UNIVERSIDADE FEDERAL FLUMINENSE
UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE

Pedro Henrique González Silva

Studies on Network Design Problems

NITERÓI

2015

UNIVERSIDADE FEDERAL FLUMINENSE
UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE

Pedro Henrique González Silva

Studies on Network Design Problems

Thesis presented to the Computing Graduate Program of the Universidade Federal Fluminense in partial fulfillment of the requirements for the degree of Doctor of Science. Topic Area: Algorithms and Optimization.

Advisor:
Carlos Alberto de Jesus Martinhon

Advisor:
Philippe Yves Paul Michelon

Co-Advisor:
Luidi Gelabert Simonetti

NITERÓI

2015

Pedro Henrique González Silva

Studies on Network Design Problems

Thesis presented to the Computing Graduate Program of the Universidade Federal Fluminense and the Laboratoire d'Informatique d'Avignon of the Université d'Avignon et des Pays de Vaucluse as partial fulfilment of the requirements for the degree of Doctor of Science. Subject Area: Algorithms and Optimization.

Approved by:

Prof. D.Sc. Carlos Alberto de Jesus Martinhon, UFF

Prof. D.Sc. Philippe Yves Paul Michelon, UAPV

Prof. D.Sc. Luidi Gelabert Simonetti, UFF

Prof. D.Sc. Luiz Satoru Ochi, UFF

Prof. D.Sc. Simone de Lima Martins, UFF

Prof. D.Sc. Van Dat Cung, Grenoble INP
(Rapporteur)

Prof. Ph.D. Abilio Pereira de Lucena Filho, UFRJ
(Rapporteur)

Niterói, September 3, 2015

Acknowledgements

First of all, thanks to my parents Gisela and Nelio, my brother Diogo and my love Mariana for all the support during the development of this thesis.
Thanks to my grandparents Judith and Pedro for all the support and incentive they gave me in my whole life and for always reminding me to go further.
I want to thank my advisors Carlos Martinhon, Philippe Michelon and Luidi Simonetti for believing in me and in this work, and also for all the scientific and personal advices during this journey.
Thanks to my friends, the new and the old ones, without you this work would have taken more time and I would have had half the fun I had. A special thanks to Pablo and Israel because their presence in Avignon made all the difference.
At last, thanks to CAPES, who partially supported this research (process BEX 9877/13-4) and also to the Laboratoire d'Informatique d'Avignon.

Resumo

Esta tese trata de dois problemas de planejamento de redes por meio de técnicas exatas, metaheurísticas e híbridas. O primeiro problema aqui estudado é o Problema de Planejamento de Redes com Rotas Ótimas para o Usuário (FCNDP-UOF), que diz respeito ao roteamento de múltiplos produtos desde sua origem até o seu destino. Para realizar este roteamento uma rede é construída, minimizando a soma dos custos de adição dos arcos selecionados mais a soma dos custos variáveis associados aos fluxos em cada arco. Além disso, uma vez que o FCNDP-UOF é um problema de dois níveis, cada mercadoria tem que ser transportada por um caminho mais curto, relativo ao comprimento dos arcos, na rede construída. Para este problema formulações matemáticas existentes foram estudadas e tiveram a força de suas relaxações lineares comparada. Depois disso, uma nova heurística e dois novos métodos híbridos foram testados. Os experimentos computacionais mostram que os algoritmos propostos para o FCNDP-UOF funcionam muito bem superando o estado da arte do problema. O segundo problema estudado é o problema de Planejamento de Expansão de Redes de Transmissão com Redimensionamento (TEP_R), no qual dado um novo conjunto de demandas e uma rede inicial, consiste na adição ou remoção de linhas de transmissão, a fim de satisfazer as novas demandas impostas, minimizando o custo operacional. Dois métodos foram desenvolvidos. O primeiro é uma decomposição de Benders onde um conjunto de variáveis contínuas é permitido no problema mestre, melhorando assim o limite da relaxação inicial. O segundo, chamado Busca Particionada em Anéis, pode ser usado tanto como método exato quanto como heurística. Experimentos computacionais mostraram o impacto destes métodos em comparação com a aplicação direta da formulação matemática em um solver comercial.

Palavras Chave: Planejamento de Redes, Métodos Exatos, Metaheurísticas, Métodos Híbridos.

Abstract

This thesis deals with two network design problems by means of exact, metaheuristic and hybrid techniques. The first problem studied here is the Fixed Charge Uncapacitated Network Design Problem with User-optimal Flow (FCNDP-UOF), which concerns routing multiple commodities from its origin to its destination by designing a network through selecting arcs, with an objective of minimizing the sum of the fixed costs of the selected arcs plus the sum of variable costs associated to the flows on each arc. Besides that, since the FCNDP-UOF is a bi-level problem, each commodity has to be transported through a shortest path, concerning the length of the edges, in the built network. To this problem existent mathematical formulations were studied and had the strength of its linear relaxations compared. After that, new heuristics and two new hybrid methods were tested. Computational experiments show that the proposed algorithms for the FCNDP-UOF worked very well leading to a new state of the art method. The second problem studied is the Transmission Expansion Planning Problem with Redesign ($\mathrm{TEP_R}$), which given a new set of loads and an initial network, consists of adding or removing transmission lines in order to satisfy the new imposed loads, while minimizing the operational cost. Two methods have been developed. The first is a Benders decomposition where a set of continuous variables is allowed in the master problem, thereby improving the limit of the initial relaxation. The second, called Ring Partition Search, can be used either as exact method and heuristic. Computational experiments showed the impact of these methods compared to the direct application of mathematical formulation in a commercial solver.

Keywords: Network Design, Exact Methods, Metaheuristics, Hybrid Methods.

List of Figures

List of Tables

List of Algorithms

Contents

Résumé étendu

Ce travail trouve sa motivation dans le grand nombre d'applications liées aux problèmes de conception de réseau, ainsi que dans leur complexités. En particulier, nous nous focalison sur deux problèmes de conception de réseau, le Fixed Charge Uncapacitated Network Design Problem with User-optimal Flow (FCNDP-UOF) et le Transmission Expansion Planning Problem with Redesign (TEP_R). Bien qu'appartenant tout deux à la classe des problèmes de conception de réseau, ils ont des structures différentes et spécifiques qui les rendent intéressants.

Le FCNDP-UOF est relatif au transport de produits dans les grands centres urbains et peut être modélisé comme un problème de programmation linéaire discret à deux niveaux. Ce type de problème implique deux agents agissant simultanément plutôt que séquentiellement lors de la prise décisions. Au niveau supérieur, le leader est chargé de choisir un sous-ensemble d'arrêtes qui seront ouvertes afin de minimiser la somme des coûts fixes (d'ouverture d'arrête) et variable (de transport des commodités sur les arrêtes). Au niveau inférieur, le suiveur doit choisir un ensemble de plus courts chemins dans le réseau, par lesquels les produits seront envoyé. L'effet d'un agent sur l'autre est indirect: la décision du suiveur est affectée par le réseau conçu par le niveau supérieur, alors que la décision du leader est affectée par les coûts variables imposés par les chemins établis au niveau inférieur.

Le TEP_R est un problème permettant d'établir une stratégie d'expansion des réseaux de transport d'électricité en ajoutant ou supprimant des lignes de transmission. Au contraire des autres problèmes de conception de réseau, tels que les problème des transport public, de transport de marchandises (problème de tournées de véhicules), transport de données (conception de réseau de télécommunication), l'ajout d'une ligne de transmission peut rendre impraticable une configuration qui avant etait réalisable. Cette caractéristique est due au fait que le gestionnaire du réseau ne peut pas choisir la façon dont les lignes de transmission seront utilisées. Il ne peut agir que sur la répartition de la production et n'affecter qu'indirectement l'acheminement de l?énergie et ne peut que choisir les angles de voltage. Cette caracteristique rend le problème a la fois très difficile et très intérêssant.

L'objectif principal de cette thèse est d'étudier ces deux problèmes et de développer des algorithmes exacts, des métaheuristiques et des méthodes hybrides. Pour le premièr problème, on a étudié trois formulations mathemátiques, deux méthodes permettant de trouver des limites inférieures (une génération de colonnes et une heuristique) et on a développé plusieurs méthodes qui ont été combinées pour obtenir une méthode de type GRASP et une méthode de type Recherche Locale Itérative. Pour le deuxième problème nous avons généré de nouvelles instances, développé deux nouvelles méthodes et testé ces deux approches comme des alternatives à la résolution directe du modèle mathématique. La première méthode est une méthode de décomposition de Benders. La seconde est une combinaison de la formulation mathématique avec un local branching.

Toutes les méthodes ont été testées intensivement. Les résultats montrent l'efficacité des méthodes par rapport à l'état de l'art de chaque problème.

Chapter 1

Introduction

The motivation for this research is the great number of practical applications related to network design problems and the complexity of these problems. In particular, this thesis focuses on two network design problems, the Fixed Charge Uncapacitated Network Design Problem with User-optimal Flow (FCNDP-UOF) and the Transmission Expansion Planning Problem with Redesign (TEP_R). Although both of them belong to the network design problems class, they have different and peculiar structures that make them interesting.

1.1 Main Objective

The main objective of this thesis is to develop exact, metaheuristic and hybrid algorithms for Network Design Problems and evaluate the perks of each technique.

1.2 Specific Objectives

In this section is presented a summarized version of the specific objectives.

1. Study the state-of-art publications;
2. Develop exact methods;
3. Develop heuristics methods and combine them in metaheuristics;
4. Combine exact and heuristic methods into hybrid methods;
5. Evaluate the quality of the developed algorithms on problem instances available in literature and on new instances generated by us;

6. Publish the achieved results.

1.3 Structure of the thesis

This thesis is organized as follows. Chapter 2 describes in detail the Fixed Charge Uncapacitated Network Design Problem with User-optimal Flow. Chapter 3 presents mathematical formulations for the FCNDP-UOF and compares the strength of its linear relaxations. Chapter 4 describes the algorithms that we have developed and combines them into metaheuristics and hybrid methods. Chapter 5 shows the computational experiments and an analysis of the results obtained. Chapter 6 concludes the studies concerning the FCNDP-UOF and gives future research directions. Chapter 7 introduces and discusses the Transmission Expansion Planning Problem with Redesign. Chapter 8 presents mathematical formulations for the TEP_R. Chapter 9 describes the developed algorithms. Chapter 10 shows the computational experiments and an analysis of the results obtained. Chapter 11 concludes the studies concerning the TEP_R and gives future research directions. Chapter 12, at last, presents a general conclusion and a general guideline to future researches.

Chapter 2

Introduction to the Fixed Charge Uncapacitated Network Design Problem with User-optimal Flow

Due to the continuous development of society, increasing quantities of commodities have to be transported in large urban centers. Therefore, network design problems arise as tools to support decision-making, aiming to meet the need of finding efficient ways to perform the transportation of each commodity from its origin to its destination. In the Fixed Charge Network Design Problem (FCNDP), a subset of edges is selected from a graph, in such a way that a given set of commodities can be transported from their origins to their destinations. The main objective is to minimize the sum of the fixed costs (due to selected edges) and variable costs (depending on the flow of goods on the edges). In addition, fixed and variable costs can be represented by linear functions and arcs are not capacitated. Belonging to a large class of network design problems, the FCNDP has several variations such as shortest path problem, minimum spanning tree problem, vehicle routing problem, travelling salesman problem and Steiner problem in graph [31]. For generic network design problems, such as FCNDP, numerous applications can be found [8, 9, 32], thus, mathematical formulations for the problem may also represent several other problems, like problems of communication, transportation, sewage systems and resource planning. It also appears in other contexts, such as flexible production systems [26] and automated manufacturing systems [20]. Finally, network design problems arise in many vehicle fleet applications that do not involve the construction of physical facilities, but rather model decision problems such as sending a vehicle through a road or not [30, 40].

This work addresses a specific variation of FCNDP, called Fixed-Charge Uncapacitated Network Design Problem with User-optimal Flows (FCNDP-UOF), which consists of adding multiple shortest path problems to the original problem. The FCNDP-UOF involves two distinct agents acting simultaneously rather than sequentially when making decisions. On the upper level, the leader (1^{st} agent) is in charge of choosing a subset of edges to be opened in order to minimize the sum of fixed and variable costs. In response, on the lower level, the follower (2^{nd} agent) must choose a set of shortest paths in the network, through which each commodity will be sent. The effect of an agent on the other is indirect: the decision of the follower is affected by the network designed on the upper level, while the leader's decision is affected by variable costs imposed by the routes settled in the lower level. The inclusion of shortest path problem constraints in a mixed integer linear programming is not straightforward. Difficulties arise both in modelling and designing efficient methods.

The FCNDP-UOF problem appears in the design of a network for hazardous materials transportation [3, 13, 14, 24]. Particularly for this kind of problem, the government defines a selection of road segments to be opened/closed to the transportation of hazardous materials assuming that the shipments in the resulting network will be done along shortest paths. In hazardous materials transportation problems, roads selected to compose the network have no costs, but the government wants to minimize the population exposure in case of an incident during a dangerous-goods transportation. This is a particular case of the FCNDP-UOF problem where, from a mathematical point of view, the fixed costs are equal to zero.

Several variants of the FCNDP-UOF can be seen on [3, 6, 13, 14, 18, 24, 34] and have been treated as part of larger problems in some applications on [22]. The work presented by Bilheimer and Grey [6] formally defines the FCNDP-UOF. Both Erkut et al. [14] and Kara et al. [24] work focus on exact methods, presenting a mathematical formulation and several metrics for the hazardous materials transportation problem. At Mauttone et al. [34], not only was presented a different model, but also a Tabu Search for the FCNDP-UOF. Both Amaldi et al. [3] and Erkut et al. [13] presented heuristic approaches to deal with the hazardous materials transportation problem. At last, Gonzalez et al. [18], presented an extension of the model proposed by Kara and Verter [24] and also a GRASP method.

According to [23, 43], the simplest versions of network design problems are $\mathcal{NP}$-hard and even the task of finding feasible solutions (for problems with budget constraint on the fixed cost) is extremely complex [44]. Therefore, heuristics methods are presented as a good alternative in the search for good solutions. Knowing that, this thesis proposes several methods for the FCNDP-UOF and at the end combinations of these methods are tested.

Chapter 3

Mathematical Formulations for the FCNDP-UOF

In this chapter we formally introduce the FCNDP-UOF and present three different mathematical formulations.

The FCNDP-UOF is defined on a graph $G = (V, E)$, where V is a set of nodes that represent the facilities and E is a set of uncapacitated and undirected edges that represent the connection between installations. Furthermore, K is the set of commodities to be transported over the network, which may represent physical goods such as raw material for industry, hazardous material or even people. For each commodity $k \in K$, there is a flow to be delivered through a shortest path between its source $o(k)$ and its destination $d(k)$. All the formulations presented in this report work with variants presenting commodities with multiple origins and destinations, and for treating such a case, it is sufficient to consider that for each pair $(o(k), d(k))$, there is a new commodity resulting from the dissociation of one into several commodities.

Two kinds of variables can be noticed for FCNDP-UOF model, one for the construction of the network and another related to representing the flow. Let y_{ij} be a binary variable, we have that $y_{ij} = 1$ if the edge (i, j) is chosen as part of the network and $y_{ij} = 0$ otherwise. In this case, x^k_{ij} denotes the commodity k flow through the arc (i, j). Although the edges have no direction, they may be referred to as arcs, because each commodity flow is directed. Treating $y = (y_{ij})$ and $x = (x^k_{ij})$, respectively, as vectors of adding edge and flow variables, a mixed integer programming formulation can be elaborated.

3.1 List of Symbols

V	Set of nodes.
E	Set of admissible edges.
K	Set of commodities.
A^E	Set of arcs obtained by bi-directing the edges in E. $A^E = \{(i,j) \wedge (j,i) \mid (i,j) \in E\}$
$\mathcal{G}$	Associated graph $G(V,E)$.
δ_i^+	Set of all arcs leaving node $i \in V$.
δ_i^-	Set of all arcs arriving at node $i \in V$.
c_ij	Length of the arc $(i,j) \in A^E$.
$e(ij)$	Edge e related to the arc (i,j).
$o(k)$	Origin node for commodity $k \in K$.
$d(k)$	Destiny node for commodity $k \in K$.
q^k	Quantity of the commodity $k \in K$ to be transported.
β_{ij}	Cost to send a general commodity through the edge $e = [i,j]$.
g_{ij}^k	Variable cost of transporting commodity k through the arc $(i,j) \in A^E$.
f_e	Fixed cost of opening the edge $e \in E$.
y_e	Indicates whether edge $e \in E$ belongs in the solution.
x_{ij}^k	Indicates whether commodity $k \in K$ passes through the arc $(i,j) \in A^E$.

Let's also define that $g_{ij}^k = q^k \beta_{ij}$, where q^k represents the amount of commodity k to be transported and β_{ij} represents the shipping cost through the edge $e = [i,j]$.

3.2 Bi-level Formulation

In the FCNDP-UOF, differently from the basic FCNDP, each commodity $k \in K$ has to be transported through a shortest path between its origin $o(k)$ and its destination $d(k)$, forcing the addition of new constraints to the general problem. Besides selecting a subset of E whose sum of fixed and variable costs is minimal (leading problem), in this variation, we also have to guarantee the shortest path constraints for each commodity $k \in K$ (follower problem). The FCNDP-UOF belongs to the class of $\mathcal{NP}$-hard problems and can be modelled as a bi-level mixed integer programming problem [10], as follows:

$$\begin{aligned} \min \quad & \sum_{e \in E} f_e y_e + \sum_{k \in K} \sum_{(i,j) \in A^E} g_{ij}^k x_{ij}^k \\ \text{s.t.} \quad & y_e \in \{0,1\}, && \forall e \in E, \qquad (3.1) \end{aligned}$$

where x_{ij}^k is a solution of the problem:

$$\begin{aligned} \min \quad & \sum_{k \in K} \sum_{(i,j) \in A^E} c_{ij} x_{ij}^k \\ \text{s.t.} \quad & \sum_{(i,j) \in \delta_i^+} x_{ij}^k - \sum_{(i,j) \in \delta_i^-} x_{ji}^k = b_i^k, && \forall i \in V, \forall k \in K, \qquad (3.2) \\ & x_{ij}^k + x_{ji}^k \leq y_e, && \forall e = [i,j] \in E, \forall k \in K, \qquad (3.3) \\ & x_{ij}^k \in \{0,1\}, && \forall (i,j) \in A^E, \forall k \in K. \qquad (3.4) \end{aligned}$$

where:

$$b_i^k = \begin{cases} -1 & \text{if } i = d(k), \\ 1 & \text{if } i = o(k), \\ 0 & \text{otherwise.} \end{cases}$$

According to constraints (3.1)-(3.4), we can notice that the set of constraints (3.1) ensures that the vector of variables y assumes only binary values. In (3.2), we have flow conservation constraints. Constraints (3.3) do not allow flow into arcs whose corresponding edges are closed. Finally, (3.4) describes the domain of the vector of variables x^k. An interesting remark is that solving the follower problem is equivalent to solving $|K|$ shortest path problems independently.

3.3 One-level Formulation

In this section we present two different one-level formulations for the FCNDP-UOF and compare their strength, through comparing its linear relaxations. The first one is a variation of the one-level formulation proposed by Kara and Verter [24] for the Hazardous Material Transportation Problem, which we address as KVV Model. The second one is a one-level formulation presented by Mautonne, Figueiredo and Labbe [34] for the FCNDP-UOF, which we address as MFL Model.

3.3.1 KVV Model

The model presented here is a variation of the model presented by [24], where the difference lies in the fact that in our problem there are fixed-costs associated to adding an edge to the solution, while in Kara and Verter's doesn't consider such fixed-costs. As noted in the *Bi-Level formulation*, the shortest path problem has to be solved on the basis of available links set by the upper level problem. Given y's values, the inner problem is unimodular [42]. Thanks to this characteristic, the inner problem can be obtained by solving the feasibility problem defined by (3.2), (3.3) and the following set of constraints:

$$c_{ij} - w_i^k + w_j^k - v_{ij}^k + \lambda_{ij}^k = 0, \qquad \forall (i,j) \in A^E, k \in K, \tag{3.5}$$

$$v_{ij}^k x_{ij}^k = 0, \qquad \forall k \in K, (i,j) \in A^E, \tag{3.6}$$

$$\lambda_{ij}^k (x_{ij}^k - y_{e(ij)}) = 0, \qquad \forall k \in K, (i,j) \in A^E, \tag{3.7}$$

$$v_{ij}^k \geq 0, \lambda_{ij}^k \geq 0, \qquad \forall k \in K, \forall (i,j) \in A^E, \tag{3.8}$$

$$w_i^k \in \mathbb{R}, \qquad \forall k \in K, \forall i \in V, \tag{3.9}$$

$$x_{ij}^k \geq 0, \qquad \forall (i,j) \in A^E, \forall k \in K, \tag{3.10}$$

where the variables v_{ij}^k, λ_{ij}^k and w_i^k are the KKT multipliers associated to constrains (3.1)-(3.4).

