub-hub achieved a better result than the model with sub-hub for just one instance (11_5_0,3).

Table 5.8: Observed gain for tLpHLRPSH over tLpHLRP (small instances with VNDS)

Instance	Gain in Ob.Func. (%)	Instance	Gain in Ob.Func. (%)	Instance	Gain in Ob.Func. (%)
9_3_0,3	3,2	10_4_0,3	2,5	11_5_0,3	-0,3
9_3_0,6	6,8	10_4_0,6	6,1	11_5_0,6	3,3
9_3_0,9	10,3	10_4_0,9	8	11_5_0,9	8,3
9_4_0,3	1,6	10_5_0,3	3	12_3_0,3	2,4
9_4_0,6	5	10_5_0,6	4,9	12_3_0,6	4,6
9_4_0,9	7	10_5_0,9	6,6	12_3_0,9	6,8
9_5_0,3	2,8	11_3_0,3	0	12_4_0,3	3,4
9_5_0,6	7,2	11_3_0,6	2,2	12_4_0,6	5,9
9_5_0,9	9,5	11_3_0,9	4,6	12_4_0,9	8,3
10_3_0,3	0	11_4_0,3	5,3	12_5_0,3	4,3
10_3_0,6	0	11_4_0,6	9,2	12_5_0,6	6,8
10_3_0,9	5	11_4_0,9	11,8	12_5_0,9	9

iii) Comparison between VNDS and Exact Method

Results for the exact method and for VNDS made it possible to confirm the gain in objective function by the model with sub-hub over the model without sub-hub (see table (5.9)).

Table 5.9: Observed gain between the tLpHLRPSH over the tLpHLRP (VNDS and exact method for small instances)

Instance	Gain in Ob.Func. (%)	Instance	Gain in Ob.Func. (%)
9_3_0,3	3,2	10_3_0,9	4,9
9_3_0,6	6,8	10_4_0,3	2,5
9_3_0,9	10,3	10_4_0,6	6
9_4_0,3	1,6	10_5_0,3	1,1
9_4_0,6	5	10_5_0,6	4,9
9_4_0,9	7	11_3_0,3	0
9_5_0,3	0,5	11_3_0,6	2,2
9_5_0,6	4,7	11_4_0,3	5,3
9_5_0,9	6,8	11_4_0,6	9,2
10_3_0,3	0	11_5_0,3	-0,3
10_3_0,6	0	-	-

Table (5.9) presents the 21 instances for which optimal solutions are achieved by the exact model. Different from table (5.4), in this table, it is compared VNDS for the model with sub-hub and the exact method for the model without sub-hub.

Though the number of optimal solution found by VNDS for the model with sub-hub is less than 50% of optimal solutions found by the exact method, using VNDS allowed to find optimal results and near optimal in an inexpressive execution time.

For those 6 instances where VNDS for the model with sub-hub did not find the optimal, results presented by VNDS are better than the optimal values achieved by tLpHLRP (model without sub-hub). Thus, the model with sub-hub remains attractive regarded to the model without sub-hub.

Among the 36 small instances analyzed, equal results between the two models were found in two instances and the better results were presented by tLpHLRPSH in all other instances.

5.2.2 Large Instances

i) VNDS approach

In tables (5.10), (5.11) and (5.12), results with the second group of 45 instances are presented. In these tables, Ob.Func. is the average value of the objective function over 30 runs, S.D is the corresponding standard deviation, E.T. is the average value of execution time in seconds, Min.Ob.Func. is the smallest objective function value found among 30 executions and Sub-hub.M.O. is the number of sub-hubs to open corresponding to Min.Ob.Func. value.

Table 5.10: Results with VNDS for circular model with large instances (100 nodes)

Instance	VNDS with sub-hub					VNDS without sub-hub			
n_p_α	**Ob.Func.**	**Std.Dev.**	**E.T.**	**Min.Ob.Func.**	**Sub-hub.M.O.**	**Ob.Func.**	**Std.Dev.**	**E.T.**	**Min.Ob.Func.**
100_8_0,3	176368	1978	93	171132	4	200597	5038	91	191614
100_8_0,6	203006	5621	92	191722	3	232910	8402	98	211621
100_8_0,9	224138	6227	90	212172	3	263098	11766	96	237398
100_10_0,3	161397	6072	111	146009	6	190651	6607	115	177312
100_10_0,6	185126	5920	110	169963	6	228604	10455	122	210185
100_10_0,9	202707	8566	110	180061	6	272027	15607	114	244254
100_12_0,3	141926	3728	126	132453	9	174111	6112	139	157281
100_12_0,6	165082	2867	128	158558	9	224482	11912	139	197730
100_12_0,9	176963	6200	128	157156	9	278771	19326	136	245115

Table 5.11: Results with VNDS for circular model with large instances (125 and 150 nodes)

Instance	VNDS with sub-hub					VNDS without sub-hub			
n_p_α	**Ob.Func.**	**Std.Dev.**	**E.T.**	**Min.Ob.Func.**	**Sub-hub.M.O.**	**Ob.Func.**	**Std.Dev.**	**E.T.**	**Min.Ob.Func.**
125_8_0,3	186737	4245	178	176609	5	225603	2663	161	218016
125_8_0,6	212793	4266	174	204202	6	259515	8378	176	236338
125_8_0,9	231487	6617	172	219144	6	289718	16724	182	246685
125_10_0,3	157739	787	197	154316	7	194545	6475	209	172184
125_10_0,6	180113	2553	200	169275	7	233757	8252	208	209921
125_10_0,9	199602	2214	196	193145	7	272151	12410	208	245563
125_12_0,3	149412	3245	234	138550	9	184942	5737	261	172776
125_12_0,6	173963	2072	222	168414	9	226669	10327	260	212164
125_12_0,9	189264	2406	226	184780	9	273807	16494	262	244857
150_8_0,3	213723	5994	296	198582	3	236713	6101	290	224458
150_8_0,6	238657	6041	293	226191	3	261768	10529	317	242049
150_8_0,9	260832	4574	286	254810	3	293906	12245	309	271319
150_10_0,3	179794	4786	344	170333	7	206541	552	348	189983
150_10_0,6	205424	3577	341	196591	6	242431	7254	367	224468
150_10_0,9	220512	4230	334	206005	6	279683	9887	356	262338
150_12_0,3	169874	6615	415	154266	9	196469	5024	450	187158
150_12_0,6	194192	7141	410	177608	9	238764	9813	454	223248
150_12_0,9	209010	9097	407	189895	9	282610	14843	457	255655

Table 5.12: Results with VNDS for circular model with large instances (175 and 200 nodes)

Instance	VNDS with sub-hub					VNDS without sub-hub			
n_p_α	**Ob.Func.**	**Std.Dev.**	**E.T.**	**Min.Ob.Func.**	**Sub-hub.M.O.**	**Ob.Func.**	**Std.Dev.**	**E.T.**	**Min.Ob.Func.**
175_8_0,3	221119	4544	455	208691	5	249240	3459	453	239237
175_8_0,6	242664	4902	442	229684	5	280080	7575	473	260321
175_8_0,9	256744	4305	434	247608	5	304723	13439	485	276038
175_10_0,3	191853	7164	562	175632	7	217387	4911	551	201472
175_10_0,6	214265	5719	527	196280	7	252927	8049	576	236472
175_10_0,9	225805	4870	525	214773	7	288126	12342	572	256921
175_12_0,3	168265	6893	638	153596	11	200229	6646	657	180112
175_12_0,6	190295	6256	615	171634	11	244252	8649	667	229381
175_12_0,9	204638	5436	601	191434	11	290111	12478	649	271350
200_8_0,3	250830	4475	739	242422	2	260013	3813	764	252061
200_8_0,6	277613	6713	754	255015	2	289483	5992	777	277147
200_8_0,9	300170	9927	745	276182	2	317241	8726	771	297072
200_10_0,3	215942	11492	840	195414	6	240753	4718	871	227408
200_10_0,6	242778	6350	858	229488	6	273097	7946	896	255946
200_10_0,9	260002	9663	833	241226	6	306456	8400	894	292099
200_12_0,3	184491	6662	1020	171118	11	217343	6073	1069	200960
200_12_0,6	206728	7781	995	183928	11	258625	7258	1070	239656
200_12_0,9	222387	9586	977	198371	11	299468	10805	1065	278182

1. Version without sub-hub

In table (5.13), coefficient of variation (CV) (see equation (5.3)) of objective function (Ob.Func.) presented in tables (5.10), (5.11) and (5.12) is showed. A small coefficient of variation is presented for each instance. That means VNDS behavior is quite stable.

$$CV = 100 \times \frac{Std.Dev.}{Ob.Func.} \tag{5.3}$$

In this version, the coefficient of variation has increased when alpha value is increased. Except in instance 200_12_06.

Table 5.13: Coefficient of variation for results with large instances (circular models)

-	Coefficient of variation					
-	$\alpha = 0{,}3$		$\alpha = 0{,}6$		$\alpha = 0{,}9$	
n_p	model with sub-hub	model without sub-hub	model with sub-hub	model without sub-hub	model with sub-hub	model without sub-hub
100_8	1,1	2,5	2,7	3,6	2,8	4,5
100_10	3,8	3,5	3,2	4,6	4,2	5,7
100_12	2,6	3,5	1,7	5,3	3,5	6,9
125_8	2,3	1,2	2	3,2	2,8	5,8
125_10	0,5	3,3	1,4	3,5	1,1	4,5
125_12	2,2	3,1	1,2	4,5	1,3	6
150_8	2,8	2,6	2,5	4	1,7	4,2
150_10	2,7	0,3	1,7	3	1,9	3,5
150_12	3,9	2,5	3,7	4,1	4,3	5,2
175_8	2	1,4	2	2,7	1,7	4,4
175_10	3,7	2,2	2,7	3,2	2,1	4,3
175_12	4	3,3	3,3	3,5	2,6	4,3
200_8	1,8	1,4	2,4	2	3,3	2,7
200_10	5,3	1,9	2,6	2,9	3,7	2,7
200_12	3,6	2,8	3,7	2,8	4,3	3,6

Variations in alpha value or in number of hubs do not present large variations in execution time (results are similar) (see table (5.14)).

We can notice that execution time is increasing when the size of instances is growing.

2. Version with sub-hub

The coefficient of variation of objective function value for instances with different alpha value presents similar results (small difference) (see table (5.13)).

Variations in alpha value or in number of hubs do not present large variations in execution time (results are similar) (see table (5.14)).

Table 5.14: increase of E.T.: circular_large instances, $\alpha = 0,6$ and $\alpha = 0,9$ compared with $\alpha = 0,3$

-	increase of execution time (%)			
-	α = 0,6		α = 0,9	
n_p	model with sub-hub	model without sub-hub	model with sub-hub	model without sub-hub
100_8	-1	7	-3	5
100_10	-0,9	6	-0,9	-0,8
100_12	1	0	1	-2
125_8	-2	9	-3	13
125_10	1	-0,4	-0,5	-0,4
125_12	-5	-0,3	-3	0,3
150_8	-1	9	-3	6
150_10	-0,8	5	-3	2
150_12	-1	0,8	-2	1
175_8	-2	4	-4	7
175_10	-6	4	-6	3
175_12	-3	1	-5	-1
200_8	2	1	0,8	0,9
200_10	2	2	-0,8	2
200_12	-2	0,09	-4	-0,3
average	-1,2	3,2	-2,4	2,38

In table (5.14), each line represents an increase (or decrease) of execution time (E.T.) (see tables (5.10), (5.11) and (5.12)) when alpha value is increased.

3. Comparison between the two VNDS versions

In large instances, the results for VNDS for tLpHLRPSH were better than those from VNDS for tLpHLRP. The smallest quantity of sub-hub generated in an instance was 2 (200-8-0,3, 200-8-0,6 and 200-8-0,9).

In table (5.15), the gain in objective function for large instances is presented. This gain is given for tLpHLRPSH over the tLpHLRP.

Table 5.15: Observed gain for tLpHLRPSH over tLpHLRP (large instances)

Instance	Gain in Ob.Func. (%)	Instance	Gain in Ob.Func. (%)	Instance	Gain in Ob.Func. (%)
100_8_0,3	12,1	125_12_0,3	19,2	175_10_0,3	11,7
100_8_0,6	12,8	125_12_0,6	23,3	175_10_0,6	15,3
100_8_0,9	14,8	125_12_0,9	30,9	175_10_0,9	21,6
100_10_0,3	15,3	150_8_0,3	9,7	175_12_0,3	16
100_10_0,6	19	150_8_0,6	8,8	175_12_0,6	22,1
100_10_0,9	25,5	150_8_0,9	11,2	175_12_0,9	29,4
100_12_0,3	18,5	150_10_0,3	12,9	200_8_0,3	3,5
100_12_0,6	26,4	150_10_0,6	15,2	200_8_0,6	4,1
100_12_0,9	36,5	150_10_0,9	21,1	200_8_0,9	5,4
125_8_0,3	17,2	150_12_0,3	13,6	200_10_0,3	10,3
125_8_0,6	18	150_12_0,6	18,7	200_10_0,6	11,1
125_8_0,9	20,1	150_12_0,9	26	200_10_0,9	15,2
125_10_0,3	18,9	175_8_0,3	11,3	200_12_0,3	15,1
125_10_0,6	22,9	175_8_0,6	13,3	200_12_0,6	20,1
125_10_0,9	26,6	175_8_0,9	15,7	200_12_0,9	25,7

In table (5.15), small gain in objective function is observed for instances with small α value (higher economy of scale in hub route). Despite this attractive hub route costs, sub-hubs are used for some transshipments and so reduce overall costs.

For instances with higher α value (small economy of scale in hub route), the number of solutions with possible alternative paths is greater, but also is the solution space and consequently, the execution time.

Execution times of VNDS for model with sub-hub were similar to model without sub-hub.

The results with large instances showed the same characteristics that the results with small instances: the superiority (positive gain) of a network with sub-hub over a network without sub-hub.

The objective function value decreases when the number of hubs is increased. That is because a higher number of arc hubs with economy of scale is offered when a higher number of hub nodes is presented. This characteristic can be seen in both models (tables 5.1 - 5.12).

5.3 Results for models with direct hub connections

In this section, results from computational tests for model MMpHLRPSH (with sub-hub) and model MMpHLRP (without sub-hub) are presented.

5.3.1 Small Instances

In tables (5.16) and (5.17), results from direct hub links models for the 36 small instances showed in the previous section are presented.

