DISSERTATION

Titel der Dissertation

Fiber To The Home,

Cost Optimal Design of Last-Mile Broadband Telecommunication Networks

Verfasser

Dipl. Ing. Peter Putz

angestrebter akademischer Grad

Doktor der Sozial- und Wirtschaftswissenschaften (Dr. rer. soc. oec.)

Wien, 2012

Studienkennzahl lt. Studienblatt:	A 084 881
Dissertationsgebiet lt. Studienblatt:	Informatik
Betreuer:	Univ.-Prof. Mag. Dr. Immanuel Bomze

Acknowledgments

I would like to thank my advisor Prof. Immanuel Bomze. He provided me with the possibility to work on the project of the University of Vienna and Telekom Austria. This was the perfect opportunity to dive into the field of operations research. On the one hand the topics arising from the project were a natural progression from my computer science background. On the other hand it was a great new challenge to focus on the more mathematical aspects. I also have particularly fond memories of Professor Bomze's highly abstract mathematical lectures.

Many thanks go to Dr. Ivana Ljubić who offered great support throughout the project. Her knowhow in the field of mathematical programming helped me a lot to get started. The continuous collaboration and the frequent discussions throughout these years ensured an instructive, challenging, and also fun experience. Moreover, her routine in all kinds of academic processes helped me to navigate the formal aspects of scientific research.

I would also like to express my gratitude to Prof. Juan-José Salazar-González from the University of La Laguna. His theoretical expertise as well as his patience and focus in our discussions were invaluable. Especially enjoyable were also multiple visits to his University allowing me to ponder on mathematical problems while sitting on the beach in Tenerife.

Lastly, I would like to thank my family and friends for their support and especially Ingrid and Arno for providing constantly friendly nagging to keep me focused on writing.

Contents

Chapter 1

Introduction

Providing future proof broadband Internet connections is currently a major infrastructural issue worldwide. More and more information is shared across the Internet and demand for higher data rates increases with new services. The *Digital Agenda for Europe*[1] of the European Commission stresses the importance of information and communications technologies and states that “Half of European productivity growth over the past 15 years was already driven by information and communications technologies [...] and this trend is likely to accelerate.” It issues the goal of achieving “internet speeds of 30 Mbps or above for all European citizens, with half European households subscribing to connections of 100 Mbps or higher” by the year 2020. The German government decided to place strong emphasis on the expansion of broadband communications in one of its latest economic stimulus packages[2]. The rather challenging aim, formulated in 2009, is to provide 75% of all households nationwide with 50 Mbps connections by the end of 2014. Reaching this goal is only possible by rolling-out fiber optic access networks on a broad scale.

The infrastructure of telecommunication networks nowadays can be seen as consisting of two layers. High speed, backbone networks interconnect cities or regions. Local access networks connect end customers via copper cables to an access point (central office) of the backbone network. In order to serve customers with higher bandwidth, telecom companies replace the copper networks with fiber optic connections. There are different strategies, distinguished by the endpoint of these new fiber optic connections.

- *Fiber-To-The-Curb* (FTTC) (or *Fiber-To-The-Node*, FTTN): The first part of the connection from the access point, or central office to the cus-

[1] *Digital Agenda* (May 2010), europa.eu/rapid/pressReleasesAction.do?reference=IP/10/581

[2] *Breitbandstrategie der Bundesregierung* (February 2009), www.zukunft-breitband.de/BBA/Navigation/Service/publikationen,did=290026.html

tomers consists of optical fibers. The second part consists of copper lines. At the transition point from fiber to copper, a *multiplexing* device has to be installed. This multiplexer receives signals from multiple customers via copper connections and aggregates them onto the fiber optic line.

- *Fiber-To-The-Building* (FTTB): The optical fiber runs all the way from the central office to a building. Multiplexing devices (usually installed in the basement) aggregate signals from the subscribers within the building via short-distance copper lines onto the fiber optic line.

- *Fiber-To-The-Home* (FTTH): The connections between the central office and the subscribers run completely over optical fiber. The connection can be *direct* or *shared*. A *direct* fiber runs directly from the central office to one individual subscriber (point to point). Alternatively, a *shared* fiber is used from the central office until close to the subscribers, where it is split into several fibers to connect the subscribers.

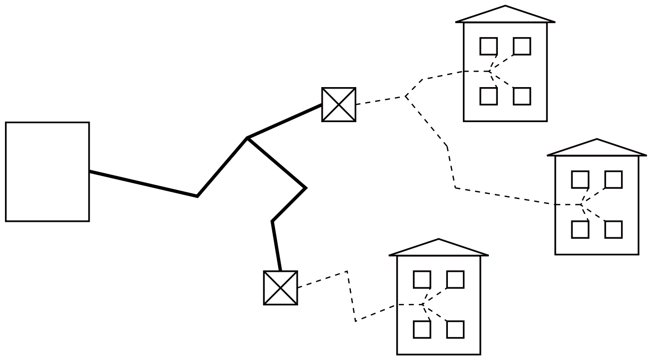
Figure 1.1 depicts these three variants. Direct FTTH allows for the highest bandwidth, followed by shared FTTH, FTTB and lastly FTTC. FTTx is used as a general label for any of these variants. In practice, also mixed scenarios are being considered. Here a subset of customers is connected via FTTH or FTTB and others are connected via FTTC. Which strategy is employed in a particular case depends on various prerequisites. For instance, it depends on how densely the planning areas are populated (e.g., urban vs. rural areas).

Many local telecommunication carriers are realizing FTTH or FTTB projects. The largest Austrian telecommunication provider, Telekom Austria Group, decided to invest one billion Euro in the modernization of the fixed net infrastructure[3]. Deutsche Telekom AG announced plans for the connection of thousands of households in ten German cities with FTTH.[4].

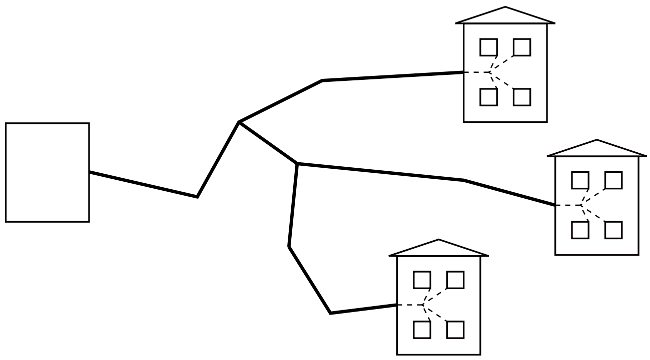
The planning of local access networks is a highly complex task. Manual planning does not allow for finding provably close-to-optimal solutions. In the last years various uncapacitated optimization problems have been proposed in the context of FTTx planning (see, e.g., [ABG$^+$11, LR11, GL11, GGL11]). These optimization problems are mainly concerned with the design of the underlying network topology, ignoring many hardware parameters. On a more detailed level, the following aspects have to be considered in addition: There are cost/capacity relations for various components, such as multiplexers, splitters, fibers and cables. There are overhead cost for trenching. Also, existing infrastructure has to be taken into account.

[3] *Press release* (July 3, 2009), www.telekomaustria.com/presse/news/2009/0703-telecommunication-infrastructure–en1.php

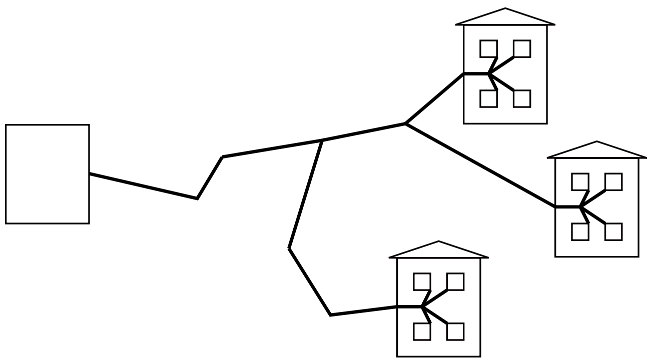
[4] *Press release* (February 28, 2011), www.telekom.com/dtag/cms/content/dt/de/996928



(a) FTTC. Fiber optic connections from the central server to two multiplexers and copper connections to the subscribers.



(b) FTTB. Fiber optic connections from the central server to the buildings that host the subscribers. Copper connections in the building.



(c) FTTH. The connection from the central server right to the subscribers is made of optical fiber.

Figure 1.1: An example, comparing FTTC, FTTB and FTTH. The rectangle to the left is the central server. Thick lines represent fiber optic connections. Crossed squares are multiplexer devices. Dashed lines represent copper connections. To the right, there are three buildings with four subscribers each.

This thesis was inspired by a project carried out by the University of Vienna and the largest telecommunications provider in Austria, Telekom Austria. This three year project dealt with the planning of local access networks. It involved practitioners from the planning department of the telecom provider and researchers from the provider and the university. A wide range of topics were addressed and a number of publications were spurred from this project: [TL08, LPSG11a, LPSG11b, GL11] and [Was11].

This thesis focuses on the topic of finding the cost optimal routes of direct fiber optic connections from the center to a set of endpoints. Applied to direct FTTH planning, the endpoints are the subscribers. For shared FTTH, the endpoint is the location at which the fibers are split apart. When dealing with FTTB connections, the endpoint is the building that hosts the subscribers. In the context of FTTC planning, the endpoints are the multiplexer devices. We do not deal with the positioning of the multiplexers themselves. The material of this thesis can be applied for mixed scenarios if the type and location of the endpoints has been decided. That is, once it is clear where multiplexers have to be set up, which buildings are FTTB connected, which groups of subscribers receive a shared FTTH connection and which subscribers receive a direct FTTH connection. These decisions are sufficient to define the endpoints and in this way mixed scenarios can be dealt with. The mainly intended audience for this work is the operations research (OR) community. In the field of OR this problem is modeled as the *Local Access Network* (LAN) design problem.

The first phase of the project focused on designing the fiber optic network between the access point and the multiplexer devices in FTTC planning. The LAN problem is approached with exact solution methods to solve the given problems optimally. The applied exact methods involve preprocessing, modeling with mixed integer programming (MIP), disaggregation of MIP formulations and Bender's decomposition. Thanks to the cooperation with the telecom company, we had access to real world data. In order to design and evaluate the methods, sample inputs with approximately 1 000 nodes and up to 67 customers where generated. Using these benchmark instances as well as smaller LAN instances from the literature allows us to evaluate the proposed methods empirically to demonstrate the practical applicability and to support and complement the theoretic results.

The second phase of the project focused on FTTH/FTTB planning and introduced the additional aspect of selecting a subset of customers in order to cover some target percentage. This is an interesting question in practice, since connecting every subscriber with FTTH/FTTB can lead to unreasonably high installation cost. To model this additional question a new extension to the LAN problem is proposed: the Prize-Collecting Local Access Network design problem

PC-LAN. In addition larger, more detailed inputs were defined together with the company. These served as more realistic and more challenging benchmark instances of PC-LAN scenarios with up to 80 000 nodes and 1 500 customers. The PC-LAN problem is approached with MIP based heuristic methods. This involves cutting plane formulations, multi-start construction heuristics and local improvement. These heuristic methods have proven to be useful when tested against the large benchmark instances.

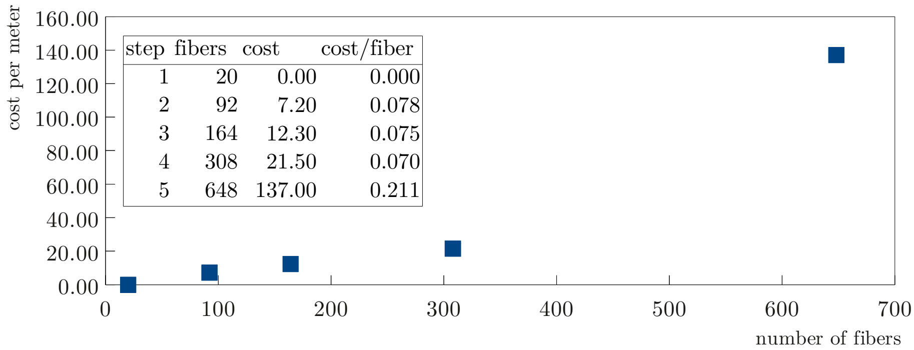
The thesis is structured in the following way. In the subsequent section the detailed requirements of FTTx problems are discussed. The Local Access Network design problem is presented as an abstract model to describe these practical problems in Section 1.2. The Prize-Collecting Local Access Network design problem is introduced in Section 1.3 to cover the aspect of choosing a certain subset of all endpoints. Chapter 2 reviews the relevant literature on related operations research problems and describes their relation to telecommunication network planning. Chapter 3 describes the exact methods to solve LAN design problems optimally. This chapter mainly represents the findings gained in the first phase of the partially industry-sponsored project. It extends upon material published in [LPSG11a]. Chapter 4 mainly constitutes material produced during the second phase of the project where the focus shifted towards the PC-LAN problem on large inputs and towards heuristic methods. Parts of this have been published in [LPSG11b]. Chapter 5 concludes this thesis with a short summary.

1.1 Modeling FTTx

In order to explain the abstract models used in operations research, four key aspects of detailed FTTx planning are presented in this section. Section 1.1.1 describes the relevant factors for establishing fiber optic connections. This involves different cable technologies and also the influence of previously existing infrastructure. Section 1.1.2 and 1.1.3 discuss the properties of Multiplexers and Splitters. These are devices to be installed at the endpoints of FTTC and FTTH/FTTB planning, respectively. Section 1.1.4 presents the concept of coverage in the context of FTTH/FTTB planning.

1.1.1 Fiber Optic Cables

Each endpoint requires a connection to the central office with a certain number of optical fibers. To establish these connection, different types of cables are available. Each type of cable is characterized by two features. One is its capacity and represents the number of optical fibers. The other is its cost.



step	fibers	cost	cost/fiber
1	20	0.00	0.000
2	92	7.20	0.078
3	164	12.30	0.075
4	308	21.50	0.070
5	648	137.00	0.211

Figure 1.2: This figure shows the *stepwise cost function* for a showcase connection. There is an existing connection providing 20 currently unused fibers. These can be used at 0 cost (step 1). In addition there is an existing duct along this connection. For new installations there is one technology of fiber optic cables available providing 72 fibers for 3.00 € per meter. It is possible to install 1, 2 or 4 cables at cost of 4.20, 6.30, or 9.50 € per meter. Including the cost for the cables themselves yields $4.20 + 3.00 = 7.20$, $6.30 + 2 \cdot 3.00 = 12.30$, and $9.50 + 4 \cdot 3.00 = 21.50$ €, respectively (steps 2, 3 and 4). Finally it is possible to dig a new trench for 100.00 €. The installation of 9 cables incurs 10.00 € in addition to the cost for the 9 cables themselves. Making up for the possibility to install $9 \cdot 72 = 648$ fibers for $100.00 + 10.00 + 3.00 \cdot 9 = 137.00$ €. For the context of this example it is assumed that the existing fiber and the cables in the existing ducts can be used simultaneously but in case new trenches are being dug, this existing infrastructure is removed and replaced by the new installations.

It has been noted above, that existing infrastructure has to be taken into account when a fiber optic connection shall be established from the access point to some endpoint. Two kinds of existing infrastructure can be distinguished. Firstly, in some cases there will be existing, currently unused installations of fiber optical cables. These can be used for very little cost. Secondly, in other cases there will be existing ducts that are not completely filled with cables. These allow for the installation of additional optical fiber cables. Again, cost for these additional fibers will be low. Both cases involve a strict limit on the capacity, i.e., there are only a certain number of unused fibers or respectively, there is only limited room in the duct for additional cables. If existing infrastructure is insufficient, new ducts have to be laid. This will typically involve, excavating new trenches and putting new ducts and cables inside. There is no strict limit on the capacity installable this way, but in addition to the cost for the ducts and cables there is a significant overhead cost for the trenches. This involves legal questions like property ownership, and cost for blocking roads while the construction works are going on.

Figure 1.2 highlights some of these aspects with a small example. Taking into account an existing fiber optic connection, existing ducts offering the possibility of relatively cheap new installations and the possibility of laying new ducts turns out to produce a non-decreasing stepwise function for the cost per number of fibers on the connection. It can be seen from the figure that if any number of fibers from 21 to 92 is needed the cost is 7.20€ per meter regardless. However if 93 fibers are needed the cost jumps to 12.30. Looking only at the steps 2, 3 and 4 in the figure, it can be seen that this section of the cost function exhibits economies of scale, i.e., the more fibers are installed the cheaper it gets per fiber $(0.078 > 0.075 > 0.070)$. This is common when entities are bought in bulks. However, over the whole range of the function it is not clear that economies of scale are given. This can clearly be seen with step 5. Due to the high overhead cost for building new fiber optic installations the cost per fiber jumps from step 4 to step 5 from 0.07 to 0.211.

In addition to what this example shows the practical situation will be further complicated by the availability of different cable technologies. They provide different numbers of fibers per cable. Others more expensive technologies can provide the same number of fibers in a cable of smaller diameter. Using these would allow to put more cables, hence more fibers in an existing duct.

All of these aspects can be described by means of a stepwise cost function for any two sites that can be connected. Instead of speaking about cables, ducts and trenches, we will only consider *modules*. Each combination of cables leads to a *module* with a given *capacity* and *cost*. The capacity of a module is simply the sum of the fibers included in the cables. The cost of a module is the sum of the cable costs plus the installation on the roads taking into account the length. A set of modules describes the stepwise cost function for the connection of two sites. Note that the set of available modules will generally differ for the connection of different sites.

1.1.2 Multiplexer Devices

In existing copper networks there is a distinct copper cable running from the access point to each building. Starting from the access point, these cables are laid in bulks and these bulks are split apart as they come closer to their endpoint. In FTTC planning a location along this connection is chosen to set up a *multiplexer* device. All the endpoints that are *behind* this point will be connected to the multiplexer via the existing copper lines. The multiplexer has to be connected with optical fiber to the access point. A multiplexer device has a limit on the number of outgoing copper lines it can support. On the other hand it needs to be connected with a certain number of optical fibers to the access point. If more copper lines are to be connected to this location, a second multiplexer device has to be installed. This, in turn, increases the number of optical fibers that have to be provided. In addition to the cost for the devices itself there is also a setup cost that is related to the conditions at the location. This involves questions such as whether the devices need to be put inside some casing to protect them or whether they are being set up inside a building. The multiplexers need a power supply and also legal questions about possible locations where devices may be installed have to be taken into account. Note that the question of deciding the locations for the multiplexer devices is not being studied in this work. Section 2.3 gives some directions and citations on publications dealing with this question. As far as FTTC in this work is considered it is assumed that the locations have previously been decided. The focus is on deciding the routing from the access point to the locations of the multiplexer devices and the cost optimal installation of optical fiber on these routes.

1.1.3 Splitter Devices

In FTTH and FTTB scenarios a *splitter* device has to be installed at the endpoint of the fiber optic connection. There is a given *splitting ratio* that describes the relation between the number of incoming optical fibers and the number of end customer devices or *subscribers* that can be connected. If there are more subscribers at a specific endpoint, a second splitter has to be installed and the number of optical fibers for the connection increases. An FTTH/FTTB endpoint can be summarized by three features. Firstly, the number of subscribers (e.g., apartments and/or offices) in the building. This is denoted as the *prize*. Secondly, the number of optical fibers required to connect this endpoint is called its *demand*. Thirdly, there is a setup *cost* of installing the appropriate number of suitable devices at the location. This cost takes into account the conditions at the site of the endpoint.

1.1.4 Coverage

Especially in the context of FTTH/FTTB the question of selecting a subset of endpoints to connect is raised. Providing FTTH/FTTB for some endpoints will be relatively cheaper than providing this level of service to other endpoints. This depends on the proximity of the endpoints and their location with respect to the access point. Also it depends on the existing infrastructure. For example there are relatively little cost associated to providing FTTH to all the apartments in an apartment building that can be directly connected to the access point via some existing, currently unused fiber optic connection. The opposite extreme example is a single household at an exposed position. In order to provide a

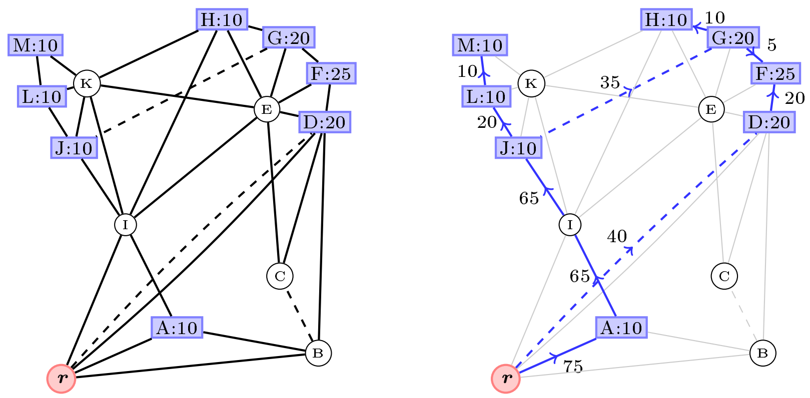
FTTH connection it would be necessary to lay new ducts and cables along a wide stretch. If there are no other customers in the proximity of this household the cost for this connection are not shared by many endpoints but must be directly attributed to this single household. Of course, due to the way new connections provide fiber optic cables in bulk, not all cases will be as obvious as these two examples. To deal with this situation, telecom companies decide not to connect 100 percent of all endpoints with a service by all means and for any cost. Instead they want to connect at least a certain percentage X. The company strives for some market share and is interested in the cost for some targeted coverage. The objective changes from finding fiber optic installations and routings to determining a subset of endpoints plus finding the installations and routings. To describe this requirement we consider the *prize*, i.e., the number of subscribers at an endpoint. The coverage condition is then expressed as: select a subset of endpoints such that X percent of the total sum of prizes is covered. Alternative definitions of prize may be used. The prize could be any number the company associates to an endpoint representing some kind of market value. For example an estimate of the expected revenue that can be achieved by serving that endpoint.

1.2 Local Access Network Design Problem

The problem of finding the routing from the access point to the endpoints and deciding the cost optimal fiber optic cable installations along the routes can be modeled as the *Local Access Network* design problem (LAN). The following paragraphs give a formal definition of the LAN design problem and explain the relation to FTTx planning.

Definition 1.2.1. *We are given an undirected, connected graph $G = (V, E)$ with a central node $r \in V$. A subset of the network nodes $K \subseteq V \setminus \{r\}$ represents customers. To each customer $k \in K$ a positive demand d_k is associated. On each edge at most one module m out of a set $M_e = \{1, 2, \dots\}$ can be installed. Each module has associated a positive capacity $u_{e,m}$ and positive cost $c_{e,m}$. The module indices are sorted by increasing capacity, i.e., $u_{e,m} < u_{e,m+1}$. The* Local Access Network design problem (LAN) *asks for an installation of at most one module per edge. The installation of modules shall allow for a single-source multiple-sink routing from r to the customers, that satisfies all the demands simultaneously. The cost for the installation of modules shall be minimal.*

The intermediate nodes $V \setminus K \setminus \{r\}$ may or may not be included in a solution, thus they are called *Steiner* nodes. Figure 1.3 shows an example of a LAN problem together with an optimal solution.



(a) Instance of the PC-LAN design problem. (b) Optimal solution of the PC-LAN instance.

Figure 1.3: Graph $G = (V, E)$ with $|V| = 14$. For each edge e the length l_e of e is the Euclidean distance. Solid lines represent modules with $u_{e,m} = 100$ and $c_{e,m} = 120l_e$. Dashed lines represent the module with $u_{e,m} = 40$ and $c_{e,m} = 10l_e$. Rectangle nodes are customers with their demands d_k written at the corresponding labels. The node r is the central office. The edge labels in Figure (b) describe a directed flow from r to the customers.

Mathematically the LAN design problem can be expressed with the following *undirected single-commodity flow*, mixed integer program (uSCF). The binary *design variables* $\boldsymbol{x}$ define the installed modules per edge, i.e., $x_{\{i,j\},m} = 1$ iff the module m is installed on the edge $\{i, j\}$. The routing is expressed by continuous *flow variables* $f_{(i,j)} \geq 0$ that define the amount of flow on the edge $\{i, j\}$ running from i to j. The undirected single-commodity flow formulation for the LAN design problem is given by (1.1)-(1.6).

The *flow conservation constraints* (1.2) ensure that every customer receives the desired amount of flow. The source of all flow is the access point r and on all other nodes there is a balance between outgoing and incoming flow. The *capacity constraints* (1.3) ensure that enough capacity is installed on every edge to support all the flow on that edge. The *disjunction constraints* (1.4) state that at most one module may be installed. This model serves as a basis for the models described later in Section 3.3.

$$\text{(uSCF)}: \quad \min \sum_{e \in E} \sum_{m \in M_e} c_{e,m} x_{e,m} \tag{1.1}$$

s.t.

$$\sum_{(i,j)\in\delta^+(i)} f_{(i,j)} - \sum_{(j,i)\in\delta^-(i)} f_{(j,i)} = \begin{cases} -d_i, & i \in K \\ \sum_{k\in K} d_k, & i = r \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in V \tag{1.2}$$

$$f_{(i,j)} + f_{(j,i)} \leq \sum_{m\in M} u_{\{i,j\},m} x_{\{i,j\},m} \quad \forall \{i,j\} \in E \tag{1.3}$$

$$\sum_{m\in M_e} x_{e,m} \leq 1 \quad \forall e \in E \tag{1.4}$$

$$x_{e,m} \in \{0,1\} \quad \forall e \in E,\ \forall m \in M_e \tag{1.5}$$

$$f_{(i,j)}, f_{(j,i)} \geq 0 \quad \forall \{i,j\} \in E. \tag{1.6}$$

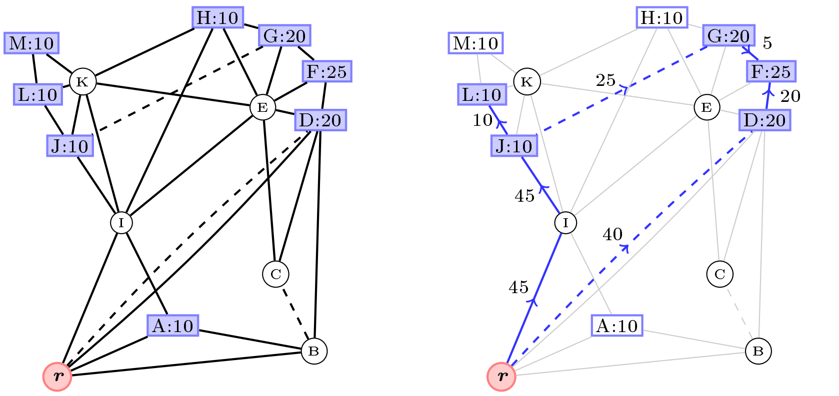
The LAN design problem relates to the problem of FTTx planning as follows. The nodes, edges and the corresponding modules represent all possible fiber optic connections that can be established. The node r represents the central office (or central server or access to the backbone network). The *customers* represent the endpoints of the FTTx planning problem. The demand of a customer is measured as the number of fibers required to serve the corresponding endpoint. In case of FTTH/FTTB the customers are the locations of buildings and the demand is given as described in Section 1.1.3. In case of FTTC the customers are already decided locations for multiplexer devices as described in Section 1.1.2. Number of fibers is also the measurement unit of capacities of the *modules*. Each module represents one step in the step cost function and the binary design variables and the disjunction constraints serve as a means to select one specific step. The cost of a module represent the cost of a certain step taking into account the distance between two sites. The flow variables specify the number of fibers for any connection. Note that the flow from r to a customer is allowed to split apart, i.e., the connection is not necessarily a single path. This kind of flow is called *bifurcated* flow.

1.3 Prize Collecting LAN

The LAN problem described in the previous section covers the aspects of finding the routing and deciding the installations along the routes. To model the coverage requirement for FTTB/FTTH planning, i.e., the additional aspect of selecting a subset of all customers, the *Prize Collecting Local Access Network* design problem is introduced. Two additional features are associated with each customer. In addition to the demand d_k we consider a cost $c_k \geq 0$ and a *prize* $p_k \geq 0\,\forall k \in K$. A subset of customers shall be selected such that at least some target prize p_0 is achieved. The sum of the cost of the installations along the edges plus the cost for the subset of customers shall be minimized. Figure 1.4 shows an example of a PC-LAN problem and a corresponding optimal solution.

Definition 1.3.1. *We are given an undirected, connected graph $G = (V, E)$ with a central node $r \in V$. A subset of nodes $K \subseteq V \setminus \{r\}$ represents customers. To each customer $k \in K$ a positive demand d_k, a positive prize p_k and a positive setup cost c_k are associated. A target prize p_0 is given. On each edge at most one module m out of a set $M_e = \{1, 2, \dots\}$ can be installed. Each module has associated a positive capacity $u_{e,m}$ and positive cost $c_{e,m}$. The module indices are sorted by increasing capacity, i.e., $u_{e,m} < u_{e,m+1}$. The* Prize-Collecting Local Access Network design problem (PC-LAN) *asks for a selection of customers to be served and an installation of at most one module per edge. The selection of customers shall cover at least the target prize p_0. The installation of modules shall allow for a single-source multiple-sink routing from r to the selected customers, that satisfies all the demands simultaneously. The cost for the installation of modules plus the cost for the selected customers shall be minimal.*

The (uSCF) model from the previous section is extended with additional binary decision variables $\boldsymbol{y}$ describing the subset of customers. Here $y_k = 1$ iff the customer k is to be connected. The *undirected single-commodity flow* formulation for the Prize Collecting LAN design problem (puSCF) is defined as follows.



(a) Instance of the PC-LAN design problem. (b) Optimal solution of the PC-LAN instance with $p_0 = 0.7 \sum_{k \in K} p_k = 80.5$.

Figure 1.4: Graph $G = (V, E)$ with $|V| = 14$. For each edge e the length l_e of e is the Euclidean distance. Solid lines represent modules with $u_{e,m} = 100$ and $c_{e,m} = 120 l_e$. Dashed lines represent the module with $u_{e,m} = 40$ and $c_{e,m} = 10 l_e$. Rectangle nodes are customers with their demands d_k written at the corresponding labels. Customer prizes and cost are defined as $p_k = d_k, c_k = d_k/2$, respectively. The node r is the central office. In Figure (b), the selected customers have a dark background and the not-selected customers have a white background. The edge labels describe a directed flow from r to the selected customers.

$$\text{(puSCF)} : \qquad \min \sum_{e \in E} \sum_{m \in M_e} c_{e,m} x_{e,m} + \sum_{k \in K} c_k y_k$$

s.t.

$$\sum_{(i,j)\in\delta^+(i)} f_{(i,j)} - \sum_{(j,i)\in\delta^-(i)} f_{(j,i)} = \begin{cases} -d_i y_i, & i \in K \\ \sum_{k\in K} d_k y_k, & i = r \\ 0, & \text{otherwise} \end{cases} \qquad \forall i \in V \tag{1.7}$$

$$\sum_{k \in K} p_k y_k \geq p_0 \tag{1.8}$$

$$f_{(i,j)} + f_{(j,i)} \leq \sum_{m \in M_{\{i,j\}}} u_{\{i,j\},m} x_{\{i,j\},m} \quad \forall \{i,j\} \in E \tag{1.9}$$

$$\sum_{m \in M_e} x_{e,m} \leq 1 \qquad \forall e \in E \tag{1.10}$$

$$x_{e,m} \in \{0,1\} \qquad \forall e \in E,\ \forall m \in M_e \tag{1.11}$$

$$y_k \in \{0,1\} \qquad \forall k \in K \tag{1.12}$$

$$f_{(i,j)}, f_{(j,i)} \geq 0 \qquad \forall \{i,j\} \in E \tag{1.13}$$

Compared to the (uSCF) model, the objective function of the (puSCF) model has an additional term representing the on-site setup cost. The set of constraints is extended by one coverage constraint (1.8) based on the prizes of customers. The flow conservation constraints (1.7) are modified in order to reflect the selected subset of customers described via the $\boldsymbol{y}$ variables.

The cost c_k represent the on-site cost determined by the splitter devices in the FTTB/FTTH problem. The prize represent the number of subscribers per endpoint, or more generally, the prize stands for any market value that the telecom company associates with this endpoint. To connect for example at least 70% of all prizes, the target prize p_0 is defined as $p_0 = 0.7 \sum_{k \in K} p_k$.

Chapter 2

Literature Review

There is a vast number of publications on the topic of network design problems in the operations research community. This section presents a review of the literature on capacitated network design, with an emphasis on publications more directly related to the topics in this thesis.

The common goal in all the problem variations is to find routes in a given graph in order to satisfy a given pattern of commodities to be transported. The commodities can generally be defined as pairs of source and sink node together with a demand. In order to allow the transportation of the demand through the graph some capacities have to be installed on the edges of the graph, incurring setup cost. In addition, there may be per-unit transportation costs.

The following list defines some terms that are commonly used to distinguish the problem variations.

single commodity vs. multiple commodities A common definition of *commodity* is a pair of source and sink nodes together with a number that specifies the amount to be transported from source to sink. An alternative description results from aggregating all source-sink pairs with a common source and defining a commodity as one source node together with a set of sink nodes and a specified demand per sink. Using this aggregated definition we define a single-commodity problem as having exactly one source. Consequently, a multi-commodity problem has many such aggregated commodities.

Note that the roles of source and sink can be exchanged, thus a problem with one sink and multiple sources is a single-commodity problem.

flow-dependent cost The simplest cost structure for network design problems is one linear term per flow variable.

fixed edge cost A more challenging cost structure is to pay one fixed amount per edge that is used in the solution. This is commonly denoted as *Fixed-Charge Network Design*. In the context of the LAN problem this would be expressed with one module per edge $|M_e| = 1 \; \forall e \in E$.

step-wise edge cost Instead of considering just one fixed charge per edge, it is common to consider more complex step-wise cost structures. Two typical variants are integer multiples and arbitrary steps:

integer multiples Assume there is one available cable technology, providing a capacity of u, incurring cost c. In order to achieve higher capacities, multiple cables can be installed. This can be modeled in a MIP with integer variables $y_e \in \mathbb{N}$, a cost function $\sum_{e \in E} c l_e y_e$, where l_e is the edge length, and capacities of $u y_e$ per edge.

Given that there are multiple available cable technologies (u_1, c_1), $(u_2, c_2), \ldots$, which can be freely combined, one would use integer variables $y_{e,1}, y_{e,2}, \ldots$. The cost function is $\sum_{e \in E} \sum_{n \in 1,2,\ldots} c_n l_e y_{e,n}$ and the capacity per edge $\sum_{n \in 1,2,\ldots} u_n y_{e,n}$. In order to express the cost function in terms of *modules* as used in Definition 1.2.1, one can compute an optimal combination of cable types for every capacity level.

arbitrary steps The description of the step-cost function used in this thesis, based on the notion of modules or levels is more general. It can be used to model not only the integer multiples function but also other non-decreasing step-cost functions including functions without economies of scale. Existing capacities are also naturally covered with this formulation. Inequalities (1.3),(1.4),(1.5) give a formal description in a MIP.

piecewise linear A yet more general form of step-cost functions with *sloped* steps is achieved via discontinuous, piecewise linear functions. This includes a flow dependent coefficient on each capacity level.

Note that in some publications the installable capacities on the edges are called *facilities* for transportation. This usage of the word facility is avoided in the present work to prevent misunderstandings with respect to the *facility location problem* which is used in the context of FTTC planning.

tree flow, non-bifurcated flow, bifurcated flow Another attribute of typical network design problems is whether a certain structure of the solution is required. One typical example is to search for tree-solutions.

A relaxation of the tree requirement is to state that there must be one unique path connecting each source-sink pair that transports the specified demand. Since the flow is not allowed to split apart, this network design variants are called non-bifurcated. Note that due to step-cost functions the unique path restriction may *not* imply a tree solution.

A further relaxation of this structure requirement is to allow bifurcated flow. Here the connection between source and sink in the solution is established via an arbitrary subgraph. The flow can split apart at nodes and rejoin at other nodes. The problems considered in this thesis are bifurcated flow problems.

triangle inequalities, Euclidean distances Some problem definitions require certain properties such as triangle inequalities or Euclidean distances on the edge lengths or on the fixed-charge costs. There is no such requirement in this thesis.

survivability There are some network design variants that consider *survivability*. This requires that in case of some failure scenario (link-failure, node-failure), the demands shall still be serviceable, or serviceable to a certain degree. Survivability is not an issue for the LAN variants considered in this work.

prize-collecting Lastly, prize-collecting aspects are an additional problem dimension that is relevant for certain applications. Instead of servicing all demands it is possible to connect only a subset of customers. Chapter 1.3 deals with a variant of the LAN problem where a certain percentage of all customers has to be connected.

These seemingly minor differences are often key to whether certain solution methods are applicable or whether these methods work well. For example, with flow based MIP models it is generally more natural to express bifurcated flow rather than non-bifurcated flow. Approximation algorithms often rely on triangle-inequalities, or Euclidean distances and on economies of scale for the performance guarantees. Tree structures will usually lend themselves more naturally to heuristic construction methods than bifurcated flows. The presence, or absence of a simple linear term in the objective function describing flow dependent cost may seem negligible upon first thought. However, in case of Benders' decomposition, the presence of flow-cost implies optimality cuts in the typically used, *natural* decomposition. The models studied in this work do not have flow-cost and thus no optimality cuts are needed. This absence of optimality cuts may be key to the observed performance of certain separation policies. This is elaborated on in Section 3.11. Considering all this, it comes clear that there is a merit in studying that specific capacitated network design problem that most closely models a given real world application.

Using the above nomenclature, the problems considered in this thesis are bifurcated single-commodity network design problems with arbitrary step-wise cost functions for the edge-capacities. Economies of scale are not given and there are no flow dependent costs. Survivability is not required.