Thanks to that we are allowed to represent the follower problem by Karush-Kuhn-Tucker's conditions of its linear relaxation, leaving us with a one level problem:

$$
\begin{aligned}
\min \quad & \sum_{e \in E} f_e y_e + \sum_{k \in K} \sum_{(i,j) \in A^E} g_{ij}^k x_{ij}^k \\
\text{s.t.} \quad & \sum_{(i,j) \in \delta_i^+} x_{ij}^k - \sum_{(i,j) \in \delta_i^-} x_{ji}^k = b_i^k, && \forall i \in V, \forall k \in K, && (3.11) \\
& x_{ij}^k + x_{ji}^k \leq y_e, && \forall e = [i,j] \in E, \forall k \in K, && (3.12) \\
& c_{ij} - w_i^k + w_j^k - v_{ij}^k + \lambda_{ij}^k = 0, && \forall (i,j) \in A^E, k \in K, && (3.13) \\
& v_{ij}^k x_{ij}^k = 0, && \forall k \in K, (i,j) \in A^E, && (3.14) \\
& \lambda_{ij}^k (x_{ij}^k - y_{e(ij)}) = 0, && \forall k \in K, (i,j) \in A^E, && (3.15) \\
& v_{ij}^k \geq 0, \lambda_{ij}^k \geq 0, && \forall k \in K, \forall (i,j) \in A^E, && (3.16) \\
& w_i^k \in \mathbb{R}, && \forall k \in K, \forall i \in V, && (3.17) \\
& x_{ij}^k \geq 0, && \forall (i,j) \in A^E, \forall k \in K, && (3.18) \\
& y_e \in \{0,1\}, && \forall e \in E. && (3.19)
\end{aligned}
$$

where:

$$
b_i^k = \begin{cases} -1 & \text{se } i = d(k), \\ 1 & \text{se } i = o(k), \\ 0 & \text{otherwise.} \end{cases}
$$

The uni-modularity of the inner problem is based on the fact that y is a set of parameters at this level. In the single-level representation of the FCNDP-UOF, however, optimal values of y and x are determined simultaneously. This structural change in the coefficient matrix causes the loss of uni-modularity, making it imperative to impose integrality on x variables.

Since we intend to focus on linear formulations in this section and constraints (3.14) and (3.15) are non-linear, a Big-M linearization method is applied, so the model can be written as a one-level mixed integer programming formulation, as follow:

$$\begin{aligned}
\min \quad & \sum_{e\in E} f_e y_e + \sum_{k\in K}\sum_{(i,j)\in A^E} g^k_{ij}x^k_{ij} && && \\
\text{s.t.} \quad & \sum_{(i,j)\in\delta_i^+} x^k_{ij} - \sum_{(i,j)\in\delta_i^-} x^k_{ji} = b^k_i, && \forall i \in V, \forall k \in K, && (3.20)\\
& x^k_{ij} + x^k_{ji} \le y_{ij}, && \forall e = [i,j] \in E, \forall k \in K, && (3.21)\\
& c_{ij} - w^k_i + w^k_j - v^k_{ij} + \lambda^k_{ij} = 0, && \forall (i,j) \in A^E, k \in K, && (3.22)\\
& v^k_{ij} \le M_{e(ij)}(1 - x^k_{ij}), && \forall k \in K, (i,j) \in A^E && (3.23)\\
& \lambda^k_{ij} \le M_{e(ij)}[1 - (y_{e(ij)} - x^k_{ij})], && \forall k \in K, \in A^E, && (3.24)\\
& v^k_{ij} \ge 0, \lambda^k_{ij} \ge 0, && \forall k \in K, \forall (i,j) \in A^E, && (3.25)\\
& w^k_i \in \mathbb{R}, && \forall k \in K, \forall i \in V, && (3.26)\\
& x^k_{ij} \in \{0,1\}, && \forall (i,j) \in A^E, \forall k \in K, && (3.27)\\
& y_e \in \{0,1\}, && \forall e = [i,j] \in E. && (3.28)
\end{aligned}$$

where:

$$b^k_i = \begin{cases} -1 & \text{se } i = d(k), \\ 1 & \text{se } i = o(k), \\ 0 & \text{otherwise.} \end{cases}$$

After analysing the model we can see that most of it, constraints (3.20), (3.21), (3.27) and (3.28) are previously explained. We can also notice that constraints (3.22), (3.23) and (3.24) represent linearized Karush-Kuhn-Tucker conditions for the follower problem, those are, shortest path problems. At last, (3.25) and (3.26) represent the domain of the variables of KKT conditions constraints.

A simple and yet useful way of approximating the value of the Big-M to this problem is for each $e = [i,j] \in E \setminus \{e\}$ keep selecting the longest edges connected to i or j until you get a path from i to j.

3.3.2 MLF Model

The FCNDP-UOF can be formulated as a one-level integer programming problem replacing the objective function and the constraints defined by (3.2)-(3.4) of the follower problem for its optimality conditions [34]. This can be done by applying the fundamental theorem of duality and the complementary slackness theorem [4] to the inner problem:

$$\pi_i^k - \pi_j^k - \lambda_{e(ij)}^k \leq c_{ij} \qquad \forall (i,j) \in A^E, k \in K, \tag{3.29}$$

$$(y_e - x_{ij}^k - x_{ji}^k)\lambda_e^k = 0, \qquad \forall e = [i,j] \in E, \forall k \in K, \tag{3.30}$$

$$(c_{ij} - \pi_i^k + \pi_j^k + \lambda_{e(ij)}^k)x_{ij}^k = 0, \qquad \forall (i,j) \in A^E, k \in K, \tag{3.31}$$

$$\lambda_e^k \geq 0, \qquad \forall e \in E, k \in K, \tag{3.32}$$

$$\pi_i^k \in \mathbb{R}, \qquad \forall i \in V, \forall k \in K, \tag{3.33}$$

$$x_{ij}^k \geq 0, \qquad \forall (i,j) \in A^E, \forall k \in K, \tag{3.34}$$

where the variables π_i^k and $\lambda_{e(ij)}^k$ are dual variables, associated to the inner problem constraints'.

Replacing the inner problem by the presented constraints, one may write the one level mathematical formulations as:

$$\min \quad \sum_{e \in E} f_e y_e + \sum_{k \in K} \sum_{(i,j) \in A^E} g_{ij}^k x_{ij}^k$$

$$\text{s.t.} \quad \sum_{(i,j) \in \delta_i^+} x_{ij}^k - \sum_{(i,j) \in \delta_i^-} x_{ji}^k = b_i^k, \qquad \forall i \in V, \forall k \in K, \tag{3.35}$$

$$x_{ij}^k + x_{ji}^k \leq y_e, \qquad \forall e = [i,j] \in E, \forall k \in K, \tag{3.36}$$

$$\pi_i^k - \pi_j^k - \lambda_{e(ij)}^k \leq c_{ij} \qquad \forall (i,j) \in A^E, k \in K, \tag{3.37}$$

$$(y_e - x_{ij}^k - x_{ji}^k)\lambda_e^k = 0, \qquad \forall e = [i,j] \in E, \forall k \in K, \tag{3.38}$$

$$(c_{ij} - \pi_i^k + \pi_j^k + \lambda_{e(ij)}^k)x_{ij}^k = 0, \qquad \forall (i,j) \in A^E, k \in K, \tag{3.39}$$

$$\lambda_e^k \geq 0, \qquad \forall e \in E, k \in K, \tag{3.40}$$

$$\pi_i^k \in \mathbb{R}, \qquad \forall i \in V, \forall k \in K, \tag{3.41}$$

$$x_{ij}^k \geq 0, \qquad \forall (i,j) \in A^E, \forall k \in K, \tag{3.42}$$

$$y_e \in \{0,1\}, \qquad \forall e \in E. \tag{3.43}$$

where:

$$b_i^k = \begin{cases} -1 & \text{if } i = d(k), \\ 1 & \text{if } i = o(k), \\ 0 & \text{otherwise.} \end{cases}$$

A disadvantage of this new formulation in comparison to the bi-level one is the loss of linearity of the model. To bypass this problem, a Big-M linearization may be used. After it, one can write the model as a one-level mixed integer linear programming problem, as follows:

$$
\begin{aligned}
\min \quad & \sum_{e\in E} f_e y_e + \sum_{k\in K}\sum_{(i,j)\in A^E} g_{ij}^k x_{ij}^k && && \\
\text{s.t.} \quad & \sum_{(i,j)\in\delta_i^+} x_{ij}^k - \sum_{(i,j)\in\delta_i^-} x_{ji}^k = b_i^k, && \forall i\in V, \forall k\in K, && (3.44)\\
& x_{ij}^k + x_{ji}^k \le y_e, && \forall e=[i,j]\in E, \forall k\in K && (3.45)\\
& \pi_i^k - \pi_j^k - \lambda_{e(ij)}^k \le c_{ij} && \forall (i,j)\in A^E, k\in K, && (3.46)\\
& \lambda_e^k + M_e y_e - M_e x_{ij}^k - M_e x_{ji}^k \le M_e, && \forall e=[i,j]\in E, k\in K, && (3.47)\\
& M_{e(ij)} x_{ij}^k - \pi_i^k + \pi_j^k + \lambda_{e(ij)}^k \le M_{e(ij)} - c_{ij}, && \forall (i,j)\in A^E, k\in K, && (3.48)\\
& \lambda_e^k \ge 0, && \forall e\in E, k\in K, && (3.49)\\
& \pi_i^k \in \mathbb{R}, && \forall i\in V, \forall k\in K, && (3.50)\\
& x_{ij}^k \in \{0,1\}, && \forall (i,j)\in A^E, \forall k\in K, && (3.51)\\
& y_e \in \{0,1\}, && \forall e\in E. && (3.52)
\end{aligned}
$$

where:

$$
b_i^k = \begin{cases} -1 & \text{if } i = d(k), \\ 1 & \text{if } i = o(k), \\ 0 & \text{otherwise.} \end{cases}
$$

However, optimality conditions for the problem in the lower level are, in fact, the optimality conditions of the shortest path problem and they could be expressed in a more compact and efficient way after applying two techniques. First we consider Bellman's optimality conditions for the shortest path problem. As can be seen in [1], Bellman's optimality conditions for the shortest path problem maybe expressed as:

Definition 1 (Bellman's Optimality Conditions for the Shortest Path Problem)**.** *Let Bellmans' equations be:*

$$
d(j) = \min\{d(i) + c_{ij} : (i,j) \in A(i)\},\ \forall j \in N \tag{3.53}
$$

If a set of distance labels d(i)'s satisfy Bellman's equations and the network contains no zero-length cycle, these distance labels are shortest path distances.

After applying Bellmans' and a simple lifting process [29], the one-level linear mathematical formulation can be written in a more efficient way as :

$$
\begin{aligned}
\min \quad & \sum_{e \in E} f_e y_e + \sum_{k \in K} \sum_{(i,j) \in A^E} g_{ij}^k x_{ij}^k && \\
\text{s.t.} \quad & \sum_{(i,j) \in \delta_i^+} x_{ij}^k - \sum_{(i,j) \in \delta_i^-} x_{ji}^k = b_i^k, && \forall i \in V, \forall k \in K, \quad (3.54) \\
& x_{ij}^k + x_{ji}^k \leq y_e, && \forall e = [i,j] \in E, \forall k \in K, \quad (3.55) \\
& \pi_i^k - \pi_j^k \leq M_{e(ij)} - y_{e(a)}(M_{e(ij)} - c_i j) - 2c_{ij} x_{ji}^k, && \forall (i,j) \in A^E, k \in K, \quad (3.56) \\
& \pi_{d(k)}^k = 0, && \forall k \in K, \quad (3.57) \\
& \pi_i^k \geq 0, && \forall i \in V \setminus \{d(k)\}, \forall k \in K, \quad (3.58) \\
& x_{ij}^k \in \{0,1\}, && \forall (i,j) \in A^E, \forall k \in K, \quad (3.59) \\
& y_e \in \{0,1\}, && \forall e \in E. \quad (3.60)
\end{aligned}
$$

where:

$$
b_i^k = \begin{cases} -1 & \text{if } i = d(k), \\ 1 & \text{if } i = o(k), \\ 0 & \text{otherwise.} \end{cases}
$$

The variables π_i^k, $k \in K$, $i \in V$, represent the shortest distance between vertex i and vertex $d(k)$. Then we define that $\pi_{d(k)}^k$ will always be equal to zero. Assuming that constraints (3.55), (3.59) and (3.60) are satisfied, it is easy to see that constraints (3.56) are equivalent to Bellman's optimality conditions for $|K|$ pairs $(o(k), d(k))$.

A simple and yet useful way of approximating the value of the Big-M to this problem is for each $e = [i,j] \in E \setminus e$ keep selecting the longest edges connected to i or j until you get a path from i to j.

3.3.3 Comparing the Linear Relaxations

As we know, an important matter when working with integer programming is how close the feasible set of the linear relaxation is to the convex hull of the original feasible set [36]. The tighter the polyhedron of the relaxation is, the stronger is the formulation. To compare the strength of KVV Model and MLF Model, we consider their linear relaxation polyhedrons, obtained by replacing the $\{0,1\}$ set by the interval $[0,1]$.

Since, to the best of our knowledge, nobody generalized the model presented in Kara et al. [24] before, no comparison between the strength of the linear relaxation of the KVV Model and the MLF Model was done. So, we dedicate this subsection to analysing them. Accordingly, the polyhedrons for the KVV Model and for the MLF Model are respectively defined as:

$$P_{KVV} = conv\left(\left\{(y,x,w,v,\lambda) \text{ satisfies: } \begin{array}{ll} (3.20), & (3.21), \\ (3.22), & (3.23),\ (3.26), \text{ for some } (f,g,c), \\ (3.24), & (3.25), \end{array}\right.\right.$$
$$\left.\left. w \in \mathbb{R}^{|K||V|},\ v \in \mathbb{R}^{+|K||A^E|},\ \lambda \in \mathbb{R}^{+|K||A^E|},\ y \in [0,1]^{|E|} \text{ and } x \in [0,1]^{|K||A^E|} \right\}\right) \tag{3.61}$$

$$P_{MLF} = conv\left(\left\{(y,x,\pi) \text{ satisfies: } \begin{array}{ll} (3.54), & (3.55), \\ (3.56), & (3.57), \end{array} (3.58), \text{ for some } (f,g,c), \right.\right.$$
$$\left.\left. \pi \in \mathbb{R}^{+|V|},\ y \in [0,1]^{|E|} \text{ and } x \in [0,1]^{|K||A^E|} \right\}\right) \tag{3.62}$$

Considering the structure of the polyhedrons, one first intuition is that $P_{MLF} \subseteq P_{KVV}$ and it could be proved as follow:

Proof ($\boldsymbol{P_{MLF} \subseteq P_{KVV}}$): Consider that we have $(y,x,\pi) \in P_{MLF}$, what we are going to show is that $\exists(v,w,\lambda) \mid (y,x,w,v,\lambda) \in P_{KVV}$.
Without loss of generality lets take v_{ij}^k and λ_{ij}^k as:

$$v_{ij}^k = M_{e(ij)}(1 - x_{ij}^k) \quad \text{and} \quad \lambda_{ij}^k = M_{e(ij)}[1 - (y_{e(ij)} - x_{ij}^k)] \tag{3.63}$$

Replacing v_{ij}^k and λ_{ij}^k in constraint (3.22):

$$\begin{aligned} c_{ij} - w_i^k + w_j^k - M_{e(ij)}(1 - x_{ij}^k) + M_{e(ij)}[1 - (y_{e(ij)} - x_{ij}^k)] = 0 \\ -w_i^k + w_j^k = M_{e(ij)}(y_{e(ij)} - 2x_{ij}^k) - c_{ij} \end{aligned} \tag{3.64}$$

Since w_i^k and w_j^k belong to $\mathbb{R}$, one can assume them as:

$$\begin{aligned} w_i^k &= -\pi_i^k + \frac{1}{2}M_{e(ij)} - c_{ij}x_{ji}^k - y_{e(ij)}M_{e(ij)} - \frac{1}{2}y_{e(ij)}c_{ij} + x_{ij}^k M_{e(ij)} + \frac{1}{2}c_{ij} \\ w_j^k &= -\pi_j^k - \frac{1}{2}M_{e(ij)} + c_{ij}x_{ji}^k + y_{e(ij)}M_{e(ij)} + \frac{1}{2}y_{e(ij)}c_{ij} - x_{ij}^k M_{e(ij)} - \frac{1}{2}c_{ij} \end{aligned} \tag{3.65}$$

Replacing w_i^k and w_j^k in constraint (3.64):

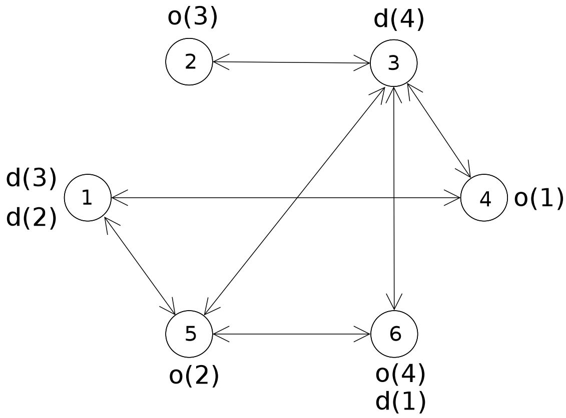
$$\pi_i^k - \pi_j^k = M_{e(ij)} - y_{e(ij)}(M_{e(ij)} - c_ij) - 2c_{ij}x_{ji}^k \tag{3.66}$$

Proving that $P_{MLF} \subseteq P_{KVV}$. □

Although not as intuitive as the previous one, it is possible to show that $P_{KVV} \not\subseteq P_{MLF}$.

In order to do this, an example is provided.

Proof **($P_{KVV} \not\subseteq P_{MLF}$):** Given the following network



let's consider $g_{ij}^k = q^k \beta_e$, where q^k is the quantity of the commodity k to be transported and β_e is a transportation cost.

e	i	j	f_e	β_e	c_e
1	1	4	34	5	83
2	1	5	23	6	48
3	2	3	32	1	36
4	3	4	3	1	98
5	3	5	18	8	33
6	3	6	12	4	95
7	5	6	1	6	38

k	q^k
1	79
2	90
3	49
4	27

The linear relaxation of KVV is:

$$
\begin{aligned}
& & x_{36}^1 &= 1 \\
y_1 &= 1 & x_{43}^1 &= 1 \\
y_2 &= 1 & x_{51}^2 &= 1 \\
y_3 &= 1 & x_{23}^3 &= 1 \\
y_4 &= 1 & x_{34}^3 &= 1 \\
y_5 &= 0.055684 & x_{41}^3 &= 1 \\
y_6 &= 1 & x_{35}^4 &= 0.055684 \\
y_7 &= 0.944316 & x_{63}^4 &= 0.055684 \\
& & x_{65}^4 &= 0.944316
\end{aligned}
$$

When trying to force this solution into the MLF model, constraints 3.56 can not be fully satisfied for commodity 1. In order to satisfy the shortest path constraints for commodity 1 the following has to be satisfied:

$$\begin{aligned}
-\pi_5^1 + \pi_6^1 &\leq 38 \\
\pi_5^1 - \pi_6^1 &\leq 38 \\
-\pi_3^1 + \pi_6^1 &\leq -95 \\
\pi_3^1 - \pi_6^1 &\leq 95 \\
-\pi_4^1 + \pi_5^1 &\leq 33 \\
\pi_3^1 - \pi_5^1 &\leq 33 \\
-\pi_3^1 + \pi_4^1 &\leq 98 \\
\pi_3^1 - \pi_4^1 &\leq -98 \\
-\pi_1^1 + \pi_5^1 &\leq 48 \\
\pi_1^1 - \pi_5^1 &\leq 48 \\
-\pi_1^1 + \pi_4^1 &\leq 83
\end{aligned} \tag{3.67}$$

The infeasibility arises when trying to satisfy those equations. The last inequation can only be satisfied along with the others if $\pi_6^1 \neq 0$, which is not possible according to constraints (3.57), proving that $P_{KVV} \not\subseteq P_{MLF}$.

□

Thanks to the information above, one can state that MLF Model is stronger than KVV Model.

Chapter 4

Algorithms for the FCNDP-UOF

This chapter focuses on presenting the different methods developed in this work. The first section presents two approaches to find better lower bounds (dual bounds) than the linear relaxation . The second section presents several heuristic methods leading to a third section where a GRASP [38] and an Iterated Local Search [28] are presented.

4.1 Relaxations

In this section we present the two approaches developed by us to find better lower bounds than the linear relaxation. The first one is a straitforward application of the Column Generation as described by Wolsey in [42]. The second one is a heuristic method in which at each iteration we come closer to the original problem.

4.1.1 Column Generation

Since the structure of this problem is very welcoming to decomposition strategies, this section presents a column generation strategy [42] to try to find a better lower bound to the FCNDP-UOF than the linear relaxation.
In the column generation scheme the idea is to generate columns at each iteration so the solution can be improved. To do that we divide the problem in master and sub-problem, where the master is responsible for using the generated columns to find a solution and the sub-problem is responsible for generating new columns.
Considering that we chose the MLF Formulation, presented in Chapter 3, to be decomposed.

