# LP approaches to survivable networks with single path routing 

Diplomarbeit bei<br>Prof. Dr. M. Grötschel

vorgelegt von
Ulrich Menne

Fachbereich Mathematik der TU Berlin, Studiengang Wirtschaftsmathematik

Berlin, den 17. Dezember 2003

## Contents

1 Introduction ..... 5
2 Graphs and polyhedra ..... 7
2.1 Basic notation ..... 7
2.2 Graph Theory ..... 7
2.3 Polyhedral Theory ..... 9
3 The network design problem ..... 10
3.1 Problem description ..... 10
3.2 Model ..... 14
3.2.1 Notation and parameters ..... 14
3.2.2 Variables ..... 17
3.2.3 Inequalities ..... 18
3.2.4 Objective function ..... 27
3.3 The model ..... 28
4 Classes of valid inequalities ..... 29
4.1 GUB cover inequalities ..... 29
4.1.1 Valid inequalities from knapsack and GUB ..... 30
4.1.2 Edge (secure) capacity constraints ..... 31
4.1.3 Port constraints ..... 32
4.1.4 Node capacity constraints ..... 33
4.2 Cutset inequalities ..... 34
4.2.1 Cut traffic inequalities ..... 36
4.2.2 Particular commodity inequalities ..... 36
4.3 Edge capacity inequalities ..... 39
4.3.1 Setup ..... 39
4.3.2 Inequality ..... 41
4.3.3 Relaxation ..... 42
4.3.4 Lifting ..... 50
5 Algorithmic approach and implementational aspects ..... 53
5.1 Branch-and-Cut ..... 53
5.1.1 Overview ..... 53
5.1.2 Node selection ..... 55
5.1.3 Variable selection ..... 55
5.2 Separation ..... 57
5.2.1 Separation of the GUB cover inequalities ..... 57
5.2.2 Separation of the hoplimit constraints ..... 58
5.2.3 Separation of the subtour elimination constraints ..... 58
5.3 Preprocessing ..... 61
5.3.1 Constraints reduction ..... 61
5.3.2 Reduction of the commodity edge flow variables ..... 62
5.3.3 Reduction of the linkdesign installation variables ..... 63
5.4 Heuristics ..... 64
5.4.1 Determination of a feasible routing ..... 64
5.4.2 Determination of a feasible linkdesign installation ..... 66
5.4.3 Additional edge exclusion ..... 67
5.4.4 Heuristic parameters ..... 68
6 Data and computational results ..... 70
6.1 Data set ..... 70
6.2 Computational results ..... 71
6.2.1 Downsizing results ..... 71
6.2.2 Branching rule comparison ..... 74
6.2.3 Hoplimit inclusion ..... 77
6.2.4 Heuristic modes ..... 77
6.2.5 Results ..... 79
7 Conclusion ..... 89
Bibliography ..... 93

## Chapter 1

## Introduction

Customer growth, increasing demands for old and new services, deregulation and the innovations in switching and transmission technologies are all placing pressure on telecommunication companies to upgrade and expand their networks. With over 50 percent of a telephone company's total investment in communication facilities and with deregulation making the market more competitive, these expansion decisions can have huge economic and strategic ramifications.
The conflictive aims towards network design are the creation of a low-cost network capable of handling the communication demands and maximum customer satisfaction in terms of quality of service. A crucial aspect of customer satisfaction is survivability, which means that the offered service has to survive certain failure situations. Common types of network failures are cable cuts caused by dredgers or other external impacts as well as defects of hardware or overloaded network.

The design of communication networks bases on an estimation of communication demands between locations. This communication traffic has to be routed over the connection links between these locations. When traffic is routed over the network it consumes capacity provided by some installed hardware. The problem can be modeled as an optimization problem where, on the one hand, the link capacity must be assigned at the lowest cost and, on the other hand, the communication flow must be routed over the network requiring the accomplishment of certain conditions. There exist different approaches towards survivability. The main strategies are protection and restoration. Protection ensures survivability by conditions on the normal operating state routing while no or little predetermined reconfiguration is required in case of a failure. Restoration techniques react in case of a failure by rerouting the affected traffic (ensuring firstly the existence of spare capacity). Another concept is the approach towards resilience, which means that there exist certain hardware technologies, installed at the locations and links of a network, which can never fail. In this thesis we focus on the latter approach. Also the exclusion of routing demands over unacceptably long paths in the network is taken into account (to decrease the probability of a failure of a path component and transmission delays) as well as certain restrictions on the network
locations.
This thesis is motivated by the network planning problem discussed in the context of the ROCOCO project ("Recherche Opérationnelle et COntraintes pour la COnception de réseaux"). The participants of this project (ILOG, France Télécom CNET, LRI and INRIA [2]) aimed at developing algorithms integrating the effectiveness of combinatorial optimization and the flexibility of Constraint Programming to optimize the dimensioning and the redimensioning of networks of telecommunications companies in particular. Robustness of optimization techniques was a main aspect in their research endeavors. Robust means that the applied algorithm must not only provide "good" solutions to problem instances of different size and numerical characteristics, but also that the algorithm must continue to work well when constraints are added or removed. Hence a network design problem is considered including a number of different additional constraints. These constraints are added arbitrarily to the base problem. No proper mathematical model integrating all constraints simultaneously has been published until now and only an informal problem description was available to us.

The fact that no publicly accessible model exists represents the initial point of this thesis. We develop a comprehensive approach to cope with the described problem including an extensive model able to handle all conditions considered within the ROCOCO project. Chapter 2 gives a short overview on notations. In the first part of Chapter 3 we present the given description, point out the relation to reality and explain the effects of certain constraints. In the second section of this chapter we introduce in detail the developed model including all constraints and restrictions on the routing and capacity installation. It is inclusively capable of modeling any possible combination of the additional constraints. In Chapter 4 we examine the associated polyhedron and elaborate valid inequalities. We show that some of them are even facet-defining for a relaxation of the problem. We apply an LP-based approach to solve the given network design problem. The developed inequalities are used in a branch-and-cut algorithm presented in Chapter 5. Also preprocessing and heuristic endeavors are introduced and described in detail in the last section of this chapter. We evaluate different strategies and measure the quality of our approach on an extensive benchmark suite built on the basis of real network design data provided by France Tèlècom R\&D [13]. This suite has been developed in the context of the prementioned ROCOCO project. The inhomogeneous structure of the benchmark reflects their ambitions towards robustness. The achieved computational results are presented in Chapter 6, together with a study of the given data. Finally, in Chapter 7 some conclusions are drawn.