2.1 Single-Commodity Network Design

The *capacitated minimum Steiner tree* problem (CMStT) is a restricted variant of the LAN problem. The solution to this single-commodity network design problem is required to form a tree. This implies a non-bifurcated flow. The *capacitated minimum spanning tree* problem (CMST) furthermore assumes that all nodes $V \setminus \{r\}$ are terminals. In [Gav85], Gavish uses integer programming formulations and Lagrangian relaxations to target the CMST problem. In [AG88], Altinkemer and Gavish present a 4-approximation algorithm for CMST. There is one available capacity and triangle inequalities are assumed to be satisfied. Their algorithm works by partitioning traveling salesman tours.

Magnanti and Mirchandani study a network design problem with a single source and a single destination and up to three available base capacities of which integer multiples can be installed in [MM93]. This can be seen as a generalized shortest path problem. They discuss which variants are polynomially solvable and which variants are NP-hard. They develop an extended MIP formulation and present computational results on inputs with up to 50 nodes and 200 arcs. Chopra, Gilboa and Sastry study a similar problem, with the extension that flow dependent costs are considered in [CGS98]. They analyze the complexity of certain problem variants.

Mateus, Luna and Sirihal consider a variant of LAN with multiple sources for the single-commodity to be distributed in [MLS00]. This allows to model multiple LAN networks simultaneously. They introduce an additional cost factor for *splicing*. These splicing cost are incurred each time the cable technology is changed at any node. They develop a Lagrangian heuristic that reduces the problem to an uncapacitated minimum cost network flow problem and test their approach on two instances with up to 217 nodes and 520 arcs.

In [BGP$^+$00], Berger, Gendron, Potvin, Raghavan and Soriano present a taboo search heuristic for the non-bifurcated LAN problem. The algorithm explores a neighborhood defined by k-shortest paths ($k = 2$) and uses an adaptive memory to collect promising paths. The algorithm is tested on instances with up to 200 nodes and a step-cost function with economies of scale.

The *single-sink by-at-bulk* network design problem (SSBB) or *single-sink* *link-installation* problem generalizes the CMStT problem by allowing for multiple steps exhibiting economies of scale and allowing a bifurcated, non-tree solution. Salman, Cheriyan, Ravi and Subramian present an approximation algorithm in [SCRS00]. They study the problem with Euclidean distances and a special case with arbitrary metric in which no source-sink path may be longer than two edges.

Randazzo, Luna and Mahey study the LAN problem with flow-cost and two modules under the assumption that the solution must be a tree in [RLM01]. This is equivalent to the CMST problem with a two-step-cost function. They apply Benders' decomposition to the multi-commodity flow formulation and test the approach on instances with up to 41 nodes, 417 arcs and 8 customer nodes.

Guha, Meyerson and Munagala [GMM01] present a randomized approximation algorithm for the SSBB problem. In [GKR03] this work is extended upon by Gupta, Kumar and Roughgarden.

Garg, Khandekar, Konjevod et al. study the integrality gap of the single-commodity flow formulation of the SSBB problem and present an approximation algorithm in [GKK+01]. This work is further extended upon in Talwar [Tal02].

Jothi and Raghavachari [JR05] present alternative approximation algorithms for the CMST and the CMStT problems as defined in [AG88].

Gamvros, Golden and Raghavan [GGR06] deal with a variant of the CMST problem where economies of scale are assumed. They employ single- and multi-commodity flow formulations, a savings based heuristic, neighborhood searches and a genetic algorithm. Their approaches exploit the tree feature and they provide computational results on instances of up to 150 nodes.

Salman, Ravi and Hooker [SRH08] consider the LAN problem with capacities defined by the sum of integer multiples of base capacities. The cost function exhibits economies of scale. They apply flow-based MIP formulations and work with relaxations obtained by approximating the noncontinuous stepwise function by its lower convex envelope. Raghavan and Stanojević [RS06] reformulate this approximation technique as a stylized branch-and-bound algorithm.

In [JR09], Jothi and Raghavachari present approximation algorithms for the non-bifurcated SSBB with an approximation ratio of 145.6. For the SSBB with bifurcation their approximation ratio is not greater than 65.49.

2.2 Multi-Commodity Network Design

Iri [Iri71] presents a description of *metric inequalities* which are of great importance for multi-commodity network design. A network design allows for a feasible multi-commodity flow, if and only if all metric inequalities are satisfied.

The *network loading* problem (NL) is a generalization of the *multi-commodity fixed charge network design* problem (MCFCND). Instead of one available capacity, NL allows for an arbitrary step-cost function. The LAN problem is a special case of NL with only a single commodity. Gavish and Altinkemer [GA90] use the NL problem to model the design of backbone networks where queuing cost have to be considered. They apply Lagrangian relaxation and cut generation.

Magnanti, Mirchandani and Vachani [MMV93] present a polyhedral study of the convex hull of the NL problem on a single arc and on the three node network under the assumption that edge capacities are defined by integer multiples. In [MMV95] the authors extend this work to the problem with capacities defined by integer multiples of two base capacities and perform a computational study on instances with up to 15 nodes.

Barahona [Bar96] formulate cut-set and partition inequalities for the NL problem with integer multiple capacities. He deals with 2-node connectivity and 2-edge connectivity for the bifurcated problem. In addition he presents a heuristic for the non-bifurcated problem and present computational results on instances with up to 64 nodes in a complete graph with 2016 commodity pairs.

Bienstock and Günlük [BG96] present a branch-and-cut approach based on cut-set, flow-cut-set and three-partition inequalities for the NL problem. Computational results on instances with up to 16 nodes and 49 edges are reported.

Amiri and Pirkul [AP97] study a problem that is related to the non-bifurcated NL problem. They explicitly model queueing cost in a MIP with a non-linear objective function and apply a Lagrangian heuristic and report computational results on four networks with up to 992 commodity pairs.

Bienstock, Chopra, Günlük and Tsai present a separation heuristic for partition inequalities. These are a subset of metric inequalities. They also present a complete description of the polyhedron of a 3-node network design problem in [BCGT98]. The computational performance is evaluated on instances with up to 27 nodes and 102 arcs.

Holmberg and Yuan present a Lagrangian heuristic for the *fixed charge network design* problem (FCND) and the problem with arbitrary stepwise cost functions in [HY98]. They demonstrate empirical results for this approach on instances with up to 150 nodes, 1000 edges and 462 commodity-pairs. This work is further extended and integrated into a branch-and-bound algorithm in [HY00].

Dahl and Stoer [DS98] study the NL problem with the additional complication that the demands must be serviceable even in the case of a single node or edge failure. They use cut-set and metric inequalities to model the problem as MIP and test their approach on instances with up to 118 nodes and 134 edges.

Günlük [Gü99] focuses on the NL problem with integer multiples of two base capacities. He presents mixed partition inequalities and compares them to other families of inequalities. In addition, results for a branch-and-cut implementation using a special knapsack branching rule on instances with up to 30 nodes and 55 edges are reported.

In [GKM99], Gabrel, Knippel and Minoux demonstrate a branch-and-cut scheme for NL. They use a max-cut heuristic to separate bipartition inequalities and a subgradient based heuristic for metric inequalities. They present empirical evidence on the advantage of adding multiple constraints per iteration on instances of up to 20 nodes and 37 edges.

Mirchandani [Mir00] studies polyhedral properties of projections of the NL problem with two available cable types.

Minoux [Min01] surveys flow-and-cut based formulations for various multi-commodity network design problems.

Crainic, Frangioni and Gendron use subgradient optimization and bundle methods for two Lagrangian relaxations of the FCND problem in [CFG01]. They compare these approaches on instances with up to 100 nodes, or up to 400 edges or up to 200 commodities.

Atamtürk [Ata01] presents facet defining inequalities of the single node fixed-charge flow polytope. He gives computational tests for this substructure of network design problems. The same author studies the polyhedra of more general network design problems with integer-multiple-capacities in [Ata02]

Hoesel, Koster, Leensel and Savelsbergh [HKLS03] present a polyhedral study of flow and path models for the non-bifurcated NL problem with uni-directed and bidirected capacity installations in integer multiples.

Gabrel, Knippel and Minoux compare greedy rerouting heuristics with Benders' decomposition based heuristics in [GKM03]. They report on computational results on NL instances with arbitrary step cost functions on instances with up to 50 nodes.

Rajan and Atamtürk [RA04] develop a column-and-cut generation procedure for a survivable multi-commodity network design problem. The computational performance of the approach is evaluated on random instances with up to 70 nodes.

Muriel and Munshi show in [MM04] that three different Lagrangian relaxations of the NL problem with flow cost and piecewise linear, nondecreasing, concave cost functions are equivalent to the linear relaxation.

Ghamlouche, Crainic and Gendreau [GCG04] describe a path-relinking procedure on a cycle-based neighborhood for the FCND problem and study the performance of different versions of the procedure on instances with up to 100 nodes, 700 arcs and 400 commodity-pairs.

Crainic, Gendron and Hernu develop an adaptive Lagrangian based heuristic for the FCND problem in [CGH04]. They show computational results on instances with up to 30 nodes, 700 arcs and 400 commodity pairs.

Costa [Cos05] presents a survey of Benders' decomposition approaches to various network design problems. He covers single- and multi-commodity problems, with and without capacities, with different variants of step-wise cost functions and with or without flow dependent costs.

Alvarez, González-Velarde and De-Alba formulate a GRASP embedded Scatter Search approach for the bifurcated fixed charge multi-commodity network design problem in [AGVDA05]. In addition to the fixed charge for edge utilization, they consider flow dependent cost per commodity and per arc. They compare their approaches on instances of up to 50 nodes, 700 edges and 100 commodity-pairs.

Avella, Mattia and Sassano [AMS07] present tight metric inequalities, which form a complete description of the convex hull of the NL polyhedron with integer multiple capacities. They develop a heuristic separation technique and test it on instances with up to 64 nodes and 2016 commodity-pairs.

Haouari, Mrad and Sherali deal with a variant of the NL problem with flow cost in [HMS07]. Here the commodities must be routed non-simultaneously, i.e., the installed capacities are shared for all commodities. They apply Benders' decomposition which leads to multiple independent min-cut subproblems and a min-cost-flow subproblem for this specific NL-variant. Computational results are presented for their branch-and-cut implementation on instances of up to 500 nodes, 2000 edges and 10 commodity-pairs.

Croxton, Gendron and Magnanti study the impact of disaggregation by commodity and by module in [CGM07]. They consider concave and non-concave piecewise linear cost functions and compare the obtainable LP-gaps for single and multi-commodity instances.

Atamtürk and Günlük study polyhedral properties of network design structures in [AG07].

Raack, Koster, Orlowski and Wessäly [RKOW07] study properties of the NL polyhedron and show that cut-set inequalities are facet defining under certain circumstances.

Alvelos and Carvalho [AC07] work on the multi-commodity flow problem. There are given arc capacities and the objective is to minimize a linear flow dependent cost function. They approach the problem with an extended version of a path based MIP model and apply column generation.

Costa, Cordeau and Gendron [CCG09] investigate the relationship between three classes of inequalities used in multi-commodity network design: Benders' cuts, metric inequalities and cut-set inequalities. They describe how cut-set inequalities and Benders' cuts associated to non-extreme rays can be strengthened by transforming them into metric inequalities and demonstrate this approach on MCFCND instances with up to 100 nodes and 400 commodity-pairs.

Frangioni and Gendron apply branch-and-price to the disaggregated flow model for the NL problem in [FG09]. They present results obtained on random instances with up to 30 nodes and 400 commodity-pairs.

Rei, Gendreau, Cordeau et al. [RGCS09] look into speeding up Benders' decomposition by combining it with local branching and demonstrate the performance of the approach on MCFCND instances with up to 20 nodes, 80 edges and 15 commodity-pairs.

Bektas, Chouman and Crainic [BCC10] work on a problem related to capacitated network design where the violation of capacity constraints is allowed and incurs a nonlinear penalty. They apply Lagrangian decomposition and compare results obtained on instances with up to 10 nodes, 60 arcs and 50 commodity pairs with results from state-of-the-art nonlinear solvers.

2.3 Further FTTx Publications

It has been said in Chapter 1 that this thesis focuses on aspects of network design in the domain of FTTH/FTTB. The question of where to position multiplexers in FTTC design is not studied in this work. An abstract mathematical problem to describe the problem of locating multiplexers is the *Connected Facility Location* problem (ConFL).

Ljubić investigates the ConFL problem by means of variable neighborhood search and branch-and-cut in [Lju07].

Tomazic and Ljubić present a greedy randomized adaptive search procedure (GRASP) for ConFL in [TL08]. They compare the achieved results to bounds computed via a branch-and-cut algorithm.

Bardossy and Raghavan present a generalized problem description that covers the ConFL, Steiner tree-star and the rent-or-buy problems in [BR10]. They evaluate their dual-based local search procedure one dense and sparse instances.

Chamberland deals with the combined planning of FTTC and FTTH networks in [Cha10]. The focus is on the detailed description of different multiplexer devices. The network structure is a tree with fixed-charge edge cost. In addition a prize collecting aspect for servicing a subset of customers is considered.

Wassermann [Was11], considers the issue of locating multiplexers in tree networks. Various side constraints arising in a practice-oriented setting are taken into account.

Kim, Lee and Han consider a mixture of LAN and ConFL on tree graphs with one or two layers of facilities in [KLH11]. The authors formulate the problem as integer program and compare a heuristic solution method to the bounds computed via branch-and-bound on instances with up to 30 nodes.

Gualandi, Malucelli and Sozzi study a problem related to ConFL in [GMS10]. The network to connect the facilities has a star-structure and, the facilities have associated capacities.

Gollowitzer and Ljubić [GL11] investigate a combination of facility location and Steiner tree problems. They present MIP formulations an perform an empirical comparison on instances with up to 1 300 nodes and 115 000 edges.

Gollowitzer, Gendron and Ljubić [GGL12] present families of valid inequalities for a more general combination of facility location and fixed-charge network design. Here, capacities along the edges and on the facilities are considered.

Contreras and Fernandez [CF12] present a survey of publications dealing with the combination of fixed-charge network design and facility location problems.

Chapter 3

Solving the Local Access Network Design Problem Exactly

This chapter presents exact and heuristic solution methods for the Local Access Network Design Problem. In Section 3.1 various exact preprocessing techniques for LAN are explained. Section 3.2 presents a transformation from the undirected LAN problem into an equivalent directed formulation. In Section 3.3 basic mathematical models are described. Section 3.4 shows how these models can be strengthened with disaggregation techniques. Section 3.5 present the application of Benders' Decomposition technique to the disaggregated models. Various ways to normalize the Benders' Decomposition are explored in Section 3.6. Other enhancements and more valid inequalities are defined in Section 3.7. The different models are summarized in a hierarchy with respect to their polyhedral inclusion in Section 3.8. Section 3.9 describes the rounding heuristic used to derive primal feasible solutions from linear relaxations. Section 3.10 reports on the results of a computational study to evaluate the performance of the disaggregation and the normalizations. This chapter details the results published in [LPSG11a].

3.1 Preprocessing

This section describes a set of preprocessing techniques. These are transformations to go from an original LAN design problem to a preprocessed LAN design problem. The preprocessed problem is *smaller* in the sense that fewer decision variables are needed to describe the problem. Preprocessing is especially important when dealing with real world data. Real world inputs typically carry a high level of detail that is not necessary for a specific optimization task. Even relatively simple methods can reduce the size of the input dramatically. This leads to a significant reduction of computer memory utilization when optimization procedures are implemented as computer programs.

The majority of the presented preprocessings utilize the fact that certain decisions will not be completely independent of each others. Consider, for example, a node of degree two. There is of course a relation between the modules installed on the two incident edges. Thus there is no need to model two edges, but instead a single joined edge is sufficient. However, care has to be taken as to how the solution from the preprocessed problem can be mapped back to the original. Other preprocessings are of the form that a certain decision will never be made in an optimal solution. Thus the respective decision variable is unnecessary and can be removed without any need for mappings. Correspondingly if a certain decision will be made in any solution there is again no need for a decision variable. It is sufficient to remember the cost of the decision and to take care about this unconditional decision in the mapping. Finally sometimes one can determine certain cases where a LAN problem is infeasible. In these cases there is no need to define a preprocessed problem and clearly no mapping is needed.

The transformations can be applied iteratively. Assume, in iteration i we are given a LAN design problem L^i. By applying a preprocessing technique we derive another, smaller problem L^{i+1}. Any feasible solution S^{i+1} for L^{i+1} with an objective value of o^{i+1} can be transformed back into a feasible solution S^i for the original problem L^i with the same objective value $o^{i+1} = o^i$. It follows that also the optimal objective value of L^i and L^{i+1} will be the same, thus these methods are *exact* preprocessings. The following listing describes a set of preprocessing steps and the corresponding mappings from L^i to L^{i+1} as well as the corresponding back-mappings from S^{i+1} to S^i, where needed. For the context of these preprocessings, the LAN problem L^i is defined as a graph $G^i = (V^i, E^i)$, a central node r^i, demands $K^i \subseteq V^i \setminus \{r^i\}, d^i \in \mathbb{R}_{\geq 0}^{|K^i|}$, modules $u^i_{e,m}, c^i_{e,m}, M^i_e$ and a newly introduced *fixed cost* term F^i. The modified objective function, including the fixed-cost term is:

$$\min \sum_{e \in E^i} \sum_{m \in M^i_e} u^i_{e,m} x^i_{e,m} + F^i.$$

Denote by the function $\mu_e : \mathbb{R}_{\geq 0} \mapsto M_e$ the *most appropriate module* on edge e for some required capacity, i.e., the cheapest module with sufficient capacity, or simply the largest module if there is no module with sufficient capacity. More formally for some requested capacity $U > 0$:

$$\mu_e(U) := \begin{cases} \arg\min_{\{m \in M_e | u_{e,m} \geq U\}} c_{e,m} & \text{if } \exists m \in M_e | \, u_{e,m} \geq U \\ |M_e| & \text{otherwise.} \end{cases} \tag{3.1}$$

Let $\mu_e^i(U)$ denote this function with respect to the i-th step in the sequence of preprocessings. The preprocessing steps primarily deal with nodes with degree zero, one and two.

(i) **Degree zero, center node:**
If the center node r^i has degree 0, the instance is infeasible.

(ii) **Degree zero, Steiner node:**
If there is a non-customer, non-center node v with degree 0, this node will certainly not be in any solution, hence it can be deleted from the instance: $V^{i+1} := V^i \setminus \{v\}$.

(iii) **Degree zero, customer node:**
If there is a customer node k with degree 0, the instance is infeasible.

(iv) **Degree one, center node:**
If the center node r^i has degree 1 and the incident edge $e = \{r^i, v\}$ provides a module with sufficient capacity for $\sum_{k \in K^i} d_k^i$, this edge will be in any solution. Therefore it can be deleted: $E^{i+1} := E^i \setminus \{e\}, V^{i+1} := V^i \setminus \{r^i\}$, the center is moved to the adjacent node: $r^{i+1} := v$ and we can easily compute the module $\tilde{m} := \mu_e^i \left(\sum_{k \in K^i} d_k^i \right)$ and only keep the cost $F^{i+1} := F^i + c_{e,\tilde{m}}^i$. For the back-mapping it must be noted that $e, \tilde{m}$ is included in the solution S^i.

If on the other hand e does not provide sufficient capacity, the problem is infeasible.

(v) **Degree one, Steiner node:**
If there is a non-customer, non-center node v with degree 1, this node will certainly not be in any solution. Therefore v and the incident edge $\{v, w\}$ can be deleted from the instance: $E^{i+1} := E^i \setminus \{\{v, w\}\}, V^{i+1} := V^i \setminus \{v\}$.

(vi) **Degree one, customer node:**
If there is a customer node k with demand d_k^i with degree 1 and the incident edge $e = \{k, v\}$ provides a module with sufficient capacity for d_k^i, this edge will be in any solution. Therefore it can be deleted from the instance: $E^{i+1} := E^i \setminus \{e\}, K^{i+1} := K^i \setminus \{k\}, V^{i+1} := V^i \setminus \{k\}$ and we only keep the cost $F^{i+1} := F^i + c^i_{e, \mu_e^i \left(d_k^i \right)}$. The demand is moved to the adjacent node v:

If v is a customer, its demand is increased to $d_v^{i+1} := d_v^i + d_k^i$. Otherwise v becomes a customer $K^{i+1} := K^{i+1} \cup \{v\}$ with demand $d_v^{i+1} := d_k^i$. For the back-mapping it must be noted that $e, \mu_e^i\left(d_k^i\right)$ is included in the solution S^i.

If the edge e does not provide sufficient capacity, the instance is infeasible.

(vii) **Degree two, Steiner node:**

If there is a non-customer, non-center node w with degree 2, then either both incident edges $\{v, w\}, \{w, z\}$ will be in the solution or none. Hence these two sequential edges can be replaced by one edge: $E^{i+1} := E^i \setminus \{\{v, w\}, \{w, z\}\} \cup \{\{v, z\}\}, V^{i+1} := V^i \setminus \{w\}$. The modules for the new edge $M_{\{v,z\}}^{i+1}$ result from installing one module from each of the two original edges $\{v, w\}, \{w, z\}$ in series. More precisely, every pair of modules $\langle m_a, m_b \rangle \in M_{\{v,w\}}^i \times M_{\{w,z\}}^i$ implies a new module $\tilde{m}$ with $u_{\{v,z\},\tilde{m}}^{i+1} := \min\left(u_{\{v,w\},m_a}^i, u_{\{w,z\},m_b}^i\right)$ and $c_{\{v,z\},\tilde{m}}^{i+1} := c_{\{v.w\},m_a}^i + c_{\{w,z\},m_b}^i$. This leads to $\left|M_{\{v,z\}}\right| = \left|M_{\{v,w\}}\right| \cdot \left|M_{\{w,z\}}\right|$ steps for the new edge $\{v, z\}$. For the back-mapping it must be recorded that if the new edge $\{v, z\}$ is in the solution S^{i+1} with the module $\tilde{m} \in M_{\{v,z\}}^{i+1}$ it implies that both edges $\{v, w\}, \{w, z\}$ are in S^i with the respective modules that were combined to make up $\tilde{m}$.

Dispensable modules are removed from $M_{\{v,z\}}^{i+1}$ in Step (ix). Note that there may already be an edge from v to z so we temporarily allow for parallel edges. See Step (viii) for a resolution.

(viii) **Parallel edges:**

Step (vii) may result in two parallel edges $e = \{v, w\}, h = \{v, w\} \in E^i$. A solution may utilize either only one of these two edges or both of them. Therefore, they can be replaced by a single edge $g = \{v, w\} : E^{i+1} = E^i \setminus \{e, h\} \cup \{g\}$. The modules for this new edge M_g^{i+1} result from all modules in M_e^i, united with all modules in M_h^i, united with all possible combinations of one module from M_e^i and one from M_h^i. More precisely, every pair of modules $\langle m_a, m_b \rangle \in M_e^i \times M_h^i$ implies a new module $\tilde{m}$ with $u_{g,\tilde{m}}^{i+1} := u_{e,m_a}^i + u_{h,m_b}^i$ and $c_{g,\tilde{m}}^{i+1} := c_{e,m_a}^i + c_{h,m_b}^i$. In summary, this leads to $\left|M_g^{i+1}\right| = \left|M_e^i\right| + \left|M_h^i\right| + \left|M_e^i\right| \cdot \left|M_h^i\right|$ steps for the new edge g. For the back-mapping it must be noted that if $g, \tilde{m}$ is in S^{i+1} it implies that the edges and modules from e, h, M_e^i, M_h^i that make up $\tilde{m}$ are in S^i.

(ix) **Dispensable modules:**

Steps (vii) and (viii) may lead to *dispensable* modules. A module $\tilde{m} \in M_e^i$ is dispensable if there exists another module $m' \in M_e^i$ with $u_{e,m'}^i \geq u_{e,\tilde{m}}^i$

and $c^i_{e,m'} \leq c^i_{e,\tilde{m}}$. A dispensable module $\tilde{m}$ will certainly not be in any solution, hence can be deleted: $M^{i+1}_e := M^i_e \setminus \{\tilde{m}\}$.

(x) **Excess modules:**
No optimal solution needs to have any installation greater than $\sum_{k \in K^i} d^i_k$. (See the proof for acyclic solutions in the following Section 3.2) Consequently, sets of excess modules $\tilde{M}_e = \left\{m \in M^i_e \mid u^i_{e,m} \geq \sum_{k \in K^i} d^i_k\right\} \subseteq M^i_e$ can be replaced by a single module $\tilde{m}$: $M^{i+1}_e = M^i_e \setminus \tilde{M}_e \cup \{\tilde{m}\}$ with $c^{i+1}_{e,\tilde{m}} = \min_{m \in \tilde{M}_e} c^i_{e,m}$ and $u^{i+1}_{e,\tilde{m}} = \sum_{k \in K^i} d^i_k$. For the back-mapping it must be noted that if $\tilde{m}$ is used on e in S^{i+1} it implies that the cheapest excess module $\arg\min_{m \in \tilde{M}_e} c^i_{e,m}$ is used on e in S^i.

Note that center or customer nodes with degree two do not allow for direct implications about the modules installed on the incident edges. Therefore there are no corresponding preprocessing steps. The preprocessing is implemented as follows: Iterate over all nodes and perform any applicable preprocessing for nodes with degree zero, one or two, i.e., Steps (i)-(vii). Preprocessing Step (vii) always triggers an attempt to apply Steps (viii) and (ix). This iteration is performed repeatedly until no more preprocessing steps for nodes with degree zero, one or two can be applied. Finally, Step (x) is performed once, in order to remove excess modules from the input.

3.2 Transformation into a Directed Problem

It is well known that the MIP formulations of uncapacitated network design problems on directed graphs often provide better lower bounds than their undirected counterparts (see e.g., [CR94]). However, the MIP approaches to LAN presented in the previous literature (see [RS06, SRH08]) involve undirected graphs. This section describes the transformation from the undirected LAN problem into a directed version of LAN. We prove that the directed version is equivalent to the original undirected definition with respect to feasibility, optimal solutions and objective values. The following sections present MIP models based on this transformation and Section 3.8 proves that these models do indeed generate better lower bounds than the corresponding undirected models.

Consider the LAN problem as defined in Section 1.2 and define a bidirected set of arcs A. Every edge in E implies a forward and a backward arc in A, i.e., $A := \{(i,j),(j,i) \mid \{i,j\} \in E\}$.

Theorem 3.2.1. *If a LAN problem is feasible, then there always exists an optimal solution $\boldsymbol{x} \in \{0,1\}^{|E|}$ and $\boldsymbol{f} \in \mathbb{R}^{|A|}_{\geq 0}$ such that the strictly positive elements of $\boldsymbol{f}$ induce a directed subgraph of $G = (V, A)$ which is cycle free.*

Proof. Let $(\boldsymbol{x}, \boldsymbol{f})$ be an optimal solution of the LAN problem such that $\boldsymbol{f}$ implies a directed cycle $C \subseteq A$. Denote by $\widetilde{a} \in A$ the arc in C with the smallest flow on C, i.e., $\widetilde{a} = \arg\min_{a \in C} f_a$. Construct a new flow $\boldsymbol{f}'$ such that $f'_a = f_a \forall a \notin C$, and $f'_a = f_a - f_{\widetilde{a}} \forall a \in C$. It follows that $f'_{\widetilde{a}} = 0$, hence the strictly positive elements of $\boldsymbol{f}'$ do not contain the cycle C. The flow $\boldsymbol{f}'$ certainly satisfies the flow conservation constraints (1.2). Since $f'_a \leq f_a \forall a \in A$ the new flow also satisfies the capacity constraints (1.3). Moreover, also the design variables $\boldsymbol{x}$ can be reduced since there is less flow along C. Denote the module in use on arc $\widetilde{a}$ in the design $\boldsymbol{x}$ by $\widetilde{m}$. After the flow-reduction, $x_{\widetilde{a},m} = 0 \forall m \in M_{\widetilde{a}}$ is compatible with $\boldsymbol{f}'$. This implies that the cost $c_{\widetilde{a},\widetilde{m}}$ must necessarily be 0, since a positive cost would contradict the optimality of $(\boldsymbol{x}, \boldsymbol{f})$. If $\boldsymbol{f}'$ implies yet another cycle, the same argument can be applied repeatedly. In each application the number of arcs with positive flow is reduced by one. This proves that in a finite number of steps each optimal solution can be transformed into an acyclic optimal solution. □

Clearly, in a solution with no cycles there will especially be no cycles of length two, i.e., no edge will have forward and backward flow. Redefine the modules on the edges in terms of symmetric modules on arcs, i.e., $c_{ij,m} = c_{ji,m} = c_{\{i,j\},m}$, $u_{ij,m} = u_{ji,m} = u_{\{i,j\},m}$ for all $m \in M_{ij} = M_{ji} = M_{\{i,j\}}$. To solve a LAN problem, we now search for the *directed solution*, i.e., for the installation of at most one module on every *arc* such that there is enough capacity to route the flow from r to every $k \in K$. Obviously a directed, cycle free solution corresponds to an equivalent undirected, cycle free solution with the same objective value. The following sections present MIP models on this directed problem.

Note that the transformation from an undirected into a directed graph is not valid for general multi-commodity network design problems. There will in general be optimal solutions that have flow in both directions on some edges. With respect to the LAN problem, which is a single-commodity problem, no edge will be used in both directions. Moreover, no edge incident to the center node r will have flow towards the node r, i.e., $x_{ir,m} = 0$ and $f_{ir} = 0$. Consequently these arcs can simply be left out from the definition of the set A. This alternative definition of A leading to an equivalent MIP is $A := \{(i,j),(j,i) \mid \{i,j\} \in E; i,j \neq r\} \cup \{(r,j) \mid \{r,j\} \in E\}$. Only for the sake of a simpler notation, the set A including these superfluous variables will be used throughout this work.

3.3 Basic MIP Models

This sections presents two basic models. The first, (SCF) is a flow model and the second, (CUT) is based on cut-set inequalities.

3.3.1 Single-Commodity Flow

The directed *single-commodity flow* formulation, (SCF), uses directed variables to describe the design and the routing. The binary design $x_{a,m}$ are equal to 1 iff the module m shall be installed on the arc a. The flow variables $\boldsymbol{f} \in \mathbb{R}^{|A|}_{\geq 0}$ are equivalently defined as in the undirected model (uSCF) in Section 1.2 and describe the amount of flow running in each direction along an edge. The arcs emanating from node i are denoted by $\delta^+(i) := \{(i,j) \in A\}$ and the arcs enter i are denoted by $\delta^-(i) := \{(j,i) \in A\}$.

$$\text{(SCF)} : \qquad \min \sum_{a \in A} \sum_{m \in M_a} c_{a,m} x_{a,m} \tag{3.2}$$

s.t.

$$\sum_{(i,j) \in \delta^+(i)} f_{(i,j)} - \sum_{(j,i) \in \delta^-(i)} f_{(j,i)} = \begin{cases} -d_i, & i \in K \\ \sum_{k \in K} d_k, & i = r \\ 0, & \text{otherwise} \end{cases} \qquad \forall i \in V \tag{3.3}$$

$$f_a \leq \sum_{m \in M} u_{a,m} x_{a,m} \qquad \forall a \in A \tag{3.4}$$

$$\sum_{m \in M_a} x_{a,m} \leq 1 \qquad \forall a \in A \tag{3.5}$$

$$x_{a,m} \in \{0,1\} \qquad \forall a \in A,\ \forall m \in M_a \tag{3.6}$$

$$f_a \geq 0 \qquad \forall a \in A. \tag{3.7}$$

Following the proof of Theorem 3.2.1 there is always a solution that does not use any pair of oppositely directed arcs (i,j) and (j,i). However, solving the (SCF) model, may produce an optimal solution that does use oppositely directed arcs even if there is another optimal solution that does not use oppositely directed arcs. In order to only produce solutions that satisfy this property the following ***subtour elimination*** constraints of length two $x_{(i,j),m} + x_{(j,i),m} \leq 1$, for all $\{i,j\} \in E$, and all $m \in M_{\{i,j\}}$, can be added to the model. Alternatively, instead of adding these constraints, the disjunction constraints (3.5) can be replaced by:

$$\sum_{m \in M_{ij}} (x_{ij,m} + x_{ji,m}) \leq 1 \quad \forall (i,j) \in A. \tag{3.8}$$

The (SCF) model contains $\mathcal{O}(|A| \cdot |M|)$ variables and constraints. In case of economies of scale, the LP relaxation of the (SCF) model has an optimal solution in which at most one of $x_{a,m}$ variables (the one with the lowest $c_{a,m}/u_{a,m}$ ratio) on every arc is non-zero (see also [SRH08]).

3.3.2 Cut-Set Model

An alternative to the flow based formulation of the previous section is to use cut-set formulations. This section recalls the cut-set formulation for LAN on directed graphs. For each subset $S \subseteq V$, denote the set of outgoing and ingoing arcs by $\delta^+(S) := \{(i,j) \in A \mid i \in S, j \in V \setminus S\}$ and $\delta^-(S) := \{(i,j) \in A \mid i \in V \setminus S, j \in S\}$, respectively. The directed cut-set formulation (CUT) is given as follows:

$$\text{(CUT)} : \qquad \min \sum_{a\in A} \sum_{m \in M_a} c_{a,m} x_{a,m} \tag{3.9}$$

s.t.

$$\sum_{a\in\delta^-(S)} \sum_{m\in M_a} u_{a,m} x_{a,m} \geq \sum_{k\in S} d_k \quad \forall S \subset V, S \cap K \neq \varnothing, r \notin S \tag{3.10}$$

$$\sum_{m\in M_a} x_{a,m} \leq 1 \qquad \forall a \in A \tag{3.11}$$

$$x_{a,m} \in \{0,1\} \quad \forall a \in A,\ \forall m \in M_a \tag{3.12}$$

The *cut-set inequalities* (3.10) state that every set of nodes, not containing r must have incoming capacity at least as large as the total demand requested inside the set. In the general case of multiple-source multiple-sink network design problems the separation problem of cut-set inequalities can be reduced to the max-cut problem and is NP-hard [Bar96]. However, the cut-set inequalities (3.10) for LAN can be separated in polynomial time as follows. For a given fractional solution $\boldsymbol{x}'$, we define the directed *support graph* $G' = (V', A')$ where $V' := V \cup \{t\}$ with an additional sink t, and $A' := A_1 \cup A_2$ being $A_1 := \{(i,j) \in A \mid \sum_{m\in M_{ij}} u_{ij,m} x'_{ij,m} > 0\}$ and $A_2 := \{(k,t) \mid k \in K\}$. The capacity associated to each arc $a = (i,j) \in A_1$ is set to $\sum_{m\in M} u_{ij,m} x'_{ij,m}$, and the capacity of each arc $a = (k,t) \in A_2$ is set to d_k. If the minimum cut between r and t in G' is less than $\sum_{k\in K} d_k$, it defines a violated inequality (3.10).

Since $x_{a,m}$ variables are binary, the cut-set inequalities can be strengthened by rounding (see Appendix A.3):

$$\sum_{a\in\delta^-(S)} \sum_{m\in M_a} \min\left(u_{a,m}, \sum_{k\in S} d_k\right) x_{a,m} \geq \sum_{k\in S} d_k.$$

3.4 Disaggregated MIP Models

This section shows how the (SCF) model can be strengthened by disaggregation. By increasing the number of variables and with the help of additional constraints, *stronger* models can be formulated. For two linear relaxation of LAN-MIP formulations, F_1 and F_2, we say that F_1 is stronger than F_2 if every x, that is feasible for F_1, is also feasible for F_2 and in addition, there exist LAN instances for which certain x are F_2-feasible but not F_1-feasible. In other words, F_1 forms a proper subset of F_2. Section 3.4.1 shows a disaggregation by commodity and Section 3.4.2 further disaggregates the model by modules. The detailed comparison of the models is a part of the later Section 3.8.

3.4.1 Multi-Commodity Flow

This section shows a disaggregation by *commodities*. Commodities in this case are source-sink pairs (r, k), $\forall k \in K$. A similar formulation is commonly used for multiple-source multiple-sink network design problems (see, e.g., [MMV95]). In this model each commodity can be directly associated to a customer $k \in K$. The continuous flow variables f_{ij}^k describe the amount of flow of commodity $k \in K$ routed through the arc (i, j). The (MCF) model reads as follows:

$$\text{(MCF)} : \qquad \min \sum_{a\in A} \sum_{m\in M_a} c_{a,m} x_{a,m} \tag{3.13}$$

s.t.

$$\sum_{(i,j)\in\delta^+(i)} f_{(i,j)}^k - \sum_{(j,i)\in\delta^-(i)} f_{(j,i)}^k = \begin{cases} -d_k, & i = k \\ d_k, & i = r \\ 0, & \text{otherwise} \end{cases} \qquad \forall i \in V,\ \forall k \in K \tag{3.14}$$

$$\sum_{k\in K} f_a^k \le \sum_{m\in M_a} u_{a,m} x_{a,m} \qquad \forall a \in A \tag{3.15}$$

$$\frac{f_a^k}{d_k} \le \sum_{m\in M_a} x_{a,m} \qquad \forall a \in A,\ \forall k \in K \tag{3.16}$$

$$\sum_{m\in M_a} x_{a,m} \le 1 \qquad \forall a \in A \tag{3.17}$$

$$x_{a,m} \in \{0, 1\} \qquad \forall a \in A,\ \forall m \in M_a \tag{3.18}$$

$$f_a^k \ge 0 \qquad \forall a \in A,\ \forall k \in K \tag{3.19}$$

The *flow conservation constraints* (3.14) describe the flow for each customer independently. The *capacity constraints* (3.15) state that the total flow per arc must not exceed the installed capacity. The *coupling constraints* (3.16) ensure that if there is flow in any module m on the arc (i, j), then the corresponding design variables need to be set to at least the given ratio per each commodity in the linear relaxation. These constraints are redundant for the integral formulation, but they improve the lower bound of the LP relaxation.