4.1.1.1 Master Model

Let P_k be the set of the paths found for each commodity $k \in K$. Define z_p^k as the binary variable associated to a path $p \in P_k$ of the commodity $k \in K$ and r_p^k as its cost. Let's also define R_{pk}^{ij}, $(ij) \in A^E$, $k \in K$, $p \in P_k$, as a multi-dimensional matrix that keeps the information of whether an edge is used in a path of the commodity k or not. Knowing that, the Master Model can be formulated as:

$$
\begin{aligned}
\min \quad & \sum_{e \in E} f_e y_e + \sum_{k \in K} \sum_{p \in P_k} r_p^k z_p^k \\
\text{s.t.} \quad & \sum_{p \in P_k} z_p^k = 1, && \forall k \in K, && (4.1)\\
& y_e \geq \sum_{p \in P_k} z_p^k R_{pk}^{ij} + z_p^k R_{pk}^{ji}, && \forall e = [i,j] \in E, \forall k \in K, && (4.2)\\
& \pi_i^k - \pi_j^k \leq M_{e(ij)} - y_{e(ij)}(M_{e(ij)} - c_{ij}) - 2c_{ij} \sum_{p \in P_k} z_p^k R_{pk}^{ji}, && \forall (i,j) \in A^E, k \in K, && (4.3)\\
& \pi_i^k = 0, && \forall i = d(k), \forall k \in K, && (4.4)\\
& \pi_i^k \geq 0, && \forall i \in V, \forall k \in K, && (4.5)\\
& z_p^k \in \{0,1\}, && \forall k \in K, \forall p \in P_k, && (4.6)\\
& y_e \in \{0,1\}, && \forall e \in E. && (4.7)
\end{aligned}
$$

The objective function minimizes the sum of the cost of choosing the best combination of paths and the opening cost of the edges used by the paths. Constraints (4.1) state that for each commodity $k \in K$, only one path is chosen. Constraints (4.2) state that if edge $(i,j) \in E$ is used by at least one of the paths chosen, then $y_{ij} = 1$. At last, constraints (4.3) ensure that the shortest path problem in this path formulation.

4.1.1.2 Subproblems

The idea behind the column generation for the FCNDP-UOF is generating paths in order to improve the solution found by the Master Problem. Considering that we intend to solve $|K|$ shortest path problems, where the objective function's coefficient is the reduced cost associated to the best solution found, one way of generating these columns would be solving the following model for each $k \in K$.

$$
\begin{aligned}
\min \quad & -\zeta_k + \sum_{(i,j) \in A^E} (g_{ij}^k - 2c_{ij}\lambda_{ji}^k + \beta_{e(ij)}^k) x_{ij}^k \\
\text{s.t.} \quad & \sum_{(i,j) \in \delta^+(i)} x_{ij}^k - \sum_{(i,j) \in \delta^-(i)} x_{ij}^k = b_i^k, && \forall i,j \in V, && (4.8)\\
& x_{ij}^k \geq 0, && \forall (i,j) \in A^E. && (4.9)
\end{aligned}
$$

where:

$$b_i^k = \begin{cases} -1 & \text{if } i = d(k), \\ 1 & \text{if } i = o(k), \\ 0 & \text{otherwise.} \end{cases}$$

In the objective function, ζ_k, λ_{ji}^k and $\beta_{e(ij)}^k$ are the dual variables associated to constraints (4.1), (4.3) and (4.2) respectively. At last constraints (4.9) state that variables x^k are non-negative.
After defining both the master and the subproblem, it is just a matter of creating the loop described above.

4.1.2 LBound Method

LBound Method, is a strategy to probably find a lower bound to the original problem stronger than the linear relaxation. In order to do that, the method consists in relaxing all variables and at each iteration a subset of y variables are turn into binary variables of the model (3.54) - (3.60). The process repeats until $\lceil 0.2|E| \rceil$ iterations are done, or more than 90% of y variable are set as binary, or an integer solution has been found. It is important to remark that when an integer solution is found, this solution is the optimal solution of the problem.
Something similar to this method might have been done by some else, but to the best of our knowledge this is a new strategy at least to this problem.
Details of the method can be seen in Algorithm 1:

The number of iterations is defined in the computational experiments chapter.
The function $LinearRelaxation()$ solves the linear relaxation of the problem and returns the solution value. The function $SolveR()$ solves a relaxation the problem with a subset of binary variables. Function $OptFound()$ verifies if the solution found by the method is integer or not. It is important to remark that the condition $nvbin > 0.9|E|$ was never reached. In line 6 the value 0.5 was chosen after several computational tests using values in the interval $[0.3; 0.7]$.
One must pay attention to the number of binary variables added at each iteration so the difficulty to solve the relaxation doesn't increase too much.

Algorithm 1: LBound

Input: K, $\mathcal{G}$
Data: $nvbin, cont \leftarrow 0$
begin
  $s_{inf} \leftarrow LinearRelaxation()$;
  $\bar{E} \leftarrow E$;
  repeat
    for $e \in \bar{E}$ **do**
      if $y_e \geq 0.5$ **then**
        $y_e \in \{0, 1\}$;
        $\bar{E} \setminus \{e\}$;
        $nvbin \leftarrow nvbin + 1$;
      end
    end
    $s_{inf} \leftarrow SolveR()$;
    $cont \leftarrow cont + 1$;
  until $cont \geq \lceil 0.2|E| \rceil$ *or* $OptFound(s_{inf}) = TRUE$ *or* $nvbin > 0.9|E|$;
  return s_{inf}
end

4.2 Heuristic Methods

This section focuses on presenting the different heuristics developed in this work. First, the Partial Decoupling Heuristic is introduced. After that, a variable fixing heuristic that uses the previously explained methods. At last the Local Branching (used as Local Search) and the Ejection Cycle (used as Perturbation) are shown so a Iterated Local Search metaheuristic could be built.

4.2.1 Partial Decoupling Heuristic

The main idea of total decoupling heuristic for the FCNDP-UOF is dissociating the problem of building a network from the shortest path problem. This disintegration, as discussed in [13], can provide worst results than when addressing both problems simultaneously. To work around this situation, the method uses what we call partial decoupling, where certain aspects of the follower problem are considered when trying to build a solution to the leading problem.
The algorithm proposed here is a small variation of the original Partial Decoupling Heuristic [18].
The Partial Decoupling Heuristic iteratively builds a network and then routes each commodity so a feasible solution can be built. In order to build the network the cost

$\bar{f}_e^k,\ e = [i,j] \in E,\ k \in K$ was defined as:

$$\bar{f}_e^k = \begin{cases} f_e + \alpha \times g_{ij}^k + (1-\alpha) \times \frac{c_{ij}+c_{ji}}{2} & \text{if } y_e = 0, \\ \alpha \times g_{ij}^k + (1-\alpha) \times \frac{c_{ij}+c_{ji}}{2} & \text{otherwise.} \end{cases} \tag{4.10}$$

Doing that, we consider whether the edge is open or not, plus a linear combination of the variable cost and the average length of the edge as the fixed cost. The α works as a scaling parameter of the importance of the g_{ij}^k and $\frac{c_{ij}+c_{ji}}{2}$ values. In the beginning of the heuristic α prioritizes the variable cost (g_{ij}^k), while in the end it prioritizes the average edge length ($\frac{c_{ij}+c_{ji}}{2}$). It is important to pay attention that $g_{ij}^k = q^k \beta_{ij}$, where q^k represents the amount of commodity k to be transported and β_{ij} represents the shipping cost through the edge $e = [i,j]$.

After building the network, another shortest path algorithm, using the edges length (c_{ij}) as cost, is applied to take every commodity from its origin $o(k)$ to its destination $d(k)$ in the built network.

In order to put the scaling parameter α in good use, the method repeats $MaxIterDP$ times and at each iteration uses a different value for α.

The procedure is further explained on Algorithm 2.

To solve the shortest path problem, the partial decoupling heuristic applies the Dijkstra algorithm. The function $DefCost()$ defines the fixed costs used in the $DijkstraLeader()$. At the $|K|$ runs, the function $DijkstraLeader()$ solves the problem of network construction, then, the shortest path problem is solved using the $DijkstraFollower()$ function, generating a feasible solution. The $\hookleftarrow$ operator in line 8, represents that all $y_e = 1$ stay with the same value and every $y_e = 0$ which the value was changed to 1 after $DijkstraLeader()$receive the new value. The notation $s \leftarrow \langle y, x \rangle$ means that the solution s is storing the values of the variables y and x that were just defined by $DijkstraLeader()$ and $DijkstraFollower()$. Since the function $DijskstraLeader()$ can open edges that at the end do not have flow, we used the function $CloseEdge()$ that sets $y_e = 0$ for every $x_{ij}^k = 0, \forall k \in K$. The $Random()$ function returns a random element from the set passed as a parameter. In order to choose the insertion order of the $|K|$ commodities, a candidate list consisting of a subset of commodities not yet routed, whose amount is greater than or equal to $\gamma\%$ $(0 \leq \gamma \leq 100)$ times the largest amount (q_k) of the commodities not routed is created through the use of the function $CandidateList()$.

Algorithm 2: Partial Decoupling Heuristic

Input: γ, K, $\mathcal{G}$

Data: $MinCost \leftarrow \infty$, $\alpha \leftarrow 1$, $y \leftarrow 0$, $x \leftarrow 0$;

begin
    $\bar{K} \leftarrow K$;
    for $numIterDP \in [1, MaxIterDP]$ **do**
        while $\bar{K} \neq \emptyset$ **do**
            $\hat{K} \leftarrow CandidateList(\bar{K}, \mathcal{G}, \gamma)$;
            $k' \leftarrow Random(\hat{K})$;
            $\bar{f}^{k'} \leftarrow DefCost(k', \alpha)$;
            $y \hookleftarrow DijkstraLeader(\bar{f}^{k'}, k')$;
            $\bar{K} \leftarrow \bar{K} \backslash \{k'\}$;
        end
        for $k \in K$ **do**
            $x^k \leftarrow DijkstraFollower(c, k)$;
        end
        $s \leftarrow \langle y, x \rangle$;
        $CloseEdge(s)$;
        if $Cost(s) < MinCost$ **then**
            $s_{best} \leftarrow s$;
            $MinCost \leftarrow Cost(s_{best})$;
        end
        $\alpha \leftarrow \alpha - \frac{1}{MaxIterDP}$;
        $\bar{K} \leftarrow K$, $x \leftarrow 0$, $y \leftarrow 0$;
    end
    return s_{best}
end

4.2.2 Variable Fixing Heuristic

The objective of the Variable Fixing Heuristic (VFH) is to find a high quality solution. The VFH starts using both the Partial Decoupling Heuristic and the LBound method. After applying those two methods, the VFH uses a relax and fix strategy to try to find a better solution.

Based on the Relax and Fix Heuristic [42], in this third part, we separate the variables in two distinct sets N_1 and N_2. N_1 is the set of relaxed variables and N_2 is the set of binary variables. Initially N_1 contains all variables, while N_2 is empty. The main idea is that at each iteration we move a subset of the variables x^k from N_1 to N_2. At the end of each iteration, if a feasible solution for the relaxed model was found, the variables y that are both zero and attend to the reduced cost criterion [42] for variable fixing, are fixed as zero. The method repeats until all x^k have been moved from N_1 to N_2 or the duality gap becomes lower than one.

In order to choose the order of x^k variables to become binary, the procedure uses a candidate list. To choose a commodity, an element is randomly selected from a candidate list consisting of the commodities whose amount to be transported is greater than or equal to $\gamma\%$ $(0 \leq \gamma \leq 100)$ times the largest amount of the commodity whose variables are not set as binary. A pseudo-code of the method is presented in Algorithm 3.

Algorithm 3: VFH

Input: γ, K, $\mathcal{G}$
Data: $MinCost \leftarrow \infty$

```
begin
    s_best ← PartialDecoupling(γ, K, G);
    s_inf ← LBound(K, G);
    MinCost ← Cost(s_best) ;
    K̄ ← K;
    if OptFound(s_inf) ≠ TRUE then
        while K̄ ≠ ∅ and |s_best − s_inf| ≥ 1 do
            k ← CandidateList(K̄, γ);
            x^k ∈ {0, 1};
            s ← SolveR(MinCost);
            if A feasible solution for the relaxed model was found then
                for e ∈ E do
                    if y_e = 0 and RCVF(y_e, s_best) = TRUE then
                        y_e ← 0;
                    end
                end
                if Cost(s) < MinCost and Feas(s) = TRUE then
                    s_best ← s ;
                    MinCost ← Cost(s_best) ;
                end
                else if Cost(s) > Cost(s_inf) and Feas(s) = FALSE then
                    s_inf ← s;
                end
            end
            else
                Exit
            end
            K̄ ← K̄ \ {k}
        end
        return s_best
    end
    else
        return s_inf
    end
end
```

The function $SolveR()$ solves a relaxation of the one level formulation (3.54)-(3.60) with a subset of binary variables, taking into consideration the primal bound $MinCost$. $MinCost$ is defined as the current best solution cost. The $RCVF()$ function returns TRUE if the Linear Relaxation cost plus the Reduced Cost (obtained in the last call of the SolveR function) of y_e is greater than the s_{best} solution, also passed as a parameter. The function $Feas()$ returns true if the solution s passed as parameter is a feasible solution to the original problem and returns false otherwise.

4.2.3 Local Branching

Introduced by Fischetti and Lodi [17], the Local Branching (LB) technique can be used as a way of improving a given feasible solution. The LB makes use of a MIP solver to explore the solution subspaces effectively. The procedure can be seen as local search, but the neighborhoods are obtained through the introduction of linear inequalities in the MIP model, called local branching cuts. More specifically, the LB searches for a local optimum by restricting the number of variables, from the feasible solution, whose values can be changed.

Formally speaking, consider a feasible solution of the FCNDP-UOP, $s = \langle \bar{y}, \bar{x} \rangle \in P$, where P is the polyhedron formed by (3.54)-(3.60). The general idea would be adding the LB constraint

$$\sum_{e \in E | \bar{y}_e = 0} y_e + \sum_{e \in E | \bar{y}_e = 1} (1 - y_e) \leq \Delta, \tag{4.11}$$

where Δ is a given positive integer parameter, indicating the number of variables y_e, $e \in E$, that are allowed to flip from one to zero and vice versa.

The strategy used here consists in applying the LB constraint only on y variables, leaving x^k variables free of LB constraints.

4.2.4 Ejection Route

Given a feasible solution, the idea behind the Ejection Route is to analyse solutions that have at least 80% of the commodities with the same path as a feasible solution passed as reference. Since exploring this solution space is computationally expensive ($\frac{|K|!}{(|K| - \lceil 0.8|K| \rceil)!}$), we decided to just sample the solution space, using the first improvement as the acceptance criteria. Meaning that when a solution with better quality than the solution passed as reference is found, this new solution becomes the reference solution. Ejection Routes' pseudo-code can be seen in Algorithm 4.

Algorithm 4: Ejection Route

```
Input: s, γ, K, 𝒢
Data: numIterER ← 0
begin
    while numIterER < maxIterER do
        s̄ ← Neighbor(s);
        if cost(s̄) < cost(s) then
            s ← s̄;
            numIterER ← 0;
        end
        else
            numIterER ← numIterER + 1;
        end
    end
    return s
end
```

The $Neighbor()$ function randomly returns a neighbor of the solution s that has at least 80% of the commodities with the same path as s. To find the new solution $\lfloor 0.2|K| \rfloor$ paths of the solution s are randomly destroyed and than those paths are reconstructed using the Partial Decoupling Heuristic. If the found solution has a better quality than the current solution, s is updated.

4.2.5 Ejection Cycle

To understand the principles of the Ejection Cycle, it is necessary to get to know a few metrics, developed by [37], to evaluate chains in a solution.

Consider a solution defined by the variables x_{ij}^k for each arc $(i,j) \in A^E$ and each commodity $k \in K$ and y_e for each edge $e \in E$. For each open edge e, where $y_e = 1$ and $x_{ij}^k > 0$ or $x_{ji}^k > 0$ for at least one commodity k, the edge *inefficiency ratio* can be defined as:

$$I_e = \frac{\displaystyle\sum_{k \in K} g_{ij}(x_{ij}^k + x_{ji}^k) + f_e}{\displaystyle\sum_{k \in K}(x_{ij}^k + x_{ji}^k)}; \qquad \forall e = [i,j] \in E. \tag{4.12}$$

The lower the value of I_e, more interesting it is to have edge e in the solution. The average inefficiency ratio is defined as:

$$\bar{I} = \frac{\sum_{e \in E} I_e y_e}{\sum_{e \in E} y_e}. \tag{4.13}$$

With these metrics we can define a set of *inefficient edges* as:

$$A_I = \{e \mid y_e = 1, I_e > \bar{I}\}. \tag{4.14}$$

As it can be seen above, the set of inefficient edges contains every edge in the solution whose inefficiency ratio is greater than the average inefficiency ratio. Our aim is to create a movement that removes flows from some of the inefficient edges in set A_I.

After evaluating the edges it is possible to construct *inefficient chains* from a subset of the *inefficient edges*. First, an edge is randomly chosen from the set A_I of inefficient edges to form a component of the inefficient chain. If the current partial inefficient chain extends from node i to node j, then an edge $(a, i) \in A_I$ or $(j, b) \in A_I$ is added to the current chain, where nodes a and b are not included in the current chain. Whenever an edge is added to a chain, it is deleted from A_I. The process of extending the current chain continues until no further extension is possible or until the chain is composed by four edges. Unless A_I is empty or contains a single arc, the process iterates with a random edge chosen to start a new chain. When the process ends, any chains containing a single edge are deleted. This is done in order to decrease the number of edges affected at each iteration of the method.

After constructing a set of *inefficient chains*, we define our movement. The movement is defined analysing each chain in the set of *inefficient chains*.

The key aspect of our perturbation is the re-routing of flow from edges of the *inefficient chain* to other edges of the network. First, a list of commodities (K_{SET}) that have a positive flow through at least one edge of the randomly selected *inefficient chain* is formed. After that, the opening cost (f_e) of each edge in the *inefficient chain* is set as infinity. After reassigning the costs, every commodity in K_{SET} has its route destroyed and reconstructed by the Partial Decoupling Heuristic taking into account the new opening costs. If a feasible solution is found the method stops, otherwise, the previously selected *inefficient chain* is removed from the set and another *inefficient chain* is randomly selected and so the process restarts. Algorithm 5 describes our Ejection Cycle procedure.

Algorithm 5: Ejection Cycle

Input: s, γ, K, $\mathcal{G}$
begin
  P $\leftarrow PInefChain(s)$;
  $\bar{s} \leftarrow \emptyset$;
  while $P \neq \emptyset$ *and* $\bar{s}$ *is not feasible* **do**
    $rchain \leftarrow Random$(P);
    $P \setminus \{rchain\}$;
    $K_{SET} \leftarrow SK(s, rchain)$;
    $\bar{s} \leftarrow$ PartialDecoupling$(\mathcal{G}, \gamma, K_{SET}, s)$;
    if $Cost(\bar{s}) \leq Cost(s)$ **then**
      $s \leftarrow \bar{s}$;
    end
  end
  return s
end

In order to clarify Algorithm 5 it is necessary to define a few things. The function $PInefChain()$ returns the set A_I of *inefficient chains* in a solution s. The function $SK()$ returns the commodities that have a positive flow in solution s through at least one arc of the *inefficient chain* passed as parameter and set the fixed costs of the edges in the $rchain$ as infinity. The function $PartialDecoupling()$ reroutes the commodities in K_{SET}. In order to do that the $DijkstraLeader()$ is applied for all $k \in K_{SET}$ and $DijkstraFollower()$ for all $k \in K$. To account those changes, now the method $PartialDecoupling()$ needs to receive two new parameters which are, the set of commodities used in $DijkstraLeader()$ and a partial solution for all $k \in K \setminus K_{SET}$.

4.3 GRASP and ILS

4.3.1 GRASP

The GRASP metaheuristic [38] is a metaheuristic based on three basic premises: greedy, randomized and adaptive. While other metaheuristics such as tabu search and genetic algorithms avail themselves of strategies based on emphasizing the local search, GRASP focuses its efforts in generating better quality solutions and then use local search only for minor improvements. The GRASP is a multi-start metaheuristic, which means that at each iteration it performs its constructive component and a complete local search. The pseudo-code of the GRASP developed by us, GRASP-DE, which uses the Partial Decoupling and the Ejection Route as its main components may be observed in Algorithm

6:

Algorithm 6: GRASP-DE

```
Input: γ, K, 𝒢
Data: s_best ← ∞
begin
    for 1 ... MaxIterGR do
        s ← PartialDecoupling(γ, K, 𝒢);
        s ← EjectionRoute(s, γ, K, 𝒢);
        s_best ← UpdateBest(s, s_best);
    end
    return s_best
end
```

GRASP-DE's constructive component is the Partial Decoupling Heuristic. To perform the local search step, the procedure uses the method Ejection Route to try to refine the best solution found on this iteration. The procedures are executed sequentially $MaxIterGR$ times.

4.3.2 Iterated Local Search

Developed by Lourenço et al. [28], the Iterated Local Search (ILS) is a metaheuristic that applies a local search method repeatedly to a set of solutions obtained by perturbing previously visited local optimal solutions. The ILS presented here uses as its components, the VFH, the Local Branching and the Ejection Cycle presented in the previously subsections. The methods are applied in a straightforward way following the scheme presented by Lourenço et al. [28]. First we execute the VFH to get a feasible solution and a lower bound. Secondly we try to improve the quality of the previous found solution through applying the Local Branching (as a Local Search) and the Ejection Cycle (as a Perturbation). For the sake of argument from now on we are going to call this ILS, VFHLB. The algorithm is described in Algorithm 7.