## Chapter 2

## Graphs and polyhedra

In this chapter, we present an outline of the elementary mathematical tools we utilized to approach our problem. We describe graph theory, since we present the given problem in this context. We further modeled the problem as a linear program and surveyed the respective polyhedron, hence we introduce some facts from the polyhedral theory. We assume some basic knowledge of linear optimization. For these fundamentals the reader is referred to [8], and especially regarding integer programming [14] and [11].

### 2.1 Basic notation

We will denote by $\mathbb{R}(\mathbb{Q}, \mathbb{Z})$ the real (rational, integer) numbers. The sets $\mathbb{R}_{+}\left(\mathbb{Q}_{+}, \mathbb{Z}_{+}\right)$ stand for the non-negative real (rational, integer) numbers. For ease of notation, we use $\mathbb{K}$ (or $\mathbb{K}_{+}$) if one of these three sets can be applied. We denote the set of positive integer numbers without zero by $\mathbb{N}=\mathbb{Z}_{+} \backslash\{0\}$. For some $n \in \mathbb{N}$, we define by $\mathbb{K}^{n}$ the set of vectors with $n$ components from $\mathbb{K}$. The transposition of a vector $x$ is $x^{T}$.

### 2.2 Graph Theory

Formally, an (undirected) graph is a triple $G=\left(V, E, \Psi_{1}\right)$ consisting of a nonempty set $V$, called the nodes (or vertices), a set $E$, called the edges (or links), and a relation of incidence $\Psi_{1}: E \rightarrow V \times V$ that associates with each edge two nodes, called its ends. Usually we just write $G=(V, E)$ and assume that the incidence relation is given implicitly in $E$. For each edge $e \in E$ there exist nodes $u, v \in V$ such that $\Psi_{1}(e)=\{u, v\}=\{v, u\}$. Two nodes that are ends of an edge are adjacent to one another (neighbors). The degree $|\delta(v)|$ of a node $v$ is the number of incident edges to $v$. An edge with identical ends is called a loop. If two edges join the same pair of ends, they are called parallel. A graph is simple if it has neither loops nor parallel edges. For a subset $W \subseteq V$ of nodes, $E(W) \subseteq E$ denotes the subset of edges with both ends in $W$.

A digraph (directed graph) is a triple $G=\left(V, A, \Psi_{1}\right)$ consisting of a nonempty set $V$, called the nodes (or vertices), a set $A$, called the arcs, and a relation of incidence $\Psi_{1}: A \rightarrow V \times V$ that associates with each arc an ordered pair of nodes, called its ends. Usually, we just write $D=(V, A)$ and assume that the incidence relation is given implicitly in $A$. For each arc $a=(u, v)$ we call $u$ the source and $v$ the target of $a$. Parallel arcs and loops are defined as for undirected graphs. Two arcs $(u, v)$ and $(v, u)$ are called associated. For an arc $(u, v)$, the arc $(v, u)$ is called its associated backward arc. A digraph where each arc has its associated backward arc is called bidirectional. We call the graph $G=(V, E)$ the underlying graph of the digraph $D=(V, A)$ if there is a bijection between the arcs of $D$ and the edges of $G$, such that for each arc $a=(u, v) \in A$ there is an edge $e=\{u, v\} \in E$, and for each edge $e=\{u, v\} \in E$ the arc $a=(u, v)$ and the arc $a^{\prime}=(v, u)$ are in $A$. The overlaying digraph $D(G)$ of a graph $G$ is the digraph obtained from $G$ by replacing each edge by two associated arcs with the same ends.

In the following, let $G=(V, E)$ denote a graph and $D=(V, A)$ a digraph.

A path $\mathfrak{p}$ in $G$ (or a directed path in $D$ ) from $v_{0}$ to $v_{l}$ is a sequence $\mathfrak{p}=$ $\left(v_{0}, e_{1}, v_{1}, \ldots, e_{l}, v_{l}\right)$ of nodes $v_{0}, \ldots, v_{l} \in V$ and edges (arcs) $e_{1}, \ldots, e_{l} \in V(\in A)$ of $G$ $(D)$, such that the nodes $v_{i-1}$ and $v_{i}$ are the ends of edge $e_{i}$ (are source and target of $e_{i}$ ) for each $1 \leq i \leq l$. Node $v_{0}$ is called the source and $v_{l}$ the target of $\mathfrak{p}$, while both are denoted as the endnodes of $\mathfrak{p}$. The nodes $v_{1}, \ldots, v_{l-1}$ are called the inner nodes of $\mathfrak{p}$. The length of a path is the number of edges (arcs). We use the notation $e \in \mathfrak{p}$ $(a \in \mathfrak{p})$ or $v \in \mathfrak{p}$, if $e \in E(a \in A)$ is an edge (arc) of $\mathfrak{p}$ or $v \in V$ is a node of $\mathfrak{p}$. We denote by $V(\mathfrak{p})$ and $E(\mathfrak{p})(A(\mathfrak{p}))$ the set of inner nodes and edges (arcs) of $\mathfrak{p}$. That is, for a path $\mathfrak{p}=\left(v_{0}, e_{1}, v_{1}, \ldots, e_{l}, v_{l}\right)$ in $G(D)$ we have $V(\mathfrak{p})=\left\{v_{1}, v_{2}, \ldots, v_{l-1}\right\}$ and $E(\mathfrak{p})=\left\{v_{1}, v_{2}, \ldots, v_{l-1}\right\}(=A(\mathfrak{p}))$. We will use the term simple path to denote paths without node repetition. A (simple) cycle is a (simple) path where the endnodes are identical. Two paths $\mathfrak{p}_{1}$ and $\mathfrak{p}_{2}$ are node-disjoint if $V\left(\mathfrak{p}_{1}\right) \cap V\left(\mathfrak{p}_{2}\right)=\emptyset$. Analogously, $\mathfrak{p}_{1}$ and $\mathfrak{p}_{2}$ are edge-disjoint (arc-disjoint) if $E\left(\mathfrak{p}_{1}\right) \cap E\left(\mathfrak{p}_{2}\right)=\emptyset\left(=A\left(\mathfrak{p}_{1}\right) \cap A\left(\mathfrak{p}_{2}\right)\right)$.