The (MCF) model contains $\mathcal{O}(|A| \cdot |M| + |A| \cdot |K|)$ variables and $\mathcal{O}(|V| \cdot |K| + |A| \cdot |M| + |A| \cdot |K|)$ constraints. The (MCF) model without the coupling constraints (3.16) is completely equivalent to the SCF model, with respect to the LP relaxation (see Section 3.8).

3.4.2 Disaggregated Multi-Commodity Flow

For the multi-commodity capacitated network design problem, Croxton et al. [CGM07] and Frangioni and Gendron [FG09] propose a disaggregation by integer values in a MIP based on the multi-commodity flow formulation. By adapting this disaggregation technique to LAN, we disaggregate flow variables with respect to modules. Beside the binary design variables, $x_{ij,m} \in \{0,1\}$, we use the disaggregated flow variables $f^k_{ij,m}$ that define the amount of flow of commodity $k \in K$, routed through the arc (i,j) using the module $m \in M_{ij}$. The (DMCF) model reads as follows:

$$\text{(DMCF)} : \qquad \min \sum_{a \in A} \sum_{m \in M_a} c_{a,m} x_{a,m} \tag{3.20}$$

s.t.

$$\sum_{(i,j) \in \delta^+(i)} \sum_{m \in M_{(i,j)}} f^k_{(i,j),m} - \sum_{(j,i) \in \delta^-(i)} \sum_{m \in M_{(j,i)}} f^k_{(j,i),m} = \begin{cases} -d_k, & i = k \\ d_k, & i = r \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in V,\ \forall k \in K \tag{3.21}$$

$$\sum_{k \in K} f^k_{a,m} \le u_{a,m} x_{a,m} \qquad \forall a \in A,\ \forall m \in M_a \tag{3.22}$$

$$\frac{f^k_{a,m}}{d_k} \le x_{a,m} \qquad \forall a \in A,\ \forall m \in M_a,\ \forall k \in K \tag{3.23}$$

$$\sum_{m \in M_a} x_{a,m} \le 1 \qquad \forall a \in A \tag{3.24}$$

$$x_{a,m} \in \{0,1\} \qquad \forall a \in A,\ \forall m \in M_a \tag{3.25}$$

$$f^k_{a,m} \ge 0 \qquad \forall a \in A,\ \forall m \in M_a,\ \forall k \in K. \tag{3.26}$$

The *flow conservation constraints* have the same meaning as for the (MCF) model in Section 3.3.1. The *capacity constraints* (3.22) ensure that the total flow over module m on arc a must not exceed the capacity of the given module m. Constraints (3.23) couple the design variables to the fraction of flow on the corresponding arc and module. Again, these coupling constraints are redundant for the MIP formulation, but they improve the objective value of the LP relaxation.

The (DMCF) model contains $\mathcal{O}(|A| \cdot |M| \cdot |K|)$ constraints and $\mathcal{O}(|A| \cdot |M| \cdot |K| + |V| \cdot |K|)$ variables. Due to this large number of variables and constraints it is unlikely that even the most sophisticated MIP solvers may solve instances of moderate size using the (DMCF) formulation directly in a typical branch-and-bound fashion. Computational experiments with the (DMCF) model confirm this claim (see Section 3.10). In order to facilitate the strength of this model it is proposed to project out the flow variables and to introduce Benders' inequalities instead, keeping the quality of lower bounds, and even improving them by rounding techniques. This is described in detail in Section 3.5.

3.4.3 Disaggregated Single-Commodity Flow

This section shows the (DSCF) model, which results from disaggregating the flow and design variables of the (SCF) model by modules. The disaggregated design variables $x_{a,m}$ are equal to 1 iff module m is used on arc a just like in the (DMCF) model above. The disaggregated flow variables $f_{a,m}$ denote the total flow routed over arc a using module m.

$$\text{(DSCF)} : \qquad \min \sum_{a \in A} \sum_{m \in M_a} c_{a,m} x_{a,m}$$

s.t.

$$\sum_{(i,j) \in \delta^+(i)} \sum_{m \in M_{(i,j)}} f_{(i,j),m} - \sum_{(j,i) \in \delta^-(i)} \sum_{m \in M_{(j,i)}} f_{(j,i),m} = \begin{cases} -d_i, & i \in K \\ \sum_{k \in K} d_k, & i = r \\ 0, & \textit{otherwise} \end{cases} \quad \forall i \in V \tag{3.27}$$

$$f_{a,m} \le u_{a,m} x_{a,m} \qquad \forall a \in A,\ \forall m \in M_a \tag{3.28}$$

$$\frac{f_{a,m}}{\sum_{k \in K} d_k} \le x_{a,m} \qquad \forall a \in A,\ \forall m \in M_a \tag{3.29}$$

$$\sum_{m \in M_a} x_{a,m} \le 1 \qquad \forall a \in A \tag{3.30}$$

$$x_{a,m} \in \{0,1\} \qquad \forall a \in A,\ \forall m \in M_a \tag{3.31}$$

$$f_{a,m} \ge 0 \qquad \forall a \in A,\ \forall m \in M_a. \tag{3.32}$$

It must be noted that this disaggregation does not lead to a stronger MIP model. It is proved in Theorem 3.8.1 that (DSCF) is is equivalent to (SCF) under the simple assumption that there are no unnecessary excess capacities, i.e., $u_{a,m} \le \sum_{k \in K} d_k$.

3.5 Benders' Decomposition

The previous section presents disaggregated flow models for the LAN problem. In the later Section 3.8 it is demonstrated, that the (DMCF) model is stronger than the (MCF) model, which in turn is stronger than the (SCF) model. This strength comes at the cost of additional variables and constraints making the models larger and more challenging to handle. A classical approach to deal with large linear programs is Benders' decomposition. See Section A.2 for a detailed explanation. The basic idea is to remove a subset of the constraints and the variables, solely used within this subset from the linear program. This forms a reduced *master* problem. The part that has been left out is treated via a series of *subproblems*. These subproblems are used to separate *Benders' inequalities* that have to be included in the master.

This section shows the Benders' decomposition approach applied to the linear relaxations of the three models, (SCF), (MCF) and (DMCF). Since the (DSCF) model is equivalent to the (SCF) model it is not considered here explicitly. The three models share the same *design part*. By relaxing integrality and all constraints concerning flow we end up with the same initial master problem for all three models:

$$\text{(MASTER)} : \qquad \min \sum_{a \in A} \sum_{m \in M_a} c_{a,m} x_{a,m} \tag{3.33}$$

s.t.

$$\sum_{m \in M_a} x_{a,m} \leq 1 \qquad \forall a \in A \tag{3.34}$$

$$x_{a,m} \in [0,1] \quad \forall a \in A,\ \forall m \in M_a. \tag{3.35}$$

A solution of this master problem is feasible for (SCF), (MCF), or (DMCF) iff there exists a compatible flow. That means flow variables that satisfy flow conservation and capacity constraints of (SCF), (MCF), or (DMCF), respectively. For (MCF) and (DMCF) a compatible flow must also satisfy the coupling constraints. The formal definitions of the three subproblems used to separate Benders' inequalities are given in the following sections.

Since the flow variables are not present in the objective functions, the subproblems are mere *feasibility problems* that ask whether there exists a flow for the given value of $\boldsymbol{x}$. Consequently, the Benders' decomposition involves only feasibility cuts and no optimality cuts. Furthermore, the flow conservation constraints are equally valid if the equality is replaced by a less-than inequality. Both transformations are applied for the dualizations of the subproblems presented in the following sections.

3.5.1 Benders' Decomposition for SCF

A solution $\boldsymbol{x}'$ of the master problem (3.33)-(3.35) defines a feasible solution for the LP-relaxation of (SCF) iff there exist flow variables $\boldsymbol{f} \in \mathbb{R}^{|A|}_{\geq 0}$ satisfying the following primal subproblem.

$$\sum_{(i,j)\in\delta^+(i)} f_{(i,j)} - \sum_{(j,i)\in\delta^-(i)} f_{(j,i)} \leq \begin{cases} -d_i, & i \in K \\ \sum_{k\in K} d_k, & i = r \\ 0, & \text{otherwise} \end{cases} \qquad \forall i \in V \tag{3.36}$$

$$f_a \leq \sum_{m\in M_a} u_{a,m} x'_{a,m} \qquad \forall a \in A \tag{3.37}$$

$$f_a \geq 0 \qquad \forall a \in A \tag{3.38}$$

These constraints correspond to (3.3), (3.4) and (3.7) for a fixed design vector $\boldsymbol{x}'$. Denote the dual variables associated with the flow conservation constraints (3.36) with $\boldsymbol{\alpha}$ and the duals associated with the capacity constraints (3.37) with $\boldsymbol{\gamma}$. This yields the following dual subproblem SCF($\boldsymbol{x}'$):

$$\text{SCF}(\boldsymbol{x}') : \qquad \min \sum_{k\in K} (\alpha_r - \alpha_k) d_k + \sum_{a\in A} \gamma_a \sum_{m\in M_a} u_{a,m} x'_{a,m} \tag{3.39}$$

s.t.

$$\alpha_i - \alpha_j + \gamma_{(i,j)} \geq 0 \qquad \forall (i,j) \in A \tag{3.40}$$

$$(\boldsymbol{\alpha}, \boldsymbol{\gamma}) \geq \mathbf{0} \tag{3.41}$$

Note that this dual subproblem SCF($\boldsymbol{x}'$) is always feasible since it contains the trivial solution $(\boldsymbol{\alpha}, \boldsymbol{\gamma}) = \mathbf{0}$ which yields an objective value of zero. If there is no other solution with a negative objective value, then the dual subproblem is bounded. It follows that also the primal subproblem is feasible and bounded. This in turn implies that $\boldsymbol{x}'$ is an optimal solution of the linear relaxation of the (SCF) model.

If on the other hand there exists another solution of the dual subproblem which yields a strictly negative objective value, denote it by $(\boldsymbol{\alpha}', \boldsymbol{\gamma}') > \mathbf{0}$. It follows that the dual subproblem SCF($\boldsymbol{x}'$) is unbounded, hence the primal subproblem is infeasible. Farkas' lemma (see Section A.1) states that a point $\boldsymbol{x}$ from the (MASTER) problem is feasible for (SCF) iff the following inequality is satisfied for every dual point $(\boldsymbol{\alpha}, \boldsymbol{\gamma})$ that satisfies (3.40)-(3.41):

$$\sum_{k\in K} (\alpha_r - \alpha_k) d_k + \sum_{a\in A} \gamma_a \sum_{m\in M_a} u_{a,m} x_{a,m} \geq 0. \tag{3.42}$$

So, in order to remove an infeasible point $\boldsymbol{x}'$ from the (MASTER) problem, we can add the following Benders' inequality:

$$\sum_{a \in A} \gamma'_a \sum_{m \in M_a} u_{a,m} x_{a,m} \geq \sum_{k \in K} (\alpha'_k - \alpha'_r) d_k. \tag{3.43}$$

Now, instead of solving the LP relaxation of (SCF) directly, it is possible to produce a solution with a cutting plane scheme: Iteratively solve the master problem, compute an unbounded direction of the dual subproblem and add the corresponding Benders' cut to the master problem until the dual subproblem is bounded. At this point the solution of the master problem $\boldsymbol{x}'$, is a solution for the LP relaxation of the (SCF) model.

Inequalities like (3.43) can be strengthened by rounding down the coefficients of the binary variables $x_{a,m}$ to the value of the right hand side (see appendix A.3). So instead of using the Benders' inequalities (3.43) in the cutting plane algorithm one can use *rounded Benders' inequalities*:

$$\sum_{a \in A} \sum_{m \in M_a} \min \left((\gamma'_a u_{a,m}), \sum_{k \in K} (\alpha'_k - \alpha'_r) d_k \right) x_{a,m} \geq \sum_{k \in K} (\alpha'_k - \alpha'_r) d_k \tag{3.44}$$

This has the potential to eventually produce an objective value of the final master problem that is greater than the objective value of the linear relaxation of the (SCF) model.

3.5.2 Benders' Decomposition for MCF

Similarly to the previous section, the primal subproblem for the Benders' decomposition of the LP relaxation of the (MCF) model is a feasibility problem that asks whether there exists a flow $\boldsymbol{f}$ that satisfies the following system:

$$\sum_{(i,j) \in \delta^+(i)} f^k_{(i,j)} - \sum_{(j,i) \in \delta^-(i)} f^k_{(j,i)} \leq \begin{cases} -d_k, & i = k \\ d_k, & i = r \\ 0, & \text{otherwise} \end{cases} \qquad \forall i \in V,\ \forall k \in K \tag{3.45}$$

$$\sum_{k \in K} f^k_a \leq \sum_{m \in M_a} u_{a,m} x'_{a,m} \qquad \forall a \in A \tag{3.46}$$

$$\frac{f^k_a}{d_k} \leq \sum_{m \in M_a} x'_{a,m} \qquad \forall a \in A,\ \forall k \in K \tag{3.47}$$

$$f^k_a \geq 0 \qquad \forall a \in A,\ \forall k \in K \tag{3.48}$$

Assigning dual variables $\boldsymbol{\alpha}, \boldsymbol{\gamma}$, and $\boldsymbol{\beta}$ to inequalities (3.45), (3.46) and (3.47), respectively yields the following dual subproblem:

MCF($\boldsymbol{x}'$) :

$$\min \sum_{k\in K} \left(\alpha_r^k - \alpha_k^k\right) d_k + \sum_{a\in A}\sum_{k\in K} \beta_a^k d_k \sum_{m\in M_a} x'_{a,m} + \sum_{a\in A} \gamma_a \sum_{m\in M_a} u_{a,m} x'_{a,m} \quad (3.49)$$

s.t.

$$\alpha_i^k - \alpha_j^k + \gamma_{(i,j)} + \beta_{(i,j)}^k \geq 0 \qquad \forall (i,j)\in A,\ \forall k \in K \quad (3.50)$$

$$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \geq 0 \quad (3.51)$$

If for some $(\boldsymbol{\alpha}', \boldsymbol{\beta}', \boldsymbol{\gamma}')$ that satisfies (3.50)-(3.51) the objective value (3.49) is negative, we know from Farkas' lemma, that the point $\boldsymbol{x}'$ is not feasible for the linear relaxation of (MCF). This point can be cut off by adding the following Benders' cut to the master LP.

$$\sum_{a\in A}\sum_{k\in K} \beta_a'^k d_k \sum_{m\in M_a} x_{a,m} + \sum_{a\in A} \gamma'_a \sum_{m\in M_a} u_{a,m} x_{a,m} \geq \sum_{k\in K} \left(\alpha_k'^k - \alpha_r'^k\right) d_k \quad (3.52)$$

In the corresponding rounded Benders' inequality, the coefficients of x_{am} are replaced by

$$\min\left(\left(\gamma'_a u_{a,m} + \sum_{k\in K} \beta_a'^k d_k\right), \sum_{k\in K} (\alpha_k'^k - \alpha_r'^k) d_k\right). \quad (3.53)$$

3.5.3 Benders' Decomposition for DMCF

The primal feasibility subproblem for the linear relaxation of the (DMCF) model is:

$$\sum_{(i,j)\in\delta^+(i)} \sum_{m\in M_{(i,j)}} f_{(i,j),m}^k - \sum_{(j,i)\in\delta^-(i)} \sum_{m\in M_{(j,i)}} f_{(j,i),m}^k \leq \begin{cases} -d_k, & i = k \\ d_k, & i = r \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in V,\ \forall k \in K \quad (3.54)$$

$$\sum_{k\in K} f_{a,m}^k \leq u_{a,m} x'_{a,m} \qquad \forall a \in A,\ \forall m \in M_a \quad (3.55)$$

$$f_{a,m}^k \leq d_k x'_{a,m} \qquad \forall a\in A,\ \forall m \in M_a,\ \forall k \in K \quad (3.56)$$

$$f_{a,m}^k \geq 0 \qquad \forall a\in A,\ \forall m \in M_a,\ \forall k \in K \quad (3.57)$$

The dual subproblem is:

DMCF(x') :

$$\min \sum_{k\in K} \left(\alpha_r^k - \alpha_k^k\right) d_k + \sum_{a\in A}\sum_{m\in M_a}\sum_{k\in K} \beta_{a,m}^k d_k x'_{a,m} + \sum_{a\in A}\sum_{m\in M_a} \gamma_{a,m} u_{a,m} x'_{a,m} \tag{3.58}$$

s.t.

$$\alpha_i^k - \alpha_j^k + \beta_{(i,j),m}^k + \gamma_{(i,j),m} \geq 0 \quad \forall (i,j)\in A,\ \forall m\in M_a,\ \forall k\in K \tag{3.59}$$

$$(\alpha,\beta,\gamma) \geq 0 \tag{3.60}$$

And for an unbounded direction $(\boldsymbol{\alpha}', \boldsymbol{\beta}', \boldsymbol{\gamma}') > \mathbf{0}$ that satisfies (3.59)-(3.60) we can derive the Benders' inequality to cut away the infeasible $\boldsymbol{x}'$:

$$\sum_{a\in A}\sum_{m\in M_a} \left(\gamma'_{a,m} u_{a,m} + \sum_{k\in K} \beta'^k_{a,m} d_k\right) x_{a,m} \geq \sum_{k\in K} (\alpha'^k_k - \alpha'^k_r) d_k \tag{3.61}$$

Also the coefficients on the left hand side can be rounded down to:

$$\begin{aligned}\sum_{a\in A}\sum_{m\in M_a} \min\left(\left(\gamma'_{a,m} u_{a,m} + \sum_{k\in K} \beta'^k_{a,m} d_k\right), \sum_{k\in K} \left(\alpha'^k_k - \alpha'^k_r\right) d_k\right) x_{a,m}& \\ \geq \sum_{k\in K} \left(\alpha'^k_k - \alpha'^k_r\right) d_k&\end{aligned} \tag{3.62}$$

This decomposition now offers the possibility to use the strength of the (DMCF) model without the need to work on the big $\boldsymbol{f} \in \mathbb{R}_{\geq 0}^{|A||M||K|}$ space in the master problem. Instead the Benders' cuts (3.61) can be generated in a piecemeal process. Besides Benders decomposition, also column generation can be used to deal with large models. For a related multi-commodity network design problem and a model similar to (DMCF) this has been done in [FG09, FG10].

3.6 Normalizations of Benders' Subproblems

In the previous section, the primal and dual subproblems for (SCF), (MCF) and (DMCF) have been given. For all three models, the aim is to determine whether a solution $\boldsymbol{x}'$ of (MASTER) is feasible, which is equivalent to the primal subproblem being infeasible, which in turn is equivalent to the dual subproblem being unbounded. If a linear program is unbounded, there will in general be infinitely many rays pointing in an unbounded direction. This opens the question of which ray to choose, since the selected ray uniquely determines the generated Benders' inequality. Costa et al. [CCG09] have proposed an approach for strengthening Benders' cuts associated to non-extreme rays for a general multicommodity network design problem. Their ideas could similarly be extended to the Benders' decompositions considered here, if non-extreme rays were generated. However, when solving an unbounded linear program with the simplex method, one retrieves always an extreme ray. Even when restricting to extreme rays, there are still multiple solutions, in general. This leaves the question, how to select a specific element out of the set of all extreme rays.

In this section several ways of generating Benders' cuts associated to extreme rays are proposed. The feasible regions of the dual subproblems from Section 3.5 are all cones containing the point $\mathbf{0}$. In addition, all three share the property, that the feasible region is the same for any master-solution $\boldsymbol{x}'$. We show different *normalization approaches* obtained by making the dual cone bounded by introducing one additional inequality. This is equivalent to making the primal subproblem feasible by introducing one additional slack variable. An extreme point that solves the normalized subproblem corresponds to an extreme ray of the original subproblem and induces a Benders' cut.

The explanation of the normalizations in this section is restricted to the (DMCF) model as described in Section 3.5.3. The extension to the other models is relatively straight forward. Denote the dual subproblem of the (DMCF) Benders' decomposition, defined in (3.58)-(3.60) by (SUB). The four models (SUBc), (SUBn), (SUBf) and (SUBcap) that will be introduced in this section, denote four different normalizations which result in bounded dual subproblems. The corresponding primal subproblems are feasible, so they can also be solved directly and are denoted by (PSUBc), (PSUBn), (PSUBf) and (PSUBcap) respectively. While it makes no difference from a theoretical perspecitve, whether the dual or the primal subproblem is solved, we are also testing for practical performance differences and therefore consider solving the primal and the dual subproblems explicitly. In total this makes nine variants which are now described in detail.

3.6.1 (SUB) Model

In order to get a violated Benders' inequality, we search for an extreme ray of the unbounded subproblem (SUB). This is also refered to as the *textbook* implementation of Benders' decomposition. As already observed in [Ben62, FSZ10], this approach has a significant drawback: it returns an arbitrarily chosen extreme ray without having any influence on the quality of the violated cut found.

An advantage of this method is that it returns a violated constraint much faster than the corresponding more sophisticated methods described below.

3.6.2 (SUBc) and (PSUBc) Models

Instead of solving the subproblem on the pointed cone, one can make the search space bounded with an additional hyperplane. The following constraint restricts the search space to a subset of the standard simplex.

$$\sum_{(i,j)\in A}\sum_{m\in M_{ij}}\sum_{k\in D}\beta^k_{ij,m} + \sum_{(i,j)\in A}\sum_{m\in M_{ij}}\gamma_{ij,m} + \sum_{i\in V}\sum_{k\in D}\alpha^k_i = 1. \tag{3.63}$$

Obviously, the (MASTER) solution $\boldsymbol{x}'$ is infeasible for (DMCF) if and only if the objective value of SUB($\boldsymbol{x}'$) extended by (3.63) has an objective value that is strictly less than zero. Furthermore, each vertex of such obtained polyhedron (except the origin) corresponds to an extreme ray of the unbounded subproblem.

One easily observes that the model (SUBc) is equivalent to the similar problem of maximizing the value of $\Theta \leq 0$ subject to constraints (3.21), (3.22) and (3.23) in which Θ is added to the left-hand side of each of them. This primal model is denoted by (PSUBc):

$$\text{(PSUBc)} : \qquad \max \Theta \tag{3.64}$$

s.t.

$$\sum_{(i,j)\in\delta^+(i)}\sum_{m\in M_{(i,j)}} f^k_{(i,j),m} - \sum_{(j,i)\in\delta^-(i)}\sum_{m\in M_{(j,i)}} f^k_{(j,i),m} + \Theta \leq \begin{cases} -d_k, & i = k \\ d_k, & i = r \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in V, \forall k \in K \tag{3.65}$$

$$\sum_{k\in K} f^k_{a,m} + \Theta \leq u_{a,m}x'_{a,m} \qquad \forall a \in A,\ \forall m \in M_a \tag{3.66}$$

$$f^k_{a,m} + \Theta \leq d_k x'_{a,m} \qquad \forall a \in A,\ \forall m \in M_a,\ \forall k \in K \tag{3.67}$$

$$f^k_{a,m} \geq 0 \qquad \forall a \in A,\ \forall m \in M_a,\ \forall k \in K \tag{3.68}$$

$$\Theta \leq 0 \tag{3.69}$$

If the optimal value for Θ is equal to zero, $\boldsymbol{x}'$ is feasible. Otherwise the dual variables $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ associated to constraints (3.65), (3.67) and (3.66) for an optimal solution of (PSUBc) define a violated Benders' inequality.

3.6.3 (SUBn) and (PSUBn) Models

The (SUBc) model used a very simple hyperplane to make the dual cone bounded. The (SUBn) model instead adds a hyperplane that leads to Benders' cuts that maximise the *violation* with respect to the current solution $\boldsymbol{x}'$. Recall that $\boldsymbol{x}'$ is infeasible iff the objective function (3.58) is strictly negative. Since $\boldsymbol{x}', \boldsymbol{d}, \boldsymbol{u}, \boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are all nonnegative, it is necessary that the term $\sum_{k \in K} \left(\alpha_r^k - \alpha_k^k\right) d_k$ is strictly negative. Thus the right hand side of a violated (DMCF) Benders' cut (3.61) is strictly positive. Obviously, the left hand side of a violated Benders' cut is smaller than the right hand side and per definition nonnegative. It follows that the ratio between the left hand side and the right hand side of (3.61) is greater or equal to 0 and less than 1 for an infeasible vector $\boldsymbol{x}'$. Therefore, we define the *violation* of a (DMCF) Benders' inequality as one minus this ratio:

$$\texttt{violation}\left(\boldsymbol{\alpha}', \boldsymbol{\beta}', \boldsymbol{\gamma}', \boldsymbol{x}'\right) := 1 - \sum_{a \in A} \sum_{m \in M_a} \frac{\sum_{k \in K} d_k \beta_{a,n}'^k + u_{a,n} \gamma_{a,n}'}{\sum_{k \in K} d_k (\alpha_k'^k - \alpha_r'^k)} \, x_{a,n}' \qquad (3.70)$$

This yields a value in the interval $(0, 1]$, where values close to 0 denote hardly violated inequalities and 1 denotes highly violated inequalities.

The (SUBn) model aims to find highly violated inequalities according to this definition. To achieve this, the (SUB) model is extended by the constraint

$$\sum_{k \in D} d_k (\alpha_k^k - \alpha_r^k) = 1,$$

which fixes the right-hand side of (3.61) to one. Minimizing the objective function (3.58) leads to a minimization of the nominator in equation (3.70) and thus maximizes the violation. The subproblem (SUBn) is bounded and its solution (if negative) always corresponds to a most violated Benders' cut according to (3.70). Again, the master solution $\boldsymbol{x}'$ is infeasible if and only if the solution of (SUBn) is strictly less than zero. The primal of SUBn, denoted by PSUBn, is related to the *maximum concurrent flow model* (see, e.g., [BR02]), and it has been used by Avella et al. [AMS07] for separation of metric inequalities.

$$\text{(PSUBn)} : \quad \max \Theta \tag{3.71}$$

s.t.

$$\sum_{(i,j)\in\delta^+(i)} \sum_{m\in M_{(i,j)}} f^k_{(i,j),m} - \sum_{(j,i)\in\delta^-(i)} \sum_{m\in M_{(j,i)}} f^k_{(j,i),m} \leq \begin{cases} -d_k - \Theta, & i = k \\ d_k + \Theta, & i = r \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in V, \forall k \in K \tag{3.72}$$

$$\sum_{k\in K} f^k_{a,m} \leq u_{a,m} x'_{a,m} \quad \forall a \in A,\ \forall m \in M_a \tag{3.73}$$

$$f^k_{a,m} \leq d_k x'_{a,m} \quad \forall a \in A,\ \forall m \in M_a,\ \forall k \in K \tag{3.74}$$

$$f^k_{a,m} \geq 0 \quad \forall a \in A,\ \forall m \in M_a,\ \forall k \in K \tag{3.75}$$

$$\Theta \leq 0 \tag{3.76}$$

3.6.4 (SUBf) and (PSUBf) Models

The (PSUBn) model uses the slack variable Θ to reduce the transported commodities. Alternatively, one can use the slack variable for all flow conservation constraints:

$$\text{(PSUBf)} : \quad \max \Theta \tag{3.77}$$

s.t.

$$\sum_{(i,j)\in\delta^+(i)} \sum_{m\in M_{(i,j)}} f^k_{(i,j),m} - \sum_{(j,i)\in\delta^-(i)} \sum_{m\in M_{(j,i)}} f^k_{(j,i),m} + \Theta \leq \begin{cases} -d_k, & i = k \\ d_k, & i = r \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in V, \forall k \in K \tag{3.78}$$

$$\sum_{k\in K} f^k_{a,m} \leq u_{a,m} x'_{a,m} \quad \forall a \in A,\ \forall m \in M_a \tag{3.79}$$

$$f^k_{a,m} \leq d_k x'_{a,m} \quad \forall a \in A,\ \forall m \in M_a,\ \forall k \in K \tag{3.80}$$

$$f^k_{a,m} \geq 0 \quad \forall a \in A,\ \forall m \in M_a,\ \forall k \in K \tag{3.81}$$

$$\Theta \leq 0 \tag{3.82}$$

This problem has a nice flow structure that can easily be recognized by an LP solver (like Cplex), therefore it is considered as another alternative normalization approach for finding a violated Benders' inequality. In the corresponding dual variant of the model, denoted by (SUBf), we extend (SUB) with $\sum_{k\in D} \sum_{i\in V} \alpha^k_i = 1$.

3.6.5 (SUBcap) and (PSUBcap) Models

The fourth normalization variant is complementary to the idea of (PSUBf). Instead of using the slack Θ to allow for a violation of the flow conservation, we use it to allow for violation of capacity and coupling constraints.

$$\text{(PSUBcap)} : \quad \max \Theta \tag{3.83}$$

s.t.

$$\sum_{(i,j)\in\delta^+(i)} \sum_{m\in M_{(i,j)}} f^k_{(i,j),m} - \sum_{(j,i)\in\delta^-(i)} \sum_{m\in M_{(j,i)}} f^k_{(j,i),m} \leq \begin{cases} -d_k, & i = k \\ d_k, & i = r \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in V, \forall k \in K \tag{3.84}$$

$$\sum_{k\in K} f^k_{a,m} + \Theta \leq u_{a,m} x'_{a,m} \quad \forall a \in A,\ \forall m \in M_a \tag{3.85}$$

$$f^k_{a,m} + \Theta \leq d_k x'_{a,m} \quad \forall a \in A,\ \forall m \in M_a,\ \forall k \in K \tag{3.86}$$

$$f^k_{a,m} \geq 0 \quad \forall a \in A,\ \forall m \in M_a,\ \forall k \in K \tag{3.87}$$

$$\Theta \leq 0 \tag{3.88}$$

This is a generalization of the *capacity reduction problem*, used by Avella et al. [AMS07] to generate the so-called *strong metric inequalities* for the multi-commodity flow model for the network loading problem.

3.6.6 Summary

Table 3.1 summarizes the nine subproblems.

Dual	Primal	Explanation
(SUB)	-	see Section 3.5.3
(SUBc)	(PSUBc)	(SUB) extended by $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})^T \mathbf{1} = 1$
(SUBn)	(PSUBn)	(SUB) extended by $\sum_{k\in D} d_k(\alpha^k_k - \alpha^k_r) = 1$
(SUBf)	(PSUBf)	(SUB) extended by $\boldsymbol{\alpha}^T \mathbf{1} = 1$
(SUBcap)	(PSUBcap)	(SUB) extended by $(\boldsymbol{\beta}, \boldsymbol{\gamma})^T \mathbf{1} = 1$

Table 3.1: Different normalization approaches for separating Benders' cuts.

3.7 Valid Inequalities - Modeling Variations

This section presents several classes of valid inequalities and separation strategies that can help speed up the branch-and-bound process.

3.7.1 Connectivity Cuts

Recall the cut-set model from Section 3.3.2. The founding idea is that every set of nodes $S \subset V, r \notin S$ must have enough incoming capacity. It can also be stated that, in order to have a connected solution, at least one module must lead into any set that contains a customer. This is formally stated with *connectivity cuts*:

$$\sum_{a \in \delta^-(S)} \sum_{m \in M_a} x_{a,m} \geq 1 \quad \forall S \subseteq V \setminus \{r\}, S \cap K \neq \varnothing, \tag{3.89}$$

Every set of nodes S with some demand, not containing r must have at least one incoming module. The separation problem of connectivity cuts is similar to that of the cut-set inequalities: For a given fractional solution x' and a chosen $k \in K$, we define the capacity associated to each arc $a \in A$ as $\sum_{m \in M_a} x'_{a,m}$. If the minimum cut between r and k in G is less than 1 it defines a violated inequality. This test has to be applied for every $k \in K$.

3.7.2 Forward and Backward Cut-Sets

To find a minimum cut in a graph between a source node s and a sink node t, we utilize the *min-cut max-flow* theorem that states the equivalence of finding a maximum flow and a minimum cut in a graph. An implementation of the *push-relabel* algorithm from Cherkassky and Goldberg [CG97] is used to compute a maximum flow. This leads to a partition of nodes $\{U, \bar{U}\}, s \in U, t \in \bar{U}$ that defines a minimum cut. Following an idea from [CGR92], one can reverse the flow and produce a *backward* cut from t to s from the partition $\{U_b, \bar{U_b}\}, t \in U_b, s \in \bar{U_b}$. In general the partitions are not equal, i.e., $U_b \neq \bar{U}$, hence by solving one flow problem, two minimum cuts can be computed. This is applied for the separation of cut-set inequalities 3.10 and connectivity cuts 3.89 to speed up the cutting plane procedure.

3.7.3 Nested Cut-Sets

Usually, a min-cut problem has multiple solutions. The forward and backward cut method described above, is one way to generate two cuts per one calculation of a maximum flow. The idea of separating *nested* cut-sets is to use multiple subsequent max-flow calculations to find a series of inequalities without solving a new LP relaxation. See [KM98]. It works as follows: Solve the min-cut problem and generate the first inequality. Increase the capacities in the min-cut problem to a *large* value. For connectivity cuts a capacity of 1 is sufficiently large. For cut-set inequalities $\sum_{k \in K} d_k$ is needed. Now resolve the min-cut problem. Check whether the second inequality derived from the new minimum cut is violated by the current fractional solution. If this is the case, it is obvious that the first and the second inequality are orthogonal to each other, i.e., at most one of the two cuts has a positive coefficient for any of the variables. By repeating this procedure, a large number of cut-sets can be produced without resolving the LP relaxation.

3.7.4 Minimum Cardinality Connectivity Cuts

Usually one prefers *sparse* inequalities, i.e., inequalities with a small number of non-zero coefficients. The *creep flow* (see [KM98]) or *minimum cardinality* (see [LWP$^+$06]) cut separation strategy, is used to produce sparse connectivity cuts 3.89. It works by adding an ϵ to all capacities prior to solving the maximum flow problem. Consider for example the relaxation of the (CUT) model: (3.9), s.t. (3.11),(3.12). The initial LP solution, before any cut-set inequalities have been added is $\boldsymbol{x} = \mathbf{0}$. Accordingly, the capacities of the max-flow/min-cut problem are all 0, hence every cut is minimal. However, when an ϵ value is added to all capacities, only a cut with the minimum cardinality of arcs is a minimum cut.

3.7.5 Disjoint Benders' Cuts

Similar to the concept of nested cut-sets, a similar separation procedure for Benders' cuts is developed. It produces several disjoint Benders' cuts from the same solution of the master problem. For the Benders' decomposition of the (DMCF) model it works as follows: Assume that $(\boldsymbol{\alpha}', \boldsymbol{\beta}', \boldsymbol{\gamma}')$ corresponds to an unbounded direction in the current Benders' subproblem DMCF($\boldsymbol{x}$) as defined by (3.58)-(3.60). Now, fix to zero the components of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ with a positive value in this solution, i.e., add inequalities $\beta^k_{a,m} = 0 \forall \beta^{k'}_{a,m} > 0$ and $\gamma_{a,m} = 0 \forall \gamma'_{a,m} > 0$ to the Benders' subproblem. Next, solve this modified subproblem with the fixed variables. If the modified subproblem is unbounded one can retrieve a second Benders' cut that is orthogonal to the first. By repeating this procedure, one may generate a set of disjoint Benders' cuts.

This separation procedure is similar to the one used to separate disjoint (*nested*) cut-sets, mentioned above. But a key difference is that the separation of nested cut-set inequalities or nested connectivity cuts is computationally relatively cheap. It basically requires solving one max-flow problem per cut. The separation of disjoint Benders' cuts however requires the solution of a large LP. In addition, one must be aware that it is unclear whether the nested cut-sets as well as the disjoint Benders' cuts are necessary to obtain the LP relaxation of the model at hand. There is a trade off between finding multiple nested cuts for *one* fractional solution $\boldsymbol{x}$ or resolving the master problem to compute a new fractional solution. In preliminary tests it turned out that solving a single Benders' subproblem is computationally more expensive than resolving the master LP. Therefore, this procedure was turned off for producing the results reported in Section 3.10.

3.7.6 Magnanti-Wong Implementation

The ideas of Magnanti and Wong [MW81] have been widely used for accelerating the separation of Benders' cuts (see, e.g., [MMW86],[RGCS09]). The authors proposed to accelerate the convergence of the basic Benders' algorithm by adding Pareto-optimal Benders' cuts. The application of their proposal to the DMCF Benders' decomposition is described below.

For a given primal master solution $\boldsymbol{x}$ and a dual subproblem solution $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$, denote the objective function (3.58) of the subproblem by $z(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{x})$. A cut $z(\boldsymbol{\alpha}'', \boldsymbol{\beta}'', \boldsymbol{\gamma}'', \boldsymbol{x}) \geq 0$ *dominates* another cut $z(\boldsymbol{\alpha}', \boldsymbol{\beta}', \boldsymbol{\gamma}', \boldsymbol{x}) \geq 0$ if and only if $z(\boldsymbol{\alpha}', \boldsymbol{\beta}', \boldsymbol{\gamma}', \boldsymbol{x}) \geq z(\boldsymbol{\alpha}'', \boldsymbol{\beta}'', \boldsymbol{\gamma}'', \boldsymbol{x})$ for all $\boldsymbol{x} \in \{0,1\}^{|A||M|}$ satisfying (3.5), and the strict inequality holds for at least one $\boldsymbol{x}$. A Benders' cut is said to be *Pareto-optimal* if no other cut dominates it. In case that there are multiple optimal solutions to the Benders' subproblem, Magnanti and Wong have proposed an approach to search for a Pareto-optimal cut by solving an additional subproblem in the separation phase:

1. Given a fractional solution $\boldsymbol{x}'$, solve the Benders' subproblem DMCF$(\boldsymbol{x}')$, given in (3.58)-(3.60), to get a violated cut defined by $(\boldsymbol{\alpha}', \boldsymbol{\beta}', \boldsymbol{\gamma}')$. If $z(\boldsymbol{\alpha}', \boldsymbol{\beta}', \boldsymbol{\gamma}', \boldsymbol{x}') = 0$, no violated cut exists. Stop.