Algorithm 7: VFHLB

```
Input: γ, Δ, K, 𝒢
Data: s_best ← ∞
begin
    s, s_inf ← VFH(𝒢, K, γ);
    s ← LB(s, Δ);
    UpdateBest(s);
    if |cost(s_best) − cost(s_inf)| ≥ 1 then
        while Stop Criterion=false do
            s ← EjectionCycle(s, γ, K, 𝒢);
            s ← LB(s, Δ);
            s_best ← UpdateBest(s, s_best);
        end
    end
    return s_best
end
```

In the VFHLB, the initial solution and the lower bound are generated by the VFH method. Then, the function LB performs the Local Branching as a Local Search and the EjectionCycle performs a perturbation. There are many possible Stop Criterion to be selected. In this case, we decided to use the number of iterations, which is going to be defined in the computational experiments chapter.

Chapter 5

Computational Experiments for the FCNDP-UOF

In this chapter we present computational experiments done using the methods presented for the FCNDP-UOF.

The algorithms were coded in Xpress Mosel using FICO Xpress Optimization Suite, on an *Intel ®Core TM i3 CPU 3250 @ 3,5GHz* computer with 8GB of RAM. Computing times are reported in seconds. In order to test the performance of the presented heuristic, we used networks data obtained from Mauttone, Labbé and Figueiredo [34].

The data used are grouped according to the number of nodes in the graph (10, 20, 30), followed by the graph density (0.3, 0.5, 0.8) and finally the amount of different commodities to be transported (5, 10, 15, 20, 30, 45).

Our computational experiments are divided in 3 sections. Section 5.1 presents the results comparing MLF and KVV mathematical formulations and their linear relaxation. In Section 5.2 the results for the Column Generation and for the LBound method are reported in comparison to the optimal solutions presented in the previously section. At last, in Section 5.3 we compare the ILS and the GRASP presented in Chapter 4.

5.1 Results for the Mathematical Formulations

For the tests presented in the next tables, we report the solution found by the solver (*Sol*), the time (*Time*), the value of the linear relaxation (*LRelax*), the time spent to find the value of the linear relaxation (*T-RL*). We also report the GAP between the solution found and the linear relaxation (*GAP MLF/RL*, *GAP KVV/RL*) for each of the formulations. The results in bold remark different linear relaxations found for the same problem instance. Underlined results indicate that after 10800 seconds, the optimal solution was not found.

The GAPs were calculated using the expression: $\frac{|Sol - LRelax|}{LRelax}$.

	MLF					KVV				
	Sol	**Time**	**LRelax**	**T-RL**	**GAP MLF/RL**	**Sol**	**Time**	**LRelax**	**T-RL**	**GAP KVV/RL**
10-0.3-5-1	3942	0.009	3942	0.004	0.000	3942	0.013	3942	0.006	0.000
10-0.3-5-2	4552	0.004	4552	0.003	0.000	4552	0.03	4552	0.005	0.000
10-0.3-5-3	5762	0.004	5762	0.016	0.000	5762	0.016	5762	0.02	0.000
10-0.3-5-4	4811	0.003	4811	0.002	0.000	4811	0.017	4811	0.005	0.000
10-0.3-5-5	4831	0.004	4831	0.003	0.000	4831	0.018	4831	0.005	0.000
10-0.3-10-1	8331	0.015	8177	0.015	0.019	8331	0.099	8177	0.013	0.019
10-0.3-10-2	8812	0.012	8812	0.006	0.000	8812	0.027	8812	0.009	0.000
10-0.3-10-3	10016	0.009	10016	0.006	0.000	10016	0.027	10016	0.023	0.000
10-0.3-10-4	8750	0.007	8750	0.005	0.000	8750	0.026	8750	0.008	0.000
10-0.3-10-5	10130	0.022	10130	0.005	0.000	10130	0.027	10130	0.009	0.000
10-0.3-15-1	12490	0.018	12490	0.019	0.000	12490	0.036	12490	0.024	0.000
10-0.3-15-2	17417	0.021	17417	0.007	0.000	17417	0.238	17417	0.011	0.000
10-0.3-15-3	12378	0.018	12378	0.007	0.000	12378	0.037	12378	0.01	0.000
10-0.3-15-4	10988	0.024	10970	0.007	0.002	10988	0.057	10970	0.02	0.002
10-0.3-15-5	9066	0.015	9066	0.024	0.000	9066	0.036	9066	0.011	0.000
20-0.3-10-1	5978	0.742	5738	0.026	0.042	5978	1.576	5738	0.075	0.042
20-0.3-10-2	10469	5.555	9562.24	0.055	0.095	10469	8.125	9562.24	0.099	0.095
20-0.3-10-3	7020	4.342	6637.33	0.035	0.058	7020	5.275	6637.33	0.064	0.058
20-0.3-10-4	5484	3.095	4849.83	0.042	0.131	5484	4.713	4849.83	0.072	0.131
20-0.3-10-5	7932	6.084	7403.2	0.035	0.071	7932	6.318	7403.2	0.066	0.071
20-0.3-20-1	9488	0.947	9430.5	0.056	0.006	9488	0.508	9430.5	0.127	0.006
20-0.3-20-2	11521	3.841	11032.3	0.064	0.044	11521	7.676	11032.3	0.123	0.044
20-0.3-20-3	8270	1.415	7969.25	0.059	0.038	8270	1.183	7969.25	0.139	0.038
20-0.3-20-4	11901	23.486	11448.3	0.101	0.040	11901	4.596	11448.3	0.188	0.040
20-0.3-20-5	9656	2.132	9379.5	0.06	0.029	9656	2.088	9379.5	0.115	0.029
20-0.3-30-1	12510	1.911	12374.5	0.102	0.011	12510	1.787	12374.5	0.235	0.011
20-0.3-30-2	14216	3.178	13946.5	0.092	0.019	14216	1.646	13946.5	0.226	0.019
20-0.3-30-3	13393	6.91	12742	0.114	0.051	13393	8.526	12742	0.248	0.051
20-0.3-30-4	14452	2.751	14304.5	0.119	0.010	14452	2.169	14304.5	0.267	0.010
20-0.3-30-5	11419	1.938	11274.5	0.09	0.013	11419	1.164	11274.5	0.21	0.013
30-0.3-15-1	7840	2.645	7718	0.093	0.016	7840	1.052	7718	0.265	0.016
30-0.3-15-2	9479	15.435	8796.12	0.15	0.078	9479	15.596	8796.12	0.357	0.078
30-0.3-15-3	7045	6.558	6878.76	0.146	0.024	7045	5.605	6878.76	0.321	0.024
30-0.3-15-4	8426	36.702	7712.12	0.229	0.093	8426	16.475	7712.12	0.427	0.093
30-0.3-15-5	8792	120.837	7951.73	0.194	0.106	8792	22.648	7951.73	0.377	0.106
30-0.3-30-1	13219	10.891	12982	0.274	0.018	13219	17.957	12982	0.805	0.018
30-0.3-30-2	13117	49.375	12149.8	0.491	0.080	13117	51.671	12149.8	1.066	0.080
30-0.3-30-3	13541	24.305	12702.2	0.449	0.066	13541	44.115	12702.2	0.913	0.066
30-0.3-30-4	12789	41.188	11646.1	0.427	0.098	12789	137.107	11646.1	1.029	0.098
30-0.3-30-5	11897	9.279	11624.5	0.258	0.023	11897	3.622	11624.5	0.728	0.023
30-0.3-45-1	15938	19.941	15582.7	0.47	0.023	15938	37.223	15582.7	1.399	0.023
30-0.3-45-2	13196	38.985	12693.8	0.605	0.040	13196	140.817	12693.8	1.542	0.040
30-0.3-45-3	18893	317.888	17614.5	0.526	0.073	18893	550.705	17614.5	1.259	0.073
30-0.3-45-4	17629	39.199	16845	0.582	0.047	17629	82.705	16845	1.394	0.047
30-0.3-45-5	16392	350.022	15237	0.636	0.076	16392	709.686	15237	1.581	0.076
Avg		25.594		0.149	0.034		42.112		0.457	0.034

Table 5.1: Computational results for MLF Formulation and KVV Formulation for 0.3 density instances

	MLF					KVV				
	Sol	**Time**	**LRelax**	**T-RL**	**GAP MLF/RL**	**Sol**	**Time**	**LRelax**	**T-RL**	**GAP KVV/RL**
10-0.5-5-1	4360	0.022	4360	0.5	0.000	4360	0.018	4360	0.009	0.000
10-0.5-5-2	1351	0.022	1351	0.021	0.000	1351	0.034	1351	0.026	0.000
10-0.5-5-3	2932	0.02	2932	0.5	0.000	2932	0.017	2932	0.008	0.000
10-0.5-5-4	4920	0.126	4475	0.019	0.099	4920	0.604	4475	0.015	0.099
10-0.5-5-5	4469	0.049	4280.5	0.006	0.044	4469	0.04	4280.5	0.01	0.044
10-0.5-10-1	7536	0.143	7363.67	0.029	0.023	7536	0.221	7363.67	0.017	0.023
10-0.5-10-2	7263	0.183	6985.67	0.01	0.040	7263	0.209	6985.67	0.031	0.040
10-0.5-10-3	5273	0.041	5273	0.02	0.000	5273	0.035	5273	0.018	0.000
10-0.5-10-4	5854	0.044	5854	0.01	0.000	5854	0.035	5854	0.017	0.000
10-0.5-10-5	4983	1.312	4520.14	0.01	0.102	4983	1.349	4520.14	0.016	0.102
10-0.5-15-1	9341	0.132	9289	0.014	0.006	9341	0.148	9289	0.026	0.006
10-0.5-15-2	6669	0.069	6669	0.025	0.000	6669	0.052	6669	0.021	0.000
10-0.5-15-3	10324	0.228	10085	0.015	0.024	10324	0.836	10085	0.023	0.024
10-0.5-15-4	6339	0.242	6150.17	0.029	0.031	6339	0.231	6150.17	0.034	0.031
10-0.5-15-5	9517	0.941	9171	0.015	0.038	9517	0.607	9171	0.027	0.038
20-0.5-10-1	4784	2.668	4388.44	0.043	0.090	4784	1.521	4388.44	0.105	0.090
20-0.5-10-2	7689	4.319	7301	0.068	0.053	7689	5.914	7301	0.127	0.053
20-0.5-10-3	6184	0.371	6068	0.047	0.019	6184	0.4	6068	0.1	0.019
20-0.5-10-4	5189	0.754	4898	0.039	0.059	5189	2.68	4898	0.104	0.059
20-0.5-10-5	6051	6.865	5673.25	0.051	0.067	6051	9.237	5673.25	0.112	0.067
20-0.5-20-1	8816	6.188	8545.67	0.099	0.032	8816	4.07	8545.67	0.223	0.032
20-0.5-20-2	8584	1.468	8371	0.114	0.025	8584	11.063	8371	0.209	0.025
20-0.5-20-3	7560	16.505	7056.25	0.131	0.071	7560	4.899	7056.25	0.265	0.071
20-0.5-20-4	7634	1.11	7549	0.111	0.011	7634	1.173	7549	0.237	0.011
20-0.5-20-5	8270	8.697	7848.25	0.134	0.054	8270	9.746	7848.25	0.302	0.054
20-0.5-30-1	10156	1.351	10109	0.166	0.005	10156	1.518	10109	0.35	0.005
20-0.5-30-2	11403	16.107	10946.8	0.2	0.042	11403	13.946	10946.8	0.439	0.042
20-0.5-30-3	11600	26.74	10750.8	0.224	0.989	11600	27.87	10750.8	0.49	0.989
20-0.5-30-4	11785	13.18	11195	0.163	0.053	11785	9.441	11195	0.396	0.053
20-0.5-30-5	9559	5.35	9103.83	0.188	0.050	9559	13.513	9103.83	0.39	0.050
30-0.5-15-1	5830	13.857	5591.33	0.297	0.043	5830	9.648	5591.33	0.809	0.043
30-0.5-15-2	6543	20.338	5861.33	0.218	0.116	6543	23.706	5861.33	0.736	0.116
30-0.5-15-3	5683	17.303	5282	0.217	0.076	5683	12.608	5282	0.639	0.076
30-0.5-15-4	5584	9.253	5403.8	0.264	0.033	5584	13.258	5403.8	0.981	0.033
30-0.5-15-5	5794	30.324	5453	0.379	0.063	5794	41.195	5453	0.944	0.063
30-0.5-30-1	8588	7.633	8161	0.493	0.052	8588	22.032	8161	2.006	0.052
30-0.5-30-2	8756	58.122	8122.75	0.874	0.078	8756	120.65	8122.75	2.119	0.078
30-0.5-30-3	10591	939.743	9222.46	1.41	0.148	10591	1904.09	9222.46	2.993	0.148
30-0.5-30-4	8114	75.07	7051.33	0.618	0.151	8114	1219.35	7051.33	1.995	0.151
30-0.5-30-5	12687	546.042	10954	0.936	0.158	12687	533.773	10954	2.185	0.158
30-0.5-45-1	10189	51.037	9687.5	0.913	0.052	10189	524.257	9687.5	3.542	0.052
30-0.5-45-2	10490	81.547	10081.2	1.361	0.041	10490	203.743	10081.2	3.789	0.041
30-0.5-45-3	13670	714.77	12472.9	1.603	0.096	13670	2581.85	12472.9	4.262	0.096
30-0.5-45-4	9637	72.56	8727.5	0.928	0.104	9637	631.575	8727.5	3.799	0.104
30-0.5-45-5	11609	17.074	11487	0.925	0.011	11609	28.737	11487	3.975	0.011
Avg		61.553		0.321	2.509		177.598		0.916	2.509

Table 5.2: Computational results for MLF Formulation and KVV Formulation for 0.5 density instances

	MLF					KVV				
	Sol	**Time**	**LRelax**	**T-RL**	**GAP MLF/RL**	**Sol**	**Time**	**LRelax**	**T-RL**	**GAP KVV/RL**
10-0.8-5-1	3619	0.034	3619	0.007	0.000	3619	0.038	3619	0.015	0.000
10-0.8-5-2	3480	0.297	3177.5	0.02	0.095	3480	0.526	3177.5	0.037	0.095
10-0.8-5-3	3018	0.344	2829.5	0.009	0.067	3018	0.226	2829.5	0.016	0.067
10-0.8-5-4	3518	0.157	3229	0.019	0.090	3518	0.356	3229	0.018	0.090
10-0.8-5-5	3871	0.061	3762.5	0.007	0.029	3871	0.276	3762.5	0.016	0.029
10-0.8-10-1	5813	0.163	5685	0.022	0.023	5813	0.126	5685	0.029	0.023
10-0.8-10-2	5040	0.29	**4831.48**	0.012	0.043	5040	0.426	**4831.28**	0.027	0.043
10-0.8-10-3	3499	0.071	3499	0.026	0.000	3499	0.076	3499	0.028	0.000
10-0.8-10-4	5364	0.062	5364	0.015	0.000	5364	0.086	5364	0.031	0.000
10-0.8-10-5	4133	3.126	3856	0.016	0.072	4133	4.156	3856	0.031	0.072
10-0.8-15-1	6822	0.194	6816	0.038	0.001	6822	0.198	6816	0.041	0.001
10-0.8-15-2	5183	0.228	5172.75	0.022	0.002	5183	0.236	5172.75	0.04	0.002
10-0.8-15-3	4523	0.23	4448.5	0.021	0.017	4523	0.716	4448.5	0.044	0.017
10-0.8-15-4	7484	0.241	7361.5	0.022	0.017	7484	0.276	7361.5	0.04	0.017
10-0.8-15-5	3843	0.104	3843	0.02	0.000	3843	0.106	3843	0.044	0.000
20-0.8-10-1	3947	0.354	3947	0.097	0.000	3947	0.566	3947	0.25	0.000
20-0.8-10-2	3743	9.408	3437.9	0.111	0.089	3743	5.867	3437.9	0.271	0.089
20-0.8-10-3	3412	0.351	3412	0.061	0.000	3412	0.248	3412	0.205	0.000
20-0.8-10-4	4086	6.159	3802.36	0.081	0.075	4086	6.366	3802.36	0.215	0.075
20-0.8-10-5	4498	3.69	4260.6	0.071	0.056	4498	12.194	4260.6	0.205	0.056
20-0.8-20-1	5796	4.305	5633.5	0.138	0.029	5796	2.366	5633.5	0.563	0.029
20-0.8-20-2	7037	83.211	5938.79	0.314	0.185	7037	143.515	5938.79	0.671	0.185
20-0.8-20-3	4596	4.237	4095	0.116	0.122	4596	14.987	4095	0.588	0.122
20-0.8-20-4	4851	2.396	4681	0.149	0.036	4851	6.309	4681	0.47	0.036
20-0.8-20-5	6086	15.279	5469.62	0.174	0.113	6086	15.556	5469.62	0.496	0.113
20-0.8-30-1	7769	3.218	7534.75	0.314	0.031	7769	13.397	7534.75	0.831	0.031
20-0.8-30-2	7681	12.023	7268.36	0.216	0.057	7681	17.751	7268.36	0.808	0.057
20-0.8-30-3	5144	16.773	4539.5	0.208	0.133	5144	19.139	4539.5	0.769	0.133
20-0.8-30-4	7188	58.195	6236.48	0.442	0.153	7188	98.558	6236.48	1.124	0.153
20-0.8-30-5	7374	16.812	6999	0.276	0.054	7374	44.429	6999	0.981	0.054
30-0.8-15-1	3061	3.093	3023.5	0.329	0.012	3061	33.67	3023.5	1.649	0.012
30-0.8-15-2	3458	2.867	3393	0.299	0.019	3458	8.024	3393	1.773	0.019
30-0.8-15-3	4729	176.273	3865.62	0.371	0.223	4729	248.011	3865.62	1.848	0.223
30-0.8-15-4	6693	33.87	6191.57	0.607	0.081	6693	78.789	6191.57	1.866	0.081
30-0.8-15-5	5991	50.065	5562.25	0.519	0.077	5991	85.277	5562.25	1.946	0.077
30-0.8-30-1	4830	11.336	4597.5	0.591	0.051	4830	5135.54	4597.5	4.161	0.051
30-0.8-30-2	6989	274.313	6155.95	1.284	0.135	6989	711.804	6155.95	5.425	0.135
30-0.8-30-3	7746	4726.12	6329.17	1.329	0.224	7751	10800	6329.17	5.039	0.225
30-0.8-30-4	8384	822.508	7322.75	1.422	0.145	8384	7707.25	7322.75	5.321	0.145
30-0.8-30-5	7428	204.24	6719.5	0.864	0.105	7428	1685.07	6719.5	4.875	0.105
30-0.8-45-1	6289	72.658	5964.5	1.03	0.054	6289	86.337	5964.5	7.498	0.054
30-0.8-45-2	8485	162.386	7861.75	2.486	0.079	8485	4650.18	7861.75	9.401	0.079
30-0.8-45-3	7751	1554.03	6455.5	1.274	0.201	17300.9	10800	6455.5	7.83	1.680
30-0.8-45-4	9419	8509.33	7389.33	1.813	0.275	9711	10800	7389.33	8.496	0.314
30-0.8-45-5	7884	141.935	7220	1.185	0.092	7884	9063	7220	7.153	0.092
Avg		377.489		0.410	0.075		1384.676		1.849	0.108

Table 5.3: Computational results for MLF Formulation and KVV Formulation for 0.8 density instances

The presented results show that the MLF formulation is a lot faster than the KVV formulation and that although we proved in Chapter 3 that the linear relaxation of MLF formulation is stronger than KVV's, we can see that for this set of problem instances the theoretical strenght made no difference, except for one problem instance.

5.2 Results for the Column Generation and the LBound Method

For the tables presented in this section, we report the GAP (GAP) between the lower bound found by the method and the optimal solutions presented in the previous section and the time ($Time$) needed to find the presented lower bound. At the end the average time spent by each method for each density is presented. Whenever a "-" symbol is presented in the GAP column, it means that the method was not able to find a lower bound in the time limit of 36000 seconds.