In these tables, Ob.Func. is the objective function value, E.T. is the execution time in seconds, N.S-H. is the number of sub-hubs to open. In the results with VNDS, Ob.Func. and Std.Dev. are respectively the average and the standard deviation of the objective function over 30 runs. P.D. is the percentage deviation from optimal solution (equation 5.1). E.T. is the average value of execution time in seconds.

Results in bold are optimal solutions. Results in italic represent an improved gain by VNDS.

Table 5.16: Results with Exact Method and VNDS for direct hub links model with small instances (9 and 10 nodes)

Instance	Exact Method with sub-hub			Exact Method without sub-hub		VNDS with sub-hub				VNDS without sub-hub			
n_p_α	**Ob.Func.**	**E.T.**	**N.S-H.**	**Ob.Func.**	**E.T.**	**Ob.Func.**	**E.T.**	**P.D.**	**Std.Dev.**	**Ob.Func.**	**E.T.**	**P.D.**	**Std.Dev.**
9_3_0,3	**436673**	38725	1	**439654**	237	442514	8	1,34	4252	**439654**	9	0	0
9_3_0,6	**524969**	26567	1	**531549**	188	532532	8	1,44	5388	**531549**	9	0	0
9_3_0,9	**611278**	40418	1	**623445**	344	627886	8	2,71	13683	**623445**	8,63	0	0
9_4_0,3	**333382**	3140	0	**333382**	104	335057	8	0,5	3870	349247	9	4,76	17250
9_4_0,6	**437421**	2016	0	**437421**	75	**437421**	8	0	0	**437421**	9	0	0
9_4_0,9	**530649**	6438	2	**541460**	109	544347	8	2,6	6565	549542	9	1,5	8788
9_5_0,3	**248754**	9	0	**248754**	1,17	251834	7,5	1,24	1827	261654	9	5,2	14275
9_5_0,6	**360143**	22	0	**360143**	0,88	364848	7,5	1,3	5221	371868	8,7	3,25	14769
9_5_0,9	**469065**	106	1	**471531**	3,9	472225	7,4	0,7	4133	487293	8,7	3,34	16640
10_3_0,3	**623285**	62975	0	**623285**	582	623744	8	0,07	2515	**623285**	8	0	0
10_3_0,6	**808600**	54051	0	**808600**	674	810431	8	0,23	5406	**808600**	8	0	0
10_3_0,9	978170 gap 4,22	172800	2	**993914**	535	994810	8	-	3949	**993914**	8	0	0
10_4_0,3	**466026**	3504	0	**466026**	137	467063	8	0,22	1420	466601	8	0,12	230
10_4_0,6	**661129**	3651	0	**661129**	46	664402	8	0,5	8492	661634	8	0,08	1923
10_4_0,9	**852831**	6538	2	**855569**	91,55	861599	8	1,03	15352	858207	8	0,31	5998
10_5_0,3	**381763**	107	0	**381763**	19,65	429822	8	12,6	6869	422067	8	10,6	0
10_5_0,6	**585140**	103	0	**585140**	3,94	627203	7,5	7,2	7689	622370	8	6,4	0
10_5_0,9	**788188**	479	1	**788359**	6,26	825657	7	4,75	10851	822673	8	4,35	0

Table 5.17: Results with Exact Method and VNDS for direct hub links model with small instances (11 and 12 nodes)

Instance	Exact Method with sub-hub			Exact Method without sub-hub		VNDS with sub-hub				VNDS without sub-hub			
n_p_α	**Ob.Func.**	**E.T.**	**N.S-H.**	**Ob.Func.**	**E.T.**	**Ob.Func.**	**E.T.**	**P.D.**	**Std.Dev.**	**Ob.Func.**	**E.T.**	**P.D.**	**Std.Dev.**
11_3_0,3	976085 gap 29,04	172800	2	**902993**	49241	*911337*	7	-	16973	936369	8	3,7	16973
11_3_0,6	1102990 gap 20,36	172800	0	**1093760**	32762	*1095930*	7,5	-	11887	1109761	8	1,5	27905
11_3_0,9	1242930 gap 16,7	172800	1	**1284520**	36736	1305172	7	-	38075	1327922	8	3,4	44255
11_4_0,3	**602759**	100216	1	**621283**	577	622626	8	3,3	34604	653141	9	5,1	32834
11_4_0,6	**809181**	39385	1	**830919**	1067	827938	7,5	2,32	32736	865313	9	4,1	47751
11_4_0,9	1014940 gap 3,10	172800	1	**1040550**	966	1053261	7	-	59304	1083597	9	4,14	54611
11_5_0,3	**487167**	1073	0	**487167**	54	541435	7	11,14	48004	494599	9	1,53	17493
11_5_0,6	**705460**	532	0	**705460**	29	725136	8	2,8	15937	730997	9	3,62	31021
11_5_0,9	**923753**	14962	0	**923753**	52	944743	9	2,27	15224	941182	9	1,89	34546
12_3_0,3	1209310 gap 42	172800	3	997873 gap 15,42	172800	*1045962*	8	-	35093	1067284	9	-	24016
12_3_0,6	1289200 gap 30,58	172800	3	1164910 gap 8	172800	*1181062*	7	-	27760	1174339	9	-	3762
12_3_0,9	1355860 gap 22,7	172800	3	**1331950**	169600	*1354226*	7	-	39038	1343002	9	0,83	3747
12_4_0,3	727104 gap 14,91	172800	1	**734087**	44423	731414	7	-	5189	740728	9	0,9	2617
12_4_0,6	957539 gap 13,32	172800	3	**913172**	11273	**904237**	7	-	5143	918135	8,5	0,5	1980
12_4_0,9	1227890 gap 18,73	172800	4	**1091370**	5474	**1079902**	7	-	5272	1094707	8	0,3	2046
12_5_0,3	595607 gap 6,27	172800	1	**599315**	6875	641311	7	-	25051	643568	8	7,38	15831
12_5_0,6	789194 gap 0,94	172800	1	**796001**	1724	844320	8	-	46758	842030	8	5,78	30599
12_5_0,9	984083 gap 0,42	172800	1	**992686**	988	1041160	9	-	53856	1054789	8	6,26	41597

Table 5.18: Average of E.T. and P.D. for results of tables (5.16) and (5.17)

-	Exact Method with sub-hub	Exact Method without sub-hub	VNDS with sub-hub		VNDS without sub-hub	
-	**E.T.**	**E.T.**	**E.T.**	**P.D.**	**E.T.**	**P.D.**
average	78450	19738	7,6	1,67	8,5	2,5

i) Exact method

1. Version without sub-hub

Regarding the model without sub-hub, optimal results were found by the exact method in 34 instances. Only for instances 12-3-0,3 (a gap of 15,42%) and 12-3-0,6 (a gap of 8%), the optimal solutions were not found (see tables (5.16) and (5.17)).

2. Version with sub-hub

For the model with sub-hub, optimal results were found by the exact method in only 22 instances (see tables (5.16) and (5.17)). Once again the model with sub-hub is more complex to solve with an exact method than the model without sub-hub.

Among the 14 instances for which the optimal value has not been found, we mainly find those with 12 nodes.

3. Comparison between the two exact versions

Large execution time is registered for model with sub-hub when compared to the model without sub-hub (see tables (5.16) and (5.17)).

A high number of optimal solutions were found by exact methods. The number of optimal solutions found by model without sub-hub is higher than number of optimal solutions found by model with sub-hub (see table (5.19)). But, a few results from the model without sub-hub present a gain over the model with sub-hub.

Table 5.19: Observed gain for MMpHLRPSH over MMpHLRP (exact methods for small instances)

Instance	Gain in Ob.Func. (%)	Instance	Gain in Ob.Func. (%)	Instance	Gain in Ob.Func. (%)
9_3_0,3	**0,6**	10_4_0,3	**0**	11_5_0,3	**0**
9_3_0,6	**1,2**	10_4_0,6	**0**	11_5_0,6	**0**
9_3_0,9	**2**	10_4_0,9	**0,3**	11_5_0,9	**0**
9_4_0,3	**0**	10_5_0,3	**0**	12_3_0,3	-21
9_4_0,6	**0**	10_5_0,6	**0**	12_3_0,6	-10,6
9_4_0,9	**2**	10_5_0,9	**0,02**	12_3_0,9	*-1,7*
9_5_0,3	**0**	11_3_0,3	*-8*	12_4_0,3	*0,9*
9_5_0,6	**0**	11_3_0,6	*-8*	12_4_0,6	*-4,8*
9_5_0,9	**0,5**	11_3_0,9	*3,23*	12_4_0,9	*-17*
10_3_0,3	**0**	11_4_0,3	**3**	12_5_0,3	*0,6*
10_3_0,6	**0**	11_4_0,6	**2,6**	12_5_0,6	*0,9*
10_3_0,9	*1,6*	11_4_0,9	*2,5*	12_5_0,9	*0,9*

In table (5.19), gain values are given in bold when both models give optimal solution whereas there are given in italic when only the model without sub-hub give the optimal solution.

Negative gain values represent results for which model with sub-hub did not find the optimal and the objective function value is lower than the one provided by model without sub-hub.

4. Scale factor analysis (α parameter)

In these models, no relationship is seen between alpha parameter and execution time (see table (5.20)). Results are rather sparse.

Table 5.20: increase of E.T.: direct hub links_small instances, $\alpha = 0,6$ and $\alpha = 0,9$ compared with $\alpha = 0,3$

-	increase of execution time (%)			
-	α = 0,6		α = 0,9	
n_p	model with sub-hub	model without sub-hub	model with sub-hub	model without sub-hub
9_3	-31	-20	4	45
9_4	-35	-27	105	4
9_5	144	-24	10	233
10_3	-14	15	174	-8
10_4	4	-66	86	-33
10_5	-3	-8	347	-68
11_3	-	-33	-	43
11_4	-60	85	72	67
11_5	-50	-46	13	-3
12_3	-	-	-	-1
12_4	-	-74	-	-87
12_5	-	-75	-	-85

In table (5.20), each line represents an increase (or decrease) of execution time (see tables (5.16) and (5.17)) when alpha value is increased.

Table 5.21: decrease of E.T.: direct hub links_small instances, $p = 4$ and $p = 5$ compared with $p = 3$

-	decrease of execution time (%)			
-	$p = 4$		$p = 5$	
n_p_α	model with sub-hub	model without sub-hub	model with sub-hub	model without sub-hub
9_p_0,3	91	56	99,97	99,5
9_p_0,6	92	60	99,91	99,5
9_p_0,9	84	68	99,7	98,8
10_p_0,3	94	76	99,8	96,6
10_p_0,6	93	93	99,8	99,4
10_p_0,9	96	82	99,7	98,8
11_p_0,3	42	98,8	99,3	99,8
11_p_0,6	77	96,7	99,7	99,91
11_p_0,9	-	97,3	91,3	99,8
12_p_0,3	-	74,29	-	96
12_p_0,6	-	93	-	99
12_p_0,9	-	96,7	-	99,4
average	55	82	74	98,87

In table (5.21), each line represents a decrease of execution time (see tables (5.16) and (5.17)) when number of hubs is increased.

In the first line of table (5.21), execution time of instance 9_4_0,3 for the model with sub-hub is decreased 91% compared to the instance 9_3_03. Whereas instance 9_5_0,3 is decrease 99,97% compared to the instance 9_3_03.

Note that when a model reaches execution time limit, a symbol "-" is put in the corresponding case.

ii) VNDS approach

1. Version without sub-hub

Among 34 instances for which the exact method found an optimal solution, VNDS found optimal solution for 24 instances (see tables (5.16), (5.17) and (5.22)). VNDS did not find optimal solutions for 10 respective instances: (9-5-0,3; 9-5-0,6; 9-5-0,9; 10-5-0,3; 10-5-0,6; 10-5-0,9; 12-4-0,3; 12-5-0,3; 12-5-0,6 and 12-5-0,9).

About the two instances where optimal was not found by the exact method:

- in instance 12-3-0,6, objective function found by exact method (1164910) was also found in 4 executions of VNDS
- in instance 12-3-0,3, the best objective value found by VNDS was 1007360 in 4 executions.

Therefore, 25 results presented by exact method were also by VNDS. Whereas, small gaps were presented by VNDS for the 11 remaining results.

For instances with standard deviation, the best results found over 30 runs are showed in table (5.22). Number of times over 30 runs that best objective value has been achieved is indicated in last column (nb. of times).

Table 5.22: Best results with VNDS for instances presenting a standard deviation (MM-pHLRP)

instance	best objective function	nb. of times	instance	best objective function	nb. of times
9_4_0,3	**333382**	16	11_4_0,9	**1040550**	15
9_4_0,9	**541460**	16	11_5_0,3	**487167**	25
9_5_0,3	250204	18	11_5_0,6	**705460**	17
9_5_0,6	361600	20	11_5_0,9	**923753**	23
9_5_0,9	472996	17	12_3_0,3	1007360	4
10_4_0,3	**466026**	4	12_3_0,6	1164910	4
10_4_0,6	**661129**	28	12_3_0,9	**1331950**	3
10_4_0,9	**855569**	25	12_4_0,3	734975	5
11_3_0,3	**902993**	6	12_4_0,6	**913172**	4
11_3_0,6	**1093760**	22	12_4_0,9	**1091370**	8
11_3_0,9	**1284520**	15	12_5_0,3	617748	2
11_4_0,3	**621283**	13	12_5_0,6	812716	9
11_4_0,6	**830919**	17	12_5_0,9	1007690	4

Note that results in bold are optimal solutions (see table (5.22)).

2. Version with sub-hub

Among 22 instances for which exact method found optimal solution, only 5 optimal solutions were found by VNDS method (see tables (5.16), (5.17) and (5.23)).

Among 17 instances where optimal solution was not found by VNDS:

- in 5 instances (9-3-0,3; 9-3-0,6; 9-3-0,9, 9-4-0,9 and 10-4-0,9), optimal value of the exact model without sub-hub was found in some executions.

- in 3 instances (9-5-0,9; 11-4-0,3 and 11-4-0,6), results found in some executions were better than the optimal value of exact model without sub-hub. Minimal objective function value found was 471390, 608674 and 814942, respectively, to compare to the model without sub-hub.

- in 9 instances(9-5-0,3; 9-5-0,6; 10-4-0,3; 10-5-0,3; 10-5-0,6; 10-5-0,9; 11-5-0,3; 11-5-0,6 and 11-5-0,9) the results were close to optimal with P.D.: (1,24%), (1,3%), (0,22%), (12,6%), (7,2%),(4,75%), (11,14%), (2,8%) and (2,27%).