A graph $\tilde{G}=(\tilde{V}, \tilde{E})$ is a subgraph of $G=(V, E)$ if $\tilde{V} \subseteq V$ and $\tilde{E} \subseteq E$. A graph $G=(V, E)$ is said to be connected if there is a path between any two nodes. A graph $G$ is $\mathbf{k}$-node(edge)-connected if there exist $k$ node (edge)-disjoint paths between any two nodes. A tree is a connected graph with no cycles. A spanning tree is a subgraph of $G$ which has the same set of nodes of $G$ and is a tree.

If $G=(V, E)$ is a graph and $X \subseteq V$, then the set of edges $\delta(X):=\{\{u, v\} \in$ $E \mid u \in X, v \notin X\}$ is a cut. For a digraph $D=(V, A)$ and a subset $X \subseteq V$ of nodes, let $\delta(X)^{+}:=\{(u, v) \in A \quad \mid \quad u \in X, v \notin X\}, \delta(X)^{-}:=\delta(V \backslash X)^{+}$and $\left.\delta(X):=\delta^{+}(X) \cup \delta\right)^{-}(X)$. The arcset $\delta(X)^{+}$is called a directed cut.

### 2.3 Polyhedral Theory

A set $X \subseteq \mathbb{K}^{n}$ is bounded, if $M \in \mathbb{K}_{+}$exists with $\|x\| \leq M$ for all $x \in X$ and some norm $\|\cdot\|: \mathbb{K}^{n} \rightarrow \mathbb{K}_{+}$. For $X \subseteq \mathbb{K}^{n}$ we define:

$$
\begin{aligned}
& \operatorname{aff}(X) \quad:=\left\{x \in \mathbb{K}^{n} \mid\right. \exists \lambda_{1}, \ldots, \lambda_{t} \in \mathbb{K} \text { and } \exists x_{1}, \ldots, x_{t} \in X, t \in \mathbb{N} \text { s.t. } \\
&\left.\sum_{i=1}^{t} \lambda_{i}=1 \text { and } x=\sum_{i=1}^{t} \lambda_{i} x_{i}\right\}, \\
& \operatorname{conv}(X):=\left\{x \in \mathbb{K}^{n} \left\lvert\, \begin{array}{l}
\exists \lambda_{1}, \ldots, \lambda_{t} \in \mathbb{K}_{+} \text {and } \exists x_{1}, \ldots, x_{t} \in X, t \in \mathbb{N} \text { s.t. } \\
\\
\\
\end{array} \sum_{i=1}^{t} \lambda_{i}=1\right. \text { and } x=\sum_{i=1}^{t} \lambda_{i} x_{i}\right\}
\end{aligned}
$$

to be the affine and convex hull of $X$, respectively. The dimension $\operatorname{dim}(W)$ of $W$ is $\operatorname{dim}(\operatorname{aff}(X))$, that is the maximum number of linear independent vectors in aff $(X)$. If $\operatorname{dim}(W)=n$, we call $W$ full-dimensional.

Given $a \in \mathbb{K}^{n} \backslash\{0\}$ and $\alpha \in \mathbb{K}$, the set $\left\{x \in \mathbb{K}^{n} \mid a^{T} x \leq \alpha\right\}$ is a halfspace and $\left\{x \in \mathbb{K}^{n} \mid a^{T} x=\alpha\right\}$ is a hyperplane. The finite intersection of halfspaces given by $\left\{x \in \mathbb{K}^{n} \quad \mid A x \leq b\right\}$ with $A \in \mathbb{K}^{m \times n}$ and $b \in \mathbb{K}^{m}$ is a polyhedron, where $\mathbb{K}^{m \times n}$ is the space of matrices with $m$ rows and $n$ columns. A bounded polyhedron is called a polytope.

For the polyhedron $\mathcal{P}=\left\{x \in \mathbb{K}^{n} \mid A x \leq b\right\}$ and $M=\{1, \ldots, m\}$, the set $M^{=}=\left\{i \in M \quad \mid \quad a_{i} x=b_{i} \forall x \in \mathcal{P}\right\}$ is called the equality set of $\mathcal{P}$ and the set $M^{\leq}=M \backslash M^{=}$the inequality set. Let $\left(A^{=}, b^{=}\right)$and $\left(A^{\leq}, b^{\leq}\right)$the corresponding rows of $(A, b)$.

The inequality $a^{T} x \leq \alpha$ for $a \in \mathbb{K}^{n}, \alpha \in \mathbb{K}$ is valid for a polyhedron $\mathcal{P}$, if $\mathcal{P} \subseteq\{x \in$ $\left.\mathbb{K}^{n} \mid a^{T} x \leq \alpha\right\}$, and is tight for $\mathcal{P}$ if it is valid and $\mathcal{F}_{a, \alpha}=\mathcal{P} \cap\left\{x \in \mathbb{K}^{n} \mid a^{T} x=\alpha\right\} \neq$ $\emptyset$. We say, $\mathcal{F}_{a, \alpha}$ is a face of $\mathcal{P}$ induced by $a^{T} x \leq \alpha$. A face $\mathcal{F} \neq \mathcal{P}$ of a polyhedron $\mathcal{P}$ is a facet of $\mathcal{P}$ if it is maximal with respect to inclusion. If $a^{T} x \leq \alpha$ is valid for $\mathcal{P}$ and $\mathcal{F}=\mathcal{P} \cap\left\{x \in \mathbb{K}^{n} \mid a^{T} x=\alpha\right\}$ is a facet of $\mathcal{P}$, we say that $a^{T} x \leq \alpha$ is facetdefining. An equivalent characterization of a facet is that, $\operatorname{dim}(\mathcal{F})=\operatorname{dim}(\mathcal{P})-1$. If a polyhedron $\mathcal{P}$ is full-dimensional, and $a^{T} x \leq \alpha$ and $b^{T} x \leq \beta$ are facet-defining with $\mathcal{F}_{a, \alpha}=\mathcal{F}_{b, \beta}$, then there exist $\lambda \in \mathbb{R}_{+}$with $\lambda a=b$ and $\lambda \alpha=\beta$.
If the polyhedron $\mathcal{P}$ is not full-dimensional and $a^{T} x \leq \alpha$ and $b^{T} x \leq \beta$ represent the same facet, then one facet can be obtained from the other by multiplication by a positive scalar and then adding multiples of equations of the linear equation system $\left(A^{=}, b^{=}\right)$of $\mathcal{P}$.