	Column Generation		LBound	
	GAP	**Time**	**GAP**	**Time**
10-0.3-5-1	0	0.094	0	0.006
10-0.3-5-2	0	0.05	0	0.006
10-0.3-5-3	0	0.08	0	0.006
10-0.3-5-4	0	0.049	0	0.008
10-0.3-5-5	0	0.144	0	0.005
10-0.3-10-1	0	0.498	0	0.028
10-0.3-10-2	0	0.404	0	0.009
10-0.3-10-3	0	0.518	0.158	0.013
10-0.3-10-4	0	1.084	0	0.009
10-0.3-10-5	0	0.791	0	0.009
10-0.3-15-1	0	1.072	0	0.013
10-0.3-15-2	0	0.573	0	0.035
10-0.3-15-3	0	0.71	0	0.013
10-0.3-15-4	0	0.963	0	0.038
10-0.3-15-5	0	0.971	0	0.013
20-0.3-10-1	0	35.176	0	0.609
20-0.3-10-2	0	509.108	0	3.23
20-0.3-10-3	0	316.727	0	4.814
20-0.3-10-4	0	344.741	0	3.04
20-0.3-10-5	0	78.355	0.017	0.936
20-0.3-20-1	0	270.95	0	0.302
20-0.3-20-2	0	176.931	0	2.944
20-0.3-20-3	0	112.011	0	0.389
20-0.3-20-4	0	1378.36	0.040	0.129
20-0.3-20-5	0	671.558	0	2.03
20-0.3-30-1	0	1987.51	0	0.561
20-0.3-30-2	0	1326.47	0	1.072
20-0.3-30-3	0	1774.48	0	2.271
20-0.3-30-4	0	2687.34	0	0.679
20-0.3-30-5	0	1529.77	0	0.558
30-0.3-15-1	0	734.39	0	1.036
30-0.3-15-2	0	880.898	0	8.849
30-0.3-15-3	0	991.19	0	3.381
30-0.3-15-4	0	3166.14	0	24.965
30-0.3-15-5	0	6396.6	0.003	19.667
30-0.3-30-1	0	13250.6	0	2.843
30-0.3-30-2	0.079	36000	0	47.993
30-0.3-30-3	-	36000	0	6.57
30-0.3-30-4	0.098	36000	0	16.537
30-0.3-30-5	-	36000	0	4.303
30-0.3-45-1	0.023	36000	0	4.176
30-0.3-45-2	-	36000	0	7.939
30-0.3-45-3	0.064	36000	0.005	43.842
30-0.3-45-4	0.040	36000	0	18.51
30-0.3-45-5	0.076	36000	0.005	33.84
Avg		8068.480		5.960

Table 5.4: Computational results for the Column Generation and the LBound Method for 0.3 density instances

	Column Generation		LBound	
	GAP	**Time**	**GAP**	**Time**
10-0.5-5-1	0	0.119	0	0.009
10-0.5-5-2	0	1.114	0	0.009
10-0.5-5-3	0	0.125	0	0.008
10-0.5-5-4	0	1.004	0	0.118
10-0.5-5-5	0	0.972	0	0.032
10-0.5-10-1	0	6.514	0	0.12
10-0.5-10-2	0	6.1	0	0.183
10-0.5-10-3	0	4.849	0	0.015
10-0.5-10-4	0	2.843	0	0.016
10-0.5-10-5	0	11.734	0	0.776
10-0.5-15-1	0	4.31	0	0.068
10-0.5-15-2	0	5.172	0	0.02
10-0.5-15-3	0	8.496	0	0.129
10-0.5-15-4	0	20.963	0	0.176
10-0.5-15-5	0	14.841	0	0.419
20-0.5-10-1	0	329.023	0	1.185
20-0.5-10-2	0	535.369	0	0.64
20-0.5-10-3	0	48.851	0	0.207
20-0.5-10-4	0	146.497	0	0.527
20-0.5-10-5	0	167.779	0.012	4.38
20-0.5-20-1	0	904.311	0	4.339
20-0.5-20-2	0	3114.89	0	2.792
20-0.5-20-3	0	1512.63	0	5.698
20-0.5-20-4	0	938.762	0	0.513
20-0.5-20-5	0	1000.07	0	2.911
20-0.5-30-1	0	3860.83	0	0.796
20-0.5-30-2	0	5214.72	0	8.587
20-0.5-30-3	0	10144.9	0	8.62
20-0.5-30-4	0	4564.65	0	3.396
20-0.5-30-5	0	14875.1	0	2.482
30-0.5-15-1	0	36000	0	6.983
30-0.5-15-2	0	8751.52	0	9.821
30-0.5-15-3	0	5710.47	0	5.722
30-0.5-15-4	-	36000	0	8.264
30-0.5-15-5	0	17371.8	0	19.863
30-0.5-30-1	0.006	36000	0	8.106
30-0.5-30-2	0.065	36000	0	33.723
30-0.5-30-3	-	36000	0	516.239
30-0.5-30-4	-	36000	0	51.8
30-0.5-30-5	-	36000	0	205.517
30-0.5-45-1	-	36000	0	13.851
30-0.5-45-2	-	36000	0	47.105
30-0.5-45-3	-	36000	0	154.834
30-0.5-45-4	-	36000	0	38.671
30-0.5-45-5	-	36000	0	9.793
Avg		11411		26.210

Table 5.5: Computational results for the Column Generation and the LBound Method for 0.5 density instances

	Column Generation		LBound	
	GAP	**Time**	**GAP**	**Time**
10-0.8-5-1	0	6.031	0	0.013
10-0.8-5-2	0	15.771	0	0.171
10-0.8-5-3	0	3.801	0	0.137
10-0.8-5-4	0	2.48	0	0.051
10-0.8-5-5	0	7.283	0	0.163
10-0.8-10-1	0	14.814	0	0.098
10-0.8-10-2	0	1.723	0.023	0.275
10-0.8-10-3	0	15.016	0	0.025
10-0.8-10-4	0	6.19	0	0.024
10-0.8-10-5	0	36.361	0	1.668
10-0.8-15-1	0	37.237	0	0.106
10-0.8-15-2	0	18.633	0	0.105
10-0.8-15-3	0	36.502	0	0.191
10-0.8-15-4	0	10.209	0	0.131
10-0.8-15-5	0	210.75	0	0.036
20-0.8-10-1	0	329.368	0	0.107
20-0.8-10-2	0	26096.9	0	4.56
20-0.8-10-3	0	2275.31	0	0.096
20-0.8-10-4	0	2511.77	0	3.728
20-0.8-10-5	0	824.617	0	2.228
20-0.8-20-1	0	1765.21	0	3.889
20-0.8-20-2	0	16218.5	0.112	21.938
20-0.8-20-3	0	4351.07	0	1.509
20-0.8-20-4	0	18389.5	0	4.833
20-0.8-20-5	0	3438.02	0	9.486
20-0.8-30-1	0	15016.4	0	1.413
20-0.8-30-2	0	15362.3	0	5.076
20-0.8-30-3	0.114	36000	0	5.454
20-0.8-30-4	0.153	36000	0	70.118
20-0.8-30-5	0	26319.9	0	9.804
30-0.8-15-1	0	29240.6	0	1.617
30-0.8-15-2	0	28724.3	0	5.664
30-0.8-15-3	-	36000	0	176.762
30-0.8-15-4	0	34148.7	0	59.054
30-0.8-15-5	0	29007.3	0	17.206
30-0.8-30-1	-	36000	0	32.095
30-0.8-30-2	-	36000	0	390.862
30-0.8-30-3	0.093	35888.7	0	946.722
30-0.8-30-4	-	36000	0	611.239
30-0.8-30-5	0.053	36000	0	105.257
30-0.8-45-1	-	36000	0	24.264
30-0.8-45-2	-	36000	0	236.999
30-0.8-45-3	-	36000	0	308.181
30-0.8-45-4	-	36000	0	1977.7
30-0.8-45-5	-	36000	0	112.556
Avg		16196.1		114.525

Table 5.6: Computational results for the Column Generation and the LBound Method for 0.8 density instances

As the results shown, the Column Generation not only was not capable of finding lower bounds for many of the problem instances (21 out of 135) before reaching the time limit, but also needed a lot more time than the LBound Method. On the other hand, the Column Generation was capable of improving most of the lower bounds founded by the linear relaxation in the previous section.
When analysing the results of the LBound method, one can state that since it improves the lower bounds found by the linear relaxations, achieving the optimal solution in many cases, and its computational time is not high, that this method is a suitable candidate to be combined with heuristic methods so the stopping criterion of these heuristics could be improved.
Just to remark, here we are not presenting a Branch-and-Price scheme, just the Column Generation as a means of finding a better lower bound than the linear relaxation.

5.3 Results for VFHLB and GRASP

In order to calibrate the parameters of the VFHLB (ILS) and the GRASP-DE (GRASP), for the experiments, we use 60% of our data so parameters over fitting could be avoided. The following $StopCriterion$, γ and Δ values were tested: $StopCriterion = \{10$ iterations; 50 iterations; 100 iterations$\}$, $\gamma = \{0.75, 0.85, 0.90\}$ and $\Delta = \{\lceil \frac{|E|}{4} \rceil, \lceil \frac{|E|}{3} \rceil, \lceil \frac{|E|}{2} \rceil\}$. After the tests the parameters were calibrated as: $StopCriterion = 10$ iterations, $\gamma = 0.85$ and $\Delta = \lceil \frac{|E|}{2} \rceil$.
We are comparing the VFHLB results with the results of the GRASP-DE presented at [18], which, to the best of our knowledge, is the best heuristic approach to solve the FCNDP-UOF as shown in [18]. For the presented tables, we report the best solution (*Best Sol*) and best time (*Best Time*) reached by each approach, the average gap (*Avg GAP*) and the gap (*GAP*) using the optimal solution. We also report the average values for time (*Avg Time*) and for solutions (*Avg Sol*). Finally, it is reported standard deviation values for time (*Dev Time*) and solution (*Dev Sol*). The results in bold represent that the optimum has been found.

	GRASP-DE								VFHLB						
	Avg Sol	**Avg Time**	**Dev Sol**	**Dev Time**	**Best Sol**	**Best Time**	**Avg GAP**	**GAP**	**Avg Sol**	**Avg Time**	**Dev Time**	**Best Sol**	**Best Time**	**Avg GAP**	**GAP**
10-0.3-5-1	**3942.00**	1.2870	0.0000	0.0329	**3942**	1.2561	0.0000	0.0000	**3942**	0.0070	0.0017	**3942**	0.0060	0.0000	0.0000
10-0.3-5-2	**4552.00**	1.3267	0.0000	0.0172	**4552**	1.3110	0.0000	0.0000	**4552**	0.0038	0.0004	**4552**	0.0030	0.0000	0.0000
10-0.3-5-3	**5762.00**	1.2470	0.0000	0.0276	**5762**	1.2420	0.0000	0.0000	**5762**	0.0040	0.0000	**5762**	0.0040	0.0000	0.0000
10-0.3-5-4	**4811.00**	1.3150	0.0000	0.0230	**4811**	1.2834	0.0000	0.0000	**4811**	0.0044	0.0009	**4811**	0.0040	0.0000	0.0000
10-0.3-5-5	**4831.00**	1.3158	0.0000	0.0418	**4831**	1.3080	0.0000	0.0000	**4831**	0.0034	0.0005	**4831**	0.0030	0.0000	0.0000
10-0.3-10-1	**8331.00**	2.6486	0.0000	0.0462	**8331**	2.6380	0.0000	0.0000	**8331**	0.0136	0.0021	**8331**	0.0120	0.0000	0.0000
10-0.3-10-2	**8812.00**	2.8110	0.0000	0.0755	**8812**	2.7941	0.0000	0.0000	**8812**	0.0128	0.0024	**8812**	0.0110	0.0000	0.0000
10-0.3-10-3	**10016.00**	2.7410	0.0000	0.0395	**10016**	2.7246	0.0000	0.0000	**10016**	0.0080	0.0007	**10016**	0.0070	0.0000	0.0000
10-0.3-10-4	**8750.00**	2.6676	0.0000	0.0804	**8750**	2.6000	0.0000	0.0000	**8750**	0.0072	0.0004	**8750**	0.0070	0.0000	0.0000
10-0.3-10-5	**10130.00**	2.7004	0.0000	0.0847	**10130**	2.6950	0.0000	0.0000	**10130**	0.0186	0.0040	**10130**	0.0160	0.0000	0.0000
10-0.3-15-1	**12490.00**	4.1740	0.0000	0.1084	**12490**	4.1657	0.0000	0.0000	**12490**	0.0186	0.0036	**12490**	0.0170	0.0000	0.0000
10-0.3-15-2	**17417.00**	4.1920	0.0000	0.0762	**17417**	4.0662	0.0000	0.0000	**17417**	0.0208	0.0013	**17417**	0.0200	0.0000	0.0000
10-0.3-15-3	**12378.00**	4.2074	0.0000	0.1048	**12378**	4.1990	0.0000	0.0000	**12378**	0.0182	0.0045	**12378**	0.0150	0.0000	0.0000
10-0.3-15-4	11007.00	4.2281	0.0000	0.0549	11007	4.1210	0.0017	0.0017	**10988**	0.0196	0.0029	**10988**	0.0170	0.0000	0.0000
10-0.3-15-5	**9066.00**	4.2565	0.0000	0.0537	**9066**	4.2060	0.0000	0.0000	**9066**	0.0158	0.0008	**9066**	0.0150	0.0000	0.0000
20-0.3-10-1	6513.58	15.6530	136.4805	0.3393	6411	15.4965	0.0896	0.0724	**5978**	0.6980	0.0098	**5978**	0.6840	0.0000	0.0000
20-0.3-10-2	10813.30	16.5735	185.6884	0.5755	10664	16.3770	0.0329	0.0186	**10469**	4.7662	0.0886	**10469**	4.6650	0.0000	0.0000
20-0.3-10-3	7286.40	15.9854	132.1352	0.3434	7200	15.6720	0.0379	0.0256	**7020**	3.7044	0.1155	**7020**	3.5470	0.0000	0.0000
20-0.3-10-4	5754.74	15.8370	116.7287	0.3310	5598	15.7103	0.0494	0.0208	**5484**	2.7238	0.0806	**5484**	2.6230	0.0000	0.0000
20-0.3-10-5	8322.00	16.0420	0.0000	0.3995	8322	16.0100	0.0492	0.0492	**7932**	14.4424	0.2933	**7932**	14.1280	0.0000	0.0000
20-0.3-20-1	**9488.00**	32.0957	0.0000	1.3602	**9488**	31.8410	0.0000	0.0000	**9488**	0.8662	0.0272	**9488**	0.8400	0.0000	0.0000
20-0.3-20-2	11699.86	31.6390	201.3070	0.9075	11607	30.9429	0.0155	0.0075	**11521**	3.3546	0.1505	**11521**	3.2080	0.0000	0.0000
20-0.3-20-3	8670.82	32.5660	222.8998	0.7159	8568	32.4357	0.0485	0.0360	**8270**	1.2644	0.0393	**8270**	1.2280	0.0000	0.0000
20-0.3-20-4	12320.58	31.9430	300.0561	1.0738	11985	31.6236	0.0353	0.0071	**11901**	21.8506	0.9442	**11901**	21.0000	0.0000	0.0000
20-0.3-20-5	10379.38	32.1230	178.5869	0.4624	10297	31.9303	0.0749	0.0664	**9656**	1.8926	0.0947	**9656**	1.8190	0.0000	0.0000
20-0.3-30-1	13244.00	49.2763	0.0000	0.7556	13244	48.6920	0.0587	0.0587	**12510**	1.4656	0.0292	**12510**	1.4280	0.0000	0.0000
20-0.3-30-2	14854.90	49.8060	364.8115	1.7615	14737	49.4076	0.0449	0.0366	**14216**	2.2224	0.1063	**14216**	2.1130	0.0000	0.0000
20-0.3-30-3	14687.52	48.1790	577.2804	1.4053	14629	47.7936	0.0967	0.0923	**13393**	5.2596	0.1448	**13393**	5.0720	0.0000	0.0000
20-0.3-30-4	15420.97	48.6160	327.7683	0.6324	15329	48.3243	0.0670	0.0607	**14452**	1.7608	0.0733	**14452**	1.6980	0.0000	0.0000
20-0.3-30-5	12599.00	51.3221	0.0000	1.0764	12599	51.0160	0.1033	0.1033	**11419**	1.3276	0.0398	**11419**	1.2950	0.0000	0.0000
30-0.3-15-1	8529.32	69.3908	263.2338	1.5946	8395	68.5680	0.0879	0.0708	**7840**	2.3482	0.0674	**7840**	2.2900	0.0000	0.0000
30-0.3-15-2	10051.33	65.7535	340.4006	1.0051	10112	64.7180	0.0604	0.0668	**9479**	11.9144	0.2141	**9479**	11.6160	0.0000	0.0000
30-0.3-15-3	7422.75	66.0270	196.0199	1.8967	7281	65.7629	0.0536	0.0335	**7045**	5.4786	0.0389	**7045**	5.4180	0.0000	0.0000
30-0.3-15-4	8775.16	66.4171	168.0749	2.3415	8654	65.8900	0.0414	0.0271	**8426**	26.4730	0.5365	**8426**	25.7670	0.0000	0.0000
30-0.3-15-5	9626.00	66.1244	0.0000	2.0463	9626	65.7300	0.0949	0.0949	**8792**	98.2168	1.1438	**8792**	97.3190	0.0000	0.0000
30-0.3-30-1	15766.28	133.4690	287.2792	2.8935	15286	132.1343	0.1927	0.1564	**13219**	9.4686	0.0500	**13219**	9.4110	0.0000	0.0000
30-0.3-30-2	14308.35	138.7550	252.7530	2.3416	13973	137.6450	0.0908	0.0653	**13117**	35.3648	0.6179	**13117**	34.8360	0.0000	0.0000
30-0.3-30-3	15504.47	139.7580	580.6050	3.8356	15412	137.8014	0.1450	0.1382	**13541**	18.5124	0.3369	**13541**	18.2120	0.0000	0.0000
30-0.3-30-4	14766.19	132.6110	254.0662	2.3091	14649	130.7544	0.1546	0.1454	**12789**	31.2224	1.5092	**12789**	29.8950	0.0000	0.0000
30-0.3-30-5	13841.41	133.6140	307.9978	3.7867	13517	133.0795	0.1634	0.1362	**11897**	8.3360	0.3231	**11897**	8.0590	0.0000	0.0000
30-0.3-45-1	18885.64	204.8792	663.5134	2.6948	18773	200.8620	0.1849	0.1779	**15938**	16.7946	0.6475	**15938**	16.4120	0.0000	0.0000
30-0.3-45-2	14455.60	206.9196	597.8858	4.3356	14200	203.6610	0.0955	0.0761	**13196**	29.0830	0.5499	**13196**	28.5450	0.0000	0.0000
30-0.3-45-3	19346.43	202.7890	340.7032	6.4728	**18893**	202.3834	0.0240	0.0000	**18893**	230.5346	8.5348	**18893**	223.8230	0.0000	0.0000
30-0.3-45-4	19162.29	215.2056	637.8094	4.2326	19048	209.7520	0.0870	0.0805	**17629**	29.6728	0.6131	**17629**	29.2020	0.0000	0.0000
30-0.3-45-5	17909.32	205.4560	231.1970	6.6092	17732	200.9360	0.0926	0.0817	**16392**	250.6620	4.4910	**16392**	246.4560	0.0000	0.0000
Avg	11171.1236	57.2432			11078.3111	56.5236	0.0528	0.0446	10537.2889	19.3746		10537.2889	18.9504	0.0000	0.0000

Table 5.7: Computational results for GRASP-DE and VFHLB approaches for 0.3 density instances