Among 14 instances where optimal was not found by the exact method, a gap was decreased in 8 instances by VNDS: 11-3-0,3; 11-3-0,6; 12-3-0,3; 12-3-0,6; 12-3-0,9; 12-4-0,3; 12-4-0,6 and 12-4-0,9.

Instances where the gap was not decreased by VNDS are:

- 10-3-0,9 and 11-3-0,9 - Solutions found by VNDS in 23 executions on 30 were equal to optimal values provided by the model without sub-hub (993914 and 1284520, respectively);

- 11-4-0,9 - The result found by VNDS (1020850) in 19 executions on 30 was better than the optimal value provided by the model without sub-hub (1040550);

- 12-5-0,9 - The result found by VNDS (992054) in 16 executions on 30 was better than the optimal value provided by the model without sub-hub (992686);

- 12-5-0,3 - The best solution found by VNDS (608597) was above of the optimal found for the optimal value provided by the model without sub-hub (599315);

- 12-5-0,6 - The best solution found by VNDS (800325) was above of the optimal found for the optimal value provided by the model without sub-hub (796001);

On instances with standard deviation, the best results found are showed in table (5.23). The number of times over 30 runs that the best objective value has been achieved is indicated in the last column (nb. of times).

Table 5.23: Best results with VNDS for instances presenting a standard deviation (MM-pHLRPSH)

instance	best objective function	nb. of times	instance	best objective function	nb. of times	instance	best objective function	nb. of times
9_3_0,3	439654	20	10_4_0,6	**661129**	25	11_5_0,6	722090	28
9_3_0,6	531549	29	10_4_0,9	855569	25	11_5_0,9	939770	27
9_3_0,9	623445	27	10_5_0,3	426339	9	12_3_0,3	997873	1
9_4_0,3	**333382**	25	10_5_0,6	625183	28	12_3_0,6	1164910	5
9_4_0,9	541460	25	10_5_0,9	822806	28	12_3_0,9	1331950	5
9_5_0,3	251404	28	11_3_0,3	902993	24	12_4_0,3	**725634**	11
9_5_0,6	362800	2	11_3_0,6	1093760	29	12_4_0,6	**900401**	18
9_5_0,9	471390	28	11_3_0,9	1284520	23	12_4_0,9	**1075040**	15
10_3_0,3	**623285**	29	11_4_0,3	608674	25	12_5_0,3	608597	8
10_3_0,6	**808600**	26	11_4_0,6	814942	25	12_5_0,6	800325	14
10_3_0,9	993914	23	11_4_0,9	1020850	19	12_5_0,9	992054	16
10_4_0,3	466690	28	11_5_0,3	503894	18	-	-	-

In tables (5.23), results in bold are optimal or best solutions found by VNDS.

3. Comparison between the two VNDS versions

In table (5.24), negative values are results for which model without sub-hub achieved a better solution than model with sub-hub.

In these VNDS versions, the model without sub-hub also found a greater number of optimal values.

However, the gain in objective function presented by the model with sub-hub over the model without sub-hub is small (average value of gain is 0,18% over the 36 instances). The solutions presented by model with sub-hub are slightly better than results for the model without sub-hub (see gain in table (5.24)).

Table 5.24: Observed gain for MMpHLRPSH over MMpHLRP (small instances with VNDS

Instance	Gain in Ob.Func. (%)	Instance	Gain in Ob.Func. (%)	Instance	Gain in Ob.Func. (%)
9_3_0,3	0	10_4_0,3	-0,1	11_5_0,3	-3
9_3_0,6	0	10_4_0,6	0	11_5_0,6	-2
9_3_0,9	0	10_4_0,9	0	11_5_0,9	-1,7
9_4_0,3	0	10_5_0,3	-1	12_3_0,3	0,9
9_4_0,6	0	10_5_0,6	-0,4	12_3_0,6	0
9_4_0,9	0	10_5_0,9	-0,0001	12_3_0,9	0
9_5_0,3	-0,4	11_3_0,3	0	12_4_0,3	1,2
9_5_0,6	-0,3	11_3_0,6	0	12_4_0,6	1,3
9_5_0,9	0,3	11_3_0,9	0	12_4_0,9	1,5
10_3_0,3	0	11_4_0,3	2	12_5_0,3	1,5
10_3_0,6	0	11_4_0,6	2	12_5_0,6	1,5
10_3_0,9	0	11_4_0,9	1,8	12_5_0,9	1,5

iii) Comparison between VNDS and Exact Method

In table (5.25), the gain in objective function for small instances is presented. This gain concerns the model with sub-hub under model without sub-hub when this one achieved the optimal result.

In table (5.25), gains are relatively small but are found in 8 of instances.

Table 5.25: Observed gain between the MMpHLRPSH over MMpHLRP (VNDS and exact method for small instances)

Instance	Gain in Ob.Func. (%)	Instance	Gain in Ob.Func. (%)	Instance	Gain in Ob.Func. (%)
9_3_0,3	0	10_4_0,3	-0,1	11_5_0,3	-3,4
9_3_0,6	0	10_4_0,6	0	11_5_0,6	-2,3
9_3_0,9	0	10_4_0,9	0	11_5_0,9	-1,7
9_4_0,3	0	10_5_0,3	-11,6	12_3_0,9	0
9_4_0,6	0	10_5_0,6	-6,8	12_4_0,3	1,1
9_4_0,9	0	10_5_0,9	-4,3	12_4_0,6	1,4
9_5_0,3	-1	11_3_0,3	0	12_4_0,9	1,5
9_5_0,6	-0,7	11_3_0,6	0	12_5_0,3	-1,5
9_5_0,9	0,0003	11_3_0,9	0	12_5_0,6	-0,5
10_3_0,3	0	11_4_0,3	2	12_5_0,9	0,0006
10_3_0,6	0	11_4_0,6	1,9	-	-
10_3_0,9	0	11_4_0,9	1,9	-	-

Number of optimal results found by VNDS for model with sub-hub was only 5. Although, the solution found by VNDS for the model with sub-hub was equal to the optimal for the model without sub-hub in 12 instances: 9-3-0,3; 9-3-0,6; 9-3-0,9; 9-4-0,9; 10-3-0,9; 10-4-0,9; 11-3-0,3; 11-3-0,6; 11-3-0,9; 12-3-0,3; 12-3-0,6 and 12-3-0,9.

In other 8 instances, VNDS results for model with sub-hub were better than optimal results for the model without sub-hub: 9-5-0,9; 11-4-0,3; 11-4-0,6; 11-4-0,9; 12-4-0,3; 12-4-0,6; 12-4-0,9 and 12-5-0,9.

Between the 36 small instances analyzed with MMpHLRPSH and MMpHLRP, results for the first model are equal to the ones obtained by the second for 18 instances. In all other instances, MMpHLRPSH presented better results.

5.3.2 Large Instances

i) VNDS approach

In tables (5.26), (5.27) and (5.28), results for the second group of 45 instances (AP instances) are presented. In these tables, Ob.Func. is the average value of the objective function over 30 runs, S.D is the corresponding standard deviation, E.T. is the average value of execution time in seconds, Min.Ob.Func. is the smallest objective function value found among 30 executions and Sub-hub.M.O. is the number of sub-hubs to open corresponding to Min.Ob.Func. value.

Table 5.26: Results with VNDS for direct hub links model with large instances (100 nodes)

Instance	VNDS with sub-hub					VNDS without sub-hub			
n_p_α	**Ob.Func.**	**Std.Dev.**	**E.T.**	**Min.Ob.Func.**	**Sub-hub.M.O.**	**Ob.Func.**	**Std.Dev.**	**E.T.**	**Min.Ob.Func.**
100_8_0,3	151627	2233	105	146122	4	178248	1868	96	171919
100_8_0,6	165957	3190	104	159410	4	189182	2656	99	181970
100_8_0,9	178392	3413	101	166454	3	199144	4094	103	190301
100_10_0,3	133344	2742	131	127292	6	161101	2740	129	153708
100_10_0,6	146087	3471	142	135766	6	170691	4794	129,5	158457
100_10_0,9	159028	2024	128	154950	6	181774	4650	145	174718
100_12_0,3	109681	3846	169	103480	9	137637	3010	157	130414
100_12_0,6	125304	3255	170	119503	9	149286	3642	160	141950
100_12_0,9	138039	3470	169	132129	9	161315	3328	157	155525

Table 5.27: Results with VNDS for direct hub links model with large instances (125 and 150 nodes)

Instance	VNDS with sub-hub					VNDS without sub-hub			
n_p_α	**Ob.Func.**	**Std.Dev.**	**E.T.**	**Min.Ob.Func.**	**Sub-hub.M.O.**	**Ob.Func.**	**Std.Dev.**	**E.T.**	**Min.Ob.Func.**
125_8_0,3	161018	5283	223	146129	5	197581	4167	197	183743
125_8_0,6	173850	3051	216	168279	5	211267	6130	201	188091
125_8_0,9	187997	4453	209	180080	5	223404	5746	208	211181
125_10_0,3	128352	3033	226	123355	7	161707	1955	226	156245
125_10_0,6	142146	1597	234	137671	8	175550	1298	218	172011
125_10_0,9	156847	1436	216	154216	7	188241	2218	226	182559
125_12_0,3	113860	2403	264	110703	9	146509	1492	256	139990
125_12_0,6	130040	4401	252	127110	9	161132	1830	292	156165
125_12_0,9	144326	2370	253	139898	9	174432	2673	315	169745
150_8_0,3	184974	4433	335	177926	3	204530	4078	327	185331
150_8_0,6	200313	6424	330	182641	3	217841	1917	329	209690
150_8_0,9	213655	6228	337	196373	3	228891	3867	337	215858
150_10_0,3	140117	3426	376	133500	7	175066	807	381	172336
150_10_0,6	153835	1474	381	149676	7	188491	1779	393	180874
150_10_0,9	168585	1313	370	163689	7	201066	1974	409	195361
150_12_0,3	130604	3685	488	123087	9	157144	1056	448	152644
150_12_0,6	145832	2221	454	140329	9	171986	1676	500	168352
150_12_0,9	159375	2837	414	150047	9	184418	3634	538	173659

Table 5.28: Results with VNDS for direct hub links model with large instances (175 and 200 nodes)

Instance	VNDS with sub-hub					VNDS without sub-hub			
n_p_α	**Ob.Func.**	**Std.Dev.**	**E.T.**	**Min.Ob.Func.**	**Sub-hub.M.O.**	**Ob.Func.**	**Std.Dev.**	**E.T.**	**Min.Ob.Func.**
175_8_0,3	202582	4659	541	195146	5	232968	6262	568,5	220707
175_8_0,6	214801	4725	522	204510	5	235891	2459	541	228436
175_8_0,9	226853	5898	553	214255	5	249708	1531	533	246851
175_10_0,3	168594	5441	650	159143	8	185065	1242	619	181236
175_10_0,6	179546	7136	688	164360	7	197032	726	610	194367
175_10_0,9	193982	8187	702	180731	7	210238	2472	632	202916
175_12_0,3	137560	8271	761	125611	11	159153	1294	704	155766
175_12_0,6	151735	7590	758	135745	11	175025	1259	711	172656
175_12_0,9	164438	7421	705	151729	11	189965	1772	715	186610
200_8_0,3	230760	7681	851	219048	2	240446	1866	853	236902
200_8_0,6	241879	6895	863	227037	2	252161	2900	858	243841
200_8_0,9	252986	7138	872	238683	2	263567	3247	840	255388
200_10_0,3	177407	11235	950	164477	6	210322	825	924	208809
200_10_0,6	193047	10773	986	178093	6	224005	2407	948	215817
200_10_0,9	213852	9976	964	198553	6	236293	3232	948	227981
200_12_0,3	150404	4563	1394	140686	12	180835	913	1085	176973
200_12_0,6	164450	4891	1291	155081	12	196767	2416	1232	190686
200_12_0,9	176903	6419	1207	167428	12	209852	2562	1197	202749

1. Version without sub-hub

In table (5.29), the coefficient of variation (CV) (equation 5.3) of objective function (Ob.Func.) presented in the tables (5.26), (5.27) and (5.28) is showed. A small coefficient of variation is presented for each instance.

Table 5.29: Coefficient of variation for results with large instances (direct hub links models)

-	Coefficient of variation					
-	$\alpha = 0{,}3$		$\alpha = 0{,}6$		$\alpha = 0{,}9$	
n_p	model with sub-hub	model without sub-hub	model with sub-hub	model without sub-hub	model with sub-hub	model without sub-hub
100_8	1,5	1	1,9	1,4	1,9	2
100_10	2	1,7	2,3	2,8	1,2	2,5
100_12	3,5	2,2	2,6	2,4	2,5	2
125_8	3,3	2,1	1,7	2,9	2,3	2,6
125_10	2,3	1,2	1,1	0,7	0,9	1,2
125_12	2,1	1	3,3	1,1	1,6	1,5
150_8	2,4	2	3,2	0,9	2,9	1,7
150_10	2,4	0,4	0,9	0,9	0,8	1
150_12	2,8	0,6	1,5	1	1,8	2
175_8	2,3	2,7	2,2	1	2,6	0,6
175_10	3,2	0,6	4	0,3	4,2	1,2
175_12	6	0,8	5	0,7	4,5	0,9
200_8	3,3	0,7	3	1,1	2,8	1,2
200_10	6,3	0,4	5,6	1	4,6	1,3
200_12	3	0,5	3	1,2	3,6	1,2

Unlike the tests on small instances, execution time is increased when the number of hubs is increased (see tables (5.26), (5.27) and (5.28). Execution time is also increased when the number of nodes is increased.

2. Version with sub-hub

VNDS for model with sub-hub has showed similar behavior to VNDS for the model without sub-hub:

- Standard deviation is also showed without large variation between the instances.

- Execution time is increased together with the number of nodes and number of hubs.

3. Comparison between the two VNDS versions

In table (5.30), the gain in objective function of the model with sub-hub under the model without sub-hub is presented for large instances. Significant gains were registered for all instances.

Similar to the results from the circular models, the best results are obtained by the version with sub-hub. The smallest number of sub-hubs generated in an instance was 2 (200-8-0,3, 200-8-0,6 and 200-8-0,9).

Although the gain has decreased if compared to the one obtained by the circular model, results showed greater gains when compared to results with small instances. Again, the superiority (positive gain) of a network with sub-hubs over a network without sub-hub is still confirmed.

Execution times of VNDS for large instances are similar for both model with sub-hub and without sub-hub.