## Chapter 3

## The network design problem

In this chapter, we present the given problem description and point out possible real world relations. Afterwards, a mathematical model is constructed which models the specified network planning problem.

The initial objective of the ROCOCO project was to design an algorithm providing best solutions and/or lower bounds on average for all instances of the applied benchmark suite and all combinations of additional constraints. For each instance and combination a CPU limit of 10 minutes was given. A wide range of techniques have been tested, such as Constraint Programming, Local Search, Linear Programming, etc. The best published upper bounds so far have been obtained by a hybrid algorithm of Constraint Programming and Local Search, presented in [6]. A newer approach comes from Alain Chabrier who applied a Heuristic Branch-and-Price-and-Cut procedure [7]. But until now, only outlines of solution approaches exist in literature and no publicly accessible model including all constraints has been published so far.

### 3.1 Problem description

In this section, we introduce the information available to us, including an attempt at explanation towards reality and importance.

The network design and routing problem studied in the ROCOCO project can be described as follows: Given is a set of nodes and a set of arcs corresponding to all potential connections between the nodes. The graph has no parallel arcs and is bidirectional. There also exists a set of directed demands representing the communication demand between two locations (nodes), which have to be routed over the network along a single path from their origin to their destination node. When these demands are routed over some arc they consume capacity in the corresponding direction. The problem consists of selecting from a discrete set of possible capacities which one to install on each arc and how often. The capacities must be chosen in such a way, that all demands can be routed over the network simultaneously without exceeding the capacities and the
capacity installation cost is minimal. Also some additional constraints concerning the routing or the capacity choices are present and arbitrarily considered.
Each arc has an indexed set of directed base capacities, out of which at most one can be chosen. Associated arcs have the same number of base capacities and the corresponding choices are linked. The capacity can be increased by an integer multiplier coming out a certain range assigned by the base capacity. Thus the overall capacity of the arc is the product of the base capacity and the multiplier. The choices of an arc determine the choices of the associated backward arc. The multiplier of associated arcs has to be the same, as well as the index of the base capacity. This does not necessarily result in the same amount of capacity for both arcs, since the sets of base capacities may vary. The overall cost for both arcs results from the product of the cost of the base capacity and the chosen multiplier.

Example 3.1 For further explanation, consider the example shown in Figure 3.1. For arc $(a, b)$ from node $a$ to node $b$ the base capacity $\operatorname{Capa}_{1}(a, b)$ is given with the associated multiplier range from 0 to 3 . This capacity is chosen, such that the choice for the arc $(b, a)$ is imposed at the same time: $(b, a)$ is equipped with $C a p a_{1}(b, a)$. The multiplier for both is the same (in our case 3 , the available maximum) and the overall costs for both were calculated by the cost of the base capacity $\operatorname{Capa}_{1}(a, b)$ times 3 . For the $\operatorname{arc}(d, a)$, the range of the multiplier associated to the base capacity $C a p a_{2}(d, a)$ is from 1 to 2 . Thus, if this capacity is chosen on the arc, it must be at least equipped once with this capacity.

Figure 3.1: Example of network with capacity installation.

Part of the benchmark suite is also the consideration of six constraints which were added to the problem in arbitrary combinations.
sec Each node, capacity and demand has an indicator which determines its security level. It can be risky or secure. When this constraint is taken into account, a secure demand must only be routed over secure nodes and arcs which are equipped with secure capacity. For the remaining demands, this constraint is of no matter. Also the secure status of the endnodes of any demand is not relevant. This constraint can be interpreted regarding the anti-interception as well as the survivability aspect. For communication demands transporting sensitive data, there could exist a set of specially shielded nodes and capacities (like ssh- or https-connections). Alternatively and under the condition of the existence of secure, indestructible nodes and links (for which we have a guarantee that they never collapse), this constraint ensures the satisfiability of certain communication demands at any rate.
nomult This constraint forbids the multiplication of the base capacities of arcs. For each arc we have to consider two cases:

If there exists a base capacity with a lower bound of the associated multiplier range greater than 0 , this capacity has to be chosen with the minimal multiplier (if we want to install any capacity on this arc). The arc can not be equipped with any other base capacity.
Otherwise the choice of capacity is free but a multiplier less than or equal to 1 is imposed.

This leads to the application of large capacities on the arcs instead of multiple smaller ones.

Example 3.2 We recall the example in Figure 3.1. It follows that for the arcs $(a, d)$ and $(d, a)$ the capacity and multiplier choice is imposed. Concerning the base capacity $\operatorname{Capa}_{2}(a, d)$, the lower bound of the associated multiplier is greater than 0 . Therefore the arcs $(a, d)$ and $(d, a)$ must be equipped with the base capacity $\operatorname{Capa}_{2}(a, d)$ and $\operatorname{Capa}_{2}(d, a)$ respectively, and the corresponding multiplier is 1 .
For the arcs $(a, b)$ and $(b, a)$, the possible multiplier regarding the capacities $\operatorname{Capa}_{1}(a, b)$ and $\operatorname{Capa}_{1}(b, a)$ respectively, is 0 or 1 (and this only if there is no other base capacity available with a lower bound of the associated multiplier greater than 0 ).