2. Set $z' := z(\boldsymbol{\alpha}', \boldsymbol{\beta}', \boldsymbol{\gamma}', \boldsymbol{x}')$.

3. Solve the new subproblem defined as:

$$\min\{z(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{x}_0) \mid (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \in \text{DMCF}(\boldsymbol{x}') \text{ and } z(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{x}') = z'\}.$$

4. Denote the solution to this subproblem by $(\boldsymbol{\alpha}'', \boldsymbol{\beta}'', \boldsymbol{\gamma}'')$. Then, the cut $z(\boldsymbol{\alpha}'', \boldsymbol{\beta}'', \boldsymbol{\gamma}'', \boldsymbol{x}) \geq 0$ is inserted into the master problem.

Of course DMCF$(\boldsymbol{x}')$ can be replaced by any of the normalization variants described in Section 3.6. The vector x_0 specifies a *core point*, i.e., a point that belongs to the relative interior of the convex hull of all binary vectors $\boldsymbol{x}$ satisfying (3.5). As already observed by Papadakos [Pap08], for the above procedure to work efficiently, one needs to start it with a different core point every time the procedure is applied. For that purpose, we start with a randomly chosen point from the interior, and later we generate a random convex combination of two incumbent solutions.

The obvious drawback of this procedure is that one has to solve two time-consuming subproblems with each separation. Furthermore, the Magnanti-Wong subproblem is computationally more expensive than solving the master problem. In the default implementation, the separation of Pareto-optimal cuts is turned off. In Section 3.10.6, there is a report on the effects obtained by applying this procedure.

3.7.7 Degree-Balance Constraints

Non-customer nodes $V \setminus (K \cup \{r\})$ cannot have incoming (or outgoing) arcs only. Therefore, we can add the following *degree-balance constraints* that only work for single source case:

$$\sum_{(l,i)\in A, l\neq j} \sum_{m\in M_{li}} x_{li,m} \geq \sum_{m\in M_{ij}} x_{ij,m} \qquad \forall (i,j) \in A, i \notin K, i \neq r \tag{3.90}$$

$$\sum_{(j,l)\in A, l\neq i} \sum_{m\in M_{jl}} x_{jl,m} \geq \sum_{m\in M_{ij}} x_{ij,m} \qquad \forall (i,j) \in A, j \notin K, j \neq r. \tag{3.91}$$

Inequality (3.90) states that if an arc (i,j) emanating from a non-customer node i is being used in the solution, there must be at least one arc entering i. Thanks to Theorem 3.2.1 the opposite arc (j,i) can be excluded from the summation on the left hand side. Inequality (3.91) states the opposite case for an arc (i,j) entering a non-customer node j.

3.7.8 Cover Inequalities

Given a cut-set inequality (3.10) defined by $S \subset V, r \in S$, define the index set $I(S) := \{(i,j,m) \mid (i,j) \in \delta^+(S), m \in M_{ij}\}$ and the demand outside of S as $B := \sum_{k\in K\setminus S} d_k$. Set $J \subset I(S)$ is called a *cover* with respect to $I(S)$ if $\sum_{(i,j,m)\in J} u_{ij,m} < B$ and a *maximal cover* if, in addition, for all J', such that $I(S) \supseteq J' \supset J$: $\sum_{(i,j,m)\in J'} u_{ij,m} \geq B$. If J is a maximal cover with respect to $I(S)$, then the following *cover inequalities* are valid:

$$\sum_{(i,j,m)\in I(S)\setminus J} x_{ij,m} \geq 1. \tag{3.92}$$

In general, the separation problem of cover inequalities is NP-hard. We show that the problem of finding the most violated cover inequality (3.92) is equivalent to solving the *precedence constrained knapsack problem.* Assume that indices $m \in M_{ij}$ are sorted according to increasing arc capacities. To model any cover J with respect to $I(S)$, define the binary variables $z_{ij,m}$ that are equal to one if and only if $(i,j,m) \in J$. For every arc $(i,j) \in \delta^+(S)$, we define $u_{ij,0} = 0$.

For a given fractional solution $\boldsymbol{x}'$ and an index set $I(S)$ induced by a cut-set inequality, the *most violated cover inequality* can be found by solving the following model:

$$\text{(KNAP)} : \qquad \max \sum_{(i,j,m)\in I(S)} x'_{ij,m} z_{ij,m}$$

s.t.

$$\sum_{(i,j,m)\in I(S)} (u_{ij,m} - u_{ij,m-1}) z_{ij,m} < B \tag{3.93}$$

$$z_{ij,m} \geq z_{ij,m+1} \qquad \forall (i,j,m) \in I(S),\ m < |M_{ij}|$$
$$z_{ij,m} \in \{0,1\} \qquad \forall (i,j,m) \in I(S)$$

Let $\boldsymbol{z}'$ be an optimal solution of model (KNAP). The corresponding cover inequality reads then as follows:

$$\sum_{(i,j,m)\in I(S)} (1 - z'_{ij,m}) x_{ij,m} \geq 1.$$

If all capacities and demands are integers, the inequality (3.93) can be replaced by $\sum_{(i,j,m)\in I(S)} (u_{ij,m} - u_{ij,m-1}) z_{ij,m} \leq B-1$. The cover inequalities are similar to the *band inequalities* for the incremental cost model in [DS98].

3.7.9 Incremental Cost versus Explicit Cost

Throughout this work the *explicit cost* (also called the *multiple choice*) model is used to describe the stepcost function of the network design. It is based on the idea of using a set of binary variables per edge, out of which at most one may be nonzero. Formally this is expressed by the definition of modules and inequalities (3.4),(3.5) and (3.6).

The *incremental cost model* was introduced by Dahl and Stoer [DS98] for the general multi-source multi-sink network design problem. It has been used to model LAN in [RS06, SRH08]. It is based on the idea of using a list of binary variables per edge that have nonincreasing values. The problem is defined by means of the incremental capacity and cost values and a feasible solution has to facilitate the increments in increasing order.

In [RS06] it is proved that for LAN both models are equivalent in terms of quality of lower bounds and their LP relaxations both approximate the monotonically increasing step cost function by its lower convex envelope. This is an application of the more general result from [CGM03], where the equivalence is shown for general minimization problems with separable non-convex piecewise linear costs (see also [KFJN04]).

3.8 Model Hierarchy

This section compares the strength of the different LP models for the LAN problem presented in this chapter. The feature of interest is the size of the polyhedron in the space of the design variable $\boldsymbol{x}$. Consider two polyhedra $\mathcal{P}_1$ and $\mathcal{P}_2$ in the space of $\boldsymbol{x}$ variables. If $\mathcal{P}_1$ is a subset $\mathcal{P}_1 \subseteq \mathcal{P}_2$ for all instances of LAN and there exist some instances for which the inclusion is strict, $\mathcal{P}_1 \subset \mathcal{P}_2$, then we call $\mathcal{P}_1$ the *stronger* formulation. In other words, there exist fractional solutions that are valid for the weaker model but not for the stronger model. The comparison includes:

(SCF) The single-commodity flow formulation described in Section 3.3.1. Denote the polyhedron of feasible points by
$\mathcal{P}_{\mathrm{SCF}} := \left\{ (\boldsymbol{x}, \boldsymbol{f}) \in [0,1]^{|A||M|} \times \mathbb{R}_{\geq 0}^{|A|} \middle| (\boldsymbol{x}, \boldsymbol{f}) \text{ satisfy } (3.3) - (3.7) \right\}.$

(CUT) The cut-set formulation from Section 3.3.2. Denote its polyhedron by
$\mathcal{P}_{\mathrm{CUT}} := \left\{ \boldsymbol{x} \in [0,1]^{|A||M|} \middle| \boldsymbol{x} \text{ satisfy } (3.10) - (3.12) \right\}.$

(CUT⁺) This denotes the (CUT) model extended by connectivity cuts (3.89). The corresponding polyhedron is defined as:
$\mathcal{P}_{\mathrm{CUT}} := \left\{ \boldsymbol{x} \in [0,1]^{|A||M|} \middle| \boldsymbol{x} \text{ satisfy } (3.10) - (3.12), (3.89) \right\}.$

(MCF) The multi-commodity flow formulation from Section 3.4.1. The polyhedron is defined as
$\mathcal{P}_{\mathrm{MCF}} := \left\{ (\boldsymbol{x}, \boldsymbol{f}) \in [0,1]^{|A||M|} \times \mathbb{R}_{\geq 0}^{|A||K|} \middle| (\boldsymbol{x}, \boldsymbol{f}) \text{ satisfy } (3.14) - (3.19) \right\}.$

(MCF⁻) This denotes the multi-commodity flow formulation without coupling constraints (3.16). Its polyhedron is defined as
$$\mathcal{P}_{\mathrm{MCF}^-} := \Big\{ (\boldsymbol{x}, \boldsymbol{f}) \in [0,1]^{|A||M|} \times \mathbb{R}_{\geq 0}^{|A||K|} \Big| \; (\boldsymbol{x}, \boldsymbol{f}) \text{ satisfy } (3.14), (3.15), (3.17), (3.18), (3.19) \Big\}.$$

(DMCF) The disaggregated multi-commodity flow formulation from Section 3.4.2. The polyhedron for (DMCF) is defined as
$\mathcal{P}_{\mathrm{DMCF}} := \left\{ (\boldsymbol{x}, \boldsymbol{f}) \in [0,1]^{|A||M|} \times \mathbb{R}_{\geq 0}^{|A||K||M|} \middle| (\boldsymbol{x}, \boldsymbol{f}) \text{ satisfy } (3.21) - (3.26) \right\}.$

(DMCF⁻) This denotes the (DMCF) model without coupling constraints (3.23).
$$\mathcal{P}_{\mathrm{DMCF}^-} := \Big\{ (\boldsymbol{x}, \boldsymbol{f}) \in [0,1]^{|A||M|} \times \mathbb{R}_{\geq 0}^{|A||K||M|} \Big| \; (\boldsymbol{x}, \boldsymbol{f}) \text{ satisfy } (3.21), (3.22), (3.24), (3.25), (3.26) \Big\}.$$

(DSCF) The disaggregated single-commodity flow formulation from Section 3.4.3. Its polyhedron is defined by
$\mathcal{P}_{\mathrm{DSCF}} := \left\{ (\boldsymbol{x}, \boldsymbol{f}) \in [0,1]^{|A||M|} \times \mathbb{R}_{\geq 0}^{|A||M|} \middle| (\boldsymbol{x}, \boldsymbol{f}) \text{ satisfy } (3.27) - (3.32) \right\}.$

(Benders) This denotes the (MASTER) problem (3.34)-(3.35) extended by the (DMCF) Benders' cuts (3.61). The corresponding polyhedron is $\mathcal{P}_{\text{Benders}} := \{\boldsymbol{x} \in [0,1]^{|A||M|} | \boldsymbol{x} \text{ satisfy } (3.34) - (3.35), (3.61)\}$.

(Benders$^+$) denotes (Benders) with rounded (DMCF) Benders' cuts (3.62), degree balance (3.90),(3.91) and cover inequalities (3.92).

$$\mathcal{P}_{\text{Benders}^+} := \Big\{\boldsymbol{x} \in [0,1]^{|A||M|} \Big| \; \boldsymbol{x} \text{ satisfy } (3.34) - (3.35), (3.61), (3.90), (3.91), (3.92)\Big\}.$$

For the matter of this polyhedral comparison we will only write $\boldsymbol{x}$ and $\boldsymbol{f}$ to denote the vectors of decision variables without their dimensions in a given model, as long as it is clear from the context. Furthermore we will refer to a variable $\boldsymbol{f}$ of appropriate dimension as a *flow* if it satisfies the models *flow conservation constraints.* For any given $(\boldsymbol{x}, \boldsymbol{f}) \in \mathcal{P}$, we call $\boldsymbol{f}$ a *feasible flow* (with respect to $\boldsymbol{x}$ and $\mathcal{P}$). In order to compare the polyhedra, we use the natural projection of the flow models onto the space of $\boldsymbol{x}$ variables. For the flow model with polyhedron $\mathcal{P}$, define: $\text{proj}_x(\mathcal{P}) := \{x \in [0,1]^{|A||M|} \mid \exists f, (x,f) \in \mathcal{P}\}$. The optimal objective value of the linear relaxation of a model is denoted by z.

The hierarchical scheme given in Figure 3.1 summarizes the relationships between the LP relaxations of the MIP models considered in this chapter. A filled arrow specifies that the target formulation is strictly stronger than the tail formulation. An empty arrow specifies that the target formulation is at least as strong as the tail formulation.

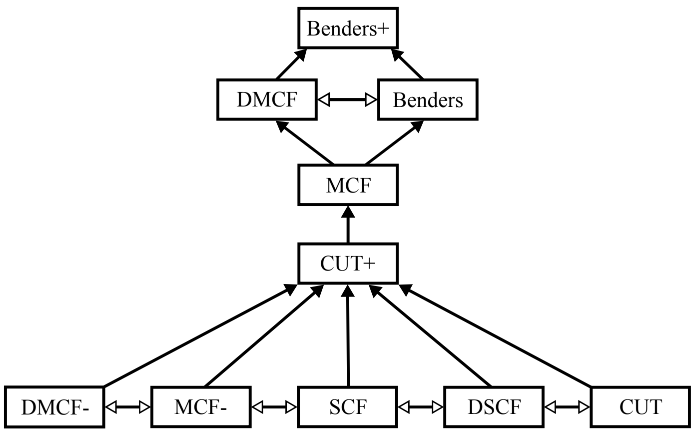


Figure 3.1: Hierarchy of LP relaxations.

The rest of this section is devoted to the proofs for the inclusions depicted in Figure 3.1. In addition to the polyhedral inclusion we can also show that there are instances of LAN for which the stronger models actually yield a greater objective value.

Theorem 3.8.1.

$$\text{proj}_x(\mathcal{P}_{\text{DSCF}}) = \text{proj}_x(\mathcal{P}_{\text{SCF}})$$

Proof. A feasible (DSCF) flow $\boldsymbol{f}$ can be aggregated to a feasible (SCF) flow $\widehat{f_a} := \sum_{m \in M_a} f_{a,m} \forall a \in A$, therefore $\text{proj}_x(\mathcal{P}_{\text{DSCF}}) \subseteq \text{proj}_x(\mathcal{P}_{\text{SCF}})$.

In the other direction, given a feasible (SCF) flow $\boldsymbol{f}$ we can define a feasible (DSCF) flow $\widehat{\boldsymbol{f}}$ like this: For some fixed $a \in A$, iterate over all modules $m \in M_a$ and raise $\widehat{f_{a,m}}$ until $\sum_{m \in M_a} \widehat{f_{a,m}} = f_a$. This decomposition can be applied independently for every arc and finally yields a flow $\widehat{\boldsymbol{f}}$ that satisfies flow conservation (3.27) and capacity constraints (3.28) for the (DSCF) model.

Furthermore, it is safe to assume that there are no excess capacities, i.e., $u_{a,m} \leq \sum_{k \in K} d_k \forall a \in A, m \in M_a$. (See preprocessing Step (x) in Section 3.1.) As a consequence, the (DSCF) capacity constraints (3.28) imply the (DSCF) coupling constraints (3.29). Thus $\widehat{\boldsymbol{f}}$ is a feasible (DSCF) flow and $\text{proj}_x(\mathcal{P}_{\text{SCF}}) \subseteq \text{proj}_x(\mathcal{P}_{\text{DSCF}})$ under the no-excess assumption. □

Theorem 3.8.2.

$$\text{proj}_x(\mathcal{P}_{\text{DMCF}^-}) = \text{proj}_x(\mathcal{P}_{\text{MCF}^-})$$

Proof. Aggregating a feasible (DMCF$^-$) flow $\boldsymbol{f}$ by modules yields a feasible (MCF$^-$) flow: $\widehat{f_a^k} := \sum_{m \in M_a} f_{a,m}^k \forall a \in A$. It follows that $\text{proj}_x(\mathcal{P}_{\text{DMCF}^-}) \subseteq \text{proj}_x(\mathcal{P}_{\text{MCF}^-})$.

Reversely, any feasible (MCF$^-$) flow $\boldsymbol{f}$ can be decomposed into a feasible (DMCF$^-$) flow $\widehat{\boldsymbol{f}}$ as follows. Choose a $k \in K$. For this single, chosen commodity the same disaggregation per modules as in the proof of Theorem 3.8.1 above, for the (DSCF) model can be applied. It yields a one-commodity flow $\widehat{\boldsymbol{f}^k} \in \mathbb{R}_{\geq 0}^{|A||M|}$. Subtracting this flow $\widehat{\boldsymbol{f}^k}$ from $\boldsymbol{f}$ yields a feasible (DMCF$^-$) solution $\left(\boldsymbol{x}, \boldsymbol{f} - \widehat{\boldsymbol{f}^k}\right)$ for the reduced LAN problem with $K' := K \setminus \{k\}$ with reduced capacities $\boldsymbol{u}' := \boldsymbol{u} - \widehat{\boldsymbol{f}^k}$. Repeated application of this argument for all customers K finally yields a (DMCF$^-$) feasible flow, thus $\text{proj}_x(\mathcal{P}_{\text{MCF}^-}) \subseteq \text{proj}_x(\mathcal{P}_{\text{DMCF}^-})$. □

Theorem 3.8.3.

$$\text{proj}_x(\mathcal{P}_{\text{MCF}^-}) = \text{proj}_x(\mathcal{P}_{\text{SCF}})$$

Proof. Aggregating a feasible (MCF$^-$) flow $\boldsymbol{f}$ by commodities yields a feasible (SCF) flow: $\widehat{f_a} := \sum_{k \in K} f_a^k \forall a \in A$, thus $\text{proj}_x(\mathcal{P}_{\text{MCF}^-}) \subseteq \text{proj}_x(\mathcal{P}_{\text{SCF}})$.

Reversely, any feasible (SCF) flow $\boldsymbol{f}$ can be decomposed into a feasible (MCF$^-$) flow $\widehat{\boldsymbol{f}}$ as follows. Initialize $\widehat{\boldsymbol{f}} := \boldsymbol{0}$. There exists a path $P \subseteq A$ in $\boldsymbol{f}$ from r to some $k \in K$ that allows the transport of a positive amount of flow. Denote the transportable flow by $\Delta = \min\{d_k, \min_{a \in P}\{f_a\}\}$. Increase the (MCF) flow by Δ to $\widehat{f_a^k} := f_a^k + \Delta \forall a \in P$. Reduce the (SCF) flow by Δ to

$f_a := f_a - \Delta \forall a \in P$. This reduced flow is feasible for the LAN problem with reduced demands $d_k := d_k - \Delta$. This reduction can be repeated for a finite number of iterations, since after each reduction either the flow on an arc or a demand becomes zero. Repeated application until $\boldsymbol{f} = \mathbf{0}$ yields a feasible (MCF$^-$) flow $\widehat{\boldsymbol{f}} \in \mathbb{R}^{|A||K|}_{\geq 0}$ that is a decomposition of the (SCF) flow $f_a = \sum_{k \in K} \widehat{f^k_a} \forall a \in A$. Thus $\text{proj}_x(\mathcal{P}_{\text{SCF}}) \subseteq \text{proj}_x(\mathcal{P}_{\text{MCF}^-})$. □

Theorem 3.8.4.

$$\text{proj}_x(\mathcal{P}_{\text{SCF}}) = \mathcal{P}_{\text{CUT}}$$

Proof. It follows directly from the min-cut max-flow theorem (see A.4), that for any $\boldsymbol{x} \in \mathcal{P}_{\text{CUT}}$ there exists a feasible single-commodity flow, therefore $\mathcal{P}_{\text{CUT}} \subseteq \text{proj}_x(\mathcal{P}_{\text{SCF}})$.

On the other hand, given a $(\boldsymbol{x}, \boldsymbol{f}) \in \mathcal{P}_{\text{SCF}}$. If we assume that $\boldsymbol{x} \notin \mathcal{P}_{\text{CUT}}$, the theorem implies that there does not exist any feasible flow $\boldsymbol{f}$. Thus the assumption must be wrong and therefore $\text{proj}_x(\mathcal{P}_{\text{SCF}}) \subseteq \mathcal{P}_{\text{CUT}}$.

□

Theorem 3.8.5. *There exist instances for which*

$$\mathcal{P}_{\text{CUT}^+} \subset \mathcal{P}_{\text{CUT}}.$$

There exist instances for which also

$$z_{\text{CUT}+} > z_{\text{CUT}}.$$

Proof. The (CUT$^+$) model is defined as $\mathcal{P}_{\text{CUT}^+} \subseteq \mathcal{P}_{\text{CUT}}$ and it has been shown above that $\mathcal{P}_{\text{CUT}} = \text{proj}_x(\mathcal{P}_{\text{SCF}})$. We can give an example that shows that $\mathcal{P}_{\text{CUT}^+} \neq \text{proj}_x(\mathcal{P}_{\text{SCF}})$. In addition, the value of the linear relaxation of the (CUT$^+$) model is greater than the LP value of the (SCF) model for this example. The (SCF) solution depicted in Figure 3.2(b) violates the connectivity cut: $x_{(\text{r},\text{v}),1} + x_{(\text{p},\text{v}),1} + x_{(\text{w},\text{v}),1} + x_{(\text{w},\text{v}),2} \geq 1$. The (CUT$^+$) solution in Figure 3.2(c) satisfies all connectivity cuts and has an objective value $z_{\text{CUT}^+} = 9.5 > 9 = z_{\text{SCF}}$. □

Theorem 3.8.6. *There exist instances for which*

$$\text{proj}_x(\mathcal{P}_{\text{MCF}}) \subset \mathcal{P}_{\text{CUT}^+}.$$

There exist instances for which also

$$z_{\text{MCF}} > z_{\text{CUT}^+}.$$

Proof. The cut-set inequalities (3.10) ensure the existence of a flow that satisfies the capacity constraints (3.15) of the (MCF) model. The connectivity cuts (3.89) ensure the existence of a set of flows of one unit from r to every k that satisfies the coupling constraints (3.16) of the (MCF) model. However, the existence of a flow that satisfies both classes of constraints is not guaranteed by (CUT$^+$) but only by the (MCF) model.

Figures 3.2(c) and (b) show the (CUT$^+$) and (MCF) LP solutions with $z_{\mathrm{MCF}} = 10.08\dot{3} > 9.5 = z_{\mathrm{CUT}^+}$. □

Theorem 3.8.7. *There exist instances for which*

$$\mathrm{proj}_x(\mathcal{P}_{\mathrm{DMCF}}) \subset \mathrm{proj}_x(\mathcal{P}_{\mathrm{MCF}}).$$

There exist instances for which also

$$z_{\mathrm{DMCF}} > z_{\mathrm{MCF}}.$$

Proof. Aggregating a feasible (DMCF) flow $\boldsymbol{f}$ by modules yields a feasible (MCF) flow: $\widehat{f_a^k} := \sum_{m \in M_a} f_{a,m}^k \forall a \in A$, therefore $\mathrm{proj}_x(\mathcal{P}_{\mathrm{DMCF}}) \subseteq \mathrm{proj}_x(\mathcal{P}_{\mathrm{MCF}})$.

Figures 3.2(d) and (e) show the (MCF) and (DMCF) LP solutions with $z_{\mathrm{DMCF}} = 10.\dot{3} > 10.08\dot{3} = z_{\mathrm{MCF}}$. Therefore, the polyhedral inclusion is strict $\mathrm{proj}_x(\mathcal{P}_{\mathrm{DMCF}}) \neq \mathrm{proj}_x(\mathcal{P}_{\mathrm{MCF}})$. □

Theorem 3.8.8.

$$\mathrm{proj}_x(\mathcal{P}_{\mathrm{DMCF}}) = \mathcal{P}_{\mathrm{Benders}}$$

Proof. The equality follows directly from the definition of Benders' decomposition. See Sections 3.5.3 and A.2. □

Theorem 3.8.9. *There exist instances for which*

$$\mathcal{P}_{\mathrm{Benders}^+} \subset \mathcal{P}_{\mathrm{Benders}}.$$

There exist instances for which also

$$z_{\mathrm{Benders}^+} > z_{\mathrm{Benders}}.$$

Proof. The (Benders$^+$) model is defined as $\mathcal{P}_{\mathrm{Benders}^+} \subseteq \mathcal{P}_{\mathrm{Benders}}$. Furthermore, the (DMCF) solution of the example in Figure 3.2(e) violates cover inequalities (3.92), e.g., for the node set $S = \{\mathrm{r}\}$ and the index set $J = \{(\mathrm{r,p},1)\}$:

$$\sum_{(i,j,m) \in I(S) \setminus J} x_{ij,m} = x_{\mathrm{r,v},1} \geq 1.$$

The (Benders$^+$) solution in Figure (f) satisfies all cover inequalities and has a greater objective value than the (DMCF) solution: $z_{\text{Benders}^+} = 12 > 10.\dot{3} = z_{\text{DMCF}}$.

□

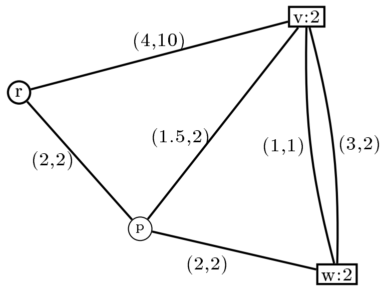
3.9 Primal Heuristic

This section describes a simple rounding heuristic. It takes a fractional solution $\boldsymbol{x}$ as input and creates an integer solution $\boldsymbol{x}'$. If $\boldsymbol{x}$ is (SCF) feasible, then $\boldsymbol{x}'$ is guaranteed to be integer feasible for the (LAN) problem. Subsequently, the heuristic creates a cheaper solution $\boldsymbol{x}''$ by means of a min-cost-flow algorithm.

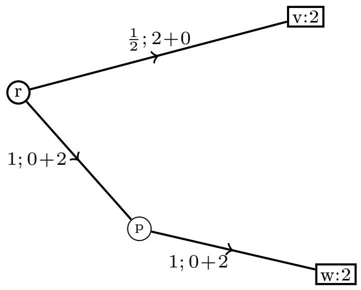
Denote the total installed capacity on an arc by $X_{ij}(\boldsymbol{x}) = \sum_{m \in M_{ij}} u_{ij,m} x_{ij,m}$. The most appropriate module to support a certain capacity $U > 0$ is again denoted by $\mu_e(U)$ as defined in equation (3.1). Initialize $\boldsymbol{x}' := \mathbf{0}$. Now for every arc (i,j) with positive capacity $U := X_{ij}(\boldsymbol{x}) > 0$, install the most appropriate module, i.e., $x'_{ij,\mu_{ij}(U)} := 1$. The resulting $\boldsymbol{x}'$ is binary and obviously satisfies the disjunction constraints (3.5). It does not decrease the capacity with respect to the fractional solution, i.e., $X_a(\boldsymbol{x}') \geq X_a(\boldsymbol{x}) \forall a \in A$. Consequently, $\boldsymbol{x}'$ satisfies the capacity constraints (3.4). This proves the implication: $\boldsymbol{x} \in \text{proj}_x(\mathcal{P}_{\text{SCF}}) \Rightarrow \boldsymbol{x}' \in \text{proj}_x(\mathcal{P}_{\text{SCF}})$.

Typically, $\boldsymbol{x}'$ is overly generous and can be improved. To this end we use an augmented graph with an additional sink t, similar to the one from Section 3.3.2: Let $G' = (V', A')$ where $V' = V \cup \{t\}$ and $A' = A \cup \{(k,t) \mid k \in K\}$. The arc capacities are set to $X_{ij}(\boldsymbol{x}')$ for all $(i,j) \in A$ and d_k for all (k,t), $k \in K$. Arc costs are defined as $\sum_{m \in M_a} c_{a,m} x'_{a,m} / u_{a,m}$ for $a \in A$ and 0 for $a \in A'$. Initialize $\boldsymbol{x}'' := \mathbf{0}$. We now compute the min-cost-flow $\boldsymbol{f} \in \mathbb{R}^{|A|}$ in G'. This induces the new incumbent candidate $\boldsymbol{x}'' : x''_{a,\mu(f_a)} := 1$ for arcs $a \in A$ with positive flow $f_a > 0$.

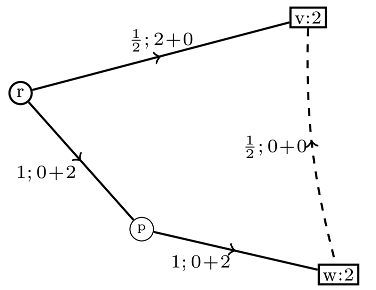
The min-cost-flow implementation based on capacity scaling and successive shortest path computation found in the commercial library LEDA, 5.2 (see [AMO93, LED]) is used. This algorithm only works for integer capacity and cost values. Therefore we round these values to the nearest integer prior to the min-cost-flow computation. A result of this rounding is that $\boldsymbol{x}''$ will, on rare occasions, be infeasible. This is easily detected by a subsequent computation of a max-flow and an infeasible $\boldsymbol{x}''$ is discarded.



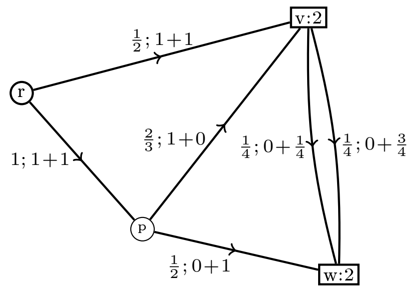
(a) Instance of the LAN design problem. The edge labels are given as (u, c). Between nodes v and w there are two available modules $m = 1 : (1, 1)$ and $m = 2 : (3, 2)$. They are shown as two parallel edges.



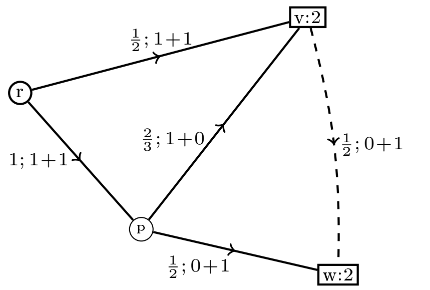
(b) LP solution of the (SCF) model. The objective value is $z_{\text{SCF}} = 1 \cdot 2 + 1/2 \cdot 10 + 1 \cdot 2 = 9$.



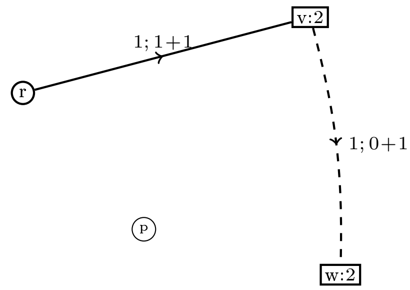
(c) LP solution of the (CUT$^+$) model. The objective value is $z_{\text{CUT}^+} = 1 \cdot 2 + 1/2 \cdot 10 + 1 \cdot 2 + 1/2 \cdot 1 = 9.5$.



(d) LP solution of the (MCF) model. The objective value is $z_{\text{MCF}} = 1 \cdot 2 + 1/2 \cdot 10 + 2/3 \cdot 2 + 1/2 \cdot 2 + 1/4 \cdot 1 + 1/4 \cdot 2 = 10.08\dot{3}$.



(e) LP solution of the (DMCF) model. The objective value is $z_{\text{DMCF}} = 1 \cdot 2 + 1/2 \cdot 10 + 2/3 \cdot 2 + 1/2 \cdot 2 + 1/2 \cdot 2 = 10.\dot{3}$.



(f) LP solution of the (Benders$^+$) model. The objective value is $z_{\text{Benders}^+} = 1 \cdot 10 + 1 \cdot 2 = 12$. This happens to be the integer optimal solution for LAN.

Figure 3.2: A LAN example that demonstrates $z_{\text{SCF}} < z_{\text{CUT}^+} < z_{\text{MCF}} < z_{\text{DMCF}} < z_{\text{Benders}^+}$. Rectangular nodes are terminals with demand. For the LP solutions (b)-(f), the arc labels are of the form $x_{a,m}; f^{\text{v}}_{a,m} + f^{\text{w}}_{a,m}$. A solid arc denotes a saturated module, i.e., the capacity constraint is satisfied with equality. A dashed arc denotes that more capacity is installed than needed for the flow.

3.10 Empirical Results for Solving LAN Instances

This section describes the algorithmic framework that was used to test the presented methods empirically and presents and discusses the obtained results. Section 3.10.1 explains the branch-and-cut algorithm. Sections 3.10.2 and 3.10.3 give some details on the two sets of benchmark instances used for the tests. The subsequent Sections 3.10.4 and 3.10.5 discuss the actual empiric results obtained. Finally, Section 3.10.6 gives a brief comment on the Magnanti-Wong method of enhancing Benders' decomposition, proposed in the literature.

3.10.1 Branch-and-Cut Algorithm

This section describes the main branch-and-cut algorithm that ties the MIP models, valid inequalities, separation algorithms and the heuristic described in the previous sections together and forms the algorithmic framework.

The algorithm starts with the LP relaxation of one of the MIP models from the polynomial hierarchy. The description here, uses the (SCF) model as basis. The overall algorithm works as follows:

1. Apply the preprocessing technique described in Section 3.1.
2. Transform the undirected LAN instance into the directed equivalent as described in Section 3.2.
3. Initialize the branch-and-bound algorithm:
    (a) Initialize the master problem with the variables and constraints of the (SCF) model.
    (b) Add *in-degree* and *out-degree* inequalities
$$\begin{aligned} \sum_{(i,k)\in\delta^-(k)} \sum_{m\in M_{ik}} x_{ik,m} &\geq 1 \qquad \forall k \in K \\ \sum_{(r,j)\in\delta^+(r)} \sum_{m\in M_{rj}} x_{rj,m} &\geq 1. \end{aligned}$$
    These inequalities are special cases of the connectivity inequalities (3.89) for singleton sets $S := \{k\} \forall k \in K$ and $S := V \setminus \{r\}$, respectively.
    (c) Add degree-balance constraints (3.90) and (3.91).
    (d) Solve the LP relaxation of the master.
4. In every n-th node of the branch-and-bound tree:

(a) As long as there are violated connectivity inequalities (3.89), add them to the master LP. Apply the techniques for forward/backward, minimum cardinality and nested cuts (see Sections 3.7.2-3.7.4). Resolve the master.

(b) If no connectivity cuts can be separated, create the (DMCF) Benders' subproblem based on the current fractional solution $\boldsymbol{x}'$. Solve the subproblem. If this results in a violated Benders' cut, add it to the master LP and resolve it.

Note that the (SCF) model is sufficient to model the LAN problem. The cuts added in Step 4 are not necessary for correctness but are used to strengthen the LP bounds. Accordingly, it is sufficient to generate cuts in every n-th node ($n > 1$) of the branch-and-bound tree. Alternatively, the computationally expensive Benders' cuts can be separated only at the root node. Furthermore, there is a time limit for each single separation of Benders' cuts of 45 seconds. In the context of multiple-source multiple-sink network design problems, these strategies are not valid. Either the master must be initialized with a much larger (multi-commodity flow) model or n must be equal to 1.

Instead of using (SCF) as the basis of the master problem, also (MCF) or (DMCF) can be used. Practically, these models proved to be too big to be solved repeatedly. Using the (CUT) model as basis does not carry this disadvantage. The master consists basically only of the disjunction constraints and the LP relaxation is easily computed. The separation Step 4b has to generate cut-set inequalities in addition. Still, this is not a very costly operation. Nevertheless, using (SCF) was superior over (CUT) in the tests. A first reason is that the commercial MIP solver Cplex, that was employed, found more helpful inequalities to strengthen the (SCF) relaxation. A second reason is that Cplex' internal heuristics found better feasible solutions with the flow model. That was noticeable, especially, late in the branch-and-bound process. Lastly, recall that the cover inequalities are derived from cut-set inequalities and are therefore not applicable in the (SCF) model. The Benders' separation procedure in Step 4b can facilitate any of the nine strategies laid out in Section 3.6.

To improve the overall performance and to avoid numerical difficulties we consider the following two standard branch-and-cut ingredients:

- Tailing Off: If the relative improvement of the lower bound is less than `Eps`% in the last `It` iterations of the separation procedure, we stop the separation and resort to branching. The general setting of (`It`,`Eps`) is $(20, 10^{-3})$. However, if only the computationally more expensive Benders' cuts were separated in recent iterations, a stricter setting of (`It'`,`Eps'`) $= (10, 10^{-3})$ is applied.

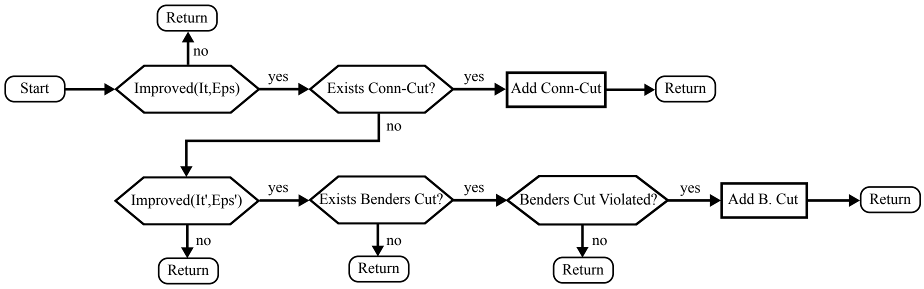


Figure 3.3: Separation of cuts in the branch-and-cut framework.