	GRASP-DE								VFHLB						
	Avg Sol	**Avg Time**	**Dev Sol**	**Dev Time**	**Best Sol**	**Best Time**	**Avg GAP**	**GAP**	**Avg Sol**	**Avg Time**	**Dev Time**	**Best Sol**	**Best Time**	**Avg GAP**	**GAP**
10-0.5-5-1	**4360.00**	1.8240	0.0000	0.0568	**4360**	1.8058	0.0000	0.0000	**4360**	0.0086	0.0005	**4360**	0.0080	0.0000	0.0000
10-0.5-5-2	**1351.00**	1.9186	0.0000	0.0632	**1351**	1.9110	0.0000	0.0000	**1351**	0.0092	0.0004	**1351**	0.0090	0.0000	0.0000
10-0.5-5-3	**2932.00**	1.8339	0.0000	0.0533	**2932**	1.8050	0.0000	0.0000	**2932**	0.0082	0.0011	**2932**	0.0070	0.0000	0.0000
10-0.5-5-4	**4920.00**	1.9230	0.0000	0.0662	**4920**	1.8890	0.0000	0.0000	**4920**	0.2516	0.0138	**4920**	0.2430	0.0000	0.0000
10-0.5-5-5	**4469.00**	1.8880	0.0000	0.0604	**4469**	1.8730	0.0000	0.0000	**4469**	0.0184	0.0009	**4469**	0.0180	0.0000	0.0000
10-0.5-10-1	7566.00	3.7367	0.0040	0.1192	7566	3.7070	0.0040	0.0040	**7536**	0.0542	0.0008	**7536**	0.0530	0.0000	0.0000
10-0.5-10-2	7575.96	3.7589	193.3202	0.0645	7442	3.7070	0.0431	0.0246	**7263**	0.3026	0.0027	**7263**	0.3000	0.0000	0.0000
10-0.5-10-3	5399.55	3.7424	131.8461	0.0968	**5273**	3.6980	0.0240	0.0000	**5273**	0.0166	0.0005	**5273**	0.0160	0.0000	0.0000
10-0.5-10-4	5983.61	3.8770	105.7580	0.0847	5901	3.8460	0.0221	0.0080	**5854**	0.0174	0.0009	**5854**	0.0170	0.0000	0.0000
10-0.5-10-5	5102.45	3.7687	66.9842	0.0719	5032	3.7240	0.0240	0.0098	**4983**	0.8284	0.0187	**4983**	0.8060	0.0000	0.0000
10-0.5-15-1	9379.00	5.6480	0.0041	0.1157	9379	5.5350	0.0041	0.0041	**9341**	0.0312	0.0013	**9341**	0.0300	0.0000	0.0000
10-0.5-15-2	7512.00	5.7720	0.0000	0.0759	7512	5.7027	0.1264	0.1264	**6669**	0.0236	0.0013	**6669**	0.0220	0.0000	0.0000
10-0.5-15-3	**10324.00**	5.9603	0.0000	0.1085	**10324**	5.9130	0.0000	0.0000	**10324**	0.3338	0.0041	**10324**	0.3300	0.0000	0.0000
10-0.5-15-4	**6339.00**	5.9380	0.0000	0.2099	**6339**	5.8100	0.0000	0.0000	**6339**	0.0810	0.0025	**6339**	0.0790	0.0000	0.0000
10-0.5-15-5	9519.00	5.9964	0.0002	0.1417	9519	5.9370	0.0002	0.0002	**9517**	4.0846	0.0354	**9517**	4.0300	0.0000	0.0000
20-0.5-10-1	**4784.00**	21.5620	0.0000	0.8304	**4784**	21.4326	0.0000	0.0000	**4784**	2.6538	0.0199	**4784**	2.6310	0.0000	0.0000
20-0.5-10-2	**7689.00**	21.8640	0.0000	0.5656	**7689**	21.7328	0.0000	0.0000	**7689**	1.9200	0.0466	**7689**	1.8770	0.0000	0.0000
20-0.5-10-3	**6184.00**	22.6760	0.0000	0.4702	**6184**	22.4492	0.0000	0.0000	**6184**	0.5824	0.0102	**6184**	0.5670	0.0000	0.0000
20-0.5-10-4	5532.91	22.4149	95.1989	0.2894	5489	22.1930	0.0663	0.0578	**5189**	1.6642	0.0275	**5189**	1.6330	0.0000	0.0000
20-0.5-10-5	6233.72	22.7810	80.4730	0.5918	6172	22.7354	0.0302	0.0200	**6051**	26.7656	0.0977	**6051**	26.6630	0.0000	0.0000
20-0.5-20-1	9964.00	46.5030	0.0000	0.9544	9964	45.8520	0.1302	0.1302	**8816**	2.9528	0.0153	**8816**	2.9320	0.0000	0.0000
20-0.5-20-2	8721.34	47.4527	150.4528	1.8322	**8584**	46.8900	0.0160	0.0000	**8584**	4.4280	0.0511	**8584**	4.3720	0.0000	0.0000
20-0.5-20-3	8354.83	45.7165	214.8412	0.9228	8305	44.6450	0.1051	0.0985	**7560**	7.0656	0.0300	**7560**	7.0130	0.0000	0.0000
20-0.5-20-4	7750.74	45.2840	100.0567	0.8360	7674	44.9217	0.0153	0.0052	**7634**	1.5694	0.0201	**7634**	1.5470	0.0000	0.0000
20-0.5-20-5	8636.00	44.8590	0.0000	1.1159	8636	44.7693	0.0443	0.0443	**8270**	6.0790	0.0509	**8270**	6.0160	0.0000	0.0000
20-0.5-30-1	12600.00	67.9890	0.0000	2.3355	12600	67.9890	0.2406	0.2406	**10156**	1.8056	0.0785	**10156**	1.7300	0.0000	0.0000
20-0.5-30-2	12932.00	68.6630	0.0000	1.9053	12932	68.6630	0.1341	0.1341	**11403**	7.2198	0.2475	**11403**	7.0420	0.0000	0.0000
20-0.5-30-3	13021.40	73.2877	334.7399	1.3527	12867	71.5700	0.1225	0.1092	**11600**	13.7846	0.3707	**11600**	13.5040	0.0000	0.0000
20-0.5-30-4	12333.56	70.8795	317.1527	1.3237	12260	68.8150	0.0465	0.0403	**11785**	6.8018	0.0628	**11785**	6.7190	0.0000	0.0000
20-0.5-30-5	10989.00	69.4657	0.0000	1.8168	10989	69.3270	0.1496	0.1496	**9559**	7.3206	0.0511	**9559**	7.2530	0.0000	0.0000
30-0.5-15-1	6824.93	104.3949	112.2020	3.3325	6744	103.9790	0.1707	0.1568	**5830**	12.4814	0.1506	**5830**	12.3470	0.0000	0.0000
30-0.5-15-2	6888.00	103.6410	0.0000	4.0814	6888	102.8119	0.0527	0.0527	**6543**	20.0704	0.1470	**6543**	19.8560	0.0000	0.0000
30-0.5-15-3	5809.89	109.4442	52.8671	2.4582	5741	107.5090	0.0223	0.0102	**5683**	14.5294	0.1577	**5683**	14.3380	0.0000	0.0000
30-0.5-15-4	6097.00	106.1190	0.0000	2.2691	6097	103.3599	0.0919	0.0919	**5584**	11.6152	0.1212	**5584**	11.4330	0.0000	0.0000
30-0.5-15-5	**5794.00**	108.9972	0.0000	3.4635	**5794**	107.9180	0.0000	0.0000	**5794**	22.8150	1.1958	**5794**	21.9610	0.0000	0.0000
30-0.5-30-1	8823.02	209.1856	151.8084	6.4735	8753	207.9380	0.0274	0.0192	**8588**	28.3988	0.5552	**8588**	27.9600	0.0000	0.0000
30-0.5-30-2	9134.00	212.8090	0.0000	3.6315	9134	211.9578	0.0432	0.0432	**8756**	56.7008	1.7342	**8756**	55.3860	0.0000	0.0000
30-0.5-30-3	10908.73	206.0050	138.0897	4.4661	**10591**	203.9450	0.0300	0.0000	**10591**	445.2772	13.4245	**10591**	432.3970	0.0000	0.0000
30-0.5-30-4	9120.14	210.4039	227.4169	6.4708	9012	209.1490	0.1240	0.1107	**8114**	74.5170	1.2355	**8114**	73.2920	0.0000	0.0000
30-0.5-30-5	13575.00	214.7650	0.0000	3.1942	13575	209.1811	0.0700	0.0700	**12687**	117.1688	2.5543	**12687**	114.8630	0.0000	0.0000
30-0.5-45-1	11160.00	332.1730	0.0000	13.8050	11160	330.1800	0.0953	0.0953	**10189**	31.7414	0.3404	**10189**	31.3220	0.0000	0.0000
30-0.5-45-2	12105.07	319.6155	248.6762	7.4214	12009	316.4510	0.1540	0.1448	**10490**	67.9630	2.6552	**10490**	65.6430	0.0000	0.0000
30-0.5-45-3	15733.00	324.8540	0.0000	6.0844	15733	315.7581	0.1509	0.1509	**13670**	150.6902	8.3190	**13670**	142.3510	0.0000	0.0000
30-0.5-45-4	10910.00	322.2408	0.0000	4.1851	10910	316.5430	0.1321	0.1321	**9637**	76.2042	2.7176	**9637**	73.6180	0.0000	0.0000
30-0.5-45-5	12870.05	314.7070	267.3752	5.6402	12593	310.3011	0.1086	0.0848	**11609**	17.7640	0.4141	**11609**	17.3510	0.0000	0.0000
Avg	8315.8204	87.7409			8270.7111	86.6185	0.0583	0.0527	7781.3333	27.7027		7781.3333	26.9241	0.0000	0.0000

Table 5.8: Computational results for GRASP-DE and VFHLB approaches for 0.5 density instances

	GRASP-DE								VFHLB						
	Avg Sol	**Avg Time**	**Dev Sol**	**Dev Time**	**Best Sol**	**Best Time**	**Avg GAP**	**GAP**	**Avg Sol**	**Avg Time**	**Dev Time**	**Best Sol**	**Best Time**	**Avg GAP**	**GAP**
10-0.8-5-1	4033.83	3.0370	108.4895	0.0314	3986	3.0249	0.1146	0.1014	**3619**	0.0142	0.0004	**3619**	0.0140	0.0000	0.0000
10-0.8-5-2	3535.68	2.9300	60.9944	0.0883	**3480**	2.9100	0.0160	0.0000	**3480**	0.1480	0.0016	**3480**	0.1460	0.0000	0.0000
10-0.8-5-3	3330.27	2.7480	112.2499	0.0442	3317	2.7150	0.1035	0.0991	**3018**	0.2422	0.0041	**3018**	0.2380	0.0000	0.0000
10-0.8-5-4	**3518.00**	2.9770	0.0000	0.0614	**3518**	2.8758	0.0000	0.0000	**3518**	0.0886	0.0009	**3518**	0.0880	0.0000	0.0000
10-0.8-5-5	3960.68	2.7730	80.1635	0.0521	3906	2.7620	0.0232	0.0090	**3871**	0.0390	0.0012	**3871**	0.0380	0.0000	0.0000
10-0.8-10-1	6031.84	5.9723	114.2613	0.0866	5902	5.8210	0.0376	0.0153	**5813**	0.0458	0.0046	**5813**	0.0430	0.0000	0.0000
10-0.8-10-2	5120.64	5.7880	107.2940	0.1479	**5040**	5.6954	0.0160	0.0000	**5040**	1.0682	0.0285	**5040**	1.0440	0.0000	0.0000
10-0.8-10-3	3975.00	5.9039	0.0000	0.1344	3975	5.8570	0.1360	0.1360	**3499**	0.0826	0.0018	**3499**	0.0810	0.0000	0.0000
10-0.8-10-4	5460.55	5.9090	116.8811	0.1723	**5364**	5.7908	0.0180	0.0000	**5364**	0.0876	0.0086	**5364**	0.0830	0.0000	0.0000
10-0.8-10-5	4225.54	5.7690	72.7043	0.1662	4192	5.7690	0.0224	0.0143	**4133**	1.1376	0.0334	**4133**	1.0860	0.0000	0.0000
10-0.8-15-1	6976.61	8.9230	90.2083	0.3180	6935	8.8338	0.0227	0.0166	**6822**	0.1478	0.0040	**6822**	0.1440	0.0000	0.0000
10-0.8-15-2	5276.29	8.8852	77.3640	0.1640	**5183**	8.6770	0.0180	0.0000	**5183**	0.1492	0.0027	**5183**	0.1450	0.0000	0.0000
10-0.8-15-3	5017.00	9.0300	0.0000	0.0780	5017	8.9940	0.1092	0.1092	**4523**	0.6238	0.0080	**4523**	0.6140	0.0000	0.0000
10-0.8-15-4	7663.62	8.9097	64.8997	0.2998	**7484**	8.8390	0.0240	0.0000	**7484**	0.6206	0.0086	**7484**	0.6070	0.0000	0.0000
10-0.8-15-5	4751.60	9.2254	85.5372	0.2468	4686	9.2070	0.2364	0.2194	**3843**	0.4682	0.0066	**3843**	0.4610	0.0000	0.0000
20-0.8-10-1	4120.80	34.3230	105.3503	0.8950	4040	34.3230	0.0440	0.0236	**3947**	0.6520	0.0107	**3947**	0.6440	0.0000	0.0000
20-0.8-10-2	3915.00	34.5080	0.0000	1.1326	3915	34.0249	0.0460	0.0460	**3743**	6.9310	0.2826	**3743**	6.7250	0.0000	0.0000
20-0.8-10-3	3480.24	34.8060	74.7532	0.5791	**3412**	34.3883	0.0200	0.0000	**3412**	0.1918	0.0033	**3412**	0.1880	0.0000	0.0000
20-0.8-10-4	4209.00	35.2740	0.0000	0.8032	4209	34.9940	0.0301	0.0301	**4086**	5.0812	0.1772	**4086**	4.9450	0.0000	0.0000
20-0.8-10-5	4542.98	35.6360	97.5143	0.7726	**4498**	35.2796	0.0100	0.0000	**4498**	4.6612	0.0678	**4498**	4.6030	0.0000	0.0000
20-0.8-20-1	6909.00	70.8823	0.0000	1.7308	6909	69.2210	0.1920	0.1920	**5796**	4.2190	0.0869	**5796**	4.1280	0.0000	0.0000
20-0.8-20-2	7635.54	71.4810	187.0284	1.0189	7590	70.3373	0.0851	0.0786	**7037**	313.3302	20.9517	**7037**	297.9690	0.0000	0.0000
20-0.8-20-3	6251.89	68.9992	89.4775	1.8381	5422	68.1810	0.3603	0.1797	**4596**	5.2952	0.1021	**4596**	5.2230	0.0000	0.0000
20-0.8-20-4	5187.00	70.2559	69.0130	2.4494	5250	69.9760	0.0693	0.0823	**4851**	2.8762	0.0466	**4851**	2.8170	0.0000	0.0000
20-0.8-20-5	6855.53	72.1322	86.2333	1.9296	6267	71.4180	0.1264	0.0297	**6086**	10.8284	0.5098	**6086**	10.5270	0.0000	0.0000
20-0.8-30-1	9425.00	105.0060	0.0000	2.1653	9425	101.2258	0.2132	0.2132	**7769**	7.7738	0.0747	**7769**	7.7040	0.0000	0.0000
20-0.8-30-2	8735.33	110.7691	126.4167	1.9805	8666	109.8900	0.1373	0.1282	**7681**	14.1722	0.2527	**7681**	13.9840	0.0000	0.0000
20-0.8-30-3	5947.89	107.2994	201.4348	2.6665	5889	106.2370	0.1563	0.1448	**5144**	14.6920	0.3542	**5144**	14.4420	0.0000	0.0000
20-0.8-30-4	8768.08	104.7711	177.5349	3.7392	8630	104.5620	0.2198	0.2006	**7188**	48.2594	2.3096	**7188**	46.6700	0.0000	0.0000
20-0.8-30-5	8175.16	108.0789	127.8169	1.4551	7942	108.0789	0.1086	0.0770	**7374**	20.4534	0.6175	**7374**	19.9750	0.0000	0.0000
30-0.8-15-1	3091.61	171.4778	66.3609	0.7593	**3061**	169.7800	0.0100	0.0000	**3061**	4.5098	0.0911	**3061**	4.4170	0.0000	0.0000
30-0.8-15-2	3506.00	160.2209	0.0000	5.1644	3506	160.2209	0.0139	0.0139	**3458**	11.7516	0.2655	**3458**	11.5390	0.0000	0.0000
30-0.8-15-3	5159.56	166.8339	44.5643	2.8985	5139	163.8840	0.0910	0.0867	**4729**	105.0818	7.0616	**4729**	100.5670	0.0000	0.0000
30-0.8-15-4	7312.13	160.7620	161.4134	3.4133	7283	159.4759	0.0925	0.0882	**6693**	53.8938	2.2803	**6693**	52.1000	0.0000	0.0000
30-0.8-15-5	6263.50	164.5370	113.3484	2.7050	6251	162.5860	0.0455	0.0434	**5991**	34.2898	1.3369	**5991**	33.3210	0.0000	0.0000
30-0.8-30-1	4871	332.1400	0.0000	9.2200	4871	330.9080	0.0085	0.0085	**4830**	27.9676	0.5595	**4830**	27.3360	0.0000	0.0000
30-0.8-30-2	7122.2	328.2900	182.3900	4.1100	**6989**	325.3570	0.0191	0.0000	**6989**	296.6414	21.9387	**6989**	279.8210	0.0000	0.0000
30-0.8-30-3	8124	337.1900	16.4300	33.6300	8112	321.8380	0.0488	0.0473	**7746**	2115.6020	49.0532	**7746**	2074.4600	0.0000	0.0000
30-0.8-30-4	**8384**	318.0600	0.0000	26.0900	**8384**	338.2490	0.0000	0.0000	**8384**	530.1420	15.6519	**8384**	520.0250	0.0000	0.0000
30-0.8-30-5	7442.8	321.4300	33.0900	17.8900	**7428**	344.3670	0.0020	0.0000	**7428**	162.6760	2.9126	**7428**	159.9620	0.0000	0.0000
30-0.8-45-1	6633.24	495.3080	118.1999	11.4544	6620	494.3174	0.0547	0.0526	**6289**	48.6748	1.2567	**6289**	47.7090	0.0000	0.0000
30-0.8-45-2	11150.60	489.6256	220.3763	15.3625	10975	489.6256	0.3142	0.2935	**8485**	377.5736	13.7328	**8485**	367.7370	0.0000	0.0000
30-0.8-45-3	9555.00	507.0021	399.7143	17.2257	9555	507.0021	0.2327	0.2327	**7751**	507.0248	20.1638	**7751**	495.2200	0.0000	0.0000
30-0.8-45-4	11214.00	492.2408	0.0000	16.3840	11214	489.3050	0.1906	0.1906	**9419**	2441.1600	31.0341	**9419**	2414.4100	0.0000	0.0000
30-0.8-45-5	8338.56	528.5251	155.1697	7.6185	8080	522.2580	0.0577	0.0249	**7884**	134.6612	1.0177	**7884**	133.4330	0.0000	0.0000
Avg	6115.6400	136.1477			6033.7111	135.9796	0.0866	0.0717	5590.1111	162.5785		5590.1111	159.2763	0.0000	0.0000

Table 5.9: Computational results for GRASP-DE and VFHLB approaches for 0.8 density instances

In Tables 5.7, 5.8, 5.9 were used 135 instances generated by Mautonne, Labbé and Figueiredo [34], whose results were published by them for only 5 instances. For these instances, the computational results suggest the efficiency of VFHLB. On average, the time spent by VFHLB was 2.31 times faster than the time spent by GRASP, being 2.954 times faster for 0.3 density networks, 3.167 times for 0.5 density networks and 0.837 times for 0.8 density networks. Also, VFHLB found all optimal solutions, while GRASP-DE found only 44 optimal solutions. Besides that, the VFHLB also improved or equalled GRASP-DE results for all 135 instances (91 improvements and 44 draws).
Another important remark is that, in Tables 5.7 and 5.8 VFHLB is faster than GRASP-DE, both in the mean of Avg Times and in the mean of Best Times. Although VFHLB lose to GRASP-DE in the mean of Avg Times and in the mean of Best Times on Table 5.9. On the other hand, GRASP-DE finds only 26 % of the optimal solutions while, as told before, VFHLB finds all optimal solutions.
The experiment also showed that, at least for the tested instances, the order of the commodities set by the candidate list in the VFHLB does not change the solution obtained at the end of the algorithm, but does affect the computational time, as standard deviation for the solutions is equal to zero after our experiment (not reported in the tables).

5.3.1 Statistical Analysis

In order to verify whether or not the differences of mean values obtained by the evaluated strategies shown in Tables 5.7,5.8 and 5.9 are statistically significant, we employed the Wilcoxon-Mann-Whitney test technique [21]. This test could be applied to compare algorithms with some random features and identify if the difference of performance between them is due to randomness.
According to [21], this statistical test is used when two independent samples are compared and whenever it is necessary to have a statistical test to reject the null hypothesis, with a significance θ level (i.e., it is possible to reject the null hypothesis with the probability of $((1-\theta) \times 100\%)$). For the sake of this analysis we considered $\theta = 0.01$. The hypotheses considered in this test are:

- Null Hypothesis (H0): there are no significant differences between the solutions found by VFHLB and the original method;
- Alternative Hypothesis (H1): there are significant differences (bilateral alternative) between the solutions found by VFHLB and the GRASP.

Table 5.10 presents the number of better average solutions found by each strategy, for each group of instances separeted by density. The number of cases where the Null Hypothesis was rejected is also shown between parentheses.

Instance	**Algorithms**	
Groups	**GRASP**	**VFHLB**
0.3	0(0)	30(29)
0.5	0(0)	34(31)
0.8	0(0)	43(33)

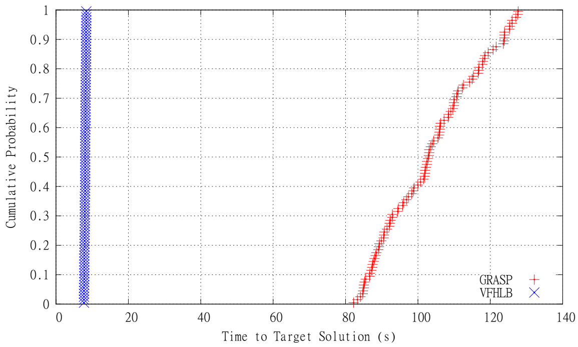
Table 5.10: Statistical Analysis of GRASP and DPRFLB

When comparing GRASP with VFHLB, we notice that almost all differences of performance (86.91% of the tests) are statistically significant. We can also observe that the VFHLB obtained 100% of the best results. These results indicate the superiority of the proposed strategy.

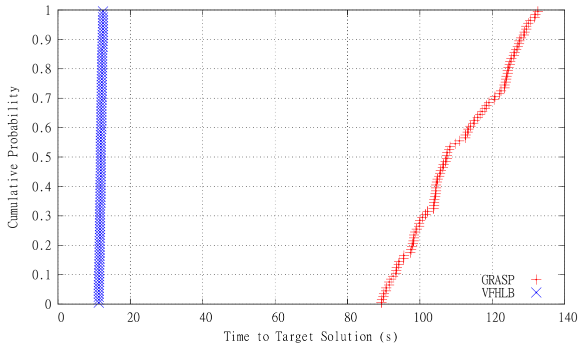
5.3.2 Complementary Analysis

Another way to analyze the behavior of algorithms with random components is provided by time-to-target plots (TTT-plots) [2]. These plots show the cumulative probability of an algorithm reaching a prefixed target solution in the indicated running time. In TTT-plots experiment, we sorted out the execution times required for each algorithm to reach a solution at least as good as a predefined target solution. After that, the i-th sorted running time, t_i, is associated with a probability $p_i = \frac{i-0.5}{100}$ and the points $z_i = (t_i; p_i)$ are plotted.

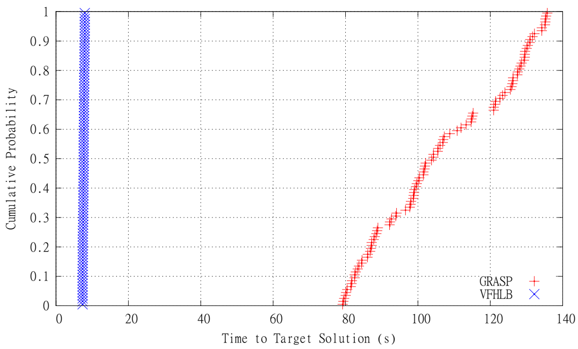
For these experiments we tested 10 of our largest instances with a medium target (1.22 times the cost of the optimal solution). Firstly we analyze the instances with 20 nodes, followed by the analysis of instances with 30 nodes.