Remark 3.3 The given problem description does not specify how to deal with two or more base capacities having a lower bound of the associated multiplier range greater than 0 . We assume that this cannot happen.
symdem This constraint states that two demands where the origin of one is the destination of the other and vice versa, have to be routed symmetrically. Hence for a demand from node $a$ to node $b$, if there exists a demand from $b$ to $a$, then the paths used to route these demands must be symmetric, i.e. that the two paths were indicated by the same node set and differ only in an opposite order of the nodes. From this it follows that demands with the same origin and destination have to use the same path, if there exists at least one reverse demand for them. This symmetric routing is required by some routing protocols, e.g. the Network Time Protocol (NTP) [1] which is used to synchronize the time of devices in a network.
bmax Each demand has a parameter limiting the length of its routing path.
This limit is normally referred to as hoplimit, as it takes into account only the amount of arcs used by the path, and not its actual physical length. It avoids unacceptably long paths and decreases the probability of a failure of a path component or transmission delays.
pmax For each node an upper bound on incoming and outgoing ports is imposed. The capacities installed on the incoming arcs consume the inports amounting to their associated multipliers and analogously for the outgoing arcs and the outports. If the particular directed capacity is zero, no ports are consumed.
This constraint ensures that the ports are not exhausted. It makes sense if the node configuration is predetermined, as well as for survivability. If for a node the number of incident arcs on which capacity can be installed is restricted, only these arcs are affected in case of a failure of this node.

Example 3.4 Recall the example of Figure 3.1: for the arc $(a, c)$ this means that the chosen capacity and associated multiplier require 2 inports at node $c$ and 2 outports at node $a$. The base capacity installed on the associated backward arc $(c, a)$ does not require any ports, because it does not provide positive capacity. Since the node $c$ has only one outport, it follows that at most one outgoing arc can be equipped with a capacity, and thus used by some routing.
tmax This constraint associates an upper traffic bound to each node. The sum of all demands routed from, to or through a node must not exceed this fixed capacity. Like the pmax constraint, this one has two aspects. The configuration on the node may be predetermined or the network designer wants to route only a certain amount of data through the node to limit the probably vulnerable data.

Remark 3.5 In the ROCOCO project, all possible combinations of these additional constraints were considered. But the restriction to some, more plausible/probable variants may be worth considering.
The bundle of tmax and pmax constraints would be convenient in case of a predetermination of the technology installed at the nodes. If there exists a restriction on
the nodes it seems plausible that the number of ports is limited as well as the traffic. The nomult can be related to these constraints as well, because it reduces port consumption of the links regarding their endnodes and thus helps to fulfill the pmax constraint.
Concerning the survivability aspect, the combination of the bmax and the sec constraint seems reasonable. The bmax constraint decreases the probability of a failure of a path component and the sec constraint ensures the satisfiability of certain communication demands at any rate. Also the restrictions on the nodes (tmax, pmax) can be imposed in this context.

### 3.2 Model

In this section we introduce some notation and our mathematical model for the described problem. The constraints are classified in two subproblems.

Base model All demands have to be routed simultaneously over a single cycle-free path from their origin to their destination (monorouting). In the target network, sufficient capacity must be installed on the edges such that these can accommodate a feasible routing of the communication demands.

Additional constraints This part of the model is dedicated to the additional constraints. We derive inequalities for the sec, pmax, bmax and tmax constraints. At first we also deduce inequalities for the symdem constraint, but we finally integrate it into the data transformation, as well as the nomult requirements.

Remark 3.6 Due to the dependency of the capacity choice for a forward and the associated backward arc, we deviate from the arc view. We consider only undirected edges which can be equipped with linkdesigns providing directed capacities. It is only necessary to make one choice per edge, while the various capacities per direction and the multiplier are mapped to some linkdesigns. The result provides the amount of directed edge capacity.

### 3.2.1 Notation and parameters

Notation 3.7 (Telecommunication network) We denote the considered telecommunication network by a graph $G=(V, E)$, with $V$ the set of nodes representing locations. The edges $e \in E$ represent the possible connections between the locations.

Notation 3.8 (Nodes) We denote the traffic limit of a node by the parameter node capacity which is consumed by the demands routed over. Each node $v \in V$ has the parameters:
$P_{v}^{-} \in \mathbb{Z}_{+}$The amount of incoming ports.
$P_{v}^{+} \in \mathbb{Z}_{+}$The amount of outgoing ports.
$C_{v} \in \mathbb{Z}_{+} \quad$ The provided capacity.
$S_{v} \in\{0,1\}$ The security status of a node is 1 if it is secure, 0 otherwise.
Notation 3.9 (Linkdesigns) The (finite) set of all linkdesigns which can be installed on an edge $\{u, v\} \in E$ is denoted by $\mathcal{L}_{\{u, v\}}$. Each edge can be equipped with at most one linkdesign. In the elements of the linkdesign set we merge the choice of the capacity and the associated multiplier. Therefore, for each edge and each base capacity of the associated arc arise as many linkdesigns as integers can be found in the associated multiplier range joint with the $\{0\}$-set. In that way we get two directed capacities per edge which are implied by the base capacities of the associated arcs and the multiplier. The transformation is analogously the same with the costs and the ports required. Other available base capacities for this arc should be dealt with analogously and the created linkdesigns should be added in the associated set $\mathcal{L}_{\{u, v\}}$.

At this point we already include the nomult constraint. If we take it into account, the set of linkdesigns is restricted to the linkdesigns with a port consumption of 1 or (when such one exists) the linkdesign with the corresponding multiplier lower bound greater than 0 .

For a linkdesign $l \in \mathcal{L}_{\{u, v\}}$ corresponding to the edge $\{u, v\} \in E$, the following parameters are given:
$C_{u v}^{l} \in \mathbb{Z}_{+} \quad$ The directed integer routing capacity from node $u$ to node $v$.
$C_{v u}^{l} \in \mathbb{Z}_{+} \quad$ The directed integer routing capacity from node $v$ to node $u$.
$P_{u v}^{l} \in \mathbb{Z}_{+} \quad$ The amount of ports required for the direction $u$ to $v$
i.e. the number of inports consumed at $v$ and the number of outports consumed at $u$.
$P_{v u}^{l} \in \mathbb{Z}_{+} \quad$ The amount of ports required for the direction $v$ to $u$. i.e. the number of inports consumed at $u$ and the number of outports consumed at $v$.
$S_{\{u, v\}}^{l} \in\{0,1\}$ The security status of a linkdesign is 1 if it is secure, 0 otherwise. $W_{\{u, v\}}^{l} \in \mathbb{Z}_{+} \quad$ The installation costs for linkdesign $l$ on the edge $\{u, v\}$.