- Degree of Violation: Assume that after solving the Benders' subproblem for a given fractional value $\boldsymbol{x}'$, we obtain a violated cut defined by a vector $(\alpha', \beta', \gamma')$. Recall the definition of the function `violation` (3.70) from Section 3.6.3. Before inserting the corresponding cut into the master LP, we normalize it by dividing it with its right-hand side (which is always positive) and calculate its violation by the current fractional solution $\boldsymbol{x}'$ as follows:

$$\texttt{violation}\left(\boldsymbol{\alpha}', \boldsymbol{\beta}', \boldsymbol{\gamma}', \boldsymbol{x}'\right) := 1 - \sum_{a \in A} \sum_{m \in M_a} \frac{\sum_{k \in K} d_k \beta'^{k}_{a,n} + u_{a,n} \gamma'_{a,n}}{\sum_{k \in K} d_k (\alpha'^{k}_{k} - \alpha'^{k}_{r})} x'_{a,n} \tag{3.94}$$

  If $\texttt{violation}(\boldsymbol{\alpha}', \boldsymbol{\beta}', \boldsymbol{\gamma}', \boldsymbol{x}') < 10^{-4}$, the cut will not be considered as sufficiently violated and will not be inserted into the system. This is done in order to avoid numerical instabilities.

The flowchart in Figure 3.3 depicts the implementation of the cut separation procedure. The described methods were implemented using C++ and Cplex 11.1 [ILO]. An Intel Core 2 personal computer with 1.8 GHz and 3.25 GB of RAM was used for testing purposes. If not mentioned otherwise, the default Cplex settings are used.

3.10.2 Salman Instances

Salman's instances form the first set of benchmark instances. They include four problems originally defined in [GA90] (problems `arpa`, `oct`, `usa`, and `ring`) and 60 randomly generated problems originally published in Salman [Sal00]. They were also used in [SRH08]. For the latter ones, there are 12 groups with 20, 30 and 40 nodes. There are 9 cable types obeying economies of scale. The cheapest cable type has a capacity of 6. See [BGP$^+$00, SRH08] for a detailed description. The convex combinations of these cable types generate up to $\lceil \sum_{k \in K} d_k/6 \rceil$ modules. The notation `e(n)(s)(d)` provides summary information on the instances: `n` denotes the number of nodes, `s` explains the location of the center node (`c` stands for *central*, `r` stands for *random position*), `d` explains the level of demand (`l` stands for *low demand*, which is randomly generated between 0 and 30; `h` stands for *high demand*, randomly generated between 0 and 60). In [RS06, SRH08] two kinds of experiments were performed: using all 9 cable types and using only 4 of them. The methods in this thesis do not depend on the number of cable types, but on the number of modules. Therefore only the more challenging variant involving all 9 cable types is considered in this work. Table 3.2 provides input information on Salman instances: each of twelve `e(n)(s)(d)` groups contains 5 instances. Using the same style of presentation as in [SRH08], the average values per group are reported. The table shows the gaps as published in [SRH08]. In addition, it gives the number of nodes $|V|$, the number of edges $|E|$, the number of customers $|K|$, and the number of modules $|M|$ in the instance. Here $|M|$ denotes an average value over all edges: $|M| = \frac{\sum_{e\in E}|M_e|}{|E|}$. The remaining four columns $|V'|, |E'|, |K'|$ and $|M'|$ present the reduction achieved with the preprocessing techniques from Section 3.1.

	s	d	gap[SRH08]	$\|V\|$	$\|E\|$	$\|K\|$	$\|M\|$	$\|V'\|$	$\|E'\|$	$\|K'\|$	$\|M'\|$
e20	c	l	0.0	20.0	40.2	9.0	12.6	18.2	37.8	8.6	12.6
e20	r	l	1.9	20.0	39.8	10.0	13.0	17.4	35.4	10.0	13.0
e20	c	h	0.7	20.0	40.2	9.2	27.6	18.2	38.0	8.6	27.6
e20	r	h	1.0	20.0	39.8	9.2	23.2	17.6	36.2	9.2	23.2
e30	c	l	7.1	30.0	58.4	16.0	24.2	26.6	54.4	14.6	24.2
e30	r	l	7.6	30.0	59.2	14.4	22.2	26.8	55.2	13.8	22.2
e30	c	h	6.2	30.0	58.4	15.8	47.4	26.2	53.6	14.2	47.4
e30	r	h	4.5	30.0	59.2	12.2	33.4	26.6	54.8	12.2	33.4
e40	c	l	14.7	40.0	80.0	19.2	27.4	36.6	75.2	19.0	27.4
e40	r	l	10.4	40.0	80.6	20.6	31.4	35.6	74.6	19.4	31.4
e40	c	h	7.9	40.0	80.0	19.2	49.4	35.8	73.8	19.0	49.4
e40	r	h	6.3	40.0	80.6	18.2	46.2	33.8	71.8	16.8	46.2
oct			0.0	25	29	14	39	16	20	14	39
ring			6.5	32	60	17	47	26	54	17	47
usa			4.8	26	39	16	44	26	39	16	44
arpa			0.0	21	26	12	35	16	21	12	35

Table 3.2: Salman's instances. The upper part of the table shows average values over 5 instances in each class `e(n)(s)(d)`. The lower part shows the four instances from Gavish and Altinkemer [GA90]. gap[SRH08] is the average gaps reported by Salman et al. [SRH08]. $|V|$,$|E|$,$|K|$ and $|M|$ are before and $|V'|$,$|E'|$,$|K'|$ and $|M'|$ are after the preprocessing.

3.10.3 Real-World Instances

The second set of inputs are real world instances based on the street map of the Austrian city Bregenz with 1014 nodes and 1191 edges as underlying network. They were used as FTTC planning scenarios. Four different sets of customers (multiplexers) with cardinalities $|K| \in \{29, 36, 45, 67\}$ were considered. With respect to the demand there are two groups. In the group with *lower* demands `L`, each customer is assigned a demand of 4 units. The group with *higher* demands `H` associates a demand randomly chosen from $\{4, 8, 12, 16, 20\}$ to each customer. There are four different sets of modules as displayed in Table 3.3. Note that these sets do not obey economies of scale. It is assumed that there are empty conduits with limited modular capacity available at low costs, but if higher capacities need to be installed, new trenches need to be prepared, which involves high investment costs.

Type	$\|M\|$	(capacity $u_{e,1}$, cost $c_{e,1}$), ...
A	2	(120, 7.0), (1020, 146.0)
B	2	(30, 2.2), (1020, 146.0)
C	3	(30, 2.2), (60, 4.0), (1020, 146.0)
D	4	(30, 2.2), (60, 4.0), (120, 7.0), (1020, 146.0)

Table 3.3: The four different sets of modules used for the real-world instances.

Taking the four sets of customers, the two groups of demands and the four sets of modules into account, yields 32 benchmark instances.

3.10.4 Solving Salman Instances

This section reports on the results with the three compact MIP models (SCF), (MCF) and (DMCF), presented in Sections 3.3.1,3.4.1 and 3.4.2. This includes a comparison the branch-and-cut approaches based on seven different Benders' cut separation models, explained in Section 3.6. The main goals of this study were: a) to compare the qualities of lower bounds obtained by solving compact models versus branch-and-cut approaches, and b) to determine whether there is a difference in the performance of the branch-and-cut approach when the textbook implementation (SUB) is compared against normalized separation approaches. For that purpose, it is necessary to ensure that the obtained results are not biased by the quality of incumbent solutions found by the MIP solver. In previous computations best known upper bounds for all instances using the heuristic described in Section 3.9 are determined. For the reported results all models are initialized with the best known upper bound and heuristic calls are turned off. For this particular test, also Cplex cuts and the presolver are turned off. Benders' cuts are separated at the root node and in each 10th node of the branch-and-bound tree. Table 3.4 provides values averaged over 5 instances per group, for `e(n)(s)(d)` instances, and the values for the four additional instances from [GA90].

Gap at the root node: Table 3.4 reports on the quality of LP relaxations of three compact models and the corresponding value of the LP relaxation at the root node of the branch-and-bound tree for the SUBc approach. The gaps between obtained lower bounds LB and the best known upper bound (provided in column UB) are given as $\frac{UB-LB}{UB} \cdot 100\%$.

These results are consistent with the theoretical discussion provided in Section 3.8. The (SUBc) approach was the one among all branch-and-cut approaches to provide the tightest lower bounds at the root node. The average (median) gap over all 64 instances of the (SUBc) approach is 6.0% (5.9%). The worst LP relaxation gap among Benders' approaches is obtained by solving the (SUB) model: the average (median) gap is 7.5% (7.5%).

Comparing compact formulations, we observe that the average (median) gap of the (SCF) model of 19.6% (18.0%) can be improved to 9.1% (9.1%) by solving the (MCF) model, which can further be improved to 5.9% (6.0%) by solving the (DMCF) formulation. However, the LP relaxation of the (DMCF) model was not solved for 4 out of 20 instances of the group e40 within the time limit.

Looking at gaps of the (SUBc) approach and the (DMCF) model, we can observe two different effects. In some cases (SUBc) produces better gaps. This results from tightening Benders' cuts by rounding down the coefficients (see groups `e20_c_l`, `e40_c_h` in Table 3.4). In other cases the gap of (SUBc) is slightly worse than the one of (DMCF). This is explained by tailing-off and violation checks. Particularly, if at some point the current Benders' cut does not satisfy the violation test (3.94) this specific cut is not added to the model and instead we resort to branching. Consequently, the lower bound at the root node will be slightly worse than the value of the LP relaxation of (DMCF).

Gap after the time limit: For the (SCF) model and for Benders' separation approaches Table 3.4 also reports the lower-bound gap after the time limit of 1000s was reached. Every single variant of our branch-and-cut approach outperforms the compact (SCF) model. The best results are obtained by solving the (SUBf) approach: the average (median) gap after 1000 seconds is 2.5% (2.5%), while (SCF) terminates at 8.0% (7.3%).

(SUBf) solves 14 out of 20 instances of group `e20` to optimality, while (SCF) finds optima only in 7 out of 20 cases. Despite the bad quality of gaps of the LP relaxation, the model (SCF) succeeds to improve the final gap by drawing the advantage of branching. The average number of branch-and-bound nodes when solving (SCF) is close to 680 000, while the number of nodes processed by our Benders' implementations varies between 212 (SUBn) and 6043 (SUBf). The two normalizations (SUBcap) and (PSUBcap) were significantly outperformed by the seven other models and therefore, the results for these two normalizations are not reported here in detail.

The rightmost column in Table 3.4 shows the average gaps reported by Salman et al. [SRH08] obtained by solving SORb2 approach. The average gaps obtained by Raghavan and Stanojević [RS06] were always worse than those obtained in [SRH08], therefore only on the latter ones are reported. In [SRH08], the authors set the time-limit to 5400 seconds and used Cplex 9.1 with default settings. For this thesis, a time-limit of 1000 seconds was used. The MIP Solver is Cplex 11.2 with Cplex cuts and presolver turned off. According to the performance evaluation tests provided in [SPE], the computer is approximately 1.2 times faster than the one from [SRH08]. Comparing the values in the column (SUBf) and the last column in Table 3.4, one may conclude that in most cases the approach described here outperforms the approach of Salman et al. [SRH08].

Table 3.5 reports on the correlations between the average time needed to solve the subproblem, the number of branch-and-bound nodes and the tightness of the bounds at the root node of the branch-and-bound tree. The average values over all 64 Salman instances for the following parameters are provided: Time_0 and Gap_0 denote the running time and the gap at the root node of the branch-and-bound tree, respectively; $Benders_0$ denotes the number of Benders' cuts separated at the root node; $Time_0/Benders_0$ provides the ratio between the total time spent and the number of Benders' cuts. The values *Benders*, *Gap* and *Time/Benders* are the corresponding values provided for the total running time of 1000 seconds. The last row shows how many branch-and-bound nodes have been processed within the time limit. For the results after 1000 seconds, the two best performing approaches are shown in bold face.

The normalized Benders' subproblems have a complicated flow structure with two kinds of capacity constraints. Therefore, the problem of solving a normalized subproblem by closing the unbounded cone with an additional constraint may become a difficult task. Row $\mathrm{Time}_0/\mathrm{Benders}_0$ of Table 3.5 provides an estimate of an average time (in seconds) needed to solve each Benders' subproblem. The fastest subproblems are (SUBf) and (PSUBn) (followed by the separation of extreme rays with the (SUB) approach). Correspondingly, these two variants are first to be finished at the root node of the branch-and-bound tree. Therefore, they are also separating the most Benders' cuts and traversing the most nodes of the branch-and-bound tree. However, the (SUBf) bounds obtained at the root node are tighter than the corresponding bounds of the

Problem				Gap at the Root Node				Gap after 1000s								Gap
				Cplex presolver off				Cplex cuts and presolver off								
	s	d	UB	DMCF	MCF	SCF	SUBc	SCF	SUB	SUBc	SUBn	SUBf	PSUBc	PSUBn	PSUBf	[SRH08]
e20	c	l	111.9	5.5	11.2	32.3	**5.4**	0.3	**0.0**	**0.0**	0.4	**0.0**	**0.0**	**0.0**	**0.0**	0.0
e20	r	l	143.3	**6.6**	10.9	33.2	6.8	4.7	1.6	2.0	2.5	**1.2**	1.7	1.5	1.5	1.9
e20	c	h	194.2	**6.0**	9.8	22.1	**6.0**	1.8	2.4	2.6	4.3 [1]	1.3	2.0	2.0	**1.2**	0.7
e20	r	h	239.8	**5.8**	9.4	19.7	**5.8**	2.8	1.6	1.7	3.8 [2]	**0.4**	0.9	1.1	0.9	1.0
e30	c	l	323.5	**6.6**	10.8	23.9	7.1 [1]	9.1	4.0	4.3 [1]	6.1 [2]	**2.7**	3.5	3.2	3.4	7.1
e30	r	l	290.1	**5.9**	9.1	22.9	**5.9**	7.8	4.0	4.4	6.0 [2]	**2.8**	4.0	3.3	3.5	7.6
e30	c	h	529.1	**5.1**	8.8	15.2	**5.1** [1]	6.3	4.4	4.4 [1]	7.5 [5]	**3.0**	3.7	3.5	3.5	6.2
e30	r	h	469.0	**4.9**	6.9	13.7	**4.9**	5.5	3.7	4.1	5.5 [3]	**2.8**	3.6	3.1	3.3	4.5
e40	c	l	409.1	**6.8**	11.0	28.1	7.1 [2]	15.4	6.5	6.5 [2]	8.6 [1]	**4.5**	6.4 [2]	5.2	5.6	14.7
e40	r	l	572.4	**5.8**	9.0	22.2	6.1 [2]	12.3	5.2	5.7 [2]	6.8 [2]	**3.9**	5.2	4.4	5.0	10.4
e40	c	h	674.8	6.1 [2]	8.2	16.0	**5.8** [4]	9.4	5.1	5.8 [4]	6.6 [2]	**3.7**	5.3 [1]	4.2	4.7 [1]	7.9
e40	r	h	745.6	**5.1** [2]	7.2	14.1	5.2 [4]	8.0	4.8 [1]	5.0 [4]	6.3 [5]	**3.4**	4.5	4.0	4.4	6.3
oct			2432.3	**6.1**	8.8	17.5	6.3	3.9	2.8	4.0	5.2	1.4	3.0	**1.2**	2.5	0.0
ring			1391.3	**7.2**	11.5	25.6	7.3	11.1	5.6	5.9	7.9	**3.8**	5.4	4.4	4.6	6.5
usa			2233.2	**7.4**	10.4	19.5	**7.4**	8.3	6.2	6.2	8.2 [1]	**4.6**	5.6	5.1	5.4	4.8
arpa			2571.4	**5.0**	9.4	15.0	5.1	1.6	1.5	2.2	4.5	**0.5**	1.7	1.7	1.3	0.0

Table 3.4: Results for Salman's instances. The upper part of the table shows average values over 5 instances in each class `e(n)(s)(d)`. The lower part shows results for the four instances from Gavish and Altinkemer [GA90]. We report the gaps to the best known upper bound obtained at the root node and after the time limit. The numbers in squared brackets denote in how many (out of 5) cases the LP solution at the root node was not found within the time limit.

(PSUBn) model, which makes the (SUBf) approach the winner, when solving this data set.

This study shows that:

- Using rounded Benders' cuts derived from the (DMCF) formulation outperforms the compact (SCF) model.
- Two important aspects decide on the quality of the Benders' approach: a) the running time needed to solve the Benders' subproblem, and b) the quality of the derived Benders' cuts. The model that succeeds to balance the trade-off between these two aspects is the most desirable one.

Average	DMCF	MCF	SCF	SUB	SUBc	SUBn	SUBf	PSUBc	PSUBn	PSUBf
Time_0	503.2	3.2	0.1	68.3	422.5	603.5	22.0	247.0	35.3	170.0
Benders_0	-	-	-	40.9	38.3	67.8	57.2	36.0	115.5	52.9
$\text{Time}_0/\text{Benders}_0$	-	-	-	1.7	11.0	8.9	0.4	6.9	0.3	3.2
Gap_0	5.9	9.1	19.6	7.5	6.0	6.9	6.4	6.1	6.9	6.2
Benders	-	-	-	740.8	164.7	159.2	**1281.5**	303.4	**1498.6**	556.0
Time/Benders	-	-	-	1.3	6.1	6.3	**0.8**	3.3	**0.7**	1.8
Gap	4.5	4.5	8.0	3.6	3.9	5.4	**2.5**	3.5	**3.0**	3.1
Nodes	1031	100 264	680 981	2152	492	212	6043	1326	3655	2501

Table 3.5: Average values over all 64 Salman's instances.

3.10.5 Solving Real-World Instances

This section shows the comparison of results obtained for the set of real-world instances derived from Bregenz, a city in Austria.

Preprocessing: The preprocessing greatly reduces the size of the graph from 1014 nodes and 1191 edges to approximately 350 nodes and 500 edges. The number of customers goes down to 28, 33, 41 and 61, respectively. Furthermore, although we start with uniform modules, we end up with non-uniform ones. The first seven columns of Table 3.6 show the detailed reduction. $|M| = \frac{\sum_{e \in E} |M_e|}{|E|}$ denotes the average number of modules per edge. $|\underline{M}| = \min_{e \in E} |M_e|$ and $|\overline{M}| = \max_{e \in E} |M_e|$ denote the smallest and largest number of modules per edge respectively.

Gap at the root node: The first test examines the strength of the three compact models, (SCF), (MCF) and (DMCF). Therefore, Cplex cuts and the presolver off are turned off and the LP relaxations at the root node are computed. The three columns labeled with *Gap at the root node* in Table 3.6 show the integrality gaps. For all 32 instances, the LP relaxation of the (SCF) model was solved within 1 or 2 seconds, but the average (median) gap over all Bregenz instances is 46.6% (42.0%). As expected, lower bounds obtained by solving the (MCF) model are significantly better: 19.1% (7.1%), but the LP relaxations of only 13 out of 32 instances were solved to optimality in less than 1000 seconds (within 383.8 seconds, on average). Finally, the average (median) gap obtained by solving the (DMCF) model is 13.9% (7.8%), but only in 3 out of 32 cases the LP relaxations were solved to optimality within the given time limit (in 55 seconds, on average). This also explains why some of the presented gaps of the (MCF) model are better than the corresponding (DMCF) ones (LP relaxations are solved by dual simplex method).

According to these experiments it can be concluded that the only compact model that can be directly solved without a row and/or column generation technique is the (SCF) model. In order to use the strength of the (DMCF) model, we apply a row-generation technique to it. A column-generation technique for a similar problem has been presented in [FG09, FG10].

Gap after the time limit: For the (SCF) model and for the seven branch-and-cut variants described above, the code is run for 1000 seconds, with default Cplex settings and the primal heuristic described in Section 3.9. Only when solving the (SUB) model, the Cplex presolver needs to be turned off. Since the separation of Benders' cuts may become a time-consuming task for instances of that size, they are only separated at the root node of the branch-and-bound tree.

Box-plots in Figure 3.4 provide an overview of the obtained gaps at the root node of the branch-and-bound tree. We observe that the huge gaps of the (SCF) model (46.6% average and 42.0% median value) can be reduced down to an average (median) value of 4.2% (3.4%), by turning on Cplex cuts and the presolver. By applying Benders' cuts, all proposed methods return better results: the average gap varies between 3.1% (PSUBf) and 3.6% (PSUBc), and the corresponding median values vary between 2.6% (PSUBf) and 2.8% (SUBc). The normalization variants (SUBcap) and (PSUBcap) were significantly outperformed by the other strategies. Typically the Benders subproblems could not be solved in the subproblem-timelimit of 45 seconds. Consequently, no detailed results for these models are reported.

Figure 3.5 shows the gaps after the time limit was reached. Looking at the overall gaps after the given time limit, we observe that it is difficult to point out the differences between particular normalization approaches when default

Cplex settings are used (see Figure 3.5(a)). Therefore, we cannot say that there is a clear winner among different Benders' approaches. Although the Benders' cuts obtained by solving the (SUBc) model are among the tightest ones (see, e.g., Figure 3.4), the separation was not finished at the root node of the branch-and-bound tree in 24 out of 32 cases. Figure 3.6 illustrates a typical situation in which (SUBc) gets stuck in the separation phase, while the (SUB) approach, for example, can draw an advantage out of branching.

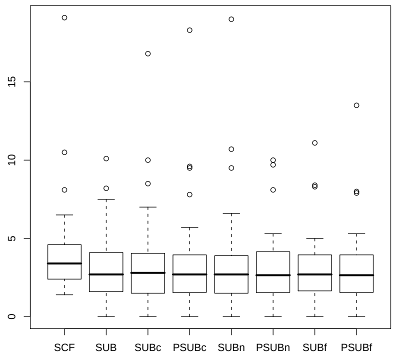
Looking at Figures 3.4(a) and 3.5(a) one could get the impression that the simple SCF model (solved with the default Cplex settings) is competitive with the much more complex Benders' approach. However, Table 3.6 presents detailed results obtained with default Cplex settings and shows that this is a wrong impression. Indeed, even on the set of real-world instances, the Benders' decomposition approach is able to outperform the (SCF) model. For 8 out of 32 instances, the branch-and-cut approach (PSUBf) is able to find the optimal solution within the given time limit, while the (SCF) model did not solve a single instance to optimality. Furthermore, for 18 out of the remaining 24 instances, better gaps were produced using the branch-and-cut approach rather than solving the (SCF) model.

3.10.6 Testing Magnanti-Wong Enhancements

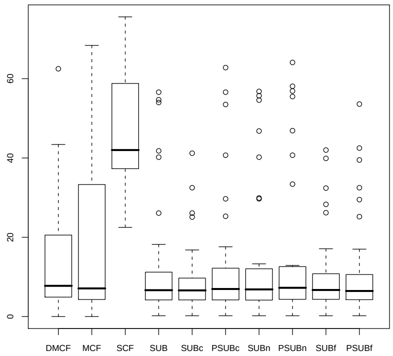
This section reports on negative results when trying to enhance the Benders' decomposition and by using the Magnanti-Wong (MW) approach, detailed in Section 3.7.6. As already observed above, if Cplex general purpose cuts are turned on, it is difficult to point out the differences between different variants of Benders' separation models. Therefore, to test the effects of applying the MW approach, we turned the Cplex cuts off. The MW approach generates most improving cuts when applied to the (SUBn) approach. For this normalization the gaps obtained within the time limit of 1000 seconds are depicted in Figure 3.7. We observe that the MW approach slows down the performance: the overall number of included Benders' cuts is reduced while there is no significant improvement in the quality of lower bounds obtained per iteration.

3.11 Conclusions

A new disaggregated flow formulation (DMCF) is presented which is a byproduct of the model introduced in [CGM07]. It induces tighter gaps than the (MCF) model which is typically used for network loading problems. Using Benders' decomposition, 8 of the 32 new single-source instances can be solved to optimality within a reasonable time limit. For 18 out of the remaining 24 instances, we see

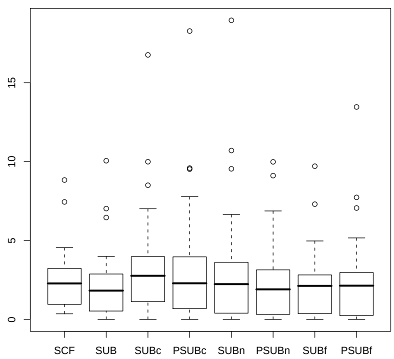


(a) With Cplex default settings.

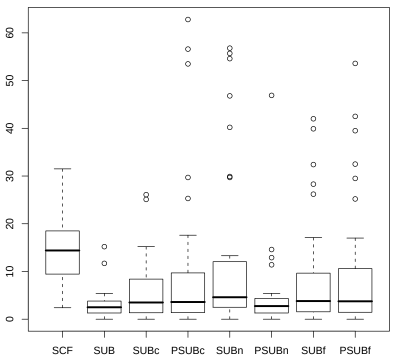


(b) By turning off Cplex cuts and the presolver.

Figure 3.4: Box-plots over 32 `Bregenz`-instances: the gaps (in %) of lower bounds at the root node of the branch-and-bound tree.



(a) With Cplex default settings.

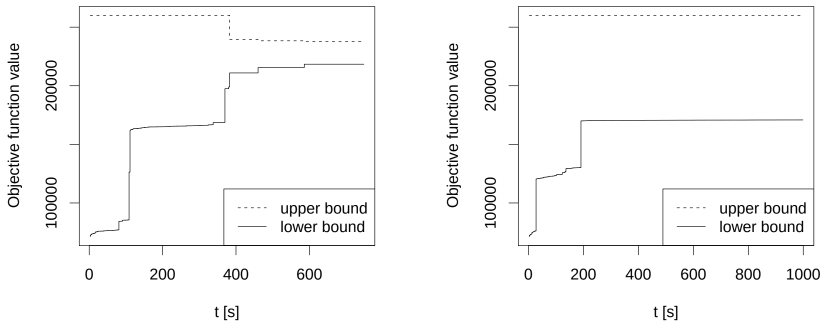


(b) By turning off Cplex cuts and the presolver.

Figure 3.5: Box-plots over 32 `Bregenz`-instances: the overall gaps (in %) obtained after 1000 seconds.

Problem							best	best	Gap at the root			Gap after 1000s							
									Cplex presolver off			with Cplex default settings							
Instance	$\|V\|$	$\|E\|$	$\|K\|$	$\|\underline{M}\|$	$\|M\|$	$\|\overline{M}\|$	**LB**	**UB**	DMCF	MCF	SCF	SCF	SUB	SUBc	SUBn	SUBf	PSUBc	PSUBn	PSUBf
29_A_H	322	488	28	2	2.0	3	**110 998.28**	**110 998.28**	3.8	3.9	58.5	1.1	**0.0**	1.1	0.3	0.2	1.2	**0.0**	**0.0**
29_B_H	322	488	28	2	2.0	4	227 009.39	228 146.84	30.6	68.4	71.1	**0.5**	0.6	7.0	6.6	2.1	7.8	0.7	2.1
29_C_H	322	488	28	3	3.0	7	55 166.67	56 779.45	24.6	10.7	25.7	3.3	3.1	3.3	2.9	2.9	4.0	**2.8**	2.9
29_D_H	322	488	28	4	4.1	10	50 447.35	52 546.87	21.7	11.6	33.1	4.5	**4.0**	5.5	5.0	4.1	5.0	4.6	4.3
36_A_H	325	490	33	2	2.0	4	**174 491.64**	**174 491.64**	3.5	3.7	42.0	1.0	0.5	0.4	0.4	0.2	0.5	0.2	**0.0**
36_B_H	325	490	33	2	2.0	4	807 815.62	819 877.83	12.8	37.1	37.4	**1.5**	2.1	4.5	5.8	2.8	5.7	1.9	2.1
36_C_H	325	490	33	3	3.0	7	192 314.14	193 821.25	14.5	56.1	60.8	**0.8**	1.0	1.8	1.8	**0.8**	2.4	0.9	**0.8**
36_D_H	325	490	33	4	4.1	11	82 432.96	84 676.01	16.0	10.6	26.0	2.8	2.6	3.2	3.0	2.7	**2.6**	3.1	2.7
45_A_H	333	498	41	2	2.0	4	**206 863.16**	**206 863.16**	3.1	2.7	43.7	1.1	0.2	1.0	**0.0**	**0.0**	0.1	**0.0**	**0.0**
45_B_H	333	498	41	2	2.0	4	851 564.22	915 891.12	33.6	40.0	39.9	7.4	**7.0**	10.0	10.7	7.3	9.6	9.1	7.7
45_C_H	333	498	41	3	3.0	7	224 778.79	230 380.49	22.3	58.9	59.9	**2.6**	6.5	8.5	9.5	5.0	9.5	6.9	7.1
45_D_H	333	498	41	4	4.1	11	100 670.85	104 114.25	19.4	19.6	27.3	3.3	3.3	3.7	3.4	**3.0**	3.7	3.5	3.6
67_A_H	351	516	61	2	2.0	4	236 879.59	238 483.81	6.4	5.9	37.2	1.2	0.8	1.8	0.7	0.8	1.2	**0.7**	**0.7**
67_B_H	351	516	61	2	2.0	4	1677 023.89	1839 517.90	43.4	67.7	64.8	**8.8**	0.1	16.8	19.0	9.7	18.3	10.0	13.5
67_C_H	351	516	61	3	3.0	7	611 825.02	625 948.67	14.2	46.8	42.0	**2.4**	3.7	5.0	5.1	3.8	5.1	4.9	5.2
67_D_H	351	516	61	4	4.1	11	138 159.10	141 382.31	26.4	29.5	22.5	**2.3**	2.6	3.4	3.1	2.4	3.4	3.2	3.1
29_A_L	322	488	28	1	1.0	1	**101 951.52**	**101 951.52**	**0.0**	**0.0**	75.6	0.4	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
29_B_L	322	488	28	2	2.0	4	38 318.98	38 472.82	7.7	7.4	55.7	1.6	0.7	1.2	0.7	2.5	0.8	**0.4**	0.5
29_C_L	322	488	28	3	3.0	4	34 927.38	35 758.16	5.1	4.6	59.1	3.2	2.3	3.3	2.3	**2.2**	2.4	2.3	2.3
29_D_L	322	488	28	3	3.0	4	34 734.91	35 533.78	4.7	4.0	60.5	3.8	**2.2**	3.1	2.4	2.4	2.3	2.3	2.3
36_A_L	325	490	33	2	2.0	2	**163 386.07**	**163 386.07**	0.6	0.6	53.4	0.4	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
36_B_L	325	490	33	2	2.0	4	**59 201.52**	**59 201.52**	6.4	6.0	36.0	0.7	**0.0**	1.5	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
36_C_L	325	490	33	3	3.0	5	57 067.61	57 797.48	7.2	6.3	39.7	2.3	1.6	2.2	1.2	1.4	**1.1**	1.3	1.5
36_D_L	325	490	33	4	4.0	5	54 881.13	56 167.45	5.8	4.6	41.9	3.1	2.3	2.7	2.3	**2.1**	2.2	2.4	2.6
45_A_L	333	498	41	2	2.0	2	**193 052.96**	**193 052.96**	0.1	0.1	56.7	0.6	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
45_B_L	333	498	41	2	2.0	4	70 211.46	70 546.13	7.8	6.8	37.7	1.3	0.7	0.6	**0.4**	0.5	0.6	0.5	0.7
45_C_L	333	498	41	3	3.0	6	67 156.12	68 224.05	7.1	5.7	40.7	2.3	1.6	2.5	**1.5**	1.9	1.9	1.9	1.7
45_D_L	333	498	41	4	4.0	6	64 901.08	66 471.38	6.4	4.8	43.8	3.3	2.4	2.8	**2.3**	**2.3**	**2.3**	2.5	2.6
67_A_L	351	516	61	2	2.0	3	**218 267.74**	**218 267.74**	1.1	1.1	54.6	0.6	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
67_B_L	351	516	61	2	2.0	4	200 000.95	201 352.36	62.5	61.1	67.9	0.9	1.0	2.8	**0.7**	0.9	2.0	**0.7**	1.5
67_C_L	351	516	61	3	3.0	7	80 926.47	82 804.18	12.2	12.0	35.5	2.5	2.3	2.8	2.2	**2.1**	2.7	2.3	2.5
67_D_L	351	516	61	4	4.1	8	79 031.51	81 906.07	12.8	13.7	41.0	4.0	3.5	4.2	3.8	**3.4**	3.9	4.2	3.8

Table 3.6: Lower bounds obtained by solving LP relaxations of three compact approaches (with Cplex presolver turned off) are compared. The overall gaps (obtained with default Cplex settings, after 1000 seconds) of the compact model (SCF) and seven branch-and-cut variants are provided.



(a) (SUB) (b) (SUBc)

Figure 3.6: Lower bound growth vs. time (CPU seconds) with models (SUB) and (SUBc) for instance `29_B_H`. (a) The first huge increase of lower bound is due to two subsequently found Benders' cuts, the second increase is due to branching. (b) The separation at the root node is not finished when solving (SUBc).

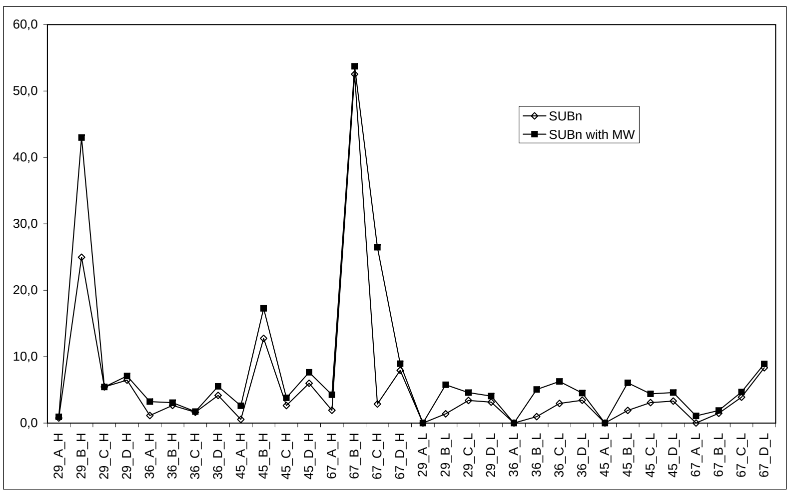


Figure 3.7: Comparing the gaps obtained within the time limit of 1000 seconds (Cplex cuts turned off): (SUBn) and (SUBn) extended by Magnanti-Wong cuts.

better gaps than the best performing compact formulation.

Comparing normalization strategies for the Benders' decomposition, we see that depending on the structure of the inputs, different normalizations are preferable. However, in contrast to a common belief, the separation of extreme rays, which is also called the *textbook implementation*, provides relatively good results across all instances.

There are several arguments explaining this observation:

1. We solve the problem starting from a compact formulation (the (SCF) model) and we use Benders' cuts only in order to improve the quality of lower bounds, i.e., they are *not necessary* for the LAN problem to have a complete MIP formulation. This is in contrast to known approaches for solving the multiple-source multiple-sink network loading, where Benders' inequalities are separated in a similar way.

2. Unlike the LAN objective function, many related problems consider flow-dependent objective values. In such cases, one has to separate both, feasibility and optimality Benders' cuts. The quality of optimality cuts is essential for such problems and therefore enhancing approaches (like those given in, e.g., [FSZ10, MW81, MW84], [RGCS09]) play a crucial role to make Benders' decomposition work.

3. The results confirm the claim of Magnanti and Wong [MW81], that the crucial role in the generation of efficient Benders' separation approaches is played by the size of the convex hull of the relaxed master problem (see Section 3.8). We show that the textbook implementation of Benders' separation is not the worst possible choice, if a "good" LP-model is used to generate the corresponding cuts. Typically there is a trade-off problem in Benders' decomposition approaches between the strength of the subproblem and the running time needed to solve it. To overcome this problem, the separation of extreme rays turns out to be a good compromise: an extreme ray is usually found much faster than an optimal extreme point of a bounded subproblem.

The algorithmic framework has been developed to solve large single-source instances arising in the design of telecommunication networks. Some ingredients in the approach exploit the single-source assumption. For example, degree-balance constraints or primal heuristic guarantee that the final solution is a directed acyclic graph. Also, based on the single-source assumption, Benders' cuts are being added to strengthen the LP-relaxation, but they are not necessary for the feasibility of a solution of the (SCF) master model. However, the presented approaches for separating Benders' cuts derived from the disaggregated formulation could also be applied to more general multi-commodity versions of the capacitated network design problem.

It would be interesting to compare the developed branch-and-cut algorithm with a stabilized column-and-row generation technique as the one proposed in [FG10]. Furthermore, one could even consider a branch-and-cut-and-price approach that combines Benders' and Dantzig-Wolfe decompositions, thereby exploiting the best of both methods.

Chapter 4

Solving the Prize Collecting Local Access Network Design Problem Heuristically

This chapter deals with the *Prize-Collecting Local Access Network design problem* (PC-LAN). This is a new combinatorial optimization problem that forms a generalization of the Local Access Network design problem (LAN). It can be used to model the deployment of broadband telecommunications systems in which optical fiber cables are installed between a *central office* and a number of *customers*. It takes into account the fact that the network does not necessarily need to connect all customers. The aim is to select the customers to be connected to the central server and to choose the link capacities to establish these connections. This question arises in the context of detailed telecommunication network planning. In fiber to the home (FTTH), or fiber to the building (FTTB) scenarios, the telecom company takes the strategic decision of fixing a percentage of customers that should be served, and aims for minimizing the total cost of the network providing this minimum service. Due to the complexity of the problem and the size of the instances in real applications, it is difficult to establish algorithmic approaches that ensure global cost-minimal solutions.