Similar to circular models, the objective function is decreased when the number of hubs is increased due to a higher number of arc hubs with economy of scale in the network. This characteristic can be seen in both models (tables 5.16 - 5.28).

Table 5.30: Observed gain for MMpHLRPSH over MMpHLRP (large instances)

Instance	Gain in Ob.Func. (%)	Instance	Gain in Ob.Func. (%)	Instance	Gain in Ob.Func. (%)
100_8_0,3	14,9	125_12_0,3	22,3	175_10_0,3	8,9
100_8_0,6	12,3	125_12_0,6	19,3	175_10_0,6	8,9
100_8_0,9	10,4	125_12_0,9	17,3	175_10_0,9	7,7
100_10_0,3	17,2	150_8_0,3	9,6	175_12_0,3	13,6
100_10_0,6	14,4	150_8_0,6	8	175_12_0,6	13,3
100_10_0,9	12,5	150_8_0,9	6,7	175_12_0,9	13,4
100_12_0,3	20,3	150_10_0,3	20	200_8_0,3	4
100_12_0,6	16,1	150_10_0,6	18,4	200_8_0,6	4,1
100_12_0,9	14,4	150_10_0,9	16,2	200_8_0,9	4
125_8_0,3	18,5	150_12_0,3	16,9	200_10_0,3	15,6
125_8_0,6	17,7	150_12_0,6	15,2	200_10_0,6	13,8
125_8_0,9	15,8	150_12_0,9	13,6	200_10_0,9	9,5
125_10_0,3	20,6	175_8_0,3	13	200_12_0,3	16,8
125_10_0,6	19	175_8_0,6	8,9	200_12_0,6	16,4
125_10_0,9	16,7	175_8_0,9	9,2	200_12_0,9	15,7

5.4 Conclusions

The obtained results with small instances and with AP instances were enough to highlight that a lower transport cost is obtained in a network with shorter alternative paths by using sub-hubs, alongside an economy of scale provided by hub links.

In a network with circular hub links, gain in objective function is more significant because distances between regions of a network without sub-hub are larger.

In a network with direct hub links, different regions are connected by direct hub links offering an economy of scale and shorter paths. Consequently, gains in network with sub-hubs are decreased but are still obtained when shorter alternative paths are offered.

Models with sub-hub and direct connection between hub have showed the lowest objective function costs. However, choose between a direct hub connection or circular hub network depends on the application and/or vehicle used.

Though, regardless of whether the decision is taken under the direct hub connections or the circular hub network, sub-hubs are crucial to achieve the lowest costs.

Optimal results were found by VNDS for the four developed model tested with different instances. For instances where VNDS did not find the optimal for models with sub-hub, results are near to optimal values or are equal to optimal solutions provided by the models without sub-hub.

In models without sub-hub, the number of optimal results found by VNDS were greater than in models with sub-hubs. However, for large instances, significant gains are registered by models with sub-hub.

Chapter 6

Conclusions and Perspectives

In this thesis, the study of hierarchical network structures in a context of hub location and routing problem is presented.

Hub-and-spoke models are based on the premises that transport cost in network is minimized when hub nodes and high-capacity means of transport are used because an economy of scale is possible.

In this study, we assume the importance of hub nodes with economy of scale for a lower transport cost. But, we do not pin in this economy of scale the only way to get an attractive network. We decide to mix economy of scale and shorter alternative paths to offer more efficient networks in terms of delivery time and, consequently, of transport cost.

The proposed mathematical models are situated in a field of literature about hub location and routing problem not yet explored (see chapter 1). The models allow the making of hierarchical networks with cluster level (region) composed of circular routes and a intersection point between two clusters (see chapter 3).

A intersection point between two clusters (multiple allocation of spoke node) is named "sub-hub" because it acts as a transshipment point between two close regions.

Models with sub-hub are compared with models without sub-hub (single allocation of nodes) trying to prove the importance of sub-hubs (see chapter 5).

In the mathematical models, the number of sub-hubs is not established. So, in the worst case, results for the model with sub-hub are equal to the model without sub-hub. But, results have confirmed the importance of sub-hub in a hub-and-spoke network structure. Models with sub-hub have showed gain in objective function over the model without sub-hub of up to 14% for small instances and 36% for large instances (variant with circular hub routes) and of 3% for small instances and 22% for large instances (variant with direct hub links).

Only an upper limit for number of sub-hubs is applied, because between two hubs it just possible to have, at most, one sub-hub. Then, a solution found by mathematical model with sub-hub could have $C_{p,2}$ sub-hubs.

Therefore, lower cost in objective function is the condition to insert one or more sub-hubs in the global solution.

Comparing results from the model with sub-hub and its version without sub-hub, the best solutions were generally obtained by the insertion of sub-hubs in network.

Results evidence that hub network structure with economy of scale in hub links and circular regional routes inside disjoint clusters can not offer the best results in time delivery and transport cost.

The concept of sub-hub emerges as an important component providing more competitive network structures in transport of goods.

Concerning the characteristics of the developed models, no limit is established for number of nodes in each cluster. Clusters can have a quantity of nodes between only one node (only hub) and $(n-p+1)$ nodes (cluster composed of all spoke nodes and its hub).

About the number of hub nodes, we want to address that in the proposed network structure, there is an upper and a lower value limit.

The upper limit is obtained when the number of hub equals number of nodes ($p = n$). Lower limit is 2 because in each hub node an incoming arc and an outgoing arc are established. Consequently, 2 hub nodes are required to create the links.

Although sub-hubs are playing the same role as hubs (aggregating demands) at regional level, no discount factor (economy of scale) is applied at regional level. So, an economy of scale in regional level could increase the gain in transport cost of a network with sub-hub over a network without sub-hub.

However, installation costs for hub and sub-hub are not taken into account in the proposed models. For this reason, although sub-hub represents a hub with lower capacities/dimensions, installation cost of this facility could decrease effective gains of a network with sub-hub.

A characteristic widely used in the hub-and-spoke literature is that customer demands are transported in circular hub network, both clockwise and counterclockwise (two way traffic). In our models, connections between two nodes are composed of directed arcs, so one way traffic is proposed.

The difference between the two situations above will depend on the need of transport. In the first situation, time delivery of some demands is minimized. However, the number of vehicles can be increased. Whereas in the second one, the number of vehicles can be decreased and delivery time can be increased. In this second situation, companies using high-capacity vehicles, liner shipping companies, for example (see chapter 2), can be more effectively managed.

Our hub-and-spoke network with sub-hub promotes liner shipping service because it relieves the flow in hub ports, decreasing delivery time and, consequently, increasing the network capacity. The proposed network also allows for development of regional services favoring a better cargo distribution across the network and helping the increasing door-to-door services.

Our computational tests trying to compare models also provided further proof about the complexity to solve these problems.

A cutting plane approach was used to insert sub-tour constraints in an exact solving approach and a Variable Neighborhood Decomposition Search (VNDS) (see chapter 4) was used to solve large instances of problems.

About the sub-tour elimination constraint, a special attention was paid to regional tours because the start point of route (hub node), the number of nodes inside a cluster and nodes included in a cluster are free.

In our implemented VNDS, each part from the problem (location, allocation, service design and routing) is solved by specific operators.

Ingredients from VNDS are included in the implemented VND. A first VND uses a shake procedure in sub-hub location problem. Whereas a decomposition of service design problem is made by means of a second VND.

In classic VNDS, shake procedure is done before applying descent operator. First, a diversification of search space is made by shake and after an intensification search is realized by the descent operator.

In our implemented VNDS, shake procedure used in VNDS and in VND are performed after the improvement phase of all neighborhoods. An intensification phase is done, followed by a diversification phase.

Results with exact method and with VNDS were fundamental to prove gains in objective function of models with sub-hub over models without sub-hub.

Regarding to the context of hub location and routing problems, perspectives for future works are numerous. But, the main points observed are highlighted below.

Mathematical models developed are included in the second and fourth categories from classification presented by Nagy and Salhi (2007) (see Table (1.2)). Development of new models for the other two categories not yet explored could validate the importance of sub-hub in all categories of hub location and routing problems.

Mathematical models allowing sub-hub allocated to more than two hubs could produce network solutions even more economic in terms of transport cost and more efficient in terms of delivery time.

Models allowing two sub-hubs allocated to a pair of close clusters could also offer a greater degree of freedom in the development of more economic networks.

A deeper study about which is the best number of hubs/sub-hubs to open in a model including installation and maintenance costs would be interesting.

Mathematical models allowing flows in both direction (no-directed arcs) could be also tested in public transport domain.

At tactical level, it will be interesting to study how many vehicles compose the fleet, how many schedules need to be created, how to synchronize these ones at transshipment points in order to manage services that companies want to offer to their customers.

These perspectives will be important contributions in the field of hub location and routing problem with multiples allocations seeking to bring theoretical models closer to practical cases.

Bibliography

Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Chapter IV Network flows. Handbooks in operations research and management science **1**, 211–369 (1989)

Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network flows: theory, algorithms, and applications (1993)

Alibeyg, A., Contreras, I., Fernández, E.: Hub network design problems with profits. Transportation Research Part E: Logistics and Transportation Review **96**, 40–59 (2016)

Altinkemer, K.: Topological design of ring networks. Computers & operations research **21**(4), 421–431 (1994)

Alumur, S., Kara, B.Y.: Network hub location problems: The state of the art. European Journal of Operational Research **190**(1), 1–21 (2008)

Alumur, S.A.: Hub location and hub network design. Ph.D. thesis, Bilkent University (2009)

Alumur, S.A., Kara, B.Y., Karasan, O.E.: Multimodal hub location and hub network design. Omega **40**(6), 927–939 (2012a)

Alumur, S.A., Nickel, S., Saldanha-da Gama, F.: Hub location under uncertainty. Transportation Research Part B: Methodological **46**(4), 529–543 (2012b)

Alumur, S.A., Yaman, H., Kara, B.Y.: Hierarchical multimodal hub location problem with time-definite deliveries. Transportation Research Part E: Logistics and Transportation Review **48**(6), 1107–1120 (2012c)

Amin-Naseri, M.R., Yazdekhasti, A., Salmasnia, A.: Robust bi-objective optimization of uncapacitated single allocation p-hub median problem using a hybrid heuristic algorithm. Neural Computing and Applications pp. 1–22 (2016)

An, Y., Zhang, Y., Zeng, B.: The reliable hub-and-spoke design problem: Models and algorithms. Transportation Research Part B: Methodological **77**, 103–122 (2015)

Andersen, M.W.: Service Network Design and Management in Liner Container Shipping Applications. DTU Transport (2010)

Arshadi Khamseh, A., Doost Mohamadi, M.: Complete/incomplete hierarchical hub center single assignment network problem. Journal of Optimization in Industrial Engineering **7**(14), 1–12 (2014)

Aykin, T.: Lagrangian relaxation based approaches to capacitated hub-and-spoke network design problem. European Journal of Operational Research **79**(3), 501–523 (1994)

Aykin, T.: The hub location and routing problem. European Journal of Operational Research **83**(1), 200–219 (1995)

Balas, E.: The prize collecting traveling salesman problem. Networks **19**(6), 621–636 (1989)

Bellmore, M., Nemhauser, G.L.: The traveling salesman problem: a survey. Operations Research **16**(3), 538–558 (1968)

Beltramin, A.: Modern branch-and-cut solvers for mixed-integer linear programming: a computational comparison (2015)

Bertsimas, D., Dunning, I., Lubin, M.: Reformulation versus cutting-planes for robust optimization. Computational Management Science **13**(2), 195–217 (2016)

Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys (CSUR) **35**(3), 268–308 (2003)

Boschetti, M.A., Maniezzo, V., Roffilli, M., Röhler, A.B.: Matheuristics: optimization, simulation and control. In: International Workshop on Hybrid Metaheuristics, pp. 171–177. Springer (2009)

Bostel, N., Dejax, P., Zhang, M.: A model and a metaheuristic method for the hub location routing problem and application to postal services. In: Industrial Engineering and Systems Management (IESM), 2015 International Conference on, pp. 1383–1389. IEEE (2015)

Boussaïd, I., Lepagnot, J., Siarry, P.: A survey on optimization metaheuristics. Information Sciences **237**, 82–117 (2013)

Brimberg, J., Hansen, P., Mladenović, N.: Decomposition strategies for large-scale continuous location–allocation problems. IMA Journal of Management Mathematics **17**(4), 307–316 (2006)

Brouer, B.D., Dirksen, J., Pisinger, D., Plum, C.E., Vaaben, B.: The vessel schedule recovery problem (vsrp)–a mip model for handling disruptions in liner shipping. European Journal of Operational Research **224**(2), 362–374 (2013)

de Camargo, R.S., de Miranda, G., Løkketangen, A.: A new formulation and an exact approach for the many-to-many hub location-routing problem. Applied Mathematical Modelling **37**(12), 7465–7480 (2013)

de Camargo, R.S., Miranda Jr, G., Ferreira, R.P.M., Luna, H.: Multiple allocation hub-and-spoke network design under hub congestion. Computers & Operations Research **36**(12), 3097–3106 (2009)

Campbell, J.: Hub location problems and the p-hub median problem. Tech. rep., Center for Business and Industrial Studies Working Paper 91-06-21, University of Missouri-St. Louis, St. Louis, MO. (1991)

Campbell, J.F.: Integer programming formulations of discrete hub location problems. European Journal of Operational Research **72**(2), 387–405 (1994)

Campbell, J.F., O'Kelly, M.E.: Twenty-five years of hub location research. Transportation Science **46**(2), 153–169 (2012)

Carpenter, T., Luss, H.: Telecommunications access network design. Handbook of optimization in telecommunications pp. 313–339 (2006)

Catanzaro, D., Gourdin, E., Labbé, M., Özsoy, F.A.: A branch-and-cut algorithm for the partitioning-hub location-routing problem. Computers & operations research **38**(2), 539–549 (2011)

Çetiner, S., Sepil, C., Süral, H.: Hubbing and routing in postal delivery systems. Annals of Operations Research **181**(1), 109–124 (2010)

Chinneck, J.W.: Practical optimization: a gentle introduction. Systems and Computer Engineering), Carleton University, Ottawa. http://www.sce.carleton.ca/faculty/chinneck/po.html (2006)

Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D.: Maritime transportation. Handbooks in operations research and management science **14**, 189–284 (2007)

Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D.: Ship routing and scheduling in the new millennium. European Journal of Operational Research **228**(3), 467–483 (2013)

Christofides, N.: Worst-case analysis of a new heuristic for the travelling salesman problem. Tech. rep., Carnegie-Mellon Univ Pittsburgh PA Management Sciences Research Group (1976)