Remark 3.10 We have a parameter for the number of ports required for each direction. Actually the consumption of ports is equal for both directions, but the special case that the capacity for one direction is 0 while the other is greater than 0 makes it necessary to split the parameter.

Example 3.11 Recalling the example shown in Figure 3.1, consider the pair of arcs between the nodes $a$ and $b$. For $\operatorname{Capa}_{1}(a, b)$ a multiplier range from 0 to 3 is given. We assume that the base capacity value of $\operatorname{Capa}_{1}(a, b)$ is $\alpha$ and the one of $\operatorname{Capa}_{1}(b, a)$ is $\beta$, it is secure and the base cost is $\varpi$. Our transformation creates for each possible multiplier a different linkdesign, so that we get 4 linkdesigns for the edge $e=\{a, b\}$ out of the base
capacity $\operatorname{Capa}_{1}(a, b)$. The edge capacity regarding a certain direction is the product of the base capacity of the associated arc with the concerned multiplier. The cost for the linkdesign is the multiplier times the cost for $\operatorname{Capa}_{1}(a, b)$. The concerned multiplier also imposes the required ports (but the associated capacities must be regarded to ensure that it is not 0). In Table 3.1 we present the linkdesigns created from $\operatorname{Capa}_{1}(a, b)$ and Capa $_{1}(b, a)$.

| LD | $C_{a b}^{l}$ | $C_{b a}^{l}$ | $P_{a b}^{l}$ | $P_{b a}^{l}$ | $S_{\{a, b\}}^{l}$ | $W_{\{a, b\}}^{l}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $l_{0}$ | 0 | 0 | 0 | 0 | 1 | 0 |
| $l_{1}$ | $\alpha$ | $\beta$ | 1 | 1 | 1 | $\varpi$ |
| $l_{2}$ | $2 \alpha$ | $2 \beta$ | 2 | 2 | 1 | $2 \varpi$ |
| $l_{3}$ | $3 \alpha$ | $3 \beta$ | 3 | 3 | 1 | $3 \varpi$ |

Table 3.1: Generated linkdesigns

Due to the nomult constraint, the set of linkdesigns $\mathcal{L}_{\{a, d\}}$ consists of only one element: The generated linkdesign corresponding to the base capacities $\operatorname{Capa}_{1}(a, d)$ and $\operatorname{Capa}_{1}(d, a)$ respectively, and the imposed multiplier 1.

Notation 3.12 (Commodities) In our model, all commodities consist of exactly one demand. In order to handle the secure routing condition we introduce a secure value for each commodity. This secure value is equal to the demand value when it has to be routed securely and otherwise 0 . We denote by $\mathcal{K}$ the set of commodities. For a commodity $k \in \mathcal{K}$ the following parameters are specified:
$o_{k} \in V \quad$ The node where the demand comes from (origin).
$d_{k} \in V \quad$ The node to which the demand has to be routed to (destination).
$U_{k} \in \mathbb{Z}_{+} \quad$ The amount of data that has to be routed from $o_{k}$ to $d_{k}$.
$S_{k} \in\{0,1\}$ The security status of a commodity is 1 if it has to be routed securely, 0 otherwise.
$U_{k}^{S} \in \mathbb{Z}_{+}$The amount of data that has to be routed from $o_{k}$ to $d_{k}$ securely, $U_{k}^{S}=U_{k} \cdot S_{k}$.
$M_{k} \in \mathbb{Z}_{+}$The maximal number of hops, i.e. the allowed number of edges of an $o_{k}-d_{k}$-path.

Remark 3.13 In literature on multicommodity network flow problems, the demands are often aggregated with respect to their source (or destination) nodes. One commodity for each node with positive supply is defined [5]. The aggregated formulation has a smaller amount of commodities compared to our approach, which often leads to a significantly reduced size of the problem.
However, in our problem no such aggregation was possible since in the aggregated version, no proper control of the fulfillment of the varying demand properties is possible. Since the requirements on the demands are quite different (concerning the sec, bmax and symdem constraints) bundling demands would be impossible.

### 3.2.2 Variables

We formulate the described problem with the following variables, where one variable corresponds to the routing of commodities in the network and the other one to the installation of linkdesigns on an edge of the network.

## Commodity edge flow variables

The routing of the commodities in our network is modeled via directed binary commodity edge flow variables. For each edge $\{u, v\} \in E$ and each commodity $k \in \mathcal{K}$, the binary variable $f_{u v}^{k}$ decides whether $k$ is routed over $\{u, v\}$ in the direction $u$ to $v$ or not. Analogous for the variable $f_{v u}^{k}$ concerning the direction $v$ to $u$. This yields for all $k \in \mathcal{K}$ and $\{u, v\} \in E$

$$
\begin{aligned}
& f_{u v}^{k}= \begin{cases}1 & \text { if } k \text { is routed over } e \text { in the direction } u \text { to } v \\
0 & \text { otherwise }\end{cases} \\
& f_{v u}^{k}= \begin{cases}1 & \text { if } k \text { is routed over } e \text { in the direction } v \text { to } u \\
0 & \text { otherwise }\end{cases}
\end{aligned}
$$

## Linkdesign installation variables

For each edge $e \in E$ and linkdesign $l \in \mathcal{L}_{e}$ we introduce the binary variable $x_{e}^{l}$ indicating if $l$ is installed on $e$. Therefore for all $e \in E$ and $l \in \mathcal{L}_{e}$

$$
x_{e}^{l}= \begin{cases}1 & \text { if } l \text { is installed on } e \\ 0 & \text { otherwise }\end{cases}
$$

Remark 3.14 Another alternative to modeling the choice of technology for the edges would be the application of an integer variable for all available associated base capacities: greater 0 iff the capacity is chosen and the value denoting the chosen multiplier. This would mean a significant decrease of linkdesigns but the range for these variables would be the union of $\{0\}$ and the multiplier range. The case of a base capacity with a lower bound of the corresponding multiplier range greater than 1 would be a problem. The 1 would be prohibited while 0 and the integral lower bound would be feasible. This gap would provoke problems in our further solution approaches.
The maximum multipliers in our data set are furthermore quite small (at most 5) and so the disprofit of the greater amount of $\{0,1\}$-variables is not that serious. On the other hand, the binary range of the variables has a lot of advantages, as we will see later.