A mixed integer programming based heuristic approach for PC-LAN is presented. It combines a cutting plane algorithm with a multi-start heuristic. The multi-start heuristic starts with fractional values of the LP-solutions and creates feasible solutions that are later improved using a local improvement strategy. A set of three new real-world benchmark instances with up to 86 000 nodes, 116 000 edges and 1 500 potential customers is used to evaluate this approach. The computational results at the end of this chapter show that this MIP-based heuristic is preferable to using the heuristic multi-start approach alone, without the MIP ingredient. Furthermore the MIP-based approach gives a certificate on the quality of the solutions by providing lower bounds to the optimal solution value. Figure 4.1 shows two deployment scenarios for a real world instance with coverage rates of 60% and 90%, respectively.

A preliminary version of the material in this chapter was presented at the *International Network Optimization Conference 2011* and appears in the proceedings [LPSG11b].

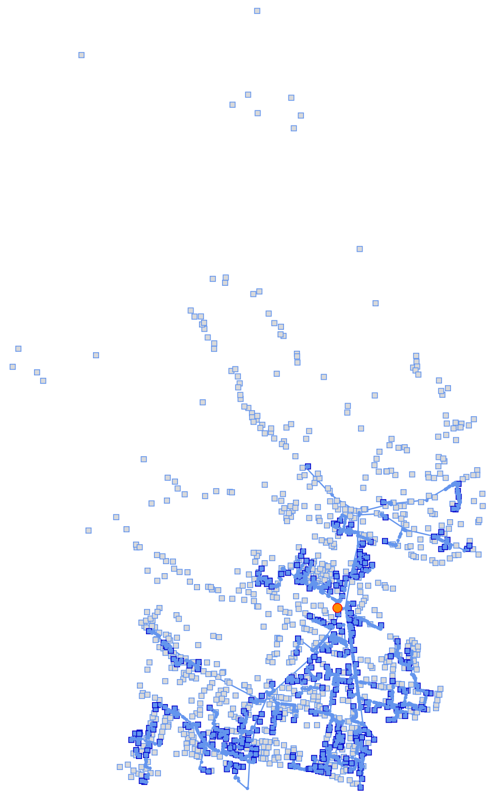
For completeness, the definition of the *Prize-Collecting Local Access Network design problem* (PC-LAN) from Section 1.3 is repeated here:

Definition 4.0.1. *We are given an undirected, connected graph $G = (V, E)$ with a central node $r \in V$. A subset of nodes $K \subseteq V \setminus \{r\}$ represents customers. To each customer $k \in K$ a positive demand d_k, a positive prize p_k and a positive setup cost c_k are associated. A target prize p_0 is given. On each edge at most one module m out of a set $M_e = \{1, 2, \dots\}$ can be installed. Each module has associated a positive capacity $u_{e,m}$ and positive cost $c_{e,m}$. The module indices are sorted by increasing capacity, i.e., $u_{e,m} < u_{e,m+1}$. The* Prize-Collecting Local Access Network design problem (PC-LAN) *asks for a selection of customers to be served and an installation of at most one module per edge. The selection of customers shall cover at least the target prize p_0. The installation of modules shall allow for a single-source multiple-sink routing from r to the selected customers, that satisfies all the demands simultaneously. The cost for the installation of modules plus the cost for the selected customers shall be minimal.*

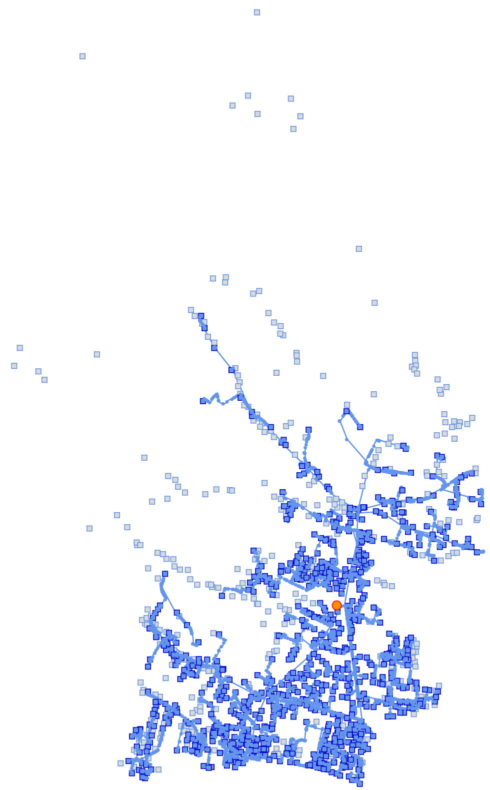
The connection from the central office to a customer can be seen as a *flow* that is allowed to split apart. Thus we are speaking of a *bifurcated* flow. As a result, an optimal solution of the problem is not necessarily a tree in the graph. The target prize p_0 can be given by means of a *coverage rate* α, $(0 < \alpha \leq 1)$: $p_0 = \alpha \sum_{k \in K} p_k$. PC-LAN contains LAN as a special case where $\alpha = 1$.

4.1 Preprocessing

This section describes a set of preprocessing steps for the PC-LAN problem. These are adaptations of the methods for the LAN problem as described in Section 3.1. The aim of the preprocessing is to transform an instance of PC-LAN denoted by L^i into a smaller instance L^{i+1} under the condition that any feasible solution S^{i+1} for L^{i+1} can be mapped back to a feasible solution of L^i with the same objective value. Additionally, if the preprocessed problem is infeasible, then also the original problem is infeasible.



(a) A solution with 60% coverage.



(b) A solution with 90% coverage.

Figure 4.1: Realistic planning scenario with coverage rates of 60% and 90%. Square nodes denote customers. The circle node denotes the *central office* r. The served customers are depicted with dark squares. Lines denote the installed connections in the solution.

In this section we denote the PC-LAN instance after i preprocessing steps by a graph $G^i = (V^i, E^i)$, a central node r^i, customers $K^i \subseteq V^i \setminus \{r^i\}$, with demand $d^i \in \mathbb{R}^{|K^i|}_{\geq 0}$, prize $p^i \in \mathbb{R}^{|K^i|}_{\geq 0}$, cost $d^i \in \mathbb{R}^{|K^i|}_{\geq 0}$, modules $u^i_{e,m}, c^i_{e,m}, M^i_e$ and a target prize p^i_0. In addition the objective function contains an additional *fixed cost* term F^i:

$$\min \sum_{e \in E^i} \sum_{m \in M^i_e} u^i_{e,m} x^i_{e,m} + F^i.$$

The list of preprocessing steps is given as follows:

(i) **Degree zero, center node:**
If the center node r^i has degree 0 and the target prize is greater than zero $p^i_0 > 0$, the instance is infeasible.

If the target prize is equal to zero p^i_0, the instance has a trivial solution of selecting no customers and making no installation which yields an objective value of zero.

(ii) **Degree zero, Steiner node:**
If there is a non-customer, non-center node v with degree 0, this node will certainly not be in any solution, hence it can be deleted from the instance: $V^{i+1} := V^i \setminus \{v\}$.

(iii) **Degree zero, customer node:**
If there is a customer node k with degree 0, it can not be in the solution. Hence, it can be deleted from the instance: $K^{i+1} := K^i \setminus \{k\}, V^{i+1} := V^i \setminus \{k\}$.

(iv) **Degree one, center node:**
If the center node r^i has degree 1 and the incident edge $e = \{r^i, v\}$ provides a module with sufficient capacity for $\sum_{k \in K^i} d^i_k$, this edge will be in any solution. Therefore it can be deleted: $E^{i+1} := E^i \setminus \{e\}, V^{i+1} := V^i \setminus \{r^i\}$, the center is moved to the adjacent node: $r^{i+1} := v$ and we can easily compute the module $\tilde{m} := \mu^i_e \left(\sum_{k \in K^i} d^i_k\right)$ and only keep the cost $F^{i+1} := F^i + c^i_{e,\tilde{m}}$. For the back-mapping it must be noted that $e, \tilde{m}$ is included in the solution S^i.

If on the other hand e does not provide sufficient capacity, the problem is infeasible.

(v) **Degree one, Steiner node:**
If there is a non-customer, non-center node v with degree 1, this node will certainly not be in any solution. Therefore v and the incident edge $\{v, w\}$ can be deleted from the instance: $E^{i+1} := E^i \setminus \{\{v, w\}\}, V^{i+1} := V^i \setminus \{v\}$.

(vi) **Degree one, customer node:**
If there is a customer node k with demand d_k^i with degree 1 and the incident edge $e = \{k, v\}$ provides a module with sufficient capacity for d_k^i and the adjacent node v is not a customer $v \notin K^i$, this edge will be in the solution iff the customer k is in the solution. Therefore the edge can be deleted from the instance, its cost can be attributed to the customer and the customer is moved to the adjacent node v: $E^{i+1} := E^i \setminus \{e\}, K^{i+1} := K^i \setminus \{k\}, V^{i+1} := V^i \setminus \{k\}$ and $K^{i+1} := K^{i+1} \cup \{v\}$ with attributes $d_v^{i+1} := d_k^i$, $p_v^{i+1} := p_k^i$ and $c_v^{i+1} := c_k^i + c^i_{e,\mu_e^i(d_k^i)}$. For the back-mapping it must be noted that the customer k is in the solution S^i if the customer v is in the solution S^{i+1}. And also the edge and module $e, \mu_e^i\left(d_k^i\right)$ are in the solution S^i in this case.

If the edge e does not provide sufficient capacity, the customer can not be in the solution and can be deleted.

(vii) **Degree two, Steiner node:**
If there is a non-customer, non-center node w with degree 2, then either both incident edges $\{v, w\}, \{w, z\}$ will be in the solution or none. Hence these two sequential edges can be replaced by one edge: $E^{i+1} := E^i \setminus \{\{v, w\}, \{w, z\}\} \cup \{\{v, z\}\}, V^{i+1} := V^i \setminus \{w\}$. The modules for the new edge $M^{i+1}_{\{v,z\}}$ result from installing one module from each of the two original edges $\{v, w\}, \{w, z\}$ in series. More precisely, every pair of modules $\langle m_a, m_b\rangle \in M^i_{\{v,w\}} \times M^i_{\{w,z\}}$ implies a new module $\tilde{m}$ with $u^{i+1}_{\{v,z\},\tilde{m}} := \min\left(u^i_{\{v,w\},m_a}, u^i_{\{w,z\},m_b}\right)$ and $c^{i+1}_{\{v,z\},\tilde{m}} := c^i_{\{v.w\},m_a} + c^i_{\{w,z\},m_b}$. This leads to $\left|M_{\{v,z\}}\right| = \left|M_{\{v,w\}}\right| \cdot \left|M_{\{w,z\}}\right|$ steps for the new edge $\{v, z\}$. For the back-mapping it must be recorded that if the new edge $\{v, z\}$ is in the solution S^{i+1} with the module $\tilde{m} \in M^{i+1}_{\{v,z\}}$ it implies that both edges $\{v, w\}, \{w, z\}$ are in S^i with the respective modules that were combined to make up $\tilde{m}$.

Dispensable modules are removed from $M^{i+1}_{\{v,z\}}$ in Step (ix). Note that there may already be an edge from v to z so we temporarily allow for parallel edges. See Step (viii) for a resolution.

(viii) **Parallel edges:**
Step (vii) may result in two parallel edges $e = \{v, w\}, h = \{v, w\} \in E^i$. Either one alone or both together can be used in a solution. Therefore they can be replaced by a single edge $g = \{v, w\} : E^{i+1} = E^i \setminus \{e, h\} \cup \{g\}$. The modules for this new edge M_g^{i+1} result from all modules in M_e^i, united with all modules in M_h^i, united with all possible combinations of one module from M_e^i and one from M_h^i. More precisely, every pair of modules

$\langle m_a, m_b \rangle \in M_e^i \times M_h^i$ implies a new module $\tilde{m}$ with $u_{g,\tilde{m}}^{i+1} := u_{e,m_a}^i + u_{h,m_b}^i$ and $c_{g,\tilde{m}}^{i+1} := c_{e,m_a}^i + c_{h,m_b}^i$. In summary, this leads to $|M_g^{i+1}| = |M_e^i| + |M_h^i| + |M_e^i| \cdot |M_h^i|$ steps for the new edge g. For the back-mapping it must be noted that if $g, \tilde{m}$ is in S^{i+1} it implies that the edges and modules from e, h, M_e^i, M_h^i that make up $\tilde{m}$ are in S^i.

(ix) **Dispensable modules:**
Steps (vii) and (viii) may lead to *dispensable* modules. A module $\tilde{m} \in M_e^i$ is dispensable if there exists another module $m' \in M_e^i$ with $u_{e,m'}^i \geq u_{e,\tilde{m}}^i$ and $c_{e,m'}^i \leq c_{e,\tilde{m}}^i$. A dispensable module $\tilde{m}$ will certainly not be in any solution, hence can be deleted: $M_e^{i+1} := M_e^i \setminus \{\tilde{m}\}$.

(x) **Excess modules:**
No optimal solution needs to have any installation greater than $\sum_{k \in K^i} d_k$. (See the proof for acyclic solutions in Section 3.2) Consequently, sets of excess modules $\tilde{M}_e = \left\{m \in M_e^i \mid u_{e,m}^i \geq \sum_{k \in K^i} d_k^i\right\} \subseteq M_e^i$ can be replaced by a single module $\tilde{m}$: $M_e^{i+1} = M_e^i \setminus \tilde{M}_e \cup \{\tilde{m}\}$ with $c_{e,\tilde{m}}^{i+1} = \min_{m \in \tilde{M}_e} c_{e,m}^i$ and $u_{e,\tilde{m}}^{i+1} = \sum_{k \in K^i} d_k^i$. For the back-mapping it must be noted that if $\tilde{m}$ is used on e in S^{i+1} it implies that the cheapest excess module $\arg\min_{m \in \tilde{M}_e} c_{e,m}^i$ is used on e in S^i.

(xi) If the set of customers provides less than the target prize $\sum_{k \in K^i} p_k^i < p_o^i$, the instance is infeasible.

Note that no preprocessing is specified for a degree one customer k when the adjacent node j is also a customer. It would be possible to do *one* such step. The problem definition would have to be adapted to accommodate for up to two customers on the same node. Suppose customer j has degree two in the original instance and hence it has degree one after the customer k has been moved onto the same node as j. It is now not immediately clear which module to choose on this unique edge incident to node j, because due to the prize collecting aspect it can not be easily decided which customer is included in the solution. It could be one of k and j, or both, or none. Therefore the degree one node j with two associated customers could not be preprocessed any further. So, firstly, this transformation from a degree one customer node with a customer neighbor to a single node would require a modified problem definition to allow for multiple customers per node. Secondly, it would only allow for *one* additional step, but not for a sequence of further preprocessing steps that may accumulated to produce any significant reduction of the size of the instance. Consequently, this preprocessing step is not considered in this work.

These preprocessings are implemented as follows: Iterate over all nodes and perform any applicable preprocessing for nodes with degree zero, one or two, i.e. steps (i)-(vii). Preprocessing step (vii) always triggers an attempt to apply steps (viii) and (ix). This iteration is performed repeatedly until no more preprocessing step for nodes with degree zero, one or two can be applied. Finally, step (x) is performed once, in order to remove excess modules from the input.

4.2 MIP Models

This section presents MIP models for the PC-LAN problem. These are adaptations of the models for LAN from Section 3.3. A single-commodity flow formulation (pSCF) and a cut-set formulation (pCUT) for PC-LAN are given explicitly. The modifications of the disaggregated models (MCF) and (DMCF), as well as the Benders' decompositions thereof are relatively straightforward and are not presented explicitly. The computational tests have shown that these larger models where not practically feasible for the set of large benchmark instances considered in this chapter.

The arguments about acyclic optimal solutions of the LAN problem from Section 3.2 hold equally well for the PC-LAN problem. Therefore, the same transformation into a directed problem is considered and the MIP models make use of the directed graph $G = (V, A)$.

4.2.1 Single-Commodity Flow

Design and flow variables for the PC-LAN models have the same meaning as for the LAN models. Binary variables $x_{a,m}$ denote whether the module m is installed on the arc a. Continuous flow variables $f_a \geq 0$ describe the amount of flow on arc $a \in A$. Compared to the LAN models, additional binary variables y_k are used, where $y_k = 1$ iff customer k is served and the prize p_k is collected. The *single-commodity flow* formulation of the PC-LAN problem (pSCF) is:

$$\text{(pSCF)} : \quad \min \sum_{a \in A} \sum_{m \in M_a} c_{a,m} x_{a,m} + \sum_{k \in K} c_k y_k \tag{4.1}$$

s.t.

$$\sum_{(i,j) \in \delta^+(i)} f_{(i,j)} - \sum_{(j,i) \in \delta^-(i)} f_{(j,i)} = \begin{cases} -d_i y_i, & i \in K \\ \sum_{k \in K} d_k y_k, & i = r \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in V \tag{4.2}$$

$$\sum_{k \in D} p_k y_k \geq p_0 \tag{4.3}$$

$$f_a \leq \sum_{m \in M_a} u_{a,m} x_{a,m} \quad \forall a \in A \tag{4.4}$$

$$\sum_{m \in M_a} x_{a,m} \leq 1 \quad \forall a \in A \tag{4.5}$$

$$x_{a,m} \in \{0,1\} \quad \forall a \in A,\ \forall m \in M_a \tag{4.6}$$

$$y_k \in \{0,1\} \quad \forall k \in K \tag{4.7}$$

$$0 \leq f_a \quad \forall a \in A \tag{4.8}$$

The objective function (4.1) adds up the cost for the network design and the cost for the customer selection. The flow conservation (4.2) takes into account which customers are selected. The coverage constraint (4.3) ensures that the selection meets the requirement of the target prize p_0. Capacity constraints (4.4) and disjunction constraints (4.5) are well known from the LAN models in Section 3.3.

4.2.2 Cut-Set Model

Similar to the cut-set (CUT) formulation for the LAN problem, the cut-set formulation (pCUT) for the (PC-LAN) problem ensures that the capacity entering any subset of nodes is large enough to support the total demand inside the subset.

$$\text{(pCUT)}: \qquad \min \sum_{a \in A} \sum_{m \in M_a} c_{a,m} x_{a,m} + \sum_{k \in K} c_k y_k \tag{4.9}$$

s.t.

$$\sum_{a \in \delta^-(S)} \sum_{m \in M_a} u_{a,m} x_{a,m} \geq \sum_{k \in S} y_k d_k \quad \forall S \subset V | S \cap K \neq \emptyset \text{ and } r \notin S \tag{4.10}$$

$$\sum_{k \in K} p_k y_k \geq p_0 \tag{4.11}$$

$$\sum_{m \in M_a} x_{a,m} \leq 1 \qquad \forall a \in A \tag{4.12}$$

$$x_{a,m} \in \{0,1\} \qquad \forall a \in A,\ \forall m \in M_a \tag{4.13}$$

$$y_k \in \{0,1\} \qquad \forall k \in K. \tag{4.14}$$

The *cut-set inequalities* (4.10) state that every subset of nodes S, containing at least one customer and not containing r, must have enough *incoming capacity* to route the total demand requested inside the set. All other constraints do also appear in the (pSCF) model above.

The separation of cut-set inequalities (4.10) for PC-LAN can be done in polynomial time as follows. For a given fractional solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$, we define the directed *support graph* $G' = (V', A')$ where $V' := V \cup \{t\}$ with an additional sink t, and $A' := A_1 \cup A_2$ being $A_1 := \{a \in A \mid \sum_{m \in M_a} u_{a,m} x^*_{a,m} > 0\}$ and $A_2 := \{(k,t) \mid k \in K\}$. The capacity associated to each arc $a \in A_1$ is set to $\sum_{m \in M_a} u_{a,m} x^*_{a,m}$, and the capacity of each arc $a = (k,t) \in A_2$ is set to $d_k y^*_k$. If the minimum cut between r and t in G' is less than $\sum_{k \in K} d_k y^*_k$, it defines a violated inequality (4.10).

Since $(\boldsymbol{x}, \boldsymbol{y})$ variables are binary and the coefficients are non-negative, the cut-set inequalities can be strengthened by rounding down some left-hand side coefficients (see appendix A.3) :

$$\sum_{a \in \delta^-(S)} \sum_{m \in M_a} \min\left(u_{a,m}, \sum_{k \in S} d_k\right) x_{a,m} \geq \sum_{k \in S} d_k y_k \quad \forall S \subseteq V | S \cap K \neq \emptyset \text{ and } r \notin S. \tag{4.15}$$

The pCUT model can be further strengthened with the following *connectivity cuts*. Every set of nodes containing at least one customer must have at least one incoming arc if the customer is included in the solution:

$$\sum_{a \in \delta^-(S)} \sum_{m \in M_a} x_{a,m} \geq y_k \qquad \forall S \subseteq V \setminus \{r\}, \forall k \in S \cap K. \tag{4.16}$$

The separation works similar to that of cut-set inequalities. Given a fractional solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$, we define a network from G where the capacity associated to each arc $a \in A$ is $\sum_{m \in M_a} x^*_{a,m}$. Then, if the minimum cut between r and any customer k in G is less than y^*_k this cut defines a violated inequality (4.16).

In general, a minimum cut problem has several optimal solutions. Especially when the dimension of the network is very large as is the case with the PC-LAN instances. Therefore, it is possible to find several violated inequalities from a given fractional solution. The methods described in Sections 3.7.3 and 3.7.4 can easily be transformed to produce nested and minimum cardinality cuts for the PC-LAN problem.

4.3 MIP-based Heuristic Approach

This section describes the MIP-based heuristic approach to find high quality feasible solutions to large-sized instances of PC-LAN. It consists of three main ingredients:

1. Cutting Plane phase: The cutting plane approach works with relaxations of a PC-LAN MIP model. In the *separation phase*, a new set of violated inequalities is inserted into the LP. The LP is resolved and the optimal LP-solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is taken as input for the following Network Construction phase.

2. Network Construction phase: First, a set of customers is selected according to the fractional values $\boldsymbol{y}^*$. Next, a network is constructed iteratively by using shortest path calculations on the graph with adapted edge weights. The fractional values $\boldsymbol{x}^*$ are taken into account for this construction.

3. Local Improvement phase: The solution found in the construction phase is subjected to a local improvement procedure. Flow routed along an *expensive* edge together with affected customers are removed, leaving a partial solution. Then the partial solution is repaired by adding new customers and extending the network design. Two different definitions of *expensive* are alternated.

Steps 2 and 3 are repeated in a multi-start fashion. The overall process is repeated within a branch-and-cut framework until the time limit is reached or an optimal solution is found.

Next, each of these ingredients are described in detail. First the notation that is used in this section is introduced. Then the three phases are presented in detail. Section 4.3.5 describes some modifications to these base algorithms applied in the multi-starting. Section 4.3.6 describes an alternative approach that follows a similar scheme without requiring a MIP solver. This alternative (non-MIP) approach may be of interest when one wants to solve a PC-LAN instance without using a MIP solver. Section 4.4 demonstrates that this non-MIP approach is competitive with the MIP-based approach in a few cases, but in most of the tests the MIP-based approach performs significantly better.

4.3.1 Notation

For the sake of a simpler description of the heuristic algorithm we use the following notation. A network design can be represented by a vector $\boldsymbol{z} \in \mathbb{N}^{|E|}$ consisting of module indices $z_e \in \{0\} \cup M_e$. For example, $z_e = 3$ means that the third available module for e is installed; $z_e = 0$ means that there is no installation on e. Capacities per edge are denoted by $\boldsymbol{g} \in \mathbb{R}_{\geq 0}$. A flow through the network is represented by a vector $\boldsymbol{f} \in \mathbb{R}_{\geq 0}^{|A|}$. The function $\mu_e : \mathbb{R}_{\geq 0} \mapsto M_e$ maps some required capacity to the index of the *most appropriate module* on edge e, i.e., the cheapest module with sufficient capacity, or the largest module if there is no module with sufficient capacity. More formally, for some required capacity $b \geq 0$ we define a function μ_e for every edge $e \in E$:

$$\mu_e(b) = \begin{cases} 0 & \text{if } b = 0 \\ \underset{\{m \in M_e | u_{e,m} \geq b\}}{\arg\min} \quad c_{e,m} & \text{if } b > 0 \text{ and } \exists m \in M_e |\, u_{e,m} \geq b \\ |M_e| & \text{otherwise} \end{cases}$$

An edge e is said to be *saturated* by a required capacity g_e if the largest module is already used on this edge and no free capacity is left, i.e., $z_e = |M_e|$ and $g_e = u_{e,z_e}$. Given a current capacity vector $\boldsymbol{g}$, a suitable design vector $\boldsymbol{z}$ and some additionally required capacity $b \geq 0$, we define the following *edge weight approximations*:

$$w_e\left(g_e, z_e, b\right) = \begin{cases} c_{e,\mu_e(g_e+b)} - c_{e,z_e}, & \text{if } g_e < u_{e,|M_e|} \\ \infty, & \text{otherwise} \end{cases} \qquad \forall e \in E. \tag{4.17}$$

Hence w_e represents the cost for expanding the installation on e from the currently selected module z_e to the module $\mu_e(g_e + b)$. If the required capacity g_e saturates the edge e, such an expansion is impossible and the edge weight is infinite.

4.3.2 Separation

The cutting plane approach starts with the linear programming relaxation of the CUT model without (4.10). This relaxation is strengthened with the cut-set inequalities (4.10) associated to all the singletons S. Other inequalities are generated in an iterative way as it is described below.

At each iteration a fractional solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is given. Cut-set inequalities (4.10) and connectivity cuts (4.16) are separated.

4.3.3 Network Construction

Starting from a fractional solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ we build a feasible solution by applying the following three procedures.

Rounding

Let $\boldsymbol{y}^*$ be the solution of a relaxed PC-LAN model. We sort the customer indices in order of decreasing fractional values y_k^*. We then define an integer feasible selection $\boldsymbol{y}$ by greedily setting indices of customers with large fractional values to one until the coverage constraint (4.11) is satisfied. The fractional vector $\boldsymbol{x}^*$ is used to compute a vector of minimum required capacities $\boldsymbol{g}^*$:

$$g_e^* = \sum_{m \in M_e} u_{e,m} \left(x_{ij,m}^* + x_{ji,m}^* \right) \qquad \forall e = \{i, j\}.$$

Note that $\boldsymbol{g}^*$ is not necessarily the undirected capacity vector of a feasible network design. This is true, for example, when using the (pCUT) model while not all cut-set inequalities (4.10) associated to every set S, have been separated so far.

Construction

Using the previously generated vector $\boldsymbol{y}$ and $\boldsymbol{g}^*$, this procedure constructs a feasible network design of the PC-LAN. Algorithm 4.3.1 describes the main steps. The initialization phase defines a *demand per node* $\boldsymbol{b} \in \mathbb{R}_{\geq 0}^{|V|}$ as:

$$b_k = \begin{cases} d_k y_k, & \text{if } k \in K \\ 0, & \text{otherwise.} \end{cases}$$

An initial network design $\boldsymbol{z}$ is defined via the most appropriate module per edge with respect to $\boldsymbol{g}^*$, i.e., $z_e := \mu_e(g_e^*)$ for all $e \in E$. The algorithm subsequently modifies $\boldsymbol{b}$, creates an undirected flow $\boldsymbol{g}$ and updates the design $\boldsymbol{z}$. In each iteration a node v with positive demand $b_v > 0$ is chosen. Denote the demand to be transported as $b := b_v$ and cancel the node demand of v: $b_v := 0$. The values of $\boldsymbol{g}$, $\boldsymbol{z}$ and b uniquely determine the edge weight approximation $\boldsymbol{w}$ via (4.17). This vector $\boldsymbol{w}$ defines the edge weights for the shortest paths calculation on G. A shortest path from v to r is computed: $SP_{\boldsymbol{w}}(v) = \langle v, v_1, v_2, \ldots, r \rangle$. Along this path, the current demand b is *transported.* Denote the remaining capacity on e by $\bar{u}$ and the maximum that can be transported by $\bar{b}$. The flow $\boldsymbol{g}$ is increased: $g_e := g_e + b$ and the necessary installations $z_e := \mu_e(g_e)$ are made for all $e \in SP_{\boldsymbol{w}}(v)$. Once Line 23 is reached, the node demand has been transported from v to r and the next iteration starts.

A special case occurs when an edge e on $SP_{\boldsymbol{w}}(v)$ does not offer sufficient remaining capacity $\bar{u}$, i.e., $b > \bar{u}$ in Line 18. Then, only this maximum available capacity $\bar{u}$ is transported on e. Now the node demands are changed appropriately at each endpoint of e. The node closer to v, denoted i, receives an additional demand of $b - \bar{u}$. The node closer to r, denoted j, receives an additional demand of $\bar{u}$. The edge e becomes saturated and the heuristic continues by picking the next randomly chosen node with positive node demand in Line 5. The heuristic terminates when $\boldsymbol{b} = \boldsymbol{0}$ and $\boldsymbol{z}$ is feasible for the chosen subset of customers represented by $\boldsymbol{y}$.

Of course, no shortest path may exist. This can be due to an infeasible input or due to the greedy decisions taken in the course of the algorithm. In this case the heuristic terminates in Line 11 without finding a feasible solution.

Flow Calculation

After the construction has produced a feasible solution $\boldsymbol{z}$, redundant capacities may have been installed along the edges. To reduce the installation cost, a minimum-cost flow problem is defined on the subgraph of G induced by $z_e > 0$. The flow cost are defined as $\frac{c_{e,z_e}}{u_{e,z_e}}$, and the capacity is set to u_{e,z_e} for all edges. The min-cost flow problem is solved and yields a directed flow vector $\boldsymbol{f}$. A new design vector $\boldsymbol{z}'$ can be derived from $\boldsymbol{f}$ by setting $z'_e := \mu_e(f_{ij} + f_{ji})$ for all $e \in E$. Clearly, $z'_e \leq z_e$ for all $e \in E$. The directed flow vector $\boldsymbol{f}$ also allows to express the design in terms of directed $x_{a,m}$ variables:

$$x_{ij,m} := \begin{cases} 1, & \text{if } f_{ij} > 0 \text{ and } m = \mu_{\{ij\}}(f_{ij}) \\ 0, & \text{otherwise.} \end{cases}.$$

4.3.4 Local Improvement

Given an integer feasible solution represented by the vector $(\boldsymbol{z}, \boldsymbol{y}, \boldsymbol{f})$, we attempt the following Local Improvement strategy. The main steps are given in Algorithm 4.3.2. Initialize the new solution $(\boldsymbol{z}', \boldsymbol{y}', \boldsymbol{f}')$ as $\boldsymbol{z}' := \boldsymbol{0}$, $\boldsymbol{y}' := \boldsymbol{y}$, $\boldsymbol{f}' := \boldsymbol{f}$. Decompose the flow on each arc a into commodity flows, i.e., compute a flow per customer per arc. Pick an edge $\tilde{e}$ maximizing c_{e,z_e}. Those customers k that have a positive flow on this edge $\tilde{e}$ are removed from the selection, i.e., set $y'_k := 0$. In addition, the flow for these customers is removed from $\boldsymbol{f}'$. Com-

Algorithm 4.3.1 Network Construction.

Input: customer selection $\boldsymbol{y} \in \{0,1\}^{|K|}$, minimum required capacity $\boldsymbol{g}^* \in \mathbb{R}^{|E|}_{\geq 0}$

1: init node demand $\boldsymbol{b} \in \mathbb{R}^{|V|}_{\geq 0} : b_v := \begin{cases} y_v d_v & \forall v \in K \\ 0 & \forall v \notin K \end{cases}$
2: init design $z_e := \mu_e(g_e^*)$ for all $e \in E$
3: init undirected flow $g_e := 0$ for all $e \in E$
4: **while** $\exists v \in V : b_v > 0$ **do**
5:     pick a random node $v \in V : b_v > 0$
6:     $b := b_v$ // *the demand to be transported*
7:     $b_v := 0$
8:     define edge weight $\boldsymbol{w} : w_e(g_e, z_e, b)\ \forall\ e \in E$ according to (4.17)
9:     compute a shortest path $SP_{\boldsymbol{w}}(v)$ from v to r in $\langle G, \boldsymbol{w} \rangle$
10:     **if** there is no shortest path **then**
11:         **return** failed
12:     **end if**
13:     **for** $e = (i,j) \in SP_{\boldsymbol{w}}(v) = \langle v, v_1, v_2, \ldots, r \rangle$ **do**
14:         $\bar{u} := u_{e,\mu_e(g_e+b)} - g_e$ // *remaining capacity*
15:         $\bar{b} := \min(b, \bar{u})$ // *maximum that can be transported*
16:         $g_e := g_e + \bar{b}$
17:         $z_e := \mu_e(g_e)$
18:         **if** $b > \bar{u}$ **then** // *insufficient remaining capacity*
19:             $b_i := b_i + b - \bar{u}$
20:             $b_j := b_j + \bar{u}$
21:             **goto** Line 5
22:         **end if**
23:     **end for**
24: **end while**

pute the required capacity per edge $\boldsymbol{g}'$ on behalf of the reduced flow $\boldsymbol{f}'$. Define the edge weight approximation $\boldsymbol{w}$ as described by equation (4.17). Let l_k be the length of the shortest path $SP_{\boldsymbol{w}}(k)$ from k to r for all currently unselected customers, i.e., $y'_k = 0$. Select customers with small l_k values and add them to a new set $\boldsymbol{y}''$ until the combined selection $\boldsymbol{y}' + \boldsymbol{y}''$ satisfies the coverage constraint $\sum_{k \in K} p_k(y'_k + y''_k) \geq p_0$. Now start the Network Construction (Algorithm 4.3.1) with the minimum required capacity $\boldsymbol{g}'$ and the set of newly selected customers $\boldsymbol{y}''$. The result is a new network design $\boldsymbol{z}'$. Set $\boldsymbol{y}' := \boldsymbol{y}' + \boldsymbol{y}''$. If the new solution $(\boldsymbol{z}', \boldsymbol{y}')$ has a smaller objective value than the currently best found solution $(\boldsymbol{z}, \boldsymbol{y})$, the new solution $(\boldsymbol{z}', \boldsymbol{y}')$ becomes the new best solution. Otherwise the edge that had been selected in Line 6 is added to a taboo list T.

In order to achieve more diverse results we alternate the two criteria in Line 6 and Line 21 as follows. The criterion for picking an edge with highest absolute cost c_{e,z_e} in Line 6 is modified to pick the edge with highest relative cost, i.e.,

$$\tilde{e} := \underset{e=\{i,j\} \in E, (f_{ij}+f_{ji})>0, e \notin T}{\arg\max} \left(c_{e,z_e} / f_e \right).$$

The criterion for choosing new customers with smallest shortest path lengths l_k in Line 21 is modified to choose customers minimizing the ratio of prizes over costs, i.e.,

$$\tilde{k} := \arg \max_{k \in K, y'_k + y''_k = 0} \left(\frac{p_k}{c_k + l_k} \right).$$

The two options for these two criteria give four variations of Algorithm 4.3.2, that are cyclically repeated until 20 edges have been considered for deletion without improving the objective value.

4.3.5 Multi-Start Modifications

As stated in Section 4.3, the Network Construction phase followed by the Local Improvement Phase is repeated in a multi-start fashion. To get a wide variation in the solutions during multi-starting the following modifications to the base algorithms provided in Sections 4.3.3 and 4.3.4 are implemented.

For some PC-LAN instances it is advantageous to send the flow to groups of customers along the same path. If economies of scale are given, as is frequently the case with this type of network design problems, larger modules have a smaller relative cost $u_{e,m}/c_{e,m}$ than smaller modules. In situations like this the Network Construction heuristic described in Section 4.3.3 should be modified to first *cluster* some neighboring demands and second search for a routing to the access point for the combined demands. On the other hand, sometimes the opposite is true: Economies of scale are not given. For exam-

Algorithm 4.3.2 Local Improvement.