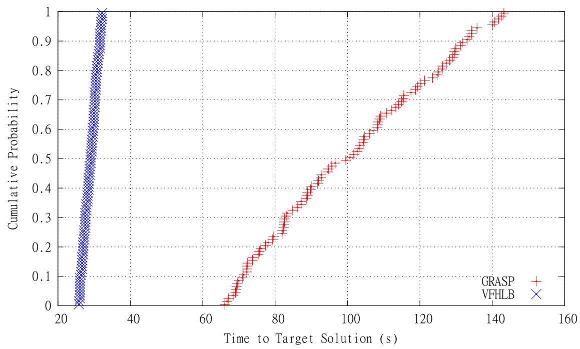
(a) TTTPlot: 20-0.8-30-1



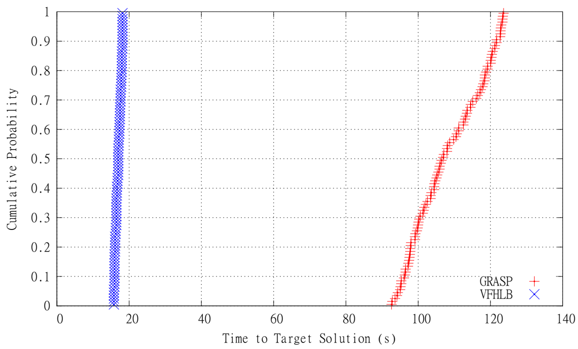
(b) TTTPlot: 20-0.8-30-2



(c) TTTPlot: 20-0.8-30-3



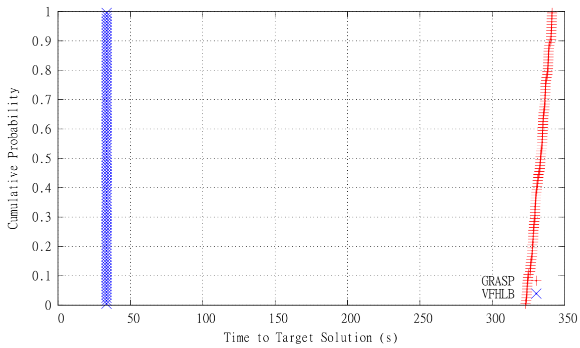
(d) TTTPlot: 20-0.8-30-4



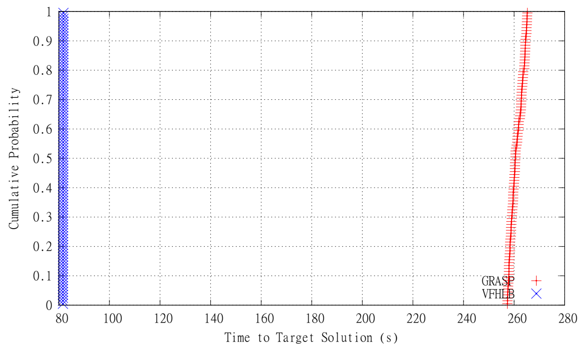
(e) TTTPlot: 20-0.8-30-5

Figure 5.1: TTT Plot - 20 Nodes Instances

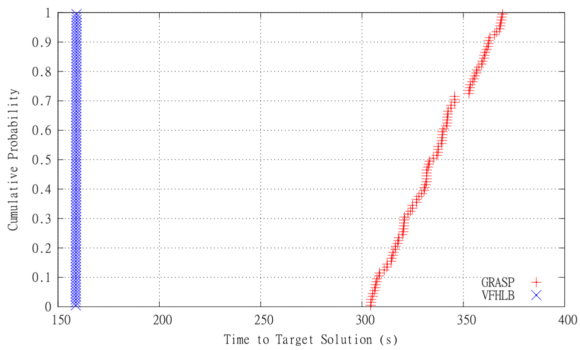
After analyzing the behavior of the methods for the selected instances of 20 nodes, through analysis of the TTTPlot Figures 5.1a to 5.1e, we conclude that the proposed strategy outperforms the GRASP, since the cumulative probability for VFHLB to find the target in less then 40 seconds is 100 %, while for GRASP it is 0 %.



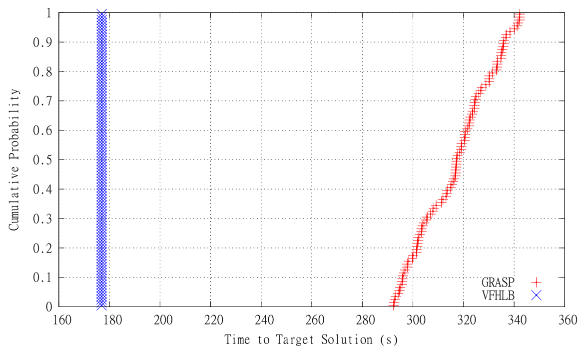
(a) TTTPlot: 30-0.8-30-1



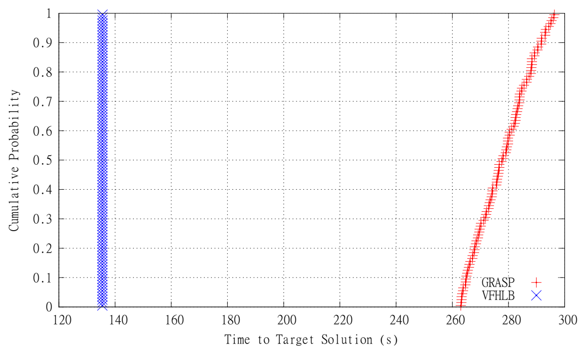
(b) TTTPlot: 30-0.8-30-2



(c) TTTPlot: 30-0.8-30-3



(d) TTTPlot: 20-0.8-30-4



(e) TTTPlot: 30-0.8-30-5

Figure 5.2: TTT Plot - 30 Nodes Instances

After analyzing the behavior of the methods for the selected instances of 30 nodes, through analysis of the TTTPlot Figures 5.2a to 5.2e, we conclude that the proposed strategy outperforms the GRASP, since the cumulative probability for VFHLB to find the target in less then 180 seconds is 100 %, while for GRASP it is 0 %.
Tests with a harder target were performed, but since GRASP-DE took to much time to reach the target, the graphics were not interesting to be presented.

Chapter 6

Conclusions - FCNDP-UOF

Firstly, we compared the MLF Model and the KVV Model theoretically and in practice, which showed us the advantages of using MLF Model instead of KVV Model. After that, besides the good values found by the linear relaxation, very few could be used as a stopping criterion for other methods (since the results were not close enough to the optimal solution), so we developed a Column Generation and the LBound method. Analysing the results we stated that LBound would be a better call to combine with other algorithms.

In the heuristics topic, we proposed new algorithms for FCNDP-UOF, a GRASP-DE and the VFHLB. The GRASP-DE was a straightforward combination of the Partial Decoupling Heuristic (as a constructive method) with the Ejection Route (used as Local Search). The VFHLB uses the VFH to build a initial solution and find a lower bound. In a second moment, a Local Branching technique and a perturbation, Ejection Cycle, are applied to reduce the solution cost.

The proposed approaches were tested on a set of instances grouped by number of nodes, graph density and number of commodities to be transported. Our results have shown the efficiency of VFHLB in comparison to the GRASP presented in [18], since the proposed algorithm finds the optimal solution for all instances and presents a best average time for the majority of the instances (125 out 135).

As future work, we intend to work on exact approaches as Benders' Decomposition and Lagrangian Relaxation since both are very effective for similar problems, as could be seen in [5, 11].

Chapter 7

Introduction to the Transmission Expansion Planning Problem with Redesign

With the growth of energy demand, upgrading energy transmission networks by adding new generators and transmission lines becomes necessary. Since, in most cases, there is the impossibility of building generators near the centers of consumption, efforts need to be focused on the construction of transmission lines. We may consider the example of Brazil, which has huge resources for power generation through hydro-power that, however, are located at great distances from consumption centers. Another feature that can not be neglected is the quantitative variation of the population, especially in countries that are experiencing a significant increase in its population, as is the case in Brazil.

The Transmission Expansion Planning problem (TEP) can be represented by a non-linear mixed integer programming model [35]. This problem is defined on an existing grid, considering some of the critical factors of the power system in question.

This combinatorial optimization problem has physical and budget constraints. Typically, operational and investment restrictions are modelled by linear constraints, but expansion restrictions are modelled through non convex functions, usually bilinear. As seen in [19] and [41], we can turn the bilinear constraints in linear ones through the use of known techniques (Big-M linearization Technique). In this way we can represent the TEP as mixed-integer linear programming problem.

As it can be seen in [27], most of the work on this theme addresses a variant in which only the addition of new transmission lines in the network is considered, i.e. all pre-existing circuits should be part of the new transmission network. However, in [35] a new approach to the problem is presented. This approach, denoted by $\mathrm{TEP_R}$, consists of, not only allowing the addition of new transmission lines, but also the removal of pre-existing transmission lines.

From an economical point of view, one can consider the cost of removing a transmission line to zero. Thus, it was shown recently in [25] and [35], that the $\mathrm{TEP_R}$ always leads to expansion plans cheaper or equal to the TEP. Despite the aforementioned advantages, we must clarify that the $\mathrm{TEP_R}$ is a problem even harder to solve than the TEP, and the authors of [35] failed to solve to optimality benchmark instances of the problem.

As we search in the literature for solution methods for the static variant of TEP, we find works describing primarily the use of metaheuristics like GRASP [7], Taboo Search [12], GRASPR [15], genetic algorithms [39] and at last a study on linear relaxations with Big-M factor [35]. However, when we searched for solving methods for $\mathrm{TEP_R}$, we have seen that, because of its difficulty, few people choose to work with this variant, leading to a small related bibliography [16, 35]. On the other hand, as stated earlier, in [35] the results show that it is not only justifiable to study it, but also encourage the study of this variant, even though it is a more complex variant.

Given the difficulty of solving efficiently the static version of the $\mathrm{TEP_R}$, this thesis presents an exact method as an alternative to the direct use of the mathematical formulation with a commercial solver. The alternative presented is a method we call Ring Partition Search (RPS). By the end of our tests, the results were quite interesting, allowing us to get good quality solutions and in less than the exact approach using the mixed integer linear programming in a commercial solver.

Chapter 8

Mathematical Formulations for the TEP_{R}

This section presents two mathematical formulations for the TEP_{R}, taking into account the Direct Current (DC) model to energy flow [35].

Before defining the mathematical formulations it is necessary to define a transmission network from a mathematical point of view. A transmission network can be represented by a connected graph $\mathcal{G}(\mathcal{B}, \mathcal{L})$, where $\mathcal{B}$ is the set of generators and $\mathcal{L}$ is the set of transmission lines. For better treatment of the circuits, we partitioned the set $\mathcal{L}$ into two disjoint subsets, where $\mathcal{L}^0$ is the set of all the transmission lines already in the network and $\mathcal{L}^1$ is the set of all transmission lines candidate to enter the network. For each transmission line $(i, j) \in \mathcal{L}$ is defined as the direction of energy flow towards $i \to j$ is positive, while towards $j \to i$ is negative. For each generator $i \in \mathcal{B}$ we can create a set of adjacent generators $N(i) \subseteq \mathcal{B}$. By using this set, we can define the subsets $N^+(i) = \{j \in \mathcal{B} : (i, j) \in \mathcal{L}\}$ and $N^-(i) = \{j \in \mathcal{B} : (j, i) \in \mathcal{L}\}$, which helps us to define the flow balancing on the network. It is necessary to also define the demand and the maximum generation of each generator, which are represented respectively by d_i and $\overline{g}_i, i \in \mathcal{B}$. When we talk about direct current, the flow of energy is proportional to the difference between the phase angles in the beginning $(\theta_i, i \in \mathcal{B})$ and in the end $(\theta_j, j \in \mathcal{B})$ of the transmission line $(i, j) \in \mathcal{L}$. The proportionality constant related to the flow in the transmission line (i, j) is called susceptance, which is represented by the symbol γ_{ij}. From a practical point of view, the susceptance is a physical characteristic of the transmission line and therefore a given input. Also, each transmission line has a capacity $(\overline{f}_{ij})$ for limiting the flow past through it. It notes that there may be transmission lines in parallel $(i, j)_1$, $(i, j)_2$, denoted by $(i, j)_1 \parallel (i, j)_2$, connecting the same generators. Finally, defining c_{ij} as the cost of adding the transmission line $(i, j) \in \mathcal{L}^1$, we have all the necessary components to represent the TEP_{R} mathematicaly. To facilitate the understanding, a list of symbols is shown below.

8.1 List of Symbols

$\mathcal{G}(\mathcal{B}, \mathcal{L})$	Associated graph $G(V, E)$.
$\mathcal{B}$	Set of generators.
$\mathcal{L}$	Set of transmission lines.
$\mathcal{L}^0$	Set of existing transmission lines.
$\mathcal{L}^1$	Set of candidate transmission lines.
$\delta^+(i)$	Set of all transmission lines beginning at generator i.
$\delta^-(i)$	Set of all transmission lines ending at generator i.
γ_{ij}	Transmission line $(i, j) \in \mathcal{L}$'s susceptance.
$\overline{f}_{ij}$	Maximum flow allowed in the transmission line $(i, j) \in \mathcal{L}$.
$\overline{g}_i$	Maximum generation allowed in the generator $i \in \mathcal{B}$.
c_{ij}	Operational cost of adding the transmission line $(i, j) \in \mathcal{L}_1$ in the network.
x_{ij}	Indicates whether or not the transmission line $(i, j) \in \mathcal{L}$ is in the final network.
f_{ij}	Indicates the flow in the transmission line $(i, j) \in \mathcal{L}$.
g_i	Indicates the generated energy in the generator $i \in \mathcal{B}$.
Θ_i	Indicates the phase angle in the generator $i \in \mathcal{B}$.

8.2 Mathematical Formulations for the TEP_R

Given the definitions made, a possible nonlinear mixed integer mathematical formulation for the static variant of TEP_R [35] can be written as:

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in\mathcal{L}^1} c_{ij}x_{ij} && && \\
\text{s.t.} \quad & \sum_{j\in\delta^+(i)} f_{ij} - \sum_{j\in\delta^-(i)} f_{ji} + g_i = d_i && \forall i \in \mathcal{B} && (8.1)\\
& f_{ij} - \gamma_{ij}x_{ij}(\theta_i - \theta_j) = 0 && \forall (i,j) \in \mathcal{L} && (8.2)\\
& |f_{ij}| \leq \overline{f}_{ij} && \forall (i,j) \in \mathcal{L} && (8.3)\\
& g_i \leq \overline{g}_i && \forall i \in \mathcal{B} && (8.4)\\
& g_i \geq 0 && \forall i \in \mathcal{B} && (8.5)\\
& \theta_i \in \mathbb{R} && \forall i \in \mathcal{B} && (8.6)\\
& f_{ij} \in \mathbb{R} && \forall (i,j) \in \mathcal{L} && (8.7)\\
& x_{ij} \in \{0,1\} && \forall (i,j) \in \mathcal{L} && (8.8)
\end{aligned}
$$

Constraint (8.1) ensures the flow balance, i.e., the entire flow coming in, less all flow out, plus the energy generated in the generator must match the demand of the generator. Constraint (8.2) regulates the flow behaviour depending on the difference between the phase angles. This phenomenon is governed by the Kirchoff Law [35]. Constraints (8.3)-(8.5) ensure respectively that flow in each transmission line and generation in each generator are larger than their lower bounds and do not exceed their upper bounds. All other constraints define the domain of each of the variables.

8.2.1 A Linear Formulation

In view of the difficulties imposed by the nonlinear constraints, in this thesis we chose to work with the linear formulation of the problem. For this we use the Big-M linearization technique. Given a constant $M_k > 0$, we can replace (8.2) by the following constraints [35]:

$$
-M_{ij}(1-x_{ij}) \leq f_{ij} - \gamma_{ij}(\theta_i - \theta_j) \leq M_{ij}(1-x_{ij}), \forall (i,j) \in \mathcal{L} \tag{8.9}
$$

In this case, the constraint (8.3) needs to be rewritten as:

$$
|f_{ij}| \leq x_{ij}\bar{f}_{ij}, \forall (i,j) \in \mathcal{L} \tag{8.10}
$$

Thus, the model we address as DC model is written as:

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in\mathcal{L}^1} c_{ij}x_{ij} & & \\
\text{s.t.} \quad & \sum_{j\in\delta^+(i)} f_{ij} - \sum_{j\in\delta^-(i)} f_{ji} + g_i = d_i & \forall i \in B \qquad & (8.11)\\
& f_{ij} - \gamma_{ij}(\theta_i - \theta_j) \le M(1 - x_{ij}) & \forall (i,j) \in \mathcal{L} \qquad & (8.12)\\
& - M(1 - x_{ij}) \le f_{ij} - \gamma_{ij}(\theta_i - \theta_j) & \forall (i,j) \in \mathcal{L} \qquad & (8.13)\\
& f_{ij} \le x_{ij}\overline{f}_{ij} & \forall (i,j) \in \mathcal{L} \qquad & (8.14)\\
& f_{ij} \ge -x_{ij}\overline{f}_{ij} & \forall (i,j) \in \mathcal{L} \qquad & (8.15)\\
& g_i \le \overline{g}_i & \forall i \in B \qquad & (8.16)\\
& g_i \ge 0 & \forall i \in B \qquad & (8.17)\\
& \theta_i \in \mathbb{R} & \forall i \in B \qquad & (8.18)\\
& f_{ij} \in \mathbb{R} & \forall (i,j) \in \mathcal{L} \qquad & (8.19)\\
& x_{ij} \in \{0,1\} & \forall (i,j) \in \mathcal{L} \qquad & (8.20)
\end{aligned}
$$

where the constraints (8.12)-(8.15) represent constraints (8.2) and (8.3). One procedure to calculate the Big-M value is described in detail in [35].

Chapter 9

Algorithms for the $\mathrm{TEP_R}$

In this chapter two exact approaches are presented. The first one is a Benders' decomposition and the second one is the Ring Partition Search.

9.1 Benders' Decomposition

In the early stages of the expansion planning through a technique of decomposition, relaxations of the original problem are solved for which convexity conditions are met and optimum solutions can be found (first hierarchical level). The relaxed restrictions are then gradually reintroduced so that at every step we become closer to the solution of the problem. So, initially, we solved a relaxed problem from which some restrictions are removed. The optimal solution obtained, as well as other relevant information, are then reused to start the settlement process of the second hierarchical level, which should contain only continuous variables. After solving the second hierarchical level new constraints are added to the problem of the first level. The process is then repeated until all necessary constraints have been added to the first level.

From a more practical point of view, we defined the first level as responsible for selecting which transmission line will be in the network, taking into consideration a relaxed flow, described by (8.12) and (8.13) before being relaxed. In the second level, the network is already built and the flow must respect again the Kirchoff law, so it will be in function of the $\omega_i \in \mathcal{B}$, leading to a real representation of the flow behaviour.

In view of the described procedure, one can define the elements of each hierarchical level.

9.1.1 First Hierarchical Level

$$\min \quad \omega + \sum_{(i,j)\in\mathcal{L}^1} c_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_{j\in\delta^+(i)} f_{ij} - \sum_{j\in\delta^-(i)} f_{ji} + g_i = d_i \qquad \forall i \in B \tag{9.1}$$

$$f_{ij} \leq x_{ij}\overline{f}_{ij} \qquad \forall (i,j) \in \mathcal{L} \tag{9.2}$$

$$f_{ij} \geq -x_{ij}\overline{f}_{ij} \qquad \forall (i,j) \in \mathcal{L} \tag{9.3}$$

$$0 \leq g_i \leq \overline{g}_i \qquad \forall i \in B \tag{9.4}$$

$$\theta_i \in \mathbb{R} \qquad \forall i \in B \tag{9.5}$$

$$g_i \geq 0 \qquad \forall i \in B \tag{9.6}$$

$$f_{ij} \in \mathbb{R} \qquad \forall (i,j) \in \mathcal{L} \tag{9.7}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{L} \tag{9.8}$$

$$\omega \geq 0 \tag{9.9}$$

$$\omega \in \mathbb{R} \tag{9.10}$$

In this formulation, following the idea of the Benders decomposition [33], we defined a variable ω, which will be used in the construction of the constraints that will be generated on the second level. Furthermore, the flow constraints have been relaxed, leading to what we call the transportation model.