### 3.2.3 Inequalities

There exist various base constraints which have to be complied by a feasible routing and linkdesign installation. We first model these conditions in the following subsection and later introduce the inequalities that correspond to the additional constraints.

## Inequalities for the base model

In this subsection we introduce the inequalities concerning the base model. We require all demands to be routed over a single cycle-free path from their origin to their destination. For each edge we can choose at most one linkdesign in a way such that the provided capacity can accommodate a feasible routing of all commodities simultaneously.

Linkdesign installation constraint. In this inequality we model the constraint that only one capacity can be chosen on each arc as stated in the given problem description. Therefore each edge $e \in E$ can only be equipped with at most one linkdesign $l \in \mathcal{L}_{e}$.

$$
\begin{equation*}
\sum_{l \in \mathcal{L}_{e}} x_{e}^{l} \leq 1 \quad \forall e \in E \tag{3.1}
\end{equation*}
$$

Flow balance constraint. To ensure that the demand between two nodes is satisfied using a single path and to state that the flow balance is fulfilled we introduce this constraint.

$$
\sum_{\{u, v\} \in \delta(v)}\left(f_{u v}^{k}-f_{v u}^{k}\right)=\left\{\begin{array}{ll}
-1 & \text { if } v=o_{k}  \tag{3.2}\\
1 & \text { if } v=d_{k} \\
0 & \text { otherwise }
\end{array} \quad \forall v \in V, k \in \mathcal{K}\right.
$$

This constraint provides for each commodity $k \in \mathcal{K}$ a single $o_{k}-d_{k}$-path since the origin node $o_{k}$ has a net outflow of 1 , the destination node $d_{k}$ a net inflow of 1 and exploiting the fact that we use binary commodity edge flow variables. The flow has no chance to get lost, because all other nodes have a net flow of 0 . But in this path can occur cycles, that is why we introduce the following constraint.

Subtour elimination constraint. To avoid cycles in our routing, we restrict for each commodity $k \in \mathcal{K}$ and each subset $W$ of $V$ the number of edges in $E(W)$ which are used by this commodity.

$$
\begin{equation*}
\sum_{\{u, v\} \in E(W)}\left(f_{u v}^{k}+f_{v u}^{k}\right) \leq|W|-1 \quad \forall W \subseteq V, k \in \mathcal{K} \tag{3.3}
\end{equation*}
$$

These constraints prevent the occurrence of cycles, because a commodity $k \in \mathcal{K}$ cannot be routed twice through a node $v \in V$.

Edge capacity constraints. In the ROCOCO problem the arc capacities must not be exceeded. Consequentially, the directed capacity of an edge provided by the installed linkdesign must not be overstepped.

$$
\begin{array}{ll}
\sum_{k \in \mathcal{K}} U_{k} f_{u v}^{k} \leq \sum_{l \in \mathcal{L}_{\{u, v\}}} C_{u v}^{l} x_{\{u, v\}}^{l} & \forall\{u, v\} \in E \\
\sum_{k \in \mathcal{K}} U_{k} f_{v u}^{k} \leq \sum_{l \in \mathcal{L}_{\{u, v\}}} C_{v u}^{l} x_{\{u, v\}}^{l} & \forall\{u, v\} \in E \tag{3.5}
\end{array}
$$

## Inequalities for the additional constraints

In this subsection we incorporate the additional constraints in the mathematical model. The nomult constraints have already been included. We express the requirements of the pmax, tmax and bmax constraints by introducing new inequalities for the model. The sec constraint is split into two parts: one concerning the edges and the other considering the secure node throughput. The symdem constraint is integrated by a commodity merge process and the adaptation of some of the preestablished model constraints.
The next two inequalities include the pmax constraint in our model.

Inport constraints. Each node provides a certain number of inports which are consumed by the linkdesigns installed on the incident edges and must not be exhausted.

$$
\begin{equation*}
\sum_{\{u, v\} \in \delta(v)} \sum_{l \in \mathcal{L}_{\{u, v\}}} P_{u v}^{l} x_{\{u, v\}}^{l} \leq P_{v}^{-} \quad \forall v \in V \tag{3.6}
\end{equation*}
$$

Outport constraints. Each node provides a certain number of outports which are consumed by the linkdesigns installed on the incident edges and must not be exhausted.

$$
\begin{equation*}
\sum_{\{u, v\} \in \delta(v)} \sum_{l \in \mathcal{L}_{\{u, v\}}} P_{v u}^{l} x_{\{u, v\}}^{l} \leq P_{v}^{+} \quad \forall v \in V \tag{3.7}
\end{equation*}
$$

The following type of inequalities integrates the tmax constraint in our model.
Node capacity constraints. The amount of data that is conducted through $v \in V$ is the sum of the values of the commodities that flow into $v$ plus the commodities that have their origin in $v$. This throughput must not exceed the node capacity.

$$
\begin{equation*}
\sum_{k \in \mathcal{K}} \sum_{\{u, v\} \in \delta(v)} U_{k} f_{u v}^{k} \leq C_{v}-\sum_{k \in \mathcal{K}: v=o_{k}} U_{k} \quad \forall v \in V \tag{3.8}
\end{equation*}
$$

Remark 3.15 This constraint differs from the usually considered switching capacity which gives a limit on the capacity of its incident edges and which is therefore normally modeled via linkdesign installation variables (e.g. [12]). The considered tmax takes into account the flow that is actually routed over the node and not only the flow that could be routed potentially like the switching capacity.

For an easier integration of the sec constraint we split it in two parts. One type of constraint focuses on the edges and the other type on the nodes.