Colorni, A., Dorigo, M., Maniezzo, V., et al.: Distributed optimization by ant colonies. In: Proceedings of the first European conference on artificial life, vol. 142, pp. 134–142. Paris, France (1991)

Contreras, I., Fernández, E., Marín, A.: Tight bounds from a path based formulation for the tree of hub location problem. Computers & Operations Research **36**(12), 3117–3127 (2009)

Contreras, I., Fernández, E., Marín, A.: The tree of hubs location problem. European Journal of Operational Research **202**(2), 390–400 (2010)

Contreras, I., Tanash, M., Vidyarthi, N.: Exact and heuristic approaches for the cycle hub location problem. Annals of Operations Research pp. 1–23 (2016)

Correia, I., Nickel, S., Saldanha-da Gama, F.: Hub and spoke network design with single-assignment, capacity decisions and balancing requirements. Applied Mathematical Modelling **35**(10), 4841–4851 (2011)

da Costa Fontes, F.F., Goncalves, G.: Hub location and routing problem with alternative paths. In: Advanced Logistics and Transport (ICALT), 2015 4th International Conference on, pp. 317–322. IEEE (2015a)

da Costa Fontes, F.F., Goncalves, G.: Routing problem with pendular and cyclic service in a hierarchical structure of hub and spoke with multiple allocation of sub-hubs. In: Industrial Engineering and Systems Management (IESM), 2015 International Conference on, pp. 561–567. IEEE (2015b)

da Costa Fontes, F.F., Goncalves, G.: Hub location and routing problem: a variable neighborhood decomposition search based solving approach. In: Proceedings of 6th International Conference on Metaheuristics and Nature Inspired Computing (META'2016), pp. 527–536 (2016a)

da Costa Fontes, F.F., Goncalves, G.: A variable neighborhood decomposition search applied to the hierarchical hub location and routing problem. Matheuristics 2016 p. 91 (2016b)

da Costa Fontes, F.F., Goncalves, G.: A new hub network design integrating deep sea and short sea services at liner shipping operations. Int. J. Shipping and Transport Logistics **9**(5), 580–600 (2017)

Crainic, T.G.: Service network design in freight transportation. European Journal of Operational Research **122**(2), 272–288 (2000)

Crainic, T.G., Kim, K.H.: 'Intermodal transportation'. Handbooks in operations research and management science **14**, 467–537 (2007)

Crainic, T.G., Laporte, G.: Planning models for freight transportation. European journal of operational research **97**(3), 409–438 (1997)

Crainic, T.G., Mancini, S., Perboli, G., Tadei, R.: Clustering-based heuristics for the two-echelon vehicle routing problem. Montreal, Canada, Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation **17**, 26 (2008)

Croes, G.A.: A method for solving traveling-salesman problems. Operations research **6**(6), 791–812 (1958)

Cunha, C.B., Silva, M.R.: A genetic algorithm for the problem of configuring a hub-and-spoke network for a LTL trucking company in Brazil. European Journal of Operational Research **179**(3), 747–758 (2007)

Dantzig, G., Fulkerson, R., Johnson, S.: Solution of a large-scale traveling-salesman problem. Journal of the operations research society of America **2**(4), 393–410 (1954)

Daskin, M.S., Maass, K.L.: The p-median problem. In: Location Science, pp. 21–45. Springer (2015)

Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische mathematik **1**(1), 269–271 (1959)

Drexl, M., Schneider, M.: A survey of variants and extensions of the location-routing problem. European Journal of Operational Research **241**(2), 283–308 (2015)

Dumas, Y., Desrosiers, J., Gelinas, E., Solomon, M.M.: An optimal algorithm for the traveling salesman problem with time windows. Operations research **43**(2), 367–371 (1995)

EghbaliZarch, M., Abedzadeh, M., Setak, M.: Differential evolution algorithm for multi-commodity and multi-level of service hub covering location problem. International Journal of Industrial Engineering Computations **4**(1), 127–138 (2013)

Elhedhli, S., Hu, F.X.: Hub-and-spoke network design with congestion. Computers & Operations Research **32**(6), 1615–1632 (2005)

Ernst, A.T., Krishnamoorthy, M.: Efficient algorithms for the uncapacitated single allocation p-hub median problem. Location science **4**(3), 139–154 (1996)

Farahani, R.Z., Asgari, N., Heidari, N., Hosseininia, M., Goh, M.: Covering problems in facility location: A review. Computers & Industrial Engineering **62**(1), 368–407 (2012)

Farahani, R.Z., Hekmatfar, M., Arabani, A.B., Nikbakhsh, E.: Hub location problems: A review of models, classification, solution techniques, and applications. Computers & Industrial Engineering **64**(4), 1096–1109 (2013)

Feo, T.A., Resende, M.G.: A probabilistic heuristic for a computationally difficult set covering problem. Operations research letters **8**(2), 67–71 (1989)

Floyd, R.W.: Algorithm 97: shortest path. Communications of the ACM **5**(6), 345 (1962)

Ford Jr, L.R.: Network flow theory. Tech. rep., RAND Corp. Santa Monica CA (1956)

Garey, M., Johnson, D.: Computers and intractability: a guide to the theory of np-completeness. W.H. Freeman and Co., New York pp. 90–91 (1979)

Gavish, B., Srikanth, K.: An optimal solution method for large-scale multiple traveling salesmen problems. Operations Research **34**(5), 698–717 (1986)

Glover, F.: Future paths for integer programming and links to artificial intelligence. Computers & operations research **13**(5), 533–549 (1986)

Glover, F., Gutin, G., Yeo, A., Zverovich, A.: Construction heuristics for the asymmetric TSP. European Journal of Operational Research **129**(3), 555–568 (2001)

Glover, F., Kochenberger, G.A.: Handbook of metaheuristics, volume 57 of international series in operations research & management science (2003)

Hamacher, H.W., Labbé, M., Nickel, S., Sonneborn, T.: Adapting polyhedral properties from facility to hub location problems. Discrete Applied Mathematics **145**(1), 104–116 (2004)

Hanafi, S., Lazić, J., Mladenović, N., Wilbaut, C., Crévits, I.: Hybrid variable neighbourhood decomposition search for 0-1 mixed integer programming problem. Electronic Notes in Discrete Mathematics **36**, 883–890 (2010)

Hansen, P., Mladenovic, N.: A tutorial on variable neighborhood search. Groupe d'études et de recherche en analyse des décisions, HEC Montréal (2003)

Hansen, P., Mladenovic, N., Brimberg, J., Moreno-Pérez, J.: Variable neighbourhood search, Handbook of Metaheuristics, Gendreau, M., Potvin, J.-Y. International Series in Operations Research & Management Sciences **146**, 61–86 (2010)

Hansen, P., Mladenović, N., Todosijević, R., Hanafi, S.: Variable neighborhood search: basics and variants. EURO Journal on Computational Optimization pp. 1–32 (2016)

Hekmatfar, M., Pishvaee, M.: Hub location problem. In: Facility Location, pp. 243–270. Springer (2009)

Henningsson, M., Holmberg, K., Yuan, D.: Ring network design. Handbook of Optimization in Telecommunications pp. 291–311 (2006)

Holland, J.H.: Adaptation in natural and artificial systems. 1975. Ann Arbor, MI, University of Michigan Press and (1992)

Horner, M.W., O'Kelly, M.E.: Embedding economies of scale concepts for hub network design. Journal of Transport Geography **9**(4), 255–265 (2001)

Johnson, D.B.: Efficient algorithms for shortest paths in sparse networks. Journal of the ACM (JACM) **24**(1), 1–13 (1977)

Kang, D., Lee, K., Park, S., Park, K., Kim, S.B.: Design of local networks using ushrs. Telecommunication Systems **14**(1-4), 197–217 (2000)

Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks IV, pages, vol. 1000 (1995)

Kim, J.G., Tcha, D.W.: Optimal design of a two-level hierarchical network with tree-star configuration. Computers & industrial engineering **22**(3), 273–281 (1992)

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simmulated annealing. science **220**(4598), 671–680 (1983)

Laporte, G.: The traveling salesman problem: An overview of exact and approximate algorithms. European Journal of Operational Research **59**(2), 231–247 (1992)

Lee, Y., Lim, B.H., Park, J.S.: A hub location problem in designing digital data service networks: Lagrangian relaxation approach. Location Science **4**(3), 185–194 (1996)

Lejeune, M.A.: A variable neighborhood decomposition search method for supply chain management planning problems. European Journal of Operational Research **175**(2), 959–976 (2006)

Li, S.: A 1.488 approximation algorithm for the uncapacitated facility location problem. Information and Computation **222**, 45–58 (2013)

Lin, C.C.: The integrated secondary route network design model in the hierarchical hub-and-spoke network for dual express services. International Journal of Production Economics **123**(1), 20–30 (2010)

Lopes, M.C., de Andrade, C.E., de Queiroz, T.A., Resende, M.G., Miyazawa, F.K.: Heuristics for a hub location-routing problem. Networks **68**(1), 54–90 (2016)

Lourenço, H.R., Martin, O.C., Stützle, T.: Iterated local search. In: Handbook of metaheuristics, pp. 320–353. Springer (2003)

Mahmutoğulları, A.İ., Kara, B.Y.: Hub location problem with allowed routing between nonhub nodes. Geographical Analysis **47**(4), 410–430 (2015)

Marianov, V., Serra, D., ReVelle, C.: Location of hubs in a competitive environment. European Journal of Operational Research **114**(2), 363–371 (1999)

Martí, R., Corberán, Á., Peiró, J.: Scatter search for an uncapacitated p-hub median problem. Computers & Operations Research **58**, 53–66 (2015)

Martí, R., Reinelt, G.: The linear ordering problem: exact and heuristic methods in combinatorial optimization, vol. 175. Springer Science & Business Media (2011)

Martínez, J.C.V., Fransoo, J.C.: Green facility location. In: Sustainable Supply Chains, pp. 219–234. Springer (2017)

Meng, Q., Wang, S.: Liner shipping service network design with empty container repositioning. Transportation Research Part E: Logistics and Transportation Review **47**(5), 695–708 (2011a)

Meng, Q., Wang, X.: Intermodal hub-and-spoke network design: incorporating multiple stakeholders and multi-type containers. Transportation Research Part B: Methodological **45**(4), 724–742 (2011b)

Mladenović, N., Brimberg, J., Hansen, P., Moreno-Pérez, J.A.: The p-median problem: A survey of metaheuristic approaches. European Journal of Operational Research **179**(3), 927–939 (2007)

Mladenović, N., Hansen, P.: Variable neighborhood search. Computers & operations research **24**(11), 1097–1100 (1997)

Mulder, J., Dekker, R.: Methods for strategic liner shipping network design. European Journal of Operational Research **235**(2), 367–377 (2014)

Nagy, G., Salhi, S.: The many-to-many location-routing problem. Top **6**(2), 261–275 (1998)

Nagy, G., Salhi, S.: Location-routing: Issues, models and methods. European Journal of Operational Research **177**(2), 649–672 (2007)

Nickel, S., Schöbel, A., Sonneborn, T.: Hub location problems in urban traffic networks. Mathematical methods on optimization in transportation systems pp. 95–107 (2001)

O'Kelly, M.E.: The location of interacting hub facilities. Transportation science **20**(2), 92–106 (1986)

O'Kelly, M.E.: A quadratic integer program for the location of interacting hub facilities. European Journal of Operational Research **32**(3), 393–404 (1987)

O'Kelly, M.E.: Hub facility location with fixed costs. Papers in Regional Science **71**(3), 293–306 (1992)

O'Kelly, M.E.: A geographer's analysis of hub-and-spoke networks. Journal of transport Geography **6**(3), 171–186 (1998)

O'Kelly, M.E., Bryan, D., Skorin-Kapov, D., Skorin-Kapov, J.: Hub network design with single and multiple allocation: A computational study. Location Science **4**(3), 125–138 (1996)

O'Kelly, M.E., Miller, H.J.: The hub network design problem: a review and synthesis. Journal of Transport Geography **2**(1), 31–40 (1994)

Osman, I.H., Kelly, J.P.: Meta-heuristics: an overview. In: Meta-Heuristics, pp. 1–21. Springer (1996)

Parejo, J.A., Ruiz-Cortés, A., Lozano, S., Fernandez, P.: Metaheuristic optimization frameworks: a survey and benchmarking. Soft Computing **16**(3), 527–561 (2011)

Parvaresh, F., Husseini, S.M., Golpayegany, S.H., Karimi, B.: Hub network design problem in the presence of disruptions. Journal of Intelligent Manufacturing **25**(4), 755–774 (2014)

Peiró, J., Corberán, Á., Martí, R.: Grasp for the uncapacitated r-allocation p-hub median problem. Computers & Operations Research **43**, 50–60 (2014)

Perboli, G., Tadei, R.: New families of valid inequalities for the two-echelon vehicle routing problem. Electronic notes in discrete mathematics **36**, 639–646 (2010)

Plum, C.E., Pisinger, D., Salazar-González, J.J., Sigurd, M.M.: Single liner shipping service design. Computers & Operations Research **45**, 1–6 (2014a)

Plum, C.E., Pisinger, D., Sigurd, M.M.: A service flow model for the liner shipping network design problem. European Journal of Operational Research **235**(2), 378–386 (2014b)

Plum, C.E.M.: Optimization of Container Line Networks with Flexible Demands. DTU Management Engineering (2013)

Rabelo, L.C., Jones, A.: Job shop scheduling. In: Encyclopedia of Operations Research and Management Science, pp. 817–830. Springer (2013)

Reese, J.: Methods for solving the p-median problem: An annotated bibliography, trinity university, mathematics faculty research, paper 28. Available in http://digitalcommons. trinity. edu/math faculty/28 (2005)

Resende, M.G., Pardalos, P.M.: Handbook of optimization in telecommunications. Springer Science & Business Media (2008)

Ribeiro, C.C., Martins, S.L., Rosseti, I.: Metaheuristics for optimization problems in computer communications. Computer Communications **30**(4), 656–669 (2007)

Rieck, J., Ehrenberg, C., Zimmermann, J.: Many-to-many location-routing with inter-hub transport and multi-commodity pickup-and-delivery. European Journal of Operational Research **236**(3), 863–878 (2014)

Rodrigue, J.P., Comtois, C., Slack, B.: The geography of transport systems. Taylor & Francis (2017)

Rodríguez-Martín, I., Salazar-González, J.J., Yaman, H.: A branch-and-cut algorithm for the hub location and routing problem. Computers & Operations Research **50**, 161–174 (2014)