Input: design $\boldsymbol{z} \in \mathbb{N}^{|E|}$, customer selection $\boldsymbol{y} \in \{0,1\}^{|K|}$, flow $\boldsymbol{f} \in \mathbb{R}_{\geq 0}^{|A|}$

1: $s := 0$ // *improvement counter*
2: $T = \varnothing$ // *taboo list*
3: **while** $s < 20$ **do**
4: $\quad \boldsymbol{z}' := \boldsymbol{0}, \boldsymbol{y}' := \boldsymbol{y}, \boldsymbol{f}' := \boldsymbol{f}$
5: $\quad$ compute $f_a^k \in \mathbb{R}_{\geq 0}^{|A| \times |K|}$ s.t. $\sum_{k \in K} f_a^k = f_a \; \forall a \in A$ // *flow decomposition*
6: $\quad \tilde{e} := \arg\max_{e=\{i,j\} \in E, (f_{ij}+f_{ji})>0, e \notin T} (c_{e,z_e})$ // *pick edge* $\tilde{e}$
7: $\quad$ **for all** $k \in K$ with $f_{(ij)}^k > 0$ or $f_{(ji)}^k > 0$ on $\tilde{e} = \{i,j\}$ **do** // *reduction*
8: $\qquad y'_k := 0$
9: $\qquad f'_a := f'_a - f_a^k \; \forall a \in A$
10: $\quad$ **end for**
11: $\quad \boldsymbol{g}' \in \mathbb{R}_{\geq 0}^{|E|} := \boldsymbol{0}$ // *required capacity*
12: $\quad$ **for all** $e = \{i,j\} \in E$ **do**
13: $\qquad g'_e := f'_{ij} + f'_{ji}$
14: $\quad$ **end for**
15: $\quad \boldsymbol{y}'' := \boldsymbol{0}$ // *additional customers*
16: $\quad$ **for all** $k \in K$ with $y'_k + y''_k = 0$ **do** // *compute shortest path lengths*
17: $\qquad$ define edge weight $\boldsymbol{w} : w_e(g'_e, z'_e, d_k) \forall \, e \in E$ according to (4.17)
18: $\qquad$ compute the shortest path $SP_{\boldsymbol{w}}(k)$ and denote the length by l_k
19: $\quad$ **end for**
20: $\quad$ **while** $\sum_{k \in K} p_k(y'_k + y''_k) < p_0$ **do**
21: $\qquad \tilde{k} := \arg\min_{k \in K, y'_k + y''_k = 0} l_k$
22: $\qquad y''_{\tilde{k}} := 1$
23: $\quad$ **end while**
24: $\quad (\boldsymbol{z}', \boldsymbol{f}') :=$ Network Construction $(\boldsymbol{g}', \boldsymbol{y}'')$
25: $\quad \boldsymbol{y}' := \boldsymbol{y}' + \boldsymbol{y}''$
26: $\quad$ **if** $\sum_{e \in E} c_{e,z'_e} + \sum_{k \in K} c_k(y'_k) < \sum_{e \in E} c_{e,z_e} + \sum_{k \in K} c_k(y_k)$ **then**
27: $\qquad \boldsymbol{z} := \boldsymbol{z}', \boldsymbol{y} := \boldsymbol{y}', \boldsymbol{f} := \boldsymbol{f}'$ // *keep new best solution*
28: $\qquad i = 0$
29: $\quad$ **else** // *no improvement*
30: $\qquad i := i + 1$
31: $\qquad T := T \cup \{\tilde{e}\}$ // *the edge from Line 6 becomes taboo*
32: $\quad$ **end if**
33: **end while**
34: **return** $(\boldsymbol{z}, \boldsymbol{y}, \boldsymbol{f})$

ple, when smaller modules represent existing infrastructure and larger modules represent new connections that involve a high setup cost. For these problems it can be crucial to facilitate the existing infrastructure as much as possible and avoid the larger modules. Under these circumstances the heuristic should do the opposite of routing clustered demands together. Instead, it would be better to split the given demands apart and route the partial demands individually in order to facilitate the existing infrastructure in an optimal way.

In order to accommodate for these two contradicting ideas, we employ several variants of the basic algorithm from Section 4.3.3. To enable a clustering of demands, the loop in Lines 13-23 of Algorithm 4.3.1 is changed so that the installation is not necessarily done along the whole path right from v to the central office r, but instead stops at some earlier node j. Two criteria are used to select j: (i) j is the first node with a positive demand $b_j > 0$ encountered along the path, or (ii) j is at most q edges away from v. Observe that the demands are clustered if criterion (i) is applied and the parameter q is set to a small number. To implement this variant, these next instructions are inserted between Lines 22 and 23:

> **if** criterion (i) or (ii) **then**
>
> $b_j := b_j + b$
>
>   **goto** Line 5
>
> **end if**.

The idea of this clustering is to merge customers that are *close* to each other with respect to the stepwise edge cost function. To provide an *anti-clustering* variant of the algorithm, two additional modifications of the algorithm are introduced. The first is a redefinition of the node demands. Instead of one number b_v per node v, we use a list of sub-demands $B_v = \{b_{v,1}, b_{v,2}, \dots\}$ for every node that can be treated independently. The initialization in Line 1 changes to

$$B_k := \{y_k d_k\} \text{ for all } k \in K$$
$$B_v := \varnothing \text{ for all } v \in V \setminus K.$$

In Line 5, *one* of the sub-demands $b_{v,t}$ of a node v with at least one positive sub-demand is chosen and Lines 6-7 become

$$b := b_{v,t}$$
$$B_v := B_v \setminus \{b_{v,t}\}.$$

The update of the node demands in case of insufficient remaining capacity in Lines 19-20 becomes:

$$B_i := B_i \cup \{b - \bar{u}\},$$
$$B_j := B_j \cup \{\bar{u}\}.$$

Whereas the update of node demands in case of criterion (i) or (ii), introduced above, becomes:

> **if** criterion (i) or (ii) **then**
> > $B_j := B_j \cup \{b\}$;
> > **goto** Line 5
>
> **end if**.

The second modification to help with anti-clustering is to initially split the demands in two $B_k = \{y_k d_k/2, y_k d_k/2\}$ or three $B_k = \{y_k d_k/3, y_k d_k/3, y_k d_k/3\}$ partial demands in Line 1.
A specific variant of the Network Construction algorithm can be chosen with four parameters:

- Activate criterion (i), or do not activate it.
- Select a value for $q \in \{1, \ldots, |V|\}$ for criterion (ii).
- Join node demands by using *one* value b_v per node, or do not join but use a list of values B_v.
- Select a splitting ratio $\in \{1, 2, 3\}$ for the initial definition of node demands.

Each time the Network Construction algorithm is executed, a specific variant is chosen by the means of a learning adaptation mechanism known as *Reactive Search Optimization*. Initially pre-specified settings for the four parameters are used. Then the settings for the parameters are varied from a diversification of the settings towards an intensification. That is, from randomly perturbed settings towards settings that have produced the best objective values so far.

4.3.6 Non-MIP variant of the heuristic

To measure the impact of the MIP information in the heuristic approach, an alternative heuristic is used which does not make use of a MIP solver. This variant is also of interest for practical purposes since a company may desire not to purchase and install a black-box MIP solver in order to heuristically solve instances. The non-MIP variant works as follows.

1. Compute a selection of customers $\boldsymbol{y}$ that satisfies the coverage constraint (4.11). This is done similarly to the customer selection in the Local Improvement Algorithm 4.3.2, Lines 15-23. Since there are no currently selected customers, this set is empty, $\boldsymbol{y}' = \mathbf{0}$. Also, initially there is no current installation, i.e., $\boldsymbol{z}' = \mathbf{0}$ and no minimum required capacity, i.e., $\boldsymbol{g}' = \mathbf{0}$. Now $\boldsymbol{y} := \boldsymbol{y}''$ denotes a customer selection, feasible with respect to inequality (4.11).

2. Apply the Network Construction phase from Section 4.3.3 on $\boldsymbol{y}$ and no initial required capacity, i.e., $\boldsymbol{g}^* = \mathbf{0}$.

These two steps compensate for the missing fractional solution in the algorithmic framework.

4.4 Computational Study

This section discusses the performance of the approaches described in Section 4.3 on a new set of instances arising from the real-world motivation of our research. The computations are performed on a computer with Intel Xeon 2.6 GHz and 3 GB RAM. Cplex 12.2 was used to solve the linear programs, and as a MIP framework.

4.4.1 Large Real-World Instances

It has been said in the introduction, that this work was inspired by a joint project with Telekom Austria. In the second phase of this project the prize-collecting aspect emerged. In addition the second phase surfaced three new inputs targeted for FTTH/FTTB planning. These instances are more challenging than the two sets used in the previous chapter (see 3.10.2-3.10.3) in two aspects. Firstly, the graphs are much larger. This naturally increases the search space and imposes a problem with respect to computer memory when implementing solution methods. Especially, sophisticated mixed integer programming methods will likely have difficulties loading instances of this size and solving linear relaxations, let alone applying branch-and-bound methods. Secondly, the stepwise cost functions on the edges are very inhomogeneous. Some have only a small number of steps, some a large number. Some functions exhibit economies of scale, some do not. This results from the fact that these instances are more realistic in the sense that they contain actually existing infrastructure and consider more cable technologies. The instances are based on real-world data we received from the telecom company.

The other features of the PC-LAN instances are generated following the procedure described here. We are given three types of nodes: physical locations of customers, location of the central office and locations of intermediate nodes. For each customer location, we are given the number of subscribers associated to this location. Usually, several splitter devices with various splitting ratios (e.g., 1:4, 1:16, 1:32) are available. Their costs obey economies of scale. For example, to connect 16 subscribers, a device must be installed that costs 2 000 € and one optical fiber should come in that building. To connect a building with 17 subscribers, a device that costs 3 000 € and 2 fibers are needed and this larger device is sufficient to support up to 32 subscribers. It is not feasible to connect only a fraction of subscribers at the customer node. Instead, decisions have to be made whether all subscribers per customer node or none of them are going to be served. This allows to pre-compute the demands and set-up costs for each customer location based on the number of subscribers.

These instances consider three types of links: existing fibers, existing ducts and public streets. Existing and currently unused optical fiber cables can be used with a very small cost. In existing ducts, a limited number of additional cables can be installed at relatively little cost. Along street segments, new trenches can be built and new ducts and cables can be laid. In addition to the cost for the ducts and cables, there is a significant overhead cost for new trenches. Different cable technologies are available. They differ in terms of the number of fibers per cable and cost per meter. Existing fiber and existing ducts can be used simultaneously. If new trenches are dug, any existing infrastructure is removed and replaced by the new installations. Taking these aspects into account, the available modules for each edge can be pre-computed. Table 4.1 lists the number of nodes, edges, customers and minimum, average and maximum number of modules per edge for the three instances: A, B and C.

	$\|V\|$	$\|E\|$	$\|K\|$	$\|\underline{M}\|$	$\|M\|$	$\|\overline{M}\|$
A	86 745	116 750	1157	3	9.0	131
B	48 247	65 304	720	3	9.0	84
C	77 329	107 696	1498	3	9.9	161

Table 4.1: The three original real-world instances. Columns $|\underline{M}|, |M|$ and $|\overline{M}|$ give the minimum, average and maximum number of modules over all edges.

	$\|V\|$	$\|E\|$	$\|K\|$	$\|\underline{M}\|$	$\|M\|$	$\|\overline{M}\|$
A	44 821	73 382	1157	3	9.1	131
B	25 600	41 926	717	3	9.1	84
C	44 542	73 483	1497	3	10.1	161

Table 4.2: The three instances after preprocessing.

This shows the high diversity in the input data. Table 4.2 shows the size of the inputs after preprocessing. The number of nodes is reduced by almost a factor of 2. Seven different values of $\alpha \in \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ are considered. Consequently, the set of benchmark instances contains 21 PC-LAN problem instances.

4.4.2 Separation Settings

Two families of inequalities (4.10),(4.16) are separated. Consequently, different separation strategies are possible and may produce different performance in the overall approach. In preliminary experiments, different configurations were tested and evaluated. The best results were achieved by alternating two configurations: In configuration (I), connectivity cuts are favored, and to this end at most two nested cut-set inequalities (4.15) and at most 2 000 nested connectivity cuts (4.16) are separated. In the second configuration (II), at most 20 nested cut-set inequalities (4.15) are separated and no connectivity cuts are generated. We apply configuration (I) iteratively until no more violated inequalities are found, or until the improvement of the objective value in the last ten iterations is too small. Then we apply configuration (II) iteratively until no more violated inequalities are found, or until the improvement is too small.

Let o be the current objective value, let o_{I} be the objective value derived ten iterations ago with configuration (I), and let o_{II} be the objective value derived ten iterations ago with configuration (II). The relative improvement for configuration (I) is said to be *too small* once $(o - o_{\text{I}})/o$ drops below $\epsilon = 10^{-4}$. Note that $(o - o_{\text{I}})/o$ may again become greater than ϵ while configuration (II) is active. Thus the algorithm may switch back to configuration (I) and vice-versa. Once no connectivity cuts, nor cut-set inequalities exist, or both values $(o - o_{\text{I}})/o$ and $(o - o_{\text{II}})/o$ are below ϵ, the algorithm resorts to branching.

4.4.3 Results

This section shows results achieved with the MIP-based approach described in Section 4.3 and the non-MIP variant from Section 4.3.6. A time limit of 10 hours is set for both approaches. The MIP-based approach applies only the Cutting Plane phase in the first 2 hours. In the remaining 8 hours, every Cutting Plane phase is followed by multi-starting Network Construction, followed by Local Improvement as long as a better solution is produced. Inside the Local Improvement, the removal of different edges is repeatedly tried, until 20 recent attempts did not improve the solution. The non-MIP approach multi-starts until the time limit of 10 hours is reached. Inside the Local Improvement the iteration continues until the solution has not improved in the 200 recent attempts.

Table 4.3 compares the performance of the MIP heuristic and the non-MIP variant. For the three instances (A, B and C) and for each coverage rate the following results are reported:

- # is the instance character.
- α is the coverage rate.

#	α	LB	UB	gap	Gap_{non}	Gap_{MIP}	$\|V^*\|$	$\|V_1^*\|$
A	0.4	894 318	1103 084	18.93	9.10	0.00	1 712	14
B	0.4	522 753	568 518	8.05	13.43	0.00	1 130	15
C	0.4	1005 973	1106 272	9.07	16.02	0.00	2 968	26
A	0.5	1245 096	1558 613	20.12	6.27	0.00	2 462	11
B	0.5	715 968	778 720	8.06	11.75	0.00	1 288	9
C	0.5	1383 417	1565 520	11.63	18.94	0.00	3 446	46
A	0.6	1617 097	2064 569	21.67	4.99	0.00	3 098	47
B	0.6	938 149	1003 232	6.49	12.84	0.00	1 654	15
C	0.6	1844 854	2249 367	17.98	6.53	0.00	4 089	55
A	0.7	2032 880	2699 669	24.70	2.30	0.00	4 196	51
B	0.7	1228 323	1322 599	7.13	7.76	0.00	1 932	19
C	0.7	2349 758	3018 622	22.16	2.73	0.00	5 691	90
A	0.8	2599 170	3433 859	24.31	0.00	0.57	5 454	108
B	0.8	1601 173	1771 221	9.60	3.72	0.00	2 521	26
C	0.8	3011 135	3927 335	23.33	0.00	1.14	7 047	100
A	0.9	3400 201	4386 960	22.49	0.00	1.60	6 776	19
B	0.9	2126 598	2349 981	9.51	2.63	0.00	3 206	31
C	0.9	4016 022	5180 962	22.49	0.00	1.17	8 552	97
A	1.0	7188 015	8584 895	16.27	0.00	3.01	9 970	18
B	1.0	3463 753	3916 245	11.55	0.00	1.08	5 286	26
C	1.0	6278 802	7384 655	14.98	0.00	2.40	11 607	112

Table 4.3: Results of the MIP-based heuristic versus the non-MIP variant.

- LB gives the lower bound obtained while running the MIP-based heuristic.
- UB gives the best upper bound obtained by the MIP-based heuristic or the non-MIP variant.
- *gap* shows the optimality gap $(UB - LB)/UB$ in percent.
- Gap_{MIP} denotes the relative distance $(UB_{\text{MIP}} - UB)/UB$ in percent, where UB_{MIP} is the upper bound obtained by the MIP-based heuristic.
- Gap_{non} denotes the relative distance $(UB_{\text{non}} - UB)/UB$ in percent, where UB_{non} is the upper bound obtained by the non-MIP variant.
- $|V^*|$ gives the number of nodes in the best solution with value UB.
- $|V_1^*|$ gives the number of nodes in the best solution with in-degree greater than 1, where edges are understood to be oriented in the direction of a flow from r to the customers.

A value of 0.00 in Gap_{MIP} and Gap_{non} implies that the corresponding heuristic approach found the best upper bound. A value of 0 in column $|V_1^*|$ would imply that the solution is a tree, thus this column gives a measure of *deviation from tree*.

For none of the 21 instances the processing of the root node of the branch-and-bound tree was completed. In all cases the time limit of 10 hours was reached before the root note could be finished. Table 4.3 shows that the MIP-based heuristic found the best solution in 14 out of the 21 instances. The average value of Gap_{MIP} is 0.52 while the average value of Gap_{non} is 5.67. So on average the MIP-based heuristic is closer to the best found solution than the non-MIP variant. And also the largest advancement of MIP over non-MIP is more pronounced than the other way around. The largest advancement of the MIP aproach over the non-MIP approach, seen for C with $\alpha = 0.5$ is $|Gap_{\text{non}} - Gap_{\text{MIP}}| = 18.94$. While the largest improvement of the non-MIP approach over the MIP-approach, seen for B with $\alpha = 1.0$ is 3.01. The MIP-based heuristic is clearly better on instances with smaller values of α, which can be seen from the relatively larger distances of Gap_{non}. On pure LAN instances ($\alpha = 1.0$), the multi-start scheme produces better results when the MIP ingredient is not used.

Figure 4.2 illustrates the performance of the two approaches on the instance B with $\alpha = 0.7$, where the MIP-based heuristic ends with a better solution than the non-MIP variant. MIP Heuristic - Upper Bound and Lower Bound show a value in every iteration of the MIP approach. Non-MIP - Upper Bound shows a value every time an improved solution is found. A value of 100 for the relative objective value in the figure corresponds to $Gap_{\text{MIP}} = 0.0$ in Table 4.3. The time is represented in hours. Note that the MIP approach starts with

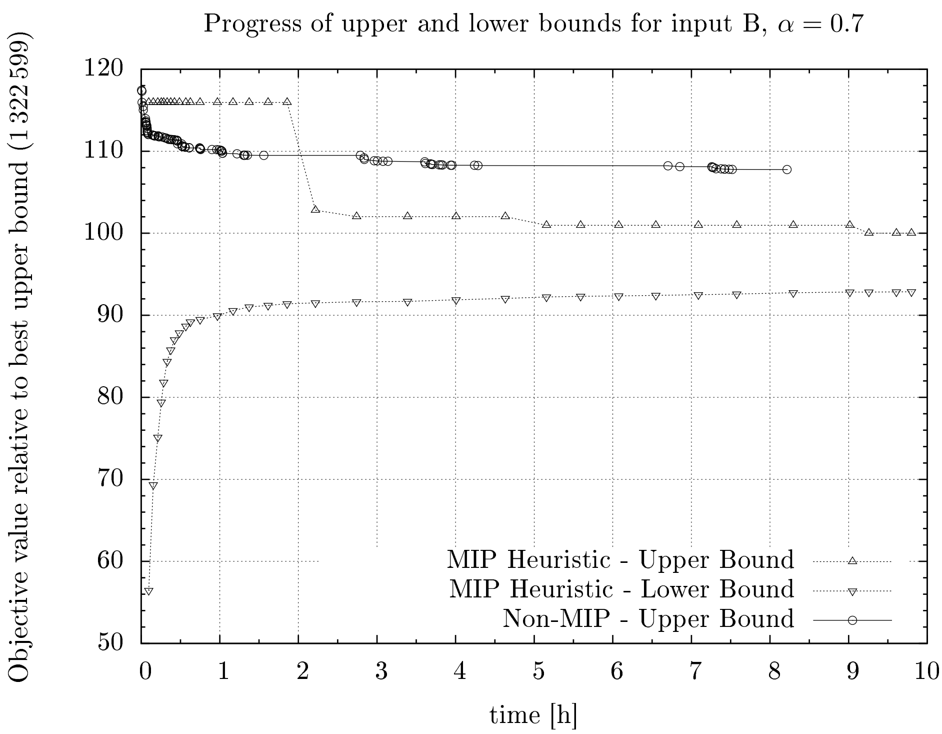


Figure 4.2: Instance B with $\alpha = 0.7$, where the MIP-based heuristic performs better than the non-MIP variant.

an initial heuristic solution computed similarly to the non-MIP variant. This initial heuristic multi-starts as long as the solution gets better in every iteration. The next 20% of the runtime, i.e., the next two hours are spent, iterating in the Cutting Plane phase to build up a set of cutting planes before Network Construction and Local Improvement algorithms are executed. We observe that the non-MIP variant slowly improves the quality of the solution and in the last almost two hours there is no more improvement. In contrast to this, the MIP approach finds a significantly better feasible solution on the first execution of Network Construction and Local Improvement after 2 hours. Furthermore, the MIP approach slowly improves the quality of upper and lower bounds providing the final gap of 7.13% between the lower and the upper bound.

Figure 4.3 illustrates the performance on the instance B with coverage rate 1.0, where the non-MIP variant ends with a better solution than the MIP-based heuristic. A value of 100 for the relative objective value in the figure corresponds to $Gap_{\mathrm{non}} = 0.0$ in Table 4.3. We observe that the non-MIP variant, again, slowly improves the quality of the solution. On the other hand, the MIP approach does not seem to draw a noticeable advantage of the information from the fractional solutions. The MIP approach behaves similar to the non-MIP variant with the exception of investing much of the runtime in cutting planes and LP solutions. Thus it does not quite achieve the same solution quality as the non-MIP variant.

Table 4.4 presents further results on the MIP-based heuristic on the 21 instances. The meaning of the columns is the following:

- Gap'_{MIP} denotes the gap of the initial solution before solving the first linear relaxation. As in the previous table, this gap has been computed with respect to the best upper bound UB, and it is a percentage.
- t'_{MIP} is the time (in seconds) to produce the initial solution.
- t_{MIP} is the total time (in seconds) of the MIP-based heuristic minus the time consumed to compute the lower bound (i.e., separation phase and MIP solver).
- n_{MIP} gives the number of improved solutions found during the MIP-based heuristic.
- n_{LP} gives the number of iterations of the cutting-plane algorithm, i.e., the number of LP solutions.
- $n_{(4.15)}$ shows the number of generated rounded cut-set inequalities (4.15).
- $n_{(4.16)}$ shows the number of generated connectivity cuts (4.16).
- $n'_{(4.15)}$ shows the average number of generated rounded cut-set inequalities (4.15) per fractional solution in the second half of the iterations of the cutting-plane algorithm.

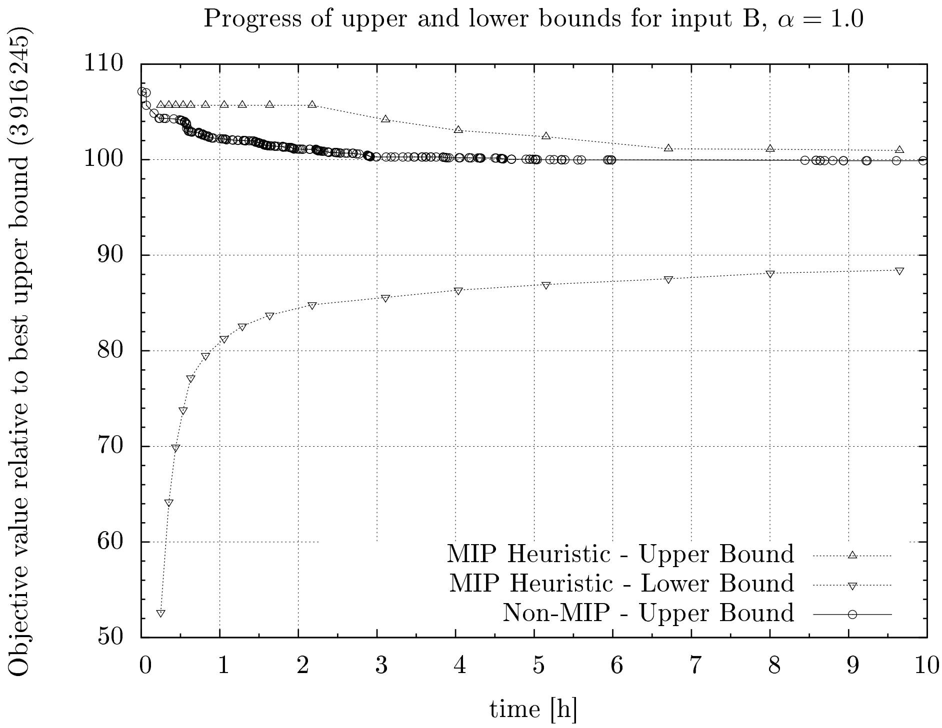


Figure 4.3: Instance B with $\alpha = 1.0$, where the non-MIP variant performs better than the MIP-based heuristic.

#	α	Gap_{MIP}	Gap'_{MIP}	t'_{MIP}	t_{MIP}	n_{MIP}	n_{LP}	$n_{(4.15)}$	$n_{(4.16)}$	$n'_{(4.15)}$	$n'_{(4.16)}$	$t_{(4.15)}$	$t_{(4.16)}$
A	0.4	0.00	11.73	68	9 411	4	17	34	10 074	2.0	492.3	99	5 999
B	0.4	0.00	19.24	15	18 477	6	125	790	4 419	6.1	9.1	918	1 603
C	0.4	0.00	24.31	459	23 623	3	37	74	8 738	2.0	43.3	447	4 886
A	0.5	0.00	10.74	23	8 938	4	17	34	11 891	2.0	595.5	74	5 102
B	0.5	0.00	18.03	86	18 818	4	89	394	5 601	3.6	13.0	421	2 022
C	0.5	0.00	26.13	455	23 863	8	24	48	9 287	2.0	105.0	423	5 953
A	0.6	0.00	9.07	142	12 188	6	15	30	11 957	2.0	658.4	79	6 933
B	0.6	0.00	22.62	30	16 329	10	61	302	6 290	8.0	8.1	485	2 318
C	0.6	0.00	13.04	426	21 522	8	20	40	12 010	2.0	271.7	291	7 059
A	0.7	0.00	7.67	303	14 365	6	13	26	12 601	2.0	826.3	84	7 316
B	0.7	0.00	15.95	91	12 675	5	33	84	7 692	3.1	87.6	110	2 994
C	0.7	0.00	8.91	1523	22 666	7	14	28	11 288	2.0	476.3	270	8 598
A	0.8	0.57	4.50	39	12 915	5	13	26	14 116	2.0	943.3	81	7 640
B	0.8	0.00	9.82	171	14 659	5	25	50	9 835	2.0	214.4	59	3 461
C	0.8	1.14	6.56	1324	20 981	8	14	28	13 683	2.0	666.5	253	10 102
A	0.9	1.60	1.77	659	8 370	2	12	24	15 576	2.0	1159.6	52	5 724
B	0.9	0.00	10.27	186	19 103	7	23	46	10 623	2.0	256.2	51	3 576
C	0.9	1.17	5.57	844	15 655	6	15	30	16 787	2.0	825.9	153	7 992
A	1.0	3.01	3.01	381	8 421	1	8	16	13 608	2.0	1514.0	15	6 204
B	1.0	1.08	5.69	461	13 909	7	16	32	12 733	2.0	553.7	17	4 120
C	1.0	2.40	3.90	3157	17 509	5	12	24	18 288	2.0	1173.4	60	8 981

Table 4.4: Details of the MIP-based heuristic.

- $n'_{(4.16)}$ shows the average number of generated connectivity cuts (4.16) per fractional solution in the second half of the iterations of the cutting-plane algorithm.
- $t_{(4.15)}$ shows the time consumed in separating rounded cut-set inequalities (4.15).
- $t_{(4.16)}$ shows the time consumed in separating connectivity cuts (4.16).

From this table it can bee seen that around half of the total time of 10 hours (36 000 seconds) in the MIP-based heuristic is consumed by the cutting-plane procedure. This involves solving linear programming relaxations and separation of inequalities. However, as previously observed, this time consumption increases the solution quality when the coverage rate α is small. The objective value of the initial heuristic solution computed before the first iteration of the cutting-plane procedure is similar to the objective value of the best solution when α is large. In particular, we even observe that on instance A with 1.0 of coverage rate, the initial solution available in the first 5 minutes of the computation was not improved during the whole 10 hours of the MIP-based heuristic. The situation is different when the coverage rate is small. Note that the number of generated inequalities of each family (4.15) and (4.16) is strongly affected by the separation settings described in Section 4.3.2. Different settings to try to find more rounded cut-set inequalities were also tested, but the overall performance of this approach was not better. As shown in the table, on average two rounded cut-set inequalities were generated from each fractional solution. This means that configuration (I) was executed once on each cutting-plane iteration. Columns $n'(4.15)$ and $n'(4.16)$ indicate the number of inequalities generated at the end of the cutting-plane process. For example, in the last 30 cutting-plane iterations only an average of 8.1 connectivity cuts are generated when solving instance B with 0.6 coverage rate.

4.5 Conclusion

A MIP-based heuristic is proposed to solve a new network design problem arising in a telecommunication context where not all customers need to be served. Instead, the company is interested in finding a good feasible solution to connect a given percentage of the customers. The problem is called *prize-collecting Local Access Network* design problem (PC-LAN) and to our knowledge this is the first method to approach it. The MIP-based heuristic separates two families of inequalities to produce fractional solutions that are used to create feasible solutions. A local search approach is used to improve each solution, and the entire approach is embedded in a multi-start framework. To measure the advantage of using fractional solution in the heuristic approach, a non-MIP multi-start variant is implemented. The two approaches are evaluated on a set of 21 instances generated from real-world data.

The experiments show that the MIP-based approach significantly outperforms the non-MIP variant for coverage rates below 80% ($\alpha = 0.8$). These coverage rates are typical for real-world applications motivating this research.

Chapter 5

Conclusions

This thesis focuses on abstract models for the *fiber to the home* (FTTH) problem. FTTH is about laying fiber optic connections on the *last mile* of telecommunication networks in order to achieve higher bandwidth. The thesis consist mainly of two parts covered in Chapters 3 and 4.

Chapter 3 suggests the *local access network* design problem (LAN) as an abstraction to model the FTTH problem. Here, we search for a minimum-cost subgraph of the input in order to support the demands of all customers. The LAN problem covers arbitrary step-cost functions for possible connections. These functions are convenient to model the fact that there are multiple available cable technologies and also takes into account that there may already be some existing infrastructure that can be facilitated in the new fiber optic network. Furthermore, it is possible to model step-cost functions that do not exhibit economies of scale.

To solve large real world instances of the LAN problem, various solution methods are proposed. Firstly, preprocessing methods that help to reduce the size of the input are described. Together with a description to map a solution of the preprocessed input back onto a solution of the original input, this is an effective tool that can reduce the size of some of the tested networks by a factor of 2. Secondly, different MIP models are described and the strength of these models with respect to their polyhedral inclusion is discussed. In order to tackle the strong but large disaggregated models, Benders' decomposition is applied. The subproblems of the Benders' decompositions are subjected to different normalizations and the practical effects of these normalizations are studied. Thirdly, a rounding heuristic is described that produces feasible solutions from LP relaxations.

These methods have been implemented and tested empirically on LAN instances from the literature and on a new set of large instances with more than

1 000 nodes, 1 000 edges and 67 customers.

Chapter 4 forms the second part of this thesis. The *prize-collecting local access network* design problem (PC-LAN) is proposed to cover a new aspect in FTTH planning. Here, the problem is extended with an additional aspect where not all customers have to be connected. Instead a *prize* is associated with each customer and we search for a solution that balances the cost for the network design and the prizes collected by connecting customers.

In order to solve the PC-LAN problem, the methods presented for the LAN problem have been adapted. In addition, a new MIP-based heuristic for PC-LAN is proposed. This includes a branch-and-bound scheme tailored for very large instances, an adaptable construction heuristic and a local improvement step. These phases are repeated in a multi-start algorithm with a learning mechanism to adapt the parameters of the heuristic.

The implementation of the proposed methods is compared to a simpler version of the heuristic without the MIP component. The tests are performed on a new set of very large PC-LAN instances with more than 80 000 nodes, 100 000 edges and almost 1 500 customers. The results show the effectiveness of the MIP-heuristic for the tested instances.

Appendix A

Appendix

This appendix repeats some basic theory from linear programming and combinatorial optimization. Appendix A.1 gives some basic results from linear programming theory. Appendix A.2 explains Benders' Decomposition and describes a cutting plane generation scheme to compute the Benders' Decomposition in practice. Appendix A.3 shows a method to strengthen inequalities of integer programs by rounding. Lastly, appendix A.4 repeats the min-cut max-flow theorem that is useful in combinatorial optimization.

A.1 Basic Linear Programming

This section recapitulates some fundamental definitions and some basic results without proofs, mostly taken from [NW88].

Consider a matrix $A \in \mathbb{R}^{m \times n}$, a vector $b \in \mathbb{R}^m$ and the polyhedron $Q = \{x \in \mathbb{R}^n \mid Ax \geq b\}$. An element $p \in Q$ is called an *extreme point* of Q if there do not exist $x_1 \neq x_2 \in Q$ such that $p = \frac{1}{2}x_1 + \frac{1}{2}x_2$. Denote the set of all extreme points of Q by $P_Q = \{p_k \mid k \in K\}$. This set is finite. Define $Q^0 = \{x \in \mathbb{R}^n \mid Ax \geq 0\}$. If $Q \neq \varnothing$, an element $r \in Q^0 \setminus \{0\}$ is called a *ray* of Q. If there do not exist $x_1, x_2 \in Q^0$ with $x_1 \neq \lambda x_2$ for any $\lambda \in \mathbb{R}$ such that $r = \frac{1}{2}x_1 + \frac{1}{2}x_2$, the ray r is called an *extreme ray* of Q. Denote the set of all extreme rays of Q by $R_Q = \{r_j | j \in J\}$. This set is finite.

Minkowski's theorem states that a polyhedron $Q = \{x \in \mathbb{R}^n \mid Ax \geq b\} \neq \varnothing$ can be written in terms of the extreme points $\{p_k \mid k \in K\}$ and extreme rays $\{r_j \mid j \in J\}$ of Q.

$$Q = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{aligned} & x = \sum_{k \in K} \lambda_k p_k + \sum_{j \in J} \mu_j r_j, \\ & \sum_{k \in K} \lambda_k = 1, \\ & \lambda_k \geq 0 \; \forall k \in K, \\ & \mu_j \geq 0 \; \forall j \in J \end{aligned} \right\}$$

For a given linear program (P) : $z = \min \left\{ c^T x \mid Ax \geq b, x \in \mathbb{R}^n_{\geq 0} \right\}$ we can write its *dual program* (D) as $w = \max \left\{ u^T b \mid A^T u \leq c, u \in \mathbb{R}^m_{\geq 0} \right\}$. The former program (P) is called the *primal*. Regarding feasibility and boundedness of such a pair of primal and dual programs, only the four combinations given in Table A.1 are possible.

	primal (P)	dual (D)
(i)	feasible, bounded	feasible, bounded
(ii)	feasible, unbounded	infeasible
(iii)	infeasible	feasible unbounded
(iv)	infeasible	infeasible

Table A.1: Possible combinations of feasibility and boundedness of a pair of primal and dual programs.

If x is a feasible solution of the primal problem and u is a feasible solution of the dual problem, it can be proved that $c^T x \geq z \geq w \geq u^T b$. This property is called *weak duality*. In case (i) of Table A.1, there exist optimal solutions x^* for the primal problem and u^* for the dual problem. *Strong duality* implies that in case (i) both solutions are attained and the duality gap is zero, i.e., $c^T x^* = z = w = u^{*T} c$.

Farkas' lemma states that a system of linear inequalities $Ax \geq b$ is feasible if and only if $u^T b \leq 0$ for any $u \in \mathbb{R}^m_{\geq 0}, A^T u \leq 0$. By Minkowski's theorem, this can equivalently be written in terms of extreme rays: $Ax \geq b$ is feasible if and only if, $r_j^T b \leq 0$ for any extreme ray $r_j \in R_{\left\{ r \in \mathbb{R}^m_{\geq 0} \mid A^T r \leq 0 \right\}}$.

A.2 Benders' Decomposition

J. F. Benders [Ben62] has presented a partitioning procedure for programming problems with mixed-variables of this type

$$\min \left\{ c^T x + f(y) \;\middle|\; Ax + F(y) \geq b, x \in \mathbb{R}^n, y \in S \right\}. \tag{A.1}$$

In this original definition from Benders', one set of variables $x \in \mathbb{R}^n$ are real valued, whereas the other variables $y \in S \subseteq \mathbb{R}^p$ are from some arbitrary subset of $\mathbb{R}^p$. Here $A \in \mathbb{R}^{m \times n}$, $f : S \mapsto \mathbb{R}$, $F : S \mapsto \mathbb{R}^m$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$.

This procedure replaces the problem (A.1) by a linear *master problem* on $\mathbb{R}^n$ and a series of *subproblems* defined on S. In an iterative fashion subproblems are solved and additional constraints are added to the master problem. In a finite number of iterations this leads to an optimal solution of the original problem. This procedure is nowadays known as the *Benders' decomposition*.

One example in Benders' work is for mixed integer programs where $S \subseteq \mathbb{Z}^p$. The application of Benders' decomposition in this thesis is to linear programs only. Specifically, it is applied to LP relaxations of mixed integer programs. Therefore, the remainder of this section will present Benders' decomposition for linear programs. Here, the decision variables are partitioned into two sets, both taking values in a real vector space.

Consider the following linear programming problem, denoted by (P):

$$\begin{aligned} z = \min c^T x + h^T y \\ \text{s.t.} \quad Ax + Gy &\geq b \\ x &\in \mathbb{R}^n_{\geq 0} \\ y &\in \mathbb{R}^p_{\geq 0}. \end{aligned} \tag{A.2}$$

For a fixed x, we have the *primal subproblem* $\mathrm{P}(x)$:

$$\begin{aligned} z_{\mathrm{P}(x)} = \min h^T y \\ \text{s.t.} \quad Gy &\geq b - Ax \\ y &\in \mathbb{R}^p_{\geq 0} \end{aligned} \tag{A.3}$$

The dual of $\mathrm{P}(x)$ is the *dual subproblem* $\mathrm{D}(x)$:

$$\begin{aligned} z_{\mathrm{D}(x)} = \max u^T (b - Ax) \\ \text{s.t.} \quad G^T u &\leq h \\ u &\in \mathbb{R}^m_{\geq 0} \end{aligned} \tag{A.4}$$

Denote the feasible region of $\mathrm{D}(x)$ by $Q = \left\{ u \in \mathbb{R}^m_{\geq 0} \;\middle|\; G^T u \leq h \right\}$, the extreme points of Q by $P_Q = \{p_k \mid k \in K\}$ and the extreme rays of Q by $R_Q = \{r_j \mid j \in J\}$.