9.1.2 Second Hierarchical Level

$$\min \quad \sum_{i\in B} \lambda_i r_i$$

$$\text{s.t.} \quad \sum_{j\in\delta^+(i)} f_{ij} - \sum_{j\in\delta^-(i)} f_{ji} + g_i = d_i - r_i \qquad \forall i \in B \tag{9.11}$$

$$f_{ij} - \gamma_{ij}(\theta_i - \theta_j) \leq M(1 - \bar{x}_{ij}) \qquad \forall (i,j) \in \mathcal{L} \tag{9.12}$$

$$-M(1 - \bar{x}_{ij}) \leq f_{ij} - \gamma_{ij}(\theta_i - \theta_j) \qquad \forall (i,j) \in \mathcal{L} \tag{9.13}$$

$$f_{ij} \leq \bar{x}_{ij}\overline{f}_{ij} \qquad \forall (i,j) \in \mathcal{L} \tag{9.14}$$

$$f_{ij} \geq -\bar{x}_{ij}\overline{f}_{ij} \qquad \forall (i,j) \in \mathcal{L} \tag{9.15}$$

$$0 \leq g_i \leq \overline{g}_i \qquad i \in B \tag{9.16}$$

$$\theta_i \in \mathbb{R} \qquad \forall i \in B \tag{9.17}$$

$$g_i \geq 0 \qquad \forall i \in B \tag{9.18}$$

$$r_i \geq 0 \qquad \forall i \in B \tag{9.19}$$

$$f_{ij} \in \mathbb{R} \qquad \forall (i,j) \in \mathcal{L} \tag{9.20}$$

In this level, we use the values of the variables x_{ij} found in the previous level as constant, ie, $\bar{x}_{ij}$ is constant equal to x_{ij}, for all $(i,j) \in \mathcal{L}$. In addition, we introduce r_i variables which work as slack variables allowing that the load may not be fulfilled. Thanks to that, we do not deal with extreme rays, for every solution is a feasible solution. Therefore, when the objective function of the sub-problem is equal to zero, then the solution is feasible for the original problem.

Once defined the model of the second hierarchical level, we get the expression that represents the constraint (Benders' Cut) which will be added to the first level problem in each iteration of the method:

$$\begin{aligned} \sum_{i \in B} d_i \beta_i + \sum_{(i,j) \in \Omega} M_{ij}(1 - x_{ij})\zeta^1_{ij} + \sum_{(i,j) \in \Omega} M_{ij}(1 - x_{ij})\zeta^2_{ij} + \\ \sum_{i \in B} \bar{g}_i \phi_i + \sum_{(i,j) \in \Omega} \rho^1_{ij} \bar{f}_{ij} x_{ij} + \sum_{(i,j) \in \Omega} \rho^2_{ij} \bar{f}_{ij} x_{ij} \leq \omega \end{aligned} \tag{9.21}$$

where the values β_i, ζ^1_{ij}, ζ^2_{ij}, ϕ_i, ρ^1_{ij} and ρ^2_{ij} are the values of the dual variables associated with constraints (9.11) - (9.15), obtained by the resolution of the second level problem.

9.2 Ring Partition Search

The idea of RPS is to divide the solution of the problem into two parts. The first part is to fix part of the variables and solve the problem. Thus solving this problem we will get an upper bound of the original problem. A natural choice of variables to be established is to fix all existing transmission lines and thus obtain a TEP solution. The second part is to partition the solution space, using a technique called Local Branching [17], and search each subspace. This is an exact approach, but can be easily used as heuristic. To do that, select a subset of the space to analyse or limit the execution time.

Formally speaking, being $\hat{x} = (\hat{x}_{ij})$ a feasible solution of the TEP_R obtained by solving the TEP model described by the model whose constraints are (8.11)-(8.20), the local branching constraints can be defined as:

$$\sum_{(i,j)\in\mathcal{L}|\hat{x}_{ij}=0} x_{ij} + \sum_{(i,j)\in\mathcal{L}|\hat{x}_{ij}=1} (1 - x_{ij}) \geq \Delta_1 \tag{9.22}$$

$$\sum_{(i,j)\in\mathcal{L}|\hat{x}_{ij}=0} x_{ij} + \sum_{(i,j)\in\mathcal{L}|\hat{x}_{ij}=1} (1 - x_{ij}) \leq \Delta_2 \tag{9.23}$$

where Δ's are non-negative integers indicating the minimum value of x_{ij}, $(i,j) \in \mathcal{L}$ variables that have to be exchanged (Δ_1) and the maximum number that may be exchanged (Δ_2) from one to zero and vice versa. These constraints allow us to make a circular search around the solution found for the TEP, as shown in Figure 9.1.

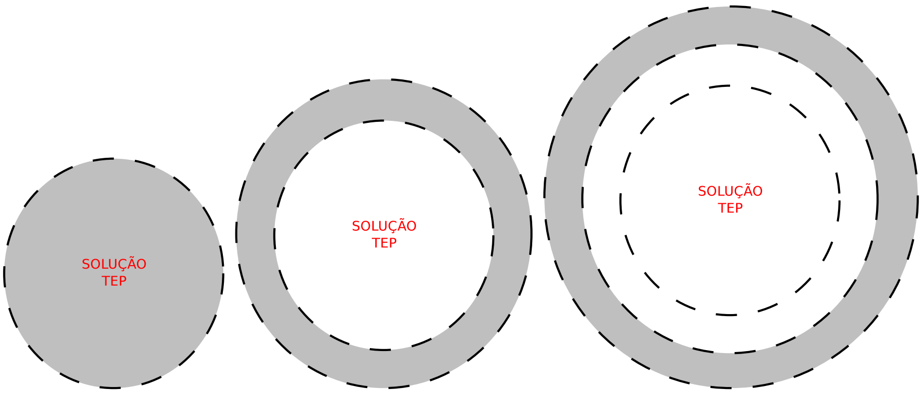


Figure 9.1: RPS Search Space Methodology

This process of creating search subspaces is repeated until the entire space has been explored. The pseudo-code of the method is presented in Algorithm 8:

Algorithm 8: RPS Pseudo-code

Input: $\mathcal{G}$, $MaxIter$, $TimeLimit$

Data: $\Delta_1 \leftarrow 1$, $CParada \leftarrow 1$, $s_{best} \leftarrow \infty$

begin
- $\bar{x} \leftarrow MIP.Solver(TEP)$;
- $s_{best} \leftarrow UpdateBest(\bar{x}, f, g, \theta, s_{best})$;
- **while** *($CParada < MaxIter$ and $gettime() < TimeLimit$)* **do**
  - $\Delta_2 \leftarrow \Delta_2 + \frac{|\Omega|}{MaxIter}$;
  - $LocalBranching(\bar{x}, \Delta_1, \Delta_2)$;
  - $s_{best} \leftarrow UpdateBest(\bar{x}, f, g, \theta, s_{best})$;
  - $\Delta_1 \leftarrow \Delta_2$;
  - $CParada \leftarrow CParada + 1$;
- **end**

end

In the given pseudo-code $getTime()$ function returns the execution time. $MIP.Solver()$ calls the solver for solving a mathematical formulation passed as a parameter and returns the value of our variable vector $x = (x_{ij})$. It is important to remark that to solve the TEP using TEP_R model one just needs to set the variables x_{ij} to one, for all transmission lines $(i, j) \in \mathcal{L}^0$. The $UpdateBest()$ verifies that the solution passed as a parameter is better or not than the best solution cost stored s_{best},if so it saves the value of the variables and updates the s_{best}. Finally, the $LocalBranching()$ adds the local branching constraints, as described by equations (9.22) and (9.23) using the passed parameters and then calls the solver to solve TEP_R's mathematical formulation with the local branching constraints created in this iteration. Remark that since we are working with rings, at each iteration the previously used local branching constraints are removed.

Chapter 10

Computational Experiments for the TEP_R

In this chapter we present computational experiments done using the methods presented for the TEP_R.

The algorithms were coded in Xpress Mosel using FICO Xpress Optimization Suite, on an *Intel ®Core TM i3 CPU 3250 @ 3,5GHz* computer with 8GB of RAM. Computing times are reported in seconds. In order to test the performance of the presented methods, we used 5 benchmark instances of the TEP and 10 instances generated by us using the multistage TEP.

After several tests, the parameter $MaxIter$ was set to 100 iterations. Both the mathematical formulation, the Benders' decomposition and RPS had their maximum execution times set as 10h.

Table 10.1 shows the comparison between the proposed algorithms and the mathematical formulation proposed by [35] presented in Chapter 8. As a matter of comparison, the value of the solution found by each method (*Solution* / *Lower Bound*) and the time spend by them in seconds (*Time*) are presented. Finally on line Avg the average time is presented for each of the three tested methods. Underlined results indicate that the optimum was achieved and proved. The column *Lower Bound* is called Lower Bound because even though we are solving the bender's first level with integer variables, for most of the cases we were not able to find the optimal solution in the given time.

	Mathematical Formulation		Benders' Decomposition		Ring Partition Search	
	Solution	**Time**	**Lower Bound**	**Time**	**Solution**	**Time**
Garver	110.000	0.203	110.000	0.084	110.000	0.211
IEEE24	152.000	4.653	152.000	10655.8	152.000	2.875
South Brazil	63.200	26.052	55.657	36000.000	63.200	7.879
South Brazil Whitout Redispatch	151.985	36000.000	127.272	36000.000	146.200	3184.540
Southeast	907.800	36000.000	284.100	36000.000	424.800	36000.000
IEEE24M_1	151000.000	3.307	151000.000	13285.000	151000.000	3.637
IEEE24M_2	325000.000	83.071	287000.000	36000.000	325000.000	41.421
IEEE24M_3	350000.000	9.395	350000.000	18360.300	350000.000	10.593
IEEE24M_4	182000.000	15.035	168000.000	36000.000	182000.000	4.304
IEEE24M_5	287000.000	42.475	244000.000	36000.000	287000.000	32.518
IEEE46M_1	63163.000	21.893	53334.000	36000.000	63163.000	16.548
IEEE46M_2	148738.000	36000.000	127272.000	36000.000	146242.000	3544.420
Colombian_1	794.644	36000.000	172.200	36000.000	296.454	36000.000
Colombian_2	409.870	36000.000	248.846	36000.000	443.494	36000.000
Colombian_3	773.385	36000.000	315.354	36000.000	562.417	36000.000
Avg		14413.739		29220.079		10056.796

Table 10.1: TEP_R computational experiments.

Table 10.1 shows the superiority of RPS in relation to mathematical formulation, both in solution quality, as in time. As for Benders' decomposition the results show its inefficiency both to find the optimum and to find good quality lower bounds.

Since the RPS can be divided in two parts, where the first consists in solving the TEP and using the solution found as a start solution, a test comparing the RPS and the Mathematical Formulation when the Mathematical Formulation starts with the same upper bound as the RPS was done.

	Mathematical Formulation		Ring Partition Search		TEP's Optimal Solution
	Solution	**Time**	**Solution**	**Time**	
Garver	110.0	0.202	110.0	0.211	110.000
IEEE24	152.0	3,653	152.0	2.875	152.000
South Brazil	63.2	21,882	63.2	7.879	63.200
South Brazil Whitout Redispatch	146.2	9097.35	146.2	3184.540	154.420
Southeast	411,442	36000.000	424.8	36000.000	424.800
IEEE24M_1	151000.0	3.217	151000.0	3.637	152000.000
IEEE24M_2	325000.0	72.679	325000.0	41.421	390000.000
IEEE24M_3	350000.0	9.745	350000.0	10.593	390000.000
IEEE24M_4	182000.0	4.446	182000.0	4.304	218000.000
IEEE24M_5	287000.0	43.963	287000.0	32.518	342000.000
IEEE46M_1	63163.0	16.811	63163.0	16.548	72870.000
IEEE46M_2	146242.0	8164.82	146242.0	3544.420	154420.000
Colombian_1	240.134	36000.000	296.5	36000.000	296.456
Colombian_2	273.907	36000.000	443.1	36000.000	443.494
Colombian_3	323.429	36000.000	562.4	36000.000	562.400
Avg		13752.555		10056.596	

Table 10.2: TEP_R computational experiments with the same initial solutions.

In view of the new experiments one can state that whenever TEP's optimal solution is near TEP_R's the RPS outperforms the mathematical formulation. On the other hand, since the Branch-and-Bound strategy can vary according to the solver criterions, sometimes when a better solution is far from the initial solution, the straightforward use of the mathematical formulation may lead to the discovery of a better solution in less time. Although not presented here, it is possible to change the order of the rings or divide them so each ring takes approximately the same time to be solved.

Chapter 11

Conclusions - TEP_R

In view of the results, we see that not only RPS has found solutions of the same quality (9 out of 15) or better (5 out of 15) then the mathematical formulation. Besides that RPS is 43% faster on average, thus proving its usefulness as an alternative to the direct application of mathematical formulation. Unfortunately when feeding the branch-and-bound with the same initial solution as the RPS, the RPS doesn't find the same or better solutions for 4 out of 15 instances, but is still faster in average. Benders' decomposition was executed until the time limit without reaching the optimal solution not even for several instances where other methods have achieved them very quickly.
Given the results, possible ways to further develop these methods are:

- Try different relaxations to the master problem of decomposition of Benders;
- Remove the artificial variables from Benders' decomposition sub problem;
- Exchange the mathematical formulation of the TEP for a metaheuristic to solve the TEP [7] as the first phase of RSP;
- Try new ring division strategies so the time to solve each ring stays approximately equal.

Chapter 12

Conclusions and Future Works

In this thesis two network design problems were studied. For the FCNDP-UOF mathematical formulations were studied, followed by the study of techniques to find efficient lower bounds. These techniques were combined with several heuristics so high quality solutions could be found in reasonable time. At the end, a GRASP and two hybrids techniques were compared, leading to the acknowledgement of a new state of the art method to the problem. For the TEP_{R}, initial solutions and branch strategies were tested leading to a new exact method and a new heuristic to the problem. The analysis of the results obtained showed that this is an interesting strategy that may also be applied to other network design problems.

As future works, studies to extend the method applied to the TEP_{R} for other network design problems may be done, possibly leading to a general MIP based framework to solve this class of problems. Besides that, decomposition techniques such as Benders' decomposition and lagrangian relaxation mighty be studied for both of the studied problems, since their structures seem to be favourable.

References

[1] Ahuja, R. K.; Magnanti, T. L.; Orlin, J. B. *Network flows: theory, algorithms, and applications.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[2] Aiex, R. M. and Resende, M. G. C. and Ribeiro, C. C. Ttt plots: a perl program to create time-to-target plots. *Optimization Letters 1*, 4 (2007), 355–366.

[3] Amaldi, E.; Bruglieri, M.; Fortz, B. On the hazmat transport network design problem. In *Proceedings of the 5th international conference on Network optimization* (Berlin, Heidelberg, 2011), INOC'11, Springer-Verlag, pp. 327–338.

[4] Bazaraa, M. S.; Jarvis, J. J.; Sherali, H. D. *Linear Programming and Network Flows.* Wiley-Interscience, 2004.

[5] Bektas, T.; Crainic, T. G.; Gendron, B. Lagrangean decomposition for the multicommodity capacitated network design problem. In *Optimization Days 2007* (May 2007).

[6] Billheimer, J. W.; Gray, P. Network Design with Fixed and Variable Cost Elements. *Transportation Science 7*, 1 (feb 1973), 49–74.

[7] Binato, S.; de Oliveira, G.; de Araujo, J. A greedy randomized adaptive search procedure for transmission expansion planning. *Power Systems, IEEE Transactions on 16*, 2 (May 2001), 247–253.

[8] Boesch, F. T. *Large-scale Networks: Theory and Design*, 1 ed. IEEE Press selected reprint series, 1976.

[9] Boyce, D.; Janson, B. A discrete transportation network design problem with combined trip distribution and assignment. *Transportation Research Part B: Methodological 14*, 1-2 (mar 1980), 147–154.

[10] Colson, B.; Marcotte, P.; Savard, G. Bilevel programming: A survey. *4OR 3*, 2 (jun 2005), 87–107.

[11] Costa, A. M.; Cordeau, J.-F.; Gendron, B. Benders, metric and cutset inequalities for multicommodity capacitated network design. *Computational Optimization and Applications 42*, 3 (2009), 371–392.

[12] Da Silva, E.; Ortiz, J.; de Oliveira, G.; Binato, S. Transmission network expansion planning under a tabu search approach. *Power Systems, IEEE Transactions on 16*, 1 (Feb 2001), 62–68.

[13] Erkut, E.; Gzara, F. Solving the hazmat transport network design problem. *Computers & Operations Research 35*, 7 (jul 2008), 2234–2247.

[14] Erkut, E.; Tjandra, S. A.; Verter, V. Hazardous Materials Transportation. In *Handbooks in Operations Research and Management Science*, vol. 14. 2007, ch. 9, pp. 539–621.

[15] Faria, H., J.; Binato, S.; Resende, M.; Falcao, D. Power transmission network design by greedy randomized adaptive path relinking. *Power Systems, IEEE Transactions on 20*, 1 (Feb 2005), 43–49.

[16] Figueiredo, R.; González, P. H.; Poss, M. Transmission expansion planning with re-design - a greedy randomized adaptive search procedure. In *International Conference on Operations Research and Enterprise Systems* (2012), no. 1st, pp. 80–88.

[17] Fischetti, M.; Lodi, A. Local branching. *Mathematical Programming 98*, 1-3 (sep 2003), 23–47.

[18] González, P. H.; Martinhon, C.; Simonetti, L.; Santos, E.; Michelon, P. Uma Metaheurística GRASP para o Problema de Planejamento de Redes com Rotas Ótimas para o Usuário. In *XLV Simpósio Brasileiro de Pesquisa Operacional* (Natal, 2013), pp. 1813–1824.

[19] Granville, S.; Pereira, M. *Analysis of the Linearized Power Flow Model in Benders Decomposition.* Stanford Univ., 1985.

[20] Graves, S. C.; Lamar, B. W. An Integer Programming Procedure for Assembly System Design Problems. *Operations Research 31*, 3 (may 1983), 522–545.

[21] Hettmansperger, T. P.; McKean, J. W. *Robust nonparametric statistical methods.* CRC Press, 1998.

[22] Holmberg, K.; Yuan, D. Optimization of Internet Protocol network design and routing. *Networks 43*, 1 (jan 2004), 39–53.

[23] Johnson, D. S.; Lenstra, J. K.; Kan, A. H. G. R. The complexity of the network design problem. *Networks 8*, 4 (jan 1978), 279–285.

[24] Kara, B. Y.; Verter, V. Designing a Road Network for Hazardous Materials Transportation. *Transportation Science 38*, 2 (may 2004), 188–196.

[25] Khodaei, A.; Shahidehpour, M.; Kamalinia, S. Transmission switching in expansion planning. *Power Systems, IEEE Transactions on 25*, 3 (Aug 2010), 1722–1733.

[26] Kimemia, J.; Gershwin, S. Network flow optimization in flexible manufacturing systems. In *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes* (1978), IEEE, pp. 633–639.

[27] Latorre, G.; Cruz, R.; Areiza, J.; Villegas, A. Classification of publications and models on transmission expansion planning. *Power Systems, IEEE Transactions on 18*, 2 (May 2003), 938–946.

[28] Lourenço, H., O. M.; S., T. *Handbook of Metaheuristics*, vol. 146 of *International Series in Operations Research & Management Science*. Springer US, Boston, MA, 2010.

[29] Luigi De Giovanni. *The Internet Protocol Network Design Problem with Reliability and Routing Constraints*. Tese de Doutorado, Politecnico di Torino, 2004.

[30] Magnanti, T. L. Combinatorial optimization and vehicle fleet planning: Perspectives and prospects. *Networks 11*, 2 (jan 1981), 179–213.

[31] Magnanti, T. L.; Wong, R. T. Network Design and Transportation Planning: Models and Algorithms. *Transportation Science 18*, 1 (feb 1984), 1–55.

[32] Mandl, C. E. A survey of mathematical optimization models and algorithms for designing and extending irrigation and wastewater networks. *Water Resources Research 17*, 4 (aug 1981), 769–775.

[33] Martin, R. K. *Large scale linear and integer optimization: a unified approach*. Springer Science & Business Media, 2012.

[34] Mauttone, A.; Labbé, M.; Figueiredo, R. M. V. A Tabu Search approach to solve a network design problem with user-optimal flows. In *V ALIO/EURO Conference on Combinatorial Optimization* (Buenos Aires, 2008), pp. 1–6.

[35] Moulin, L.; Poss, M.; Sagastizábal, C. Transmission expansion planning with re-design. *Energy Systems 1*, 2 (2010), 113–139.

[36] Nemhauser, G. L.; Wolsey, L. A. *Integer and combinatorial optimization*. Wiley-Interscience, 1999.

[37] Paraskevopoulos, D. C.; Bektas, T.; Crainic, T. G.; Potts, C. N. A Cycle-Based Evolutionary Algorithm for the Fixed-Charge Capacitated Multi-Commodity Network Design Problem. Tech. rep., Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, Montreal, 2013.

[38] Resende, M.; Ribeiro, C. *Handbook of Metaheuristics*. Springer US, 2003.

[39] Silva, I.; Rider, M.; Romero, R.; Garcia, A.; Murari, C. Algoritmo genético de chu-beasley para o problema de planejamento da expansão de sistema da transmissão de energia elétrica. In *Anais do VII Simpósio Brasileiro de Automação Inteligente* (2005), no. 7th.

[40] Simpson, R. W. *Scheduling and routing models for airline systems*. Massachusetts Institute of Technology, Flight Transportation Laboratory, 1969.

[41] Villasana, R.; Garver, L.; Salon, S. Transmission network planning using linear programming. *Power Apparatus and Systems, IEEE Transactions on PAS-104*, 2 (Feb 1985), 349–356.

[42] Wolsey, L. A. *Integer programming*. Wiley-Interscience, New York, NY, USA, 1998.

[43] Wong, R. T. *Accelerating Benders decomposition for network design.* Tese de Doutorado, Massachusetts Institute of Technology, 1978.

[44] Wong, R. T. Worst-Case Analysis of Network Design Problem Heuristics. *SIAM Journal on Algebraic Discrete Methods 1*, 1 (mar 1980), 51–63.