Rodríguez-Martín, I., Salazar-González, J.J., Yaman, H.: Hierarchical survivable network design problems. Electronic Notes in Discrete Mathematics **52**, 229–236 (2016a)

Rodríguez-Martín, I., Salazar-González, J.J., Yaman, H.: The ring/κ-rings network design problem: Model and branch-and-cut algorithm. Networks **68**(2), 130–140 (2016b)

de Sá, E.M., de Camargo, R.S., de Miranda, G.: An improved Benders decomposition algorithm for the tree of hubs location problem. European Journal of Operational Research **226**(2), 185–202 (2013)

Saboury, A., Ghaffari-Nasab, N., Barzinpour, F., Jabalameli, M.S.: Applying two efficient hybrid heuristics for hub location problem with fully interconnected backbone and access networks. Computers & Operations Research **40**(10), 2493–2507 (2013)

Şahin, G., Süral, H.: A review of hierarchical facility location models. Computers & Operations Research **34**(8), 2310–2331 (2007)

Shi, J., Fonseka, J.P.: Design of hierarchical self-healing ring networks. In: Communications, 1994. ICC'94, SUPERCOMM/ICC'94, Conference Record,'Serving Humanity Through Communications.'IEEE International Conference on, pp. 478–482. IEEE (1994)

Skorin-Kapov, D., Skorin-Kapov, J., O'Kelly, M.E.: Tight linear programming relaxations of uncapacitated p-hub median problems. European Journal of Operational Research **94**(3), 582–593 (1996)

Sörensen, K., Glover, F.W.: Metaheuristics. In: Encyclopedia of operations research and management science, pp. 960–970. Springer (2013)

Sörensen, K., Sevaux, M., Glover, F.: A history of metaheuristics. In: ORBEL29-29th Belgian conference on Operations Research (2015)

Stahlbock, R., Voß, S.: Operations research at container terminals: a literature update. Or Spectrum **30**(1), 1–52 (2008)

Stopford, M.: Maritime Economics 3e. Routledge (2009)

Sun, J.U.: A hierarchical bio-inspired computing for the hub location-routing problem in parcel service. International Journal of Applied Engineering Research **11**(8), 5357–5362 (2016)

Sung, C., Jin, H.: Dual-based approach for a hub network design problem under non-restrictive policy. European Journal of Operational Research **132**(1), 88–105 (2001)

Talbi, E.G.: Metaheuristics: from design to implementation, vol. 74. John Wiley & Sons (2009)

Tamir, A.: The k-centrum multi-facility location problem. Discrete Applied Mathematics **109**(3), 293–307 (2001)

Thomadsen, T.: Hierarchical network design. PhD tesis in Informatics and Mathematical Modelling-Technical University of Denmark (2005)

Thomadsen, T., Larsen, J.: A hub location problem with fully interconnected backbone and access networks. Computers & operations research **34**(8), 2520–2531 (2007)

Torkestani, S.S., Seyedhosseini, S.M., Makui, A., Shahanaghi, K.: Hierarchical facility location and hub network problems: A literature review. Journal of Industrial and Systems Engineering **9**, 1–22 (2016)

Urošević, D., Brimberg, J., Mladenović, N.: Variable neighborhood decomposition search for the edge weighted k-cardinality tree problem. Computers & Operations Research **31**(8), 1205–1213 (2004)

Wagner, B.: An exact solution procedure for a cluster hub location problem. European journal of operational research **178**(2), 391–401 (2007)

Wang, S., Meng, Q.: Robust schedule design for liner shipping services. Transportation Research Part E: Logistics and Transportation Review **48**(6), 1093–1106 (2012)

Wang, S., Meng, Q.: Liner shipping network design with deadlines. Computers & Operations Research **41**, 140–149 (2014)

Wieberneit, N.: Service network design for freight transportation: a review. OR spectrum **30**(1), 77–112 (2008)

Yaman, H.: The hierarchical hub median problem with single assignment. Transportation Research Part B: Methodological **43**(6), 643–658 (2009)

Yaman, H.: Allocation strategies in hub networks. European Journal of Operational Research **211**(3), 442–451 (2011)

Yaman, H., Carello, G.: Solving the hub location problem with modular link capacities. Computers & operations research **32**(12), 3227–3245 (2005)

Zheng, J., Meng, Q., Sun, Z.: Impact analysis of maritime cabotage legislations on liner hub-and-spoke shipping network design. European Journal of Operational Research **234**(3), 874–884 (2014)

Appendices

Appendix A

Instances

This appendix presents, for each instance we defined, the cost or distance matrix and the demand matrix, respectively.

A.1 (Nine Nodes)

0	360.56	600.00	1200.00	1341.64	1513.27	1529.71	1700.00	1676.31
360.56	0	360.56	1044.03	1044.03	1392.84	1300.00	1392.84	1414.21
600.00	360.56	0	1341.64	1200.00	1700.00	1529.71	1513.27	1603.12
1200.00	1044.03	1341.64	0	600.00	360.56	424.26	854.40	640.31
1341.64	1044.03	1200.00	600.00	0	854.40	424.26	360.56	412.31
1513.27	1392.84	1700.00	360.56	854.40	0	500.00	1000.00	707.11
1529.71	1300.00	1529.71	424.26	424.26	500.00	0	500.00	223.61
1700.00	1392.84	1513.27	854.40	360.56	1000.00	500.00	0	316.23
1676.31	1414.21	1603.12	640.31	412.31	707.11	223.61	316.23	0

0	5	2	10	3	8	10	7	3
15	0	3	6	12	11	15	5	4
2	3	0	4	8	10	13	10	7
5	6	4	0	5	8	3	4	6
3	4	9	5	0	7	13	2	5
3	7	12	6	4	0	6	9	4
7	4	3	2	6	5	0	5	11
2	5	7	5	4	6	5	0	12
12	3	6	4	7	4	8	4	0

A.2 (Ten Nodes)

0	1204.16	282.84	905.54	1403.57	1216.55	2220.36	2376.97	2502.00	2570.99
1204.16	0	1044.03	360.56	282.84	300.00	2469.82	2319.48	2823.12	2668.33
282.84	1044.03	0	707.11	1204.16	1000.00	2002.50	2118.96	2302.17	2334.52
905.54	360.56	707.11	0	500.00	316.23	2184.03	2100.00	2529.82	2418.68
1403.57	282.84	1204.16	500.00	0	223.61	2370.65	2158.70	2729.47	2529.82
1216.55	300.00	1000.00	316.23	223.61	0	2193.17	2022.37	2549.51	2376.97
2220.36	2469.82	2002.50	2184.03	2370.65	2193.17	0	600.00	360.56	424.26
2376.97	2319.48	2118.96	2100.00	2158.70	2022.37	600.00	0	854.40	424.26
2502.00	2823.12	2302.17	2529.82	2729.47	2549.51	360.56	854.40	0	500.00
2570.99	2668.33	2334.52	2418.68	2529.82	2376.97	424.26	424.26	500.00	0

0	3	4	7	8	6	3	7	5	10
6	0	2	10	8	3	5	5	6	9
4	1	0	10	10	5	7	9	4	8
8	6	7	0	7	5	10	1	6	2
6	9	10	7	0	1	3	10	9	5
3	8	6	6	2	0	6	2	4	8
4	4	10	3	2	3	0	1	1	7
5	2	3	9	6	1	8	0	9	4
5	2	9	8	6	4	1	5	0	10
1	3	5	5	6	9	4	7	9	0

A.3 (Eleven Nodes)

0	509.90	1000.00	509.90	360.56	854.40	1900.00	2195.45	2236.07	2507.99	2555.39
509.90	0	509.90	200.00	500.00	500.00	2061.55	2088.06	2302.17	2451.53	2435.16
1000.00	509.90	0	509.90	854.40	360.56	2147.09	1902.63	2280.35	2300.00	2220.36
509.90	200.00	509.90	0	360.56	360.56	1868.15	1897.37	2102.38	2256.10	2247.22
360.56	500.00	854.40	360.56	0	600.00	1612.45	1835.76	1910.50	2154.07	2195.45
854.40	500.00	360.56	360.56	600.00	0	1788.85	1627.88	1941.65	2009.98	1964.69
1900.00	2061.55	2147.09	1868.15	1612.45	1788.85	0	1100.00	500.00	1077.03	1334.17
2195.45	2088.06	1902.63	1897.37	1835.76	1627.88	1100.00	0	761.58	412.31	360.56
2236.07	2302.17	2280.35	2102.38	1910.50	1941.65	500.00	761.58	0	608.28	900.00
2507.99	2451.53	2300.00	2256.10	2154.07	2009.98	1077.03	412.31	608.28	0	316.23
2555.39	2435.16	2220.36	2247.22	2195.45	1964.69	1334.17	360.56	900.00	316.23	0

0	6	2	6	5	6	10	10	3	3	5
2	0	6	9	8	8	3	8	10	4	10
6	2	0	9	1	7	1	1	8	3	1
3	9	9	0	3	3	3	7	1	7	2
6	8	3	9	0	2	6	9	5	7	4
6	1	7	4	8	0	2	5	8	8	9
2	2	1	8	5	2	0	3	2	2	6
10	5	6	8	9	8	6	0	3	5	3
5	6	8	2	7	10	9	3	0	1	8
8	10	8	2	9	3	10	5	9	0	7
3	6	7	1	4	9	6	6	9	6	0

A.4 (Twelve Nodes)

0	400.00	800.00	360.56	670.82	1414.21	1523.15	1700.00	1746.42	1878.83	1897.37	608.28
400.00	0	400.00	360.56	360.56	1414.21	1414.21	1746.42	1700.00	1746.42	1811.08	223.61
800.00	400.00	0	670.82	360.56	1523.15	1414.21	1878.83	1746.42	1700.00	1811.08	223.61
360.56	360.56	670.82	0	400.00	1100.00	1170.47	1414.21	1414.21	1523.15	1552.42	565.69
670.82	360.56	360.56	400.00	0	1170.47	1100.00	1523.15	1414.21	1414.21	1500.00	400.00
1414.21	1414.21	1523.15	1100.00	1170.47	0	400.00	360.56	360.56	670.82	565.69	1552.42
1523.15	1414.21	1414.21	1170.47	1100.00	400.00	0	670.82	360.56	360.56	400.00	1500.00
1700.00	1746.42	1878.83	1414.21	1523.15	360.56	670.82	0	400.00	800.00	608.28	1897.37
1746.42	1700.00	1746.42	1414.21	1414.21	360.56	360.56	400.00	0	400.00	223.61	1811.08
1878.83	1746.42	1700.00	1523.15	1414.21	670.82	360.56	800.00	400.00	0	223.61	1811.08
1897.37	1811.08	1811.08	1552.42	1500.00	565.69	400.00	608.28	223.61	223.61	0	1900.00
608.28	223.61	223.61	565.69	400.00	1552.42	1500.00	1897.37	1811.08	1811.08	1900.00	0

0	5	2	10	3	8	10	7	6	5	11	7
15	0	3	6	12	11	15	5	2	11	12	9
2	3	0	4	8	10	13	10	12	2	7	3
5	6	4	0	5	8	3	4	3	5	8	9
3	4	9	5	0	7	13	2	6	3	3	12
3	7	12	6	4	0	6	9	2	1	9	7
7	4	3	2	6	5	0	5	7	13	8	6
2	5	7	5	4	6	5	0	3	8	7	4
3	5	2	3	5	12	7	3	0	5	6	8
5	8	9	3	3	5	4	6	4	0	5	6
7	4	6	10	4	7	3	9	12	2	0	5
3	5	10	11	4	6	3	9	9	5	13	0

Appendix B

Report of Exact Method

<u>**Model with Circular Hub Route and Sub-hubs**</u>

Report of exact method for computational test for instance 9_3_0.3.

Network Generated:

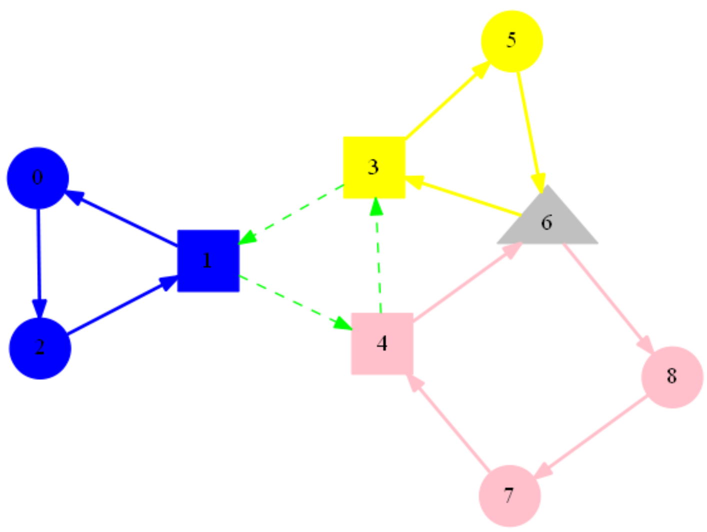


Figure B.1: Network generated by exact method for instance 9_3_0.3.