Observe that (A.2) is feasible if and only if there exists a $x \in \mathbb{R}^n_{\geq 0}$ such that the problem $\mathrm{P}(x)$ is feasible. By explicitly enforcing the feasibility of (P), the following problem is equivalent to (A.2):

$$
\begin{aligned}
& \min c^T x + z_{\mathrm{P}(x)} \\
\text{s.t.} \quad & \varnothing \neq \left\{ y \in \mathbb{R}^p_{\geq 0} \;\middle|\; Gy \geq b - Ax \right\} \\
& x \in \mathbb{R}^n_{\geq 0}
\end{aligned}
$$

Applying Farkas' lemma leads to

$$
\begin{aligned}
& \min c^T x + z_{\mathrm{P}(x)} \\
\text{s.t.} \quad & 0 \geq r_j^T (b - Ax) \quad \forall r_j \in R_Q \\
& x \in \mathbb{R}^n_{\geq 0}
\end{aligned}
$$

This is a minimization over all x such that $\mathrm{P}(x)$ is feasible. If in addition $\mathrm{P}(x)$ is bounded for any $x \in \mathbb{R}^n_{\geq 0}$, also the dual $\mathrm{D}(x)$ is feasible and bounded and we can rely on strong duality, i.e., $z_{\mathrm{P}(x)} = z_{\mathrm{D}(x)} = \max \left\{ p_k^T (b - Ax) \;\middle|\; p_k \in P_Q \right\}$. Hence the problem can be rewritten as

$$
\begin{aligned}
& \min c^T x + \mu \\
\text{s.t.} \quad & \mu \geq p_k^T (b - Ax) \quad \forall p_k \in P_Q \\
& 0 \geq r_j^T (b - Ax) \quad \forall r_j \in R_Q \\
& x \in \mathbb{R}^n_{\geq 0} \\
& \mu \in \mathbb{R}
\end{aligned}
\tag{A.5}
$$

If $\mathrm{P}(x)$ is unbounded for some x, the dual is infeasible: $Q = \varnothing$ and $P_Q = \varnothing$. It follows that μ is unrestricted and therefore also (A.5) is unbounded. This proves the equivalence of (A.2) and (A.5).

The constraints of the form $\mu \geq p_k^T (b - Ax) \forall p_k \in P_Q$ are called *optimality constraints* and the inequalities $0 \geq r_j^T (b - Ax) \forall r_{\in} R_Q$ are called *feasibility constraints*. This linear program in x contains a large number of constraints and in general the sets P_Q and R_Q are not known. However, only a subset of these constraints is needed to describe an optimal solution.

A.2.1 Separation of Benders' Cuts

The previous section showed the equivalence of the two linear programs (A.2) and (A.5). This section shows how to find necessary elements of P_Q and R_Q iteratively. Denote by $P_Q^t \subseteq P_Q$ and $R_Q^t \subseteq R_Q$ the sets of extreme points and extreme rays known in iteration t. The following *relaxed master* program is a relaxation of (A.5):

$$\min c^T x + \mu$$

$$\begin{aligned} \text{s.t.} \quad & \mu \geq p_k^T(b - Ax) \quad \forall p_k \in P_Q^t \\ & 0 \geq r_j^T(b - Ax) \quad \forall r_j \in R_Q^t \\ & x \in \mathbb{R}_{\geq 0}^n \\ & \mu \in \mathbb{R} \end{aligned} \tag{A.6}$$

The algorithm to solve (A.2) by iteratively increasing P_Q^t and R_Q^t in (A.6) works as follows.

1. In iteration $t = 0$, both sets are empty: $P_Q^0 = \varnothing, R_Q^0 = \varnothing$.

2. Solve (A.6) to set (x_t, μ_t).

3. If (A.6) is infeasible $\rightarrow$ Stop(*infeasible*).

4. Solve the dual subproblem $\mathrm{D}(x_t)$.

   (a) If $\mathrm{D}(x_t)$ is infeasible $\rightarrow$ Stop(*unbounded*).

   (b) If $\mathrm{D}(x_t)$ is feasible and unbounded, we get an extreme ray r of Q. Add a feasibility cut: $R_Q^{t+1} = R_Q^t \cup \{r\}$.

   (c) Otherwise, $\mathrm{D}(x_t)$ is feasible and bounded and we get an extreme point p of Q.

      i. If $\mu_t = p^T(b - Ax_t) \rightarrow$ Stop(optimum).

      ii. Otherwise, $\mu_t < p^T(b - Ax_t)$. Add an optimality cut: $P_Q^{t+1} = P_Q^t \cup \{p\}$

5. Set $t = t + 1$, go to step 2.

A.3 Tightening by Rounding

This section recapitulates a method to tighten inequalities in integer programs.

Consider the following inequality in a minimization problem:

$$\sum_{i \in I} a_i x_i \geq b \tag{A.7}$$

where $\boldsymbol{x} \in \{0,1\}^{|I|}$ is a vector of binary variables, with nonnegative coefficients $\boldsymbol{a} \in \mathbb{R}^{|I|}_{\geq 0}$ and a nonnegative right hand side $b \geq 0$. Assume that the coefficient of the variable with index i is greater than the right hand side, i.e., $a_i > b$ for some fixed $i \in I$. The constraint can be satisfied by setting $x_i = 1$. This remains true if a_i is replaced by b. More generally, the constraint (A.7) is equivalent to:

$$\sum_{i \in I} \min(a_i, b) x_i \geq b \tag{A.8}$$

with respect to the integer program. However this rounded form (A.8) is stronger with respect to the linear relaxation $\left(\boldsymbol{x} \in [0,1]^{|I|}\right)$. A trivial example where the objective value of the LP is actually larger when using the rounded inequality is the following: Consider a binary program using only one variable: $\min\{x \mid ax \geq b, x \in \{0,1\}\}$. Assume that the coefficient is greater than the right hand side, i.e., $a > b$. The linear relaxation of the problem will yield the value $x = b/a < 1$, whereas the linear relaxation of the *rounded* problem: $\min\{x \mid bx \geq b, x \in \{0,1\}\}$ will lead to a stronger solution $x = b/b = 1$, which in fact is the integer optimal solution.

This method can be extended to cases where some coefficients on the left hand side are negative. Consider the following inequality:

$$\sum_{i \in I} a_i x_i - \sum_{j \in J} d_j y_j \quad \geq \quad b \tag{A.9}$$

$$\sum_{i \in I} a_i x_i \quad \geq \quad b + \sum_{j \in J} d_j y_j \tag{A.10}$$

with binary variables $\boldsymbol{x} \in \{0,1\}^{|I|}, \boldsymbol{y} \in \{0,1\}^{|J|}$, nonnegative coefficients $\boldsymbol{a} \in \mathbb{R}^{|I|}_{\geq 0}, \boldsymbol{d} \in \mathbb{R}^{|J|}_{\geq 0}$ and a nonnegative right hand side $b \geq 0$. The $\boldsymbol{y}$ variables have negative coefficients in (A.9). Moving them to the right hand side, we observe that the maximum value of the right hand side of (A.10) is $b + \sum_{j \in J} d_j$ for $\boldsymbol{y} = \mathbf{1}$. Now we can apply the same argument as above and provide the rounded inequality

$$\sum_{i \in I} \min\left(a_i, b + \sum_{j \in J} d_j\right) x_i \geq b + \sum_{j \in J} d_j y_j \tag{A.11}$$

which is equivalent for the integer program, but strengthens the LP.

A.4 Min-Cut Max-Flow

This section repeats a classical theorem from combinatorial optimization that states the equivalence between the *maximum flow* problem and the *minimum cut* problem (see [NW88]).

Consider a directed graph $G = (V, A)$ with capacities on the arcs $\boldsymbol{u} \in \mathbb{R}^{|A|}_{\geq 0}$ and two special nodes: a *source* $s \in V$ and a *sink* $t \in V$. The *maximum flow* problem asks for a flow $\boldsymbol{f} \in \mathbb{R}^{|A|}_{\geq 0}$, subject to $\sum_{(ij)\in\delta^+(i)} f_{ij} - \sum_{(ji)\in\delta^-(i)} f_{ji} = 0$ $\forall i \in V \setminus \{s, t\}$ that does not exceed the capacities $\boldsymbol{f} \leq \boldsymbol{u}$ and maximizes the amount of flow $\sum_{(it)\in\delta^-(t)} f_{it}$ through the graph.

The *minimum cut* problem asks for a partition of V: $\{U, \bar{U}\}, U \cap \bar{U} = \varnothing$, $U \cup \bar{U} = V, s \in U, t \in \bar{U}$ such that the set of crossing arcs $\delta^+(U) = \{(i,j) \in A | i \in U, j \in \bar{U}\}$ minimizes the capacity $\sum_{a\in\delta^+(U)} u_a$.

Obviously every flow has to cross any cut in the graph. Therefore no flow through the graph can be greater than the capacity of any cut:

$$\sum_{a\in\delta^-(t)} f_a \leq \sum_{a\in\delta^+(U)} u_a$$

for any feasible flow $\boldsymbol{f}$ and any set $U : \{s\} \subseteq U \subseteq V \setminus \{t\}$. Furthermore, the *max-flow min-cut theorem* states that:

Theorem A.4.1. *The value of a maximum flow equals the capacity of a minimum cut.*

The proof can be found for example in [NW88].

Appendix B

List of Symbols

α	Coverage rate *alpha*. Defines the target prize $p_0 = \alpha \sum_{k \in K} p_k.$, page 76
A	The set of arcs of a graph $G = (V, A)$, page 29
a	An arc of a graph $a \in A$, page 29
c_k	The cost associated with a customer $c_k \geq 0$, page 12
$c_{e,m}$	The cost of a module $c_{e,m} \geq 0, e \in E, m \in M_e$, page 9
$\delta^+(i)$	The set of arcs emanating from node i, page 31
$\delta^+(i)$	The set of arcs emanating from the node i, page 11
$\delta^+(S)$	The set of arcs emanating from the set S, page 32
$\delta^-(i)$	The set of arcs entering node i, page 31
$\delta^-(i)$	The set of arcs entering the node i, page 11
$\delta^-(S)$	The set of arcs entering the set S, page 32
d_k	The demand of a customer $d_k \geq 0$, page 9
E	The set of edges of a graph $G = (V, E)$, page 9
e	An edge of a graph $e \in E$, page 9
f_a	A real valued flow variable associated to the arc a, page 11
G	An undirected graph $G = (V, E)$ or a directed graph $G = (V, A)$, page 9
i	A node in a graph $i \in V$, page 11

$\{i, j\}$	An (undirected) edge in a graph $\{i, j\} \in E$, page 11
(i, j)	A (directed) arc in a graph $(i, j) \in A$, page 29
j	A node in a graph $j \in V$, page 11
K	The set of customers $K \subset V$, page 9
k	A customer $k \in K$, page 9
m	The index of a module $m \in M_e$, page 9
μ_e	A function mapping from a requested capacity to the index of the most appropriate module for the edge e, page 27, page 85
M_e	The set of indices of the modules available for the edge $e, M_e = \{1, 2, \dots\}$, page 9
p_0	The target prize $p_0 \geq 0$, page 12
p_k	The prize associated with a customer $p_k \geq 0$, page 12
r	The backbone access node $r \in V$, page 9
$u_{e,m}$	The capacity of a module $c_{e,m} \geq 0, e \in E, m \in M_e$, page 9
V	The set of nodes of a graph $G = (V, E)$, page 9
v	A vertex of a graph $v \in V$, page 9
$x_{a,m}$	A binary decision variable associated to the arc a and the module m, page 31
$x_{e,m}$	A binary decision variable associated to the edge e and the module m, page 11
y_k	A binary decision variable associated to the customer k, page 12

Appendix C

Abstract

Within recent years the request for broadband telecommunication networks has been constantly increasing. A strategy employed by telecommunication companies to increase the bandwidth on the *last mile* of the network is to lay optical fiber directly to the end customer. This strategy is denoted as *fiber to the home* (FTTH).

In this thesis the *local access network* design problem (LAN) and its prize-collecting variant (PC-LAN) are used to formalize the planning of FTTH networks. The LAN problem asks for a cost minimal solution and allows to model different cable technologies, existing infrastructure and the overhead cost incurred by building new connections. In addition, the PC-LAN problem covers the aspect, that not all customers must necessarily be connected with FTTH, but instead we search for a subset of customers in order to maximize profits.

To solve LAN and PC-LAN instances, the following operations research methods are employed: Preprocessing, mixed integer programming, model strengthening by variable disaggregation, Benders' decomposition and adaptive multi-start heuristics.

In a project between University of Vienna and Telekom Austria, large real world data sets for FTTH planning were investigated and the methods presented in this thesis have been designed. These solution methods have been implemented as computer programs and empirically verified to be reasonable approaches to FTTH network design problems.

Anhang D

Zusammenfassung

In den letzten Jahren gab es zunehmenden Bedarf für breitbandige Telekommunikations Netzwerke. Eine von Telekommunikationsunternehmen angewandte Strategie um die Bandbreite entlang der *last-mile* des Netzwerks zu erhöhen ist, Glasfaserkabel direkt bis zum Endkunden zu verlegen. Diese Strategie wird *fiber to the home* (FTTH) genannt.

In der vorliegenden Arbeit wird das *local access network* design problem (LAN) und die Variante mit *prize-collecting* (PC-LAN) verwendet, um das Problem der FTTH Planung zu modellieren. Das LAN Problem zielt darauf ab eine kostenminimale Lösung zu finden und gestattet es sowohl verschiedene Kabeltechnologien und existierende Infrastruktur, als auch die Zusatzkosten zu modellieren, die anfallen wenn neue Verbindungen hergestellt werden. Darüber hinaus, erlaubt das PC-LAN Problem den Aspekt zu modellieren, dass nicht unbedingt alle Kunden mit FTTH versorgt werden müssen. Stattdessen wird eine Teilmenge der Kunden versorgt mit dem Ziel den Profit zu maximieren.

Um LAN und PC-LAN Problem Instanzen zu lösen, werden folgende Methoden des Operations Research angewandt: Preprocessing, ganzzahlige Programmierung, Stärkung der mathematischen Modelle durch Disaggregation der Variablen, Benders' Dekomposition und adaptive Multi-Start-Heuristiken.

In einem Projekt von Universität Wien und Telekom Austria wurden große FTTH Datensätze untersucht und die hier vorgestellten Methoden entworfen. Diese Lösungsansätze wurden als Computerprogramme implementiert und ihre Tauglichkeit zur Behandlung von FTTH Planungsfragen konnte gezeigt werden.

Appendix E

Curriculum Vitae

Personal Data

Name	Peter Putz
Title	Dipl. Ing.
Address	Neudeggergasse 10/11
	Austria 1080 Vienna
Email	p.putz@yahoo.de
Date of Birth	31.03.1978
Place of Birth	Vienna, Austria

Education

1984-1988	Volksschule, 1220 Vienna
1988-1992	Gymnasium, 1210 Vienna
1992-1997	HTL Mödling, Möbel- und Innenausbau, Matura
1997-2007	Vienna University of Technology, Dipl. Ing.
2007-	University of Vienna, Dr. rer. soc. oec.

Publications and Conference Proceedings

I. Ljubić, P. Putz, and J. J. Salazar-González. Exact Approaches to the Single-Source Network Loading Problem. *Networks*, 59(1):89–106, 2011. DOI:10.1002/net.20481.

I. Ljubić, P. Putz, and J. J. Salazar-González. A Heuristic Algorithm for a Prize-Collecting Local Access Network Design Problem. In J. Pahl, T. Reiners, and S. Voss, editors, *Network Optimization*, volume 6701 of *LNCS*. Springer, 2011. DOI:10.1007/978-3-642-21527-8_17.

Appendix F

Bibliography

[ABG$^+$11] A. Arulselvan, A. Bley, S. Gollowitzer, I. Ljubić, and O. Maurer. MIP Modeling of Incremental Connected Facility Location. In J. Pahl, T. Reiners, and S. Voss, editors, *Proceedings of the International Network Optimization Conference (INOC)*, volume 6701 of *Lecture Notes in Computer Science*. Springer, 2011. DOI:10.1007/978-3-642-21527-8_54.

[AC07] F. Alvelos and J. M. V. de Carvalho. An Extended Model and a Column Generation Algorithm for the Planar Multicommodity Flow Problem. *Networks*, 50(1):3–16, 2007. DOI:10.1002/net.20161.

[AG88] K. Altinkemer and B. Gavish. Heuristics with Constant Error Guarantees for the Design of Tree Networks. *Management Science*, 34(3):331–341, 1988. DOI:10.1287/mnsc.34.3.331.

[AG07] A. Atamtürk, , and O. Günlük. Network Design Arc Set with Variable Upper Bounds. *Networks*, 50(1):17–28, 2007. DOI:10.1002/net.20162.

[AGVDA05] A. M. Alvarez, J. L. González-Velarde, and K. De-Alba. Grasp Embedded Scatter Search for the Multicommodity Capacitated Network Design Problem. *Journal of Heuristics*, 11(3):233–257, 2005. DOI:10.1007/s10732-005-1509-4.

[AMO93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows*. Prentice Hall, 1993.

[AMS07] P. Avella, S. Mattia, and A. Sassano. Metric Inequalities and the Network Loading Problem. *Discrete Optimization*, 4(1):103–114, 2007. DOI:10.1016/j.disopt.2006.10.002.

[AP97] A. Amiri and H. Pirkul. Routing and Capacity Assignment in Backbone Communication Networks. *Computers & Operations Research*, 24(3):275–287, 1997. DOI:10.1016/S0305-0548(96)00049-4.

[Ata01] A. Atamtürk. Flow Pack Facets of the Single Node Fixed-Charge Flow Polytope. *Operations Research Letters*, 29(3):107–114, October 2001. DOI:10.1016/S0167-6377(01)00100-6.

[Ata02] A. Atamtürk. On Capacitated Network Design Cut-Set Polyhedra. *Mathematical Programming*, 92:425–437, 2002. DOI:10.1007/s101070100284.

[Bar96] F. Barahona. Network Design Using Cut Inequalities. *SIAM Journal on Optimization*, 6(3):823–837, 1996. DOI:10.1137/S1052623494279134.

[BCC10] T. Bektaş, M. Chouman, and T. G. Crainic. Lagrangean-based Decomposition Algorithms for Multicommodity Network Design Problems with Penalized Constraints. *Networks*, 55(3):171–180, 2010. DOI:10.1002/net.20362.

[BCGT98] D. Bienstock, S. Chopra, O. Günlük, and C.-Y. Tsai. Mininum Cost Capacity Installation for Multicommodity Flows. *Mathematical Programming*, 81(2):177–199, 1998. DOI:10.1007/BF01581104.

[Ben62] J. F. Benders. Partitioning Procedures for Solving Mixed-Variables Programming Problems. *Numerische Mathematik*, 4(3):238–252, Sept. 1962. DOI:10.1007/s10287-004-0020-y.

[BG96] D. Bienstock and O. Günlük. Capacitated Network Design - Polyhedral Structure and Computation. *INFORMS Journal on Computing*, 8:243–259, 1996. DOI:10.1287/ijoc.8.3.243.

[BGP+00] D. Berger, B. Gendron, J.-Y. Potvin, S. Raghavan, and P. Soriano. Tabu Search for a Network Loading Problem with Multiple Facilities. *Journal of Heuristics*, 6(2):253–267, 2000. DOI:10.1023/A:1009679511137.

[BR02] D. Bienstock and O. Raskina. Asymptotic Analysis of the Flow Deviation Method for the Maximum Concurrent Flow Problem. *Mathematical Programming*, 91(3):479–492, 2002. DOI:10.1007/s101070100254.

[BR10] M. G. Bardossy and S. Raghavan. Dual-Based Local Search for the Connected Facility Location and Related Problems. *INFORMS Journal on Computing*, 22(4):584–602, 2010. DOI:10.1287/ijoc.1090.0375.

[CCG09] A. M. Costa, J.-F. Cordeau, and B. Gendron. Benders, Metric and Cutset Inequalities for Multicommodity Capacitated Network Design. *Computational Optimization and Applications*, 42(3):371–392, April 2009. DOI:10.1007/s10589-007-9122-0.

[CF12] I. Contreras and E. Fernández. General Network Design: A Unified View of Combined Location and Network Design Problems. *European Journal of Operational Research*, 219(3):680–697, 2012. DOI:10.1016/j.ejor.2011.11.009.

[CFG01] T. G. Crainic, A. Frangioni, and B. Gendron. Bundle-based Relaxation Methods for Multicommodity Capacitated Fixed Charge Network Design. *Discrete Applied Mathematics*, 112(1-3):73–99, 2001. DOI:10.1016/S0166-218X(00)00310-3.

[CG97] B. V. Cherkassky and A. V. Goldberg. On Implementing Push-Relabel Method for the Maximum Flow Problem. *Algorithmica*, 19:390–410, 1997. DOI:10.1007/PL00009180.

[CGH04] T. G. Crainic, B. Gendron, and G. Hernu. A Slope Scaling/Lagrangean Perturbation Heuristic with Long-Term Memory for Multicommodity Capacitated Fixed-Charge Network Design. *Journal of Heuristics*, 10(5):525–545, 2004. DOI:10.1023/B:HEUR.0000045323.83583.bd.

[CGM03] K. L. Croxton, B. Gendron, and T. L. Magnanti. A Comparison of Mixed-Integer Programming Models for Nonconvex Piecewise Linear Cost Minimization Problems. *Management Science*, 49:1268–1273, 2003. DOI:10.1287/mnsc.49.9.1268.16570.

[CGM07] K. L. Croxton, B. Gendron, and T. L. Magnanti. Variable Disaggregation in Network Flow Problems with Piecewise Linear Costs. *Operations Research*, 55(1):146–157, 2007. DOI:10.1287/opre.1060.0314.

[CGR92] S. Chopra, E. R. Gorres, and M. R. Rao. Solving the Steiner Tree Problem on a Graph Using Branch and Cut. *ORSA Journal on Computing*, 4(3):320–335, 1992. DOI:10.1287/ijoc.4.3.320.

[CGS98] S. Chopra, I. Gilboa, and S. T. Sastry. Source Sink Flows with Capacity Installation in Batches. *Discrete Applied Mathematics*, 85(3):165–192, 1998. DOI:10.1016/S0166-218X(98)00024-9.

[Cha10] S. CHAMBERLAND. Global Access Network Evolution. *IEEE/ACM Transactions on Networking*, 18(1):136–149, 2010. DOI:10.1109/TNET.2009.2021430.

[Cos05] A. M. COSTA. A Survey on Benders Decomposition Applied to Fixed-Charge Network Design Problems. *Computers & Operations Research*, 32(6):1429–1450, 2005. DOI:10.1016/j.cor.2003.11.012.

[CR94] S. CHOPRA and M. R. RAO. The Steiner Tree Problem I: Formulations, Compositions and Extension of Facets. *Mathematical Programming*, 64(1–3):209–229, 1994. DOI:10.1007/BF01582573.

[DS98] G. DAHL and M. STOER. A Cutting Plane Algorithm for Multicommodity Survivable Network Design Problems. *INFORMS Journal on Computing*, 10(1):1–11, 1998. DOI:10.1287/ijoc.10.1.1.

[FG09] A. FRANGIONI and B. GENDRON. 0-1 Reformulations of the Multicommodity Capacitated Network Design Problem. *Discrete Applied Mathematics*, 157(6):1229–1241, 2009. DOI:10.1016/j.dam.2008.04.022.

[FG10] A. FRANGIONI and B. GENDRON. A Stabilized Structured Dantzig-Wolfe Decomposition Method. Technical Report 2010–02, CIRRELT, 2010.

[FSZ10] M. FISCHETTI, D. SALVAGNIN, and A. ZANETTE. A Note on the Selection of Benders' Cuts. *Mathematical Programming*, 124:175–182, 2010. DOI:10.1007/s10107-010-0365-7.

[GA90] B. GAVISH and K. ALTINKEMER. Backbone Network Design Tools with Economic Tradeoffs. *ORSA Journal on Computing*, 2(3):236–252, 1990. DOI:10.1287/ijoc.2.3.236.

[Gav85] B. GAVISH. Augmented Lagrangean Based Algorithms for Centralized Network Design. *IEEE Transactions on Communications*, 33(12):1247–1257, 1985. DOI:10.1109/TCOM.1985.1096250.

[GCG04] I. GHAMLOUCHE, T. G. CRAINIC, and M. GENDREAU. Path Relinking, Cycle-based Neighbourhoods and Capacitated Multicommodity Network Design. *Annals of Operations Research*, 131(1-4):109–133, 2004. DOI:10.1023/B:ANOR.0000039515.90453.1d.

[GGL11] S. GOLLOWITZER, L. GOUVEIA, and I. LJUBIĆ. A Node Splitting Technique for Two Level Network Design Problems with Transition Nodes. In J. PAHL, T. REINERS, and S. VOSS, editors, *Proceedings of the International Network Optimization Conference*

(INOC), volume 6701 of *Lecture Notes in Computer Science*, pages 57–70. Springer, 2011. DOI:10.1007/978-3-642-21527-8_8.

[GGL12] S. GOLLOWITZER, B. GENDRON, and I. LJUBIĆ. Capacitated Network Design with Facility Location. In *2nd International Symposium on Combinatorial Optimization*. 2012.

[GGR06] I. GAMVROS, B. GOLDEN, and S. RAGHAVAN. The Multilevel Capacitated Minimum Spanning Tree Problem. *INFORMS Journal on Computing*, 18(3):348–365, 2006. DOI:10.1287/ijoc.1040.0123.

[GKK+01] N. GARG, R. KHANDEKAR, G. KONJEVOD, R. RAVI, F. SALMAN, and A. SINHA. On the Integrality Gap of a Natural Formulation of the Single-Sink Buy-at-bulk Network Design Problem. [PS]. In K. AARDAL and B. GERARDS, editors, *Proceedings of the 8th Conference on Integer Programming & Combinatorial Optimization (IPCO)*, volume 2081 of *Lecture Notes in Computer Science 2081*, pages 170–184. Springer Berlin / Heidelberg, 2001. DOI:10.1007/3-540-45535-3_14.

[GKM99] V. GABREL, A. KNIPPEL, and M. MINOUX. Exact Solution of Multicommodity Network Optimization Problems with General Step Cost Functions. *Operations Research Letters*, 25(1):15–23, August 1999. DOI:10.1016/S0167-6377(99)00020-6.

[GKM03] V. GABREL, A. KNIPPEL, and M. MINOUX. A Comparison of Heuristics for the Discrete Cost Multicommodity Network Optimization Problem. *Journal of Heuristics*, 9(5):429–445, Nov 2003. DOI:10.1023/B:HEUR.0000004812.23590.a2.

[GKR03] A. GUPTA, A. KUMAR, and T. ROUGHGARDEN. Simpler and Better Approximation Algorithms for Network Design. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, STOC '03, pages 365–372. ACM, New York, NY, USA, 2003. DOI:10.1145/780542.780597.

[GL11] S. GOLLOWITZER and I. LJUBIĆ. MIP Models for Connected Facility Location: A Theoretical and Computational Study. *Computers & Operations Research*, 38(2):435–449, 2011. DOI:10.1016/j.cor.2010.07.002.

[GMM01] S. GUHA, A. MEYERSON, and K. MUNAGALA. A Constant Factor Approximation for the Single Sink Edge Installation Problem. In *Proceedings of the thirty-third annual ACM symposium on Theory* *of computing*, STOC '01, pages 383–388. ACM, New York, NY, USA, 2001. DOI:10.1145/380752.380827.

[GMS10] S. GUALANDI, F. MALUCELLI, and D. L. SOZZI. On the Design of the Next Generation Access Networks. In A. LODI, M. MILANO, and P. TOTH, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, 7th International Conference, CPAIOR 2010, Bologna, Italy, June 14-18, 2010. Proceedings*, volume 6140 of *Lecture Notes in Computer Science*, pages 162–175. Springer, 2010. DOI:10.1007/978-3-642-13520-0_20.

[Gü99] O. GÜNLÜK. A Branch-and-Cut Algorithm for Capacitated Network Design Problems. *Mathematical Programming*, 86(1):17–39, 1999. DOI:10.1007/s101070050077.

[HKLS03] S. VAN HOESEL, A. KOSTER, R. L. M. J. VAN DE LEENSEL, and M. W. P. SAVELSBERGH. Bidirected and Unidirected Capacity Installation in Telecommunication Networks. *Discrete Applied Mathematics*, 133(1-3):103–121, 2003. DOI:10.1016/S0166-218X(03)00436-0.

[HMS07] M. HAOUARI, M. MRAD, and H. SHERALI. Optimum Synthesis of Discrete Capacitated Networks with Multi-Terminal Commodity Flow Requirements. *Optimization Letters*, 1:341–354, 2007. DOI:10.1007/s11590-006-0030-5.

[HY98] K. HOLMBERG and D. YUAN. A Lagrangean Approach to Network Design Problems. *International Transactions in Operational Research*, 5(6):529–539, 1998. DOI:10.1111/j.1475-3995.1998.tb00135.x.

[HY00] K. HOLMBERG and D. YUAN. A Lagrangian Heuristic Based Branch-and-Bound Approach for the Capacitated Network Design Problem. *Operations Research*, 48(3):461–481, May/June 2000. DOI:10.1287/opre.48.3.461.12439.

[ILO] I. ILOG. CPLEX. `http://www.ilog.com/products/cplex/`; visited on May 1st 2010.

[Iri71] M. IRI. On an Extension of the Maximum-Flow Minimum-Cut Theorem to Multicommodity Flows. *J. Operations Research Society of Japan*, 13(3):129–135, January 1971.

[JR05] R. JOTHI and B. RAGHAVACHARI. Approximation Algorithms for the Capacitated Minimum Spanning Tree Problem and its Variants in Network Design. *ACM Transactions on Algorithms*, 1(2):265–282, 2005. DOI:10.1145/1103963.1103967.

[JR09] R. Jothi and B. Raghavachari. Improved Approximation Algorithms for the Single-Sink Buy-at-Bulk Network Design Problems. *Journal of Discrete Algorithms*, 7:249–255, June 2009. DOI:10.1016/j.jda.2008.12.003.

[KFJN04] A. B. Keha, I. R. de Farias Jr., and G. L. Nemhauser. Models for Representing Piecewise Linear Cost Functions. *Operations Research Letters*, 32(1):44–48, 2004. DOI:10.1016/S0167-6377(03)00059-2.

[KLH11] Y. Kim, Y. Lee, and J. Han. A Splitter Location-Allocation Problem in Designing Fiber Optic Access Networks. *European Journal of Operational Research*, 210(2):425 – 435, 2011. DOI:10.1016/j.ejor.2010.10.003.

[KM98] T. Koch and A. Martin. Solving Steiner Tree Problems in Graphs to Optimality. *Networks*, 32(3):207–232, 1998. DOI:10.1002/(SICI)1097-0037(199810)32:3<207::AID-NET5>3.0.CO;2-O.

[LED] A. S. S. G. LEDA. LEDA Library for Efficient Data Types and Algorithms. www.algorithmic-solutions.com; visited on May 1st 2010.

[Lju07] I. Ljubić. A Hybrid VNS for Connected Facility Location. In T. Bartz-Beielstein, M. Blesa Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, and M. Sampels, editors, *Hybrid Metaheuristics*, volume 4771 of *Lecture Notes in Computer Science*, pages 157–169. Springer Berlin / Heidelberg, 2007. DOI:10.1007/978-3-540-75514-2_12.

[LPSG11a] I. Ljubić, P. Putz, and J. J. Salazar-González. Exact Approaches to the Single-Source Network Loading Problem. *Networks*, 59(1):89–106, 2011. DOI:10.1002/net.20481.

[LPSG11b] I. Ljubić, P. Putz, and J. J. Salazar-González. A Heuristic Algorithm for a Prize-Collecting Local Access Network Design Problem. In J. Pahl, T. Reiners, and S. Voss, editors, *Network Optimization*, volume 6701 of *LNCS*. Springer, 2011. DOI:10.1007/978-3-642-21527-8_17.

[LR11] M. Leitner and G. R. Raidl. Branch-and-Cut-and-Price for Capacitated Connected Facility Location. *Journal of Mathematical Modelling and Algorithms*, 2011. DOI:10.1007/s10852-011-9153-5.

[LWP$^+$06] I. Ljubić, R. Weiskircher, U. Pferschy, G. W. Klau, P. Mutzel, and M. Fischetti. An Algorithmic Framework for the Exact Solution of the Prize-Collecting Steiner Tree Problem. *Mathematical Programming*, 105:427–449, 2006. DOI:10.1007/s10107-005-0660-x.

[Min01] M. Minoux. Discrete Cost Multicommodity Network Optimization Problems and Exact Solution Methods. *Annals of Operations Research*, 106(1):19–46, September 2001. DOI:10.1023/A:1014554606793.

[Mir00] P. Mirchandani. Projections of the Capacitated Network Loading Problem. *European Journal of Operational Research*, 122(3):534 – 560, 2000. DOI:10.1016/S0377-2217(99)00083-1.

[MLS00] G. R. Mateus, H. P. L. Luna, and A. B. Sirihal. Heuristics for Distribution Network Design in Telecommunication. *Journal of Heuristics*, 6(1):131–148, 2000. DOI:10.1023/A:1009669927855.

[MM93] T. L. Magnanti and P. Mirchandani. Shortest Paths, Single Origin-Destination Network Design, and Associated Polyhedra. *Networks*, 23(2):103–121, 1993. DOI:10.1002/net.3230230205.

[MM04] A. Muriel and F. Munshi. Capacitated Multicommodity Network Flow Problems with Piecewise Linear Concave Costs. *IIE Transactions*, 36(7):683 – 696, July 2004. DOI:10.1080/07408170490447357.

[MMV93] T. L. Magnanti, P. Mirchandani, and R. Vachani. The Convex Hull of Two Core Capacitated Network Design Problems. *Mathematical Programming*, 60(2):233–250, 1993. DOI:10.1007/BF01580612.

[MMV95] T. L. Magnanti, P. Mirchandani, and R. Vachani. Modeling and Solving the Two-Facility Capacitated Network Loading Problem. *Operations Research*, 43(1):142–157, 1995. DOI:10.1016/0966-8349(95)97847-D.

[MMW86] T. L. Magnanti, P. Mireault, and R. T. Wong. Tailoring Benders Decomposition for Uncapacitated Network Design. In G. Gallo and C. Sandi, editors, *Netflow at Pisa*, volume 26 of *Mathematical Programming Studies*, pages 112–154. Springer Berlin Heidelberg, 1986. DOI:10.1007/BFb0121090.

[MW81] T. L. Magnanti and R. T. Wong. Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria. *Operations Research*, 29(3):464–484, 1981. DOI:10.1287/opre.29.3.464.

[MW84] T. L. Magnanti and R. T. Wong. Network Design and Transportation Planning: Models and Algorithms. *Transportation Science*, 18(1):1–55, 1984. DOI:10.1287/trsc.18.1.1.

[NW88] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, Inc., 1988.

[Pap08] N. Papadakos. Practical Enhancements to the Magnanti-Wong Method. *Operations Research Letters*, 36(4):444 – 449, 2008. DOI:10.1016/j.orl.2008.01.005.

[RA04] D. Rajan and A. Atamtürk. A Directed Cycle-based Column-and-Cut Generation Method for Capacitated Survivable Network Design. *Networks*, 43(4):201–211, 2004. DOI:10.1002/net.20004.

[RGCS09] W. Rei, M. Gendreau, J.-F. Cordeau, and P. Soriano. Accelerating Benders Decomposition by Local Branching. *INFORMS Journal on Computing*, 21(2):333–345, 2009. DOI:10.1287/ijoc.1080.0296.

[RKOW07] C. Raack, A. M. Koster, S. Orlowski, and R. Wessäly. Capacitated Network Design using General Flow-Cutset Inequalities. Technical report, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2007.

[RLM01] C. D. Randazzo, H. P. L. Luna, and P. Mahey. Benders Decomposition for Local Access Network Design with Two Technologies. *Discrete Mathematics and Theoretical Computer Science*, 4(2):235–246, 2001.

[RS06] S. Raghavan and D. Stanojević. A Note on Search by Objective Relaxation. In *Telecommunications Planning: Innovations in Pricing, Network Design and Management*, volume 33 of *Operations Research/Computer Science Interfaces Series*, pages 181–201. Springer US, 2006. DOI:10.1007/0-387-29234-9_10.

[Sal00] F. S. Salman. *Selected Problems in Network Design: Exact and Approximate Solution Methods*. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, 2000.

[SCRS00] F. S. Salman, J. Cheriyan, R. Ravi, and S. Subramanian. Approximating the Single-Sink Link-Installation Problem in Network Design. *SIAM Journal on Optimization*, 11(3):595–610, 2000. DOI:10.1137/S1052623497321432.

[SPE] SPEC. Standard Performance Evaluation Corporation. www.spec.org; visited on May 1st 2010.

[SRH08] F. S. Salman, R. Ravi, and J. N. Hooker. Solving the Capacitated Local Access Network Design Problem. *INFORMS Journal on Computing*, 20(2):243–254, 2008. DOI:10.1287/ijoc.1070.0237.

[Tal02] K. Talwar. The Single-Sink Buy-at-Bulk LP Has Constant Integrality Gap. In G. Goos, J. Hartmanis, and J. van Leeuwen, editors, *Proceedings of the 9th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, volume 2337 of *LNCS*, pages 475 – 486. Springer, 2002. DOI:10.1007/3-540-47867-1_33.

[TL08] A. Tomazic and I. Ljubić. A GRASP Algorithm for the Connected Facility Location Problem. In *SAINT*, pages 257–260. 2008. DOI:10.1109/SAINT.2008.64.

[Was11] B. Wassermann. *Operations Research in Action: A Project for Designing Telecommunication Access Networks*. Ph.D. thesis, Universität Wien, 2011. permalink.obvsg.at/AC08958101.