Routing:

Optimal solution: 463198

x[i][j]: THE NODE i IS ALLOCATED TO NODE j

x[0][1] =1

x[1][1] =1

x[2][1] =1

x[3][3] =1

x[4][4] =1

x[5][3] =1

x[6][3] =1

x[6][4] =1
x[7][4] =1
x[8][4] =1
x[j][j]: j IS A HUB WITH ALLOCATION OR u[j]: j IS AN ISOLATED HUB
x[1][1] =1
x[3][3] =1
x[4][4] =1
t[j]: j IS A SUB_HUB
t[6] =1
x[i][j]: i IS A SPOKE
x[0][1] =1
x[2][1] =1
x[5][3] =1
x[6][3] =1
x[6][4] =1
x[7][4] =1
x[8][4] =1
y[i][j]: THERE ARE AN ARC_NON_HUB BETWEEN THE NODES i AND j
y[0][2] =1
y[1][0] =1
y[2][1] =1
y[3][5] =1
y[4][6] =1
y[5][6] =1
y[6][3] =1
y[6][8] =1
y[7][4] =1
y[8][7] =1
b[i][j]: THERE ARE AN ARC_HUB BETWEEN THE HUB_NODES i AND j
b[1][4] =1
b[3][1] =1
b[4][3] =1
FLOW BETWEEN THE ORIGIN 0 AND THE DESTINATION 1
fl[0][1] * s[0][1][0]][2] =5
fl[0][1] * s[0][1][2]][1] =5
FLOW BETWEEN THE ORIGIN 0 AND THE DESTINATION 2
fl[0][2] * s[0][2][0]][2] =2
FLOW BETWEEN THE ORIGIN 0 AND THE DESTINATION 3

fl[0][3] * s[0][3][0]][2] =10

fl[0][3] * w[0][3][1]][4] =10 ← 1-4 is an arc hub

fl[0][3] * s[0][3][2]][1] =10

fl[0][3] * w[0][3][4]][3] =10 ← 4-3 is an arc hub

FLOW BETWEEN THE ORIGIN 0 AND THE DESTINATION 4

fl[0][4] * s[0][4][0]][2] =3

fl[0][4] * w[0][4][1]][4] =3 ← 1-4 is an arc hub

fl[0][4] * s[0][4][2]][1] =3

FLOW BETWEEN THE ORIGIN 0 AND THE DESTINATION 5

fl[0][5] * s[0][5][0]][2] =8

fl[0][5] * w[0][5][1]][4] =8 ← 1-4 is an arc hub

fl[0][5] * s[0][5][2]][1] =8

fl[0][5] * s[0][5][3]][5] =8

fl[0][5] * w[0][5][4]][3] =8 ← 4-3 is an arc hub

FLOW BETWEEN THE ORIGIN 0 AND THE DESTINATION 6

fl[0][6] * s[0][6][0]][2] =10

fl[0][6] * w[0][6][1]][4] =10 ← 1-4 is an arc hub

fl[0][6] * s[0][6][2]][1] =10

fl[0][6] * s[0][6][4]][6] =10

FLOW BETWEEN THE ORIGIN 0 AND THE DESTINATION 7

fl[0][7] * s[0][7][0]][2] =7

fl[0][7] * w[0][7][1]][4] =7 ← 1-4 is an arc hub

fl[0][7] * s[0][7][2]][1] =7

fl[0][7] * s[0][7][4]][6] =7

fl[0][7] * s[0][7][6]][8] =7

fl[0][7] * s[0][7][8]][7] =7

FLOW BETWEEN THE ORIGIN 0 AND THE DESTINATION 8

fl[0][8] * s[0][8][0]][2] =3

fl[0][8] * w[0][8][1]][4] =3 ← 1-4 is an arc hub

fl[0][8] * s[0][8][2]][1] =3

fl[0][8] * s[0][8][4]][6] =3

fl[0][8] * s[0][8][6]][8] =3

FLOW BETWEEN THE ORIGIN 1 AND THE DESTINATION 0

fl[1][0] * s[1][0][1]][0] =15

FLOW BETWEEN THE ORIGIN 1 AND THE DESTINATION 2

fl[1][2] * s[1][2][0]][2] =3

fl[1][2] * s[1][2][1]][0] =3

FLOW BETWEEN THE ORIGIN 1 AND THE DESTINATION 3

fl[1][3] * w[1][3][1]][4] =6 ← 1-4 is an arc hub

fl[1][3] * w[1][3][4]][3] =6 ← 4-3 is an arc hub

FLOW BETWEEN THE ORIGIN 1 AND THE DESTINATION 4

fl[1][4] * w[1][4][1]][4] =12 ← 1-4 is an arc hub

FLOW BETWEEN THE ORIGIN 1 AND THE DESTINATION 5

fl[1][5] * w[1][5][1]][4] =11 ← 1-4 is an arc hub

fl[1][5] * s[1][5][3]][5] =11

fl[1][5] * w[1][5][4]][3] =11 ← 4-3 is an arc hub

FLOW BETWEEN THE ORIGIN 1 AND THE DESTINATION 6

fl[1][6] * w[1][6][1]][4] =15 ← 1-4 is an arc hub

fl[1][6] * s[1][6][4]][6] =15

FLOW BETWEEN THE ORIGIN 1 AND THE DESTINATION 7

fl[1][7] * w[1][7][1]][4] =5 ← 1-4 is an arc hub

fl[1][7] * s[1][7][4]][6] =5

fl[1][7] * s[1][7][6]][8] =5

fl[1][7] * s[1][7][8]][7] =5

FLOW BETWEEN THE ORIGIN 1 AND THE DESTINATION 8

fl[1][8] * w[1][8][1]][4] =4 ← 1-4 is an arc hub

fl[1][8] * s[1][8][4]][6] =4

fl[1][8] * s[1][8][6]][8] =4

FLOW BETWEEN THE ORIGIN 2 AND THE DESTINATION 0

fl[2][0] * s[2][0][1]][0] =2

fl[2][0] * s[2][0][2]][1] =2

FLOW BETWEEN THE ORIGIN 2 AND THE DESTINATION 1

fl[2][1] * s[2][1][2]][1] =3

FLOW BETWEEN THE ORIGIN 2 AND THE DESTINATION 3

fl[2][3] * w[2][3][1]][4] =4 ← 1-4 is an arc hub

fl[2][3] * s[2][3][2]][1] =4

fl[2][3] * w[2][3][4]][3] =4 ← 4-3 is an arc hub

FLOW BETWEEN THE ORIGIN 2 AND THE DESTINATION 4

fl[2][4] * w[2][4][1]][4] =8 ← 1-4 is an arc hub

fl[2][4] * s[2][4][2]][1] =8

FLOW BETWEEN THE ORIGIN 2 AND THE DESTINATION 5

fl[2][5] * w[2][5][1]][4] =10 ← 1-4 is an arc hub

fl[2][5] * s[2][5][2]][1] =10

fl[2][5] * s[2][5][3]][5] =10

fl[2][5] * w[2][5][4]][3] =10 ← 4-3 is an arc hub

FLOW BETWEEN THE ORIGIN 2 AND THE DESTINATION 6

fl[2][6] * w[2][6][1]][4] =13 ← 1-4 is an arc hub

fl[2][6] * s[2][6][2]][1] =13

fl[2][6] * s[2][6][4]][6] =13

FLOW BETWEEN THE ORIGIN 2 AND THE DESTINATION 7

fl[2][7] * w[2][7][1]][4] =10 ← 1-4 is an arc hub

fl[2][7] * s[2][7][2]][1] =10

fl[2][7] * s[2][7][4]][6] =10

fl[2][7] * s[2][7][6]][8] =10

fl[2][7] * s[2][7][8]][7] =10

FLOW BETWEEN THE ORIGIN 2 AND THE DESTINATION 8

fl[2][8] * w[2][8][1]][4] =7 ← 1-4 is an arc hub

fl[2][8] * s[2][8][2]][1] =7

fl[2][8] * s[2][8][4]][6] =7

fl[2][8] * s[2][8][6]][8] =7

FLOW BETWEEN THE ORIGIN 3 AND THE DESTINATION 0

fl[3][0] * s[3][0][1]][0] =5

fl[3][0] * w[3][0][3]][1] =5 ← 3-1 is an arc hub

FLOW BETWEEN THE ORIGIN 3 AND THE DESTINATION 1

fl[3][1] * w[3][1][3]][1] =6 ← 3-1 is an arc hub

FLOW BETWEEN THE ORIGIN 3 AND THE DESTINATION 2

fl[3][2] * s[3][2][0]][2] =4

fl[3][2] * s[3][2][1]][0] =4

fl[3][2] * w[3][2][3]][1] =4 ← 3-1 is an arc hub

FLOW BETWEEN THE ORIGIN 3 AND THE DESTINATION 4

fl[3][4] * w[3][4][1]][4] =5 ← 1-4 is an arc hub

fl[3][4] * w[3][4][3]][1] =5 ← 3-1 is an arc hub

FLOW BETWEEN THE ORIGIN 3 AND THE DESTINATION 5

fl[3][5] * s[3][5][3]][5] =8

FLOW BETWEEN THE ORIGIN 3 AND THE DESTINATION 6

fl[3][6] * s[3][6][3]][5] =3

fl[3][6] * s[3][6][5]][6] =3

FLOW BETWEEN THE ORIGIN 3 AND THE DESTINATION 7

fl[3][7] * s[3][7][3]][5] =4

fl[3][7] * s[3][7][5]][6] =4

fl[3][7] * s[3][7][6]][8] =4

fl[3][7] * s[3][7][8]][7] =4

FLOW BETWEEN THE ORIGIN 3 AND THE DESTINATION 8

fl[3][8] * s[3][8][3]][5] =6

fl[3][8] * s[3][8][5]][6] =6

fl[3][8] * s[3][8][6]][8] =6

FLOW BETWEEN THE ORIGIN 4 AND THE DESTINATION 0

fl[4][0] * s[4][0][1]][0] =3

fl[4][0] * w[4][0][3]][1] =3 ← 3-1 is an arc hub

fl[4][0] * w[4][0][4]][3] =3 ← 4-3 is an arc hub

FLOW BETWEEN THE ORIGIN 4 AND THE DESTINATION 1

fl[4][1] * w[4][1][3]][1] =4 ← 3-1 is an arc hub

fl[4][1] * w[4][1][4]][3] =4 ← 4-3 is an arc hub

FLOW BETWEEN THE ORIGIN 4 AND THE DESTINATION 2

fl[4][2] * s[4][2][0]][2] =9

fl[4][2] * s[4][2][1]][0] =9

fl[4][2] * w[4][2][3]][1] =9 ← 3-1 is an arc hub

fl[4][2] * w[4][2][4]][3] =9 ← 4-3 is an arc hub

FLOW BETWEEN THE ORIGIN 4 AND THE DESTINATION 3

fl[4][3] * w[4][3][4]][3] =5 ← 4-3 is an arc hub

FLOW BETWEEN THE ORIGIN 4 AND THE DESTINATION 5

fl[4][5] * s[4][5][3]][5] =7

fl[4][5] * w[4][5][4]][3] =7 ← 4-3 is an arc hub

FLOW BETWEEN THE ORIGIN 4 AND THE DESTINATION 6

fl[4][6] * s[4][6][4]][6] =13

FLOW BETWEEN THE ORIGIN 4 AND THE DESTINATION 7

fl[4][7] * s[4][7][4]][6] =2

fl[4][7] * s[4][7][6]][8] =2

fl[4][7] * s[4][7][8]][7] =2

FLOW BETWEEN THE ORIGIN 4 AND THE DESTINATION 8

fl[4][8] * s[4][8][4]][6] =5

fl[4][8] * s[4][8][6]][8] =5

FLOW BETWEEN THE ORIGIN 5 AND THE DESTINATION 0

fl[5][0] * s[5][0][1]][0] =3

fl[5][0] * w[5][0][3]][1] =3 ← 3-1 is an arc hub

fl[5][0] * s[5][0][5]][6] =3

fl[5][0] * s[5][0][6]][3] =3

FLOW BETWEEN THE ORIGIN 5 AND THE DESTINATION 1

fl[5][1] * w[5][1][3]][1] =7 ← 3-1 is an arc hub

fl[5][1] * s[5][1][5]][6] =7

fl[5][1] * s[5][1][6]][3] =7

FLOW BETWEEN THE ORIGIN 5 AND THE DESTINATION 2

fl[5][2] * s[5][2][0]][2] =12

fl[5][2] * s[5][2][1]][0] =12

fl[5][2] * w[5][2][3]][1] =12 ← 3-1 is an arc hub

fl[5][2] * s[5][2][5]][6] =12

fl[5][2] * s[5][2][6]][3] =12

FLOW BETWEEN THE ORIGIN 5 AND THE DESTINATION 3

fl[5][3] * s[5][3][5]][6] =6

fl[5][3] * s[5][3][6]][3] =6

FLOW BETWEEN THE ORIGIN 5 AND THE DESTINATION 4

fl[5][4] * s[5][4][5]][6] =4

fl[5][4] * s[5][4][6]][8] =4

fl[5][4] * s[5][4][7]][4] =4

fl[5][4] * s[5][4][8]][7] =4

FLOW BETWEEN THE ORIGIN 5 AND THE DESTINATION 6

fl[5][6] * s[5][6][5]][6] =6

FLOW BETWEEN THE ORIGIN 5 AND THE DESTINATION 7

fl[5][7] * s[5][7][5]][6] =9

fl[5][7] * s[5][7][6]][8] =9

fl[5][7] * s[5][7][8]][7] =9

FLOW BETWEEN THE ORIGIN 5 AND THE DESTINATION 8

fl[5][8] * s[5][8][5]][6] =4

fl[5][8] * s[5][8][6]][8] =4

FLOW BETWEEN THE ORIGIN 6 AND THE DESTINATION 0

fl[6][0] * s[6][0][1]][0] =7

fl[6][0] * w[6][0][3]][1] =7 ← 3-1 is an arc hub

fl[6][0] * s[6][0][6]][3] =7

FLOW BETWEEN THE ORIGIN 6 AND THE DESTINATION 1

fl[6][1] * w[6][1][3]][1] =4 ← 3-1 is an arc hub

fl[6][1] * s[6][1][6]][3] =4

FLOW BETWEEN THE ORIGIN 6 AND THE DESTINATION 2

fl[6][2] * s[6][2][0]][2] =3

fl[6][2] * s[6][2][1]][0] =3

fl[6][2] * w[6][2][3]][1] =3 ← 3-1 is an arc hub

fl[6][2] * s[6][2][6]][3] =3

FLOW BETWEEN THE ORIGIN 6 AND THE DESTINATION 3

fl[6][3] * s[6][3][6]][3] =2

FLOW BETWEEN THE ORIGIN 6 AND THE DESTINATION 4

fl[6][4] * s[6][4][6]][8] =6

fl[6][4] * s[6][4][7]][4] =6

fl[6][4] * s[6][4][8]][7] =6

FLOW BETWEEN THE ORIGIN 6 AND THE DESTINATION 5

fl[6][5] * s[6][5][3]][5] =5

fl[6][5] * s[6][5][6]][3] =5

FLOW BETWEEN THE ORIGIN 6 AND THE DESTINATION 7

fl[6][7] * s[6][7][6]][8] =5

fl[6][7] * s[6][7][8]][7] =5

FLOW BETWEEN THE ORIGIN 6 AND THE DESTINATION 8

fl[6][8] * s[6][8][6]][8] =11

FLOW BETWEEN THE ORIGIN 7 AND THE DESTINATION 0

fl[7][0] * s[7][0][1]][0] =2

fl[7][0] * w[7][0][3]][1] =2 ← 3-1 is an arc hub

fl[7][0] * w[7][0][4]][3] =2 ← 4-3 is an arc hub

fl[7][0] * s[7][0][7]][4] =2

FLOW BETWEEN THE ORIGIN 7 AND THE DESTINATION 1

fl[7][1] * w[7][1][3]][1] =5 ← 3-1 is an arc hub

fl[7][1] * w[7][1][4]][3] =5 ← 4-3 is an arc hub

fl[7][1] * s[7][1][7]][4] =5

FLOW BETWEEN THE ORIGIN 7 AND THE DESTINATION 2

fl[7][2] * s[7][2][0]][2] =7

fl[7][2] * s[7][2][1]][0] =7

fl[7][2] * w[7][2][3]][1] =7 ← 3-1 is an arc hub

fl[7][2] * w[7][2][4]][3] =7 ← 4-3 is an arc hub

fl[7][2] * s[7][2][7]][4] =7

FLOW BETWEEN THE ORIGIN 7 AND THE DESTINATION 3

fl[7][3] * w[7][3][4]][3] =5 ← 4-3 is an arc hub

fl[7][3] * s[7][3][7]][4] =5

FLOW BETWEEN THE ORIGIN 7 AND THE DESTINATION 4

fl[7][4] * s[7][4][7]][4] =4

FLOW BETWEEN THE ORIGIN 7 AND THE DESTINATION 5

fl[7][5] * s[7][5][3]][5] =6

fl[7][5] * w[7][5][4]][3] =6 ← 4-3 is an arc hub

fl[7][5] * s[7][5][7]][4] =6

FLOW BETWEEN THE ORIGIN 7 AND THE DESTINATION 6

fl[7][6] * s[7][6][4]][6] =5

fl[7][6] * s[7][6][7]][4] =5

FLOW BETWEEN THE ORIGIN 7 AND THE DESTINATION 8

fl[7][8] * s[7][8][4]][6] =12

fl[7][8] * s[7][8][6]][8] =12

fl[7][8] * s[7][8][7]][4] =12

FLOW BETWEEN THE ORIGIN 8 AND THE DESTINATION 0

fl[8][0] * s[8][0][1]][0] =12

fl[8][0] * w[8][0][3]][1] =12 ← 3-1 is an arc hub

fl[8][0] * w[8][0][4]][3] =12 ← 4-3 is an arc hub

fl[8][0] * s[8][0][7]][4] =12

fl[8][0] * s[8][0][8]][7] =12

FLOW BETWEEN THE ORIGIN 8 AND THE DESTINATION 1

fl[8][1] * w[8][1][3]][1] =3 ← 3-1 is an arc hub

fl[8][1] * w[8][1][4]][3] =3 ← 4-3 is an arc hub

fl[8][1] * s[8][1][7]][4] =3

fl[8][1] * s[8][1][8]][7] =3

FLOW BETWEEN THE ORIGIN 8 AND THE DESTINATION 2

fl[8][2] * s[8][2][0]][2] =6

fl[8][2] * s[8][2][1]][0] =6

fl[8][2] * w[8][2][3]][1] =6 ← 3-1 is an arc hub

fl[8][2] * w[8][2][4]][3] =6 ← 4-3 is an arc hub

fl[8][2] * s[8][2][7]][4] =6

fl[8][2] * s[8][2][8]][7] =6

FLOW BETWEEN THE ORIGIN 8 AND THE DESTINATION 3

fl[8][3] * w[8][3][4]][3] =4 ← 4-3 is an arc hub

fl[8][3] * s[8][3][7]][4] =4

fl[8][3] * s[8][3][8]][7] =4

FLOW BETWEEN THE ORIGIN 8 AND THE DESTINATION 4

fl[8][4] * s[8][4][7]][4] =7

fl[8][4] * s[8][4][8]][7] =7

FLOW BETWEEN THE ORIGIN 8 AND THE DESTINATION 5

fl[8][5] * s[8][5][3]][5] =4

fl[8][5] * w[8][5][4]][3] =4 ← 4-3 is an arc hub

fl[8][5] * s[8][5][7]][4] =4

fl[8][5] * s[8][5][8]][7] =4

FLOW BETWEEN THE ORIGIN 8 AND THE DESTINATION 6

fl[8][6] * s[8][6][4]][6] =8

fl[8][6] * s[8][6][7]][4] =8

fl[8][6] * s[8][6][8]][7] =8

FLOW BETWEEN THE ORIGIN 8 AND THE DESTINATION 7

fl[8][7] * s[8][7][8]][7] =4

Appendix C

Data Structure

The data structure used for representing a solution is composed by dynamic 1 dimensional arrays and dynamic 2 dimensional arrays:

C.1 CLUSTER_WITH_SUB-HUB - int[ ][ ]

In this array, the number of each node is recorded.

The number of clusters is defined by the number of hubs (p), because each cluster must be constituted by only one hub.

In this array, each cluster is represented by one line (see Figure (C.1).

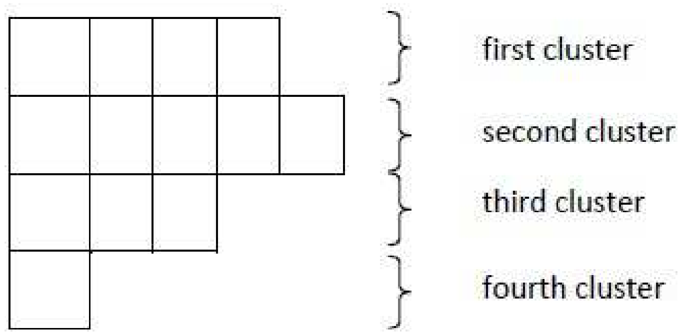


Figure C.1: CLUSTER_WITH_SUB-HUB array composed of 4 clusters

where the configuration of each cluster is:

- first item: number of nodes used as hub;
- second item until last one: number of nodes used as spokes.

In our problem, it is possible to consider an isolated hub (see Figure (C.1), fourth cluster) that is to say a cluster just composed by only one item.

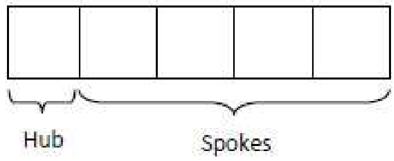


Figure C.2: One line of CLUSTER_WITH_SUB-HUB array

Example of an instance with 10 nodes and 4 hubs could have, initially, a solution represented by Figure (C.3) below.

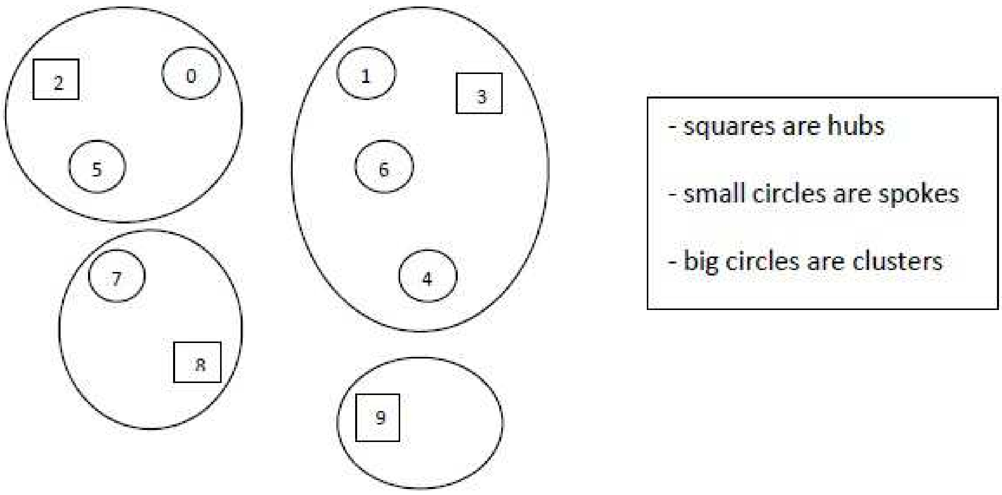


Figure C.3: Clusters composed of hubs and spokes

Figure (C.4) represents the CLUSTER_WITH_SUB-HUB array according to Figure (C.3).

2	0	5	
3	1	4	6
8	7		
9			

Figure C.4: CLUSTER_WITH_SUB-HUB array corresponding to Figure (C.3)

In algorithm (10), some spoke nodes are chosen as sub-hubs and they are included in a second cluster. In other words, sub-hubs nodes are intersection points between two close clusters (Figure C.5).

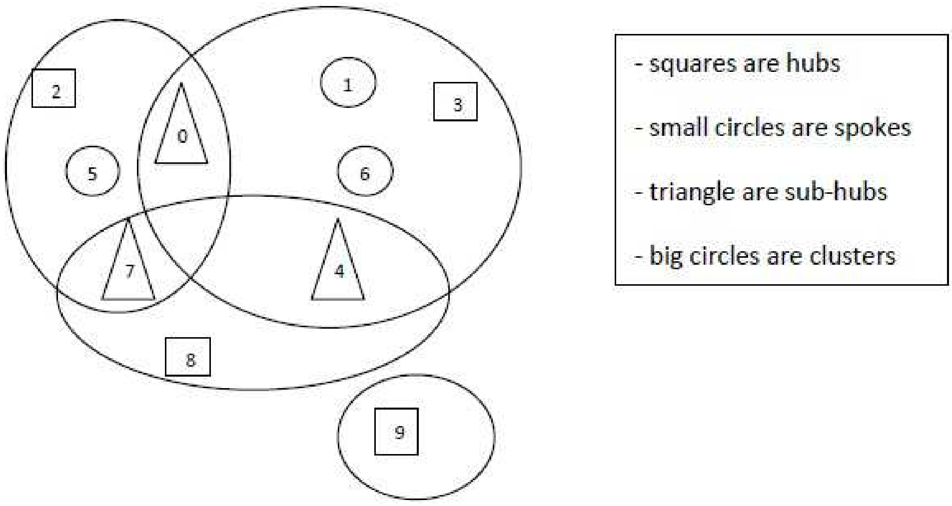


Figure C.5: Close clusters with sub-hubs

Figure (C.6) represents the CLUSTER_WITH_SUB-HUB array according the Figure (C.5).

Some lines in CLUSTER_WITH_SUB-HUB array can be increased in the algorithm (10) (choose sub-hub).

2	5	0	7	
3	1	6	0	4
8	7	4		
9				

Figure C.6: CLUSTER_WITH_SUB-HUB array corresponding to Figure (C.5)

Where in Figure (C.6), the close clusters and respective sub-hubs are:

- clusters 0 and 1 with sub-hub 0;
- clusters 0 and 2 with sub-hub 7;
- cluster 1 and 2 with sub-hub 4.

In this example (Figure C.6), it is possible to highlight that sub-hubs were inserted in the last position of each line and in the ordered sequence that each pair close clusters is analyzed (first cluster with the second, and first cluster with the third. After that, second cluster with the third etc...).

In *improve_hub* (see algorithm 17 and algorithm 25), one spoke node of a cluster can be chosen as hub and the position between the hub and the spoke node chosen is switched.

Note that each line from CLUSTER_WITH_SUB-HUB array can be increased or decreased with *improve_sub_hub* (see algorithm 17), shake *remove_sub_hub* (algorithm 17), *spoke_nodes_allocations* (see algorithm (23)) and *shift_spoke_node* shake-operator (see algorithm (24)).

For the example above (Figure C.6), one sequence of new solutions could be:

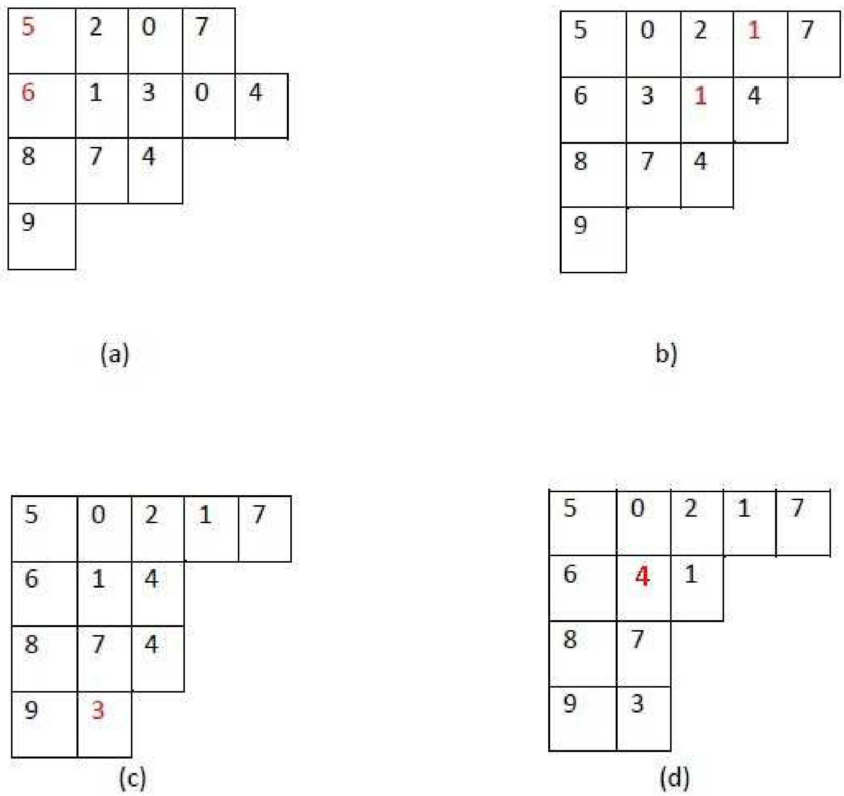


Figure C.7: Procedures executed in *improve_hub* (a), *improve_sub_hub* (b), *shift_spoke_node* or *spoke_nodes_allocations* (c) and *remove_sub_hub* (d).

Figure (C.7) is a possible sequence of procedures executed:

- In Figure (C.7)(a), nodes 5 and 6 became hubs through *improve_hub*. Sub-hub node can not be choose as hub;

- In Figure (C.7)(b), node 0 is no longer a sub-hub and node 1 becomes a sub-hub. Node 0 is removed from cluster 1. In its original cluster (cluster 0), node 0 is inserted in right side of hub. Whereas, node 1 is inserted between sub-hubs (cluster 0 and cluster 1), in the position of the previous one. These procedures can be done by *improve_sub_hub* because the node 1 was a spoke node.

- In Figure (C.7)(c), spoke node 3 is removed from cluster 1 and it is inserted in the cluster 3. This procedure can be done by *spoke_nodes_allocations* and *shift_spoke_node* only for spoke node.
- In Figure (C.7)(d), sub-hub 4 is deleted by *remove_sub_hub*. Spoke node 4 is inserted on the right side from the hub of its original cluster.