Edge secure capacity constraints. If a secure demand $k \in \mathcal{K}$ with $S_{k}=1$ is routed over an edge $e \in E$ it requires this edge to be equipped with a secure linkdesign $l \in \mathcal{L}_{e}$ with $S_{e}^{l}=1$. We achieve this by introducing a secure capacity which is only provided by secure linkdesigns and consumed by secure commodities. Over an edge $e \in E$ regarding a certain direction, only as much secure data can be routed as directed secure capacity is provided. The amount of secure capacity is determined by the installed linkdesigns and their secure status.

$$
\begin{align*}
& \sum_{k \in \mathcal{K}} U_{k}^{S} f_{u v}^{k} \leq \sum_{l \in \mathcal{L}_{\{u, v\}}} S_{\{u, v\}}^{l} C_{u v}^{l} x_{\{u, v\}}^{l} \quad \forall\{u, v\} \in E  \tag{3.9}\\
& \sum_{k \in \mathcal{K}} U_{k}^{S} f_{v u}^{k} \leq \sum_{l \in \mathcal{L}_{\{u, v\}}} S_{\{u, v\}}^{l} C_{v u}^{l} x_{\{u, v\}}^{l} \quad \forall\{u, v\} \in E \tag{3.10}
\end{align*}
$$

Node secure capacity constraints. If a node $v \in V$ is risky, a secure commodity $k \in \mathcal{K}$ must not be routed over it. We ensure this constraint by prohibiting that a secure commodity is routed over edges incident to $v$ (except the case that $v=o_{k}$ or $v=d_{k}$ ).

$$
f_{u v}^{k} \leq 1-S_{k}+ \begin{cases}1 & \text { if } \begin{array}{l}
u \in\left\{o_{k}, d_{k}\right\}, \\
v \in\left\{o_{k}, d_{k}\right\}
\end{array}  \tag{3.11}\\
S_{v} & \text { if } \begin{array}{l}
u \in\left\{o_{k}, d_{k}\right\}, \\
v \notin\left\{o_{k}, d_{k}\right\}
\end{array} \quad \forall\{u, v\} \in E, k \in \mathcal{K} \text {. } \quad \begin{array}{l}
u \notin\left\{o_{k}, d_{k}\right\}, \\
v \in\left\{o_{k}, d_{k}\right\}
\end{array} \\
S_{u} & \text { if } \\
S_{u} S_{v} & \text { otherwise }\end{cases}
$$

$$
f_{v u}^{k} \leq 1-S_{k}+ \begin{cases}1 & \text { if } \begin{array}{l}
u \in\left\{o_{k}, d_{k}\right\} \\
v \in\left\{o_{k}, d_{k}\right\}
\end{array}  \tag{3.12}\\
S_{v} & \text { if } \left.\begin{array}{l}
u \in\left\{o_{k}, d_{k}\right\}, \\
v \notin\left\{o_{k}, d_{k}\right\}
\end{array} \quad \forall\{u, v\} \in E, k \in \mathcal{K}\right\} \\
S_{u} \quad \text { if } \begin{array}{l}
u \notin\left\{o_{k}, d_{k}\right\}, \\
v \in\left\{o_{k}, d_{k}\right\}
\end{array} \\
S_{u} S_{v} \quad \text { otherwise }\end{cases}
$$

The constraint is modeled in these inequalities. Let $k \in K$ and $e \in E$. In case of a risky commodity $\left(S_{k}=0\right)$ they hold in any case, as well as when the origin and the destination of the commodity are equal to the endnodes of $e$. When the commodity $k$ is secure the potential choice of edge $e$ for routing depends on the security status of the endnodes. If one endnode is the origin or destination of $k$, $e$ can only be used to route $k$ if the other node is secure. In the case of a secure commodity where any of the endnodes of $e$ is neither origin nor destination of it, the edge is prohibited to use for routing $k$ if one of the node is not secure.

This constraint has actually not to be included in the formulation, because practically (3.11) and (3.12) are used in terms of variable reduction.

Path length limitation constraint. The bmax constraint is represented by this class of inequalities which claim that the number of edges over which a commodity $k \in \mathcal{K}$ is routed must not exceed the predetermined limit.

$$
\begin{equation*}
\sum_{\{u, v\} \in E}\left(f_{u v}^{k}+f_{v u}^{k}\right) \leq M_{k} \quad \forall k \in \mathcal{K} \tag{3.13}
\end{equation*}
$$

In the next paragraph we describe the mechanism used to integrate the symmetrical routing constraint symdem in the model.

Symmetry constraints. First we present some notation to simplify the discussion: For $k, k^{\prime} \in \mathcal{K}$ :

- $k, k^{\prime}$ are parallel if $o_{k}=o_{k^{\prime}}$ and $d_{k}=d_{k^{\prime}}$.
- $k, k^{\prime}$ are coupled if $o_{k}=d_{k^{\prime}}$ and $d_{k}=o_{k^{\prime}}$.
- $k, k^{\prime}$ are connected if $k, k^{\prime}$ are parallel or coupled.
- $k$ is isolated if no commodity is coupled with $k$.

We define commodity subsets, consisting of all commodities with the same origin and destination nodes:

$$
\mathcal{K}_{p q}:=\left\{k \in \mathcal{K} \mid o_{k}=p, d_{k}=q\right\} \quad \forall p, q \in V
$$

The symdem constraint demands that two coupled commodities have to be routed symmetrically. This means that if one commodity uses an edge $e \in$ $E$ in a certain direction, the other commodity must use the same edge but in the opposite direction. This constraint also implies that parallel commodities have to use the same edges in equal direction if at least one coupled commodity exists. For simplification we assume from now on that there exist only parallel commodities if, at the same time, also at least one coupled commodity exists for them. Therefore we can formalize the requirements as follows:

$$
\begin{align*}
f_{u v}^{k}=f_{v u}^{k^{\prime}} & \forall\{u, v\} \in E, k, k^{\prime} \in \mathcal{K}: k, k^{\prime} \text { coupled }  \tag{3.14}\\
f_{u v}^{k}=f_{u v}^{k^{\prime}} & \forall\{u, v\} \in E, k, k^{\prime} \in \mathcal{K}: k, k^{\prime} \text { parallel }  \tag{3.15}\\
f_{v u}^{k}=f_{v u}^{k^{\prime}} & \forall\{u, v\} \in E, k, k^{\prime} \in \mathcal{K}: k, k^{\prime} \text { parallel } \tag{3.16}
\end{align*}
$$

We use the variable-reducing character of this constraint by modifying the data and expanding a part of our model inequalities. Merging connected commodities into a new commodity scales down our problem size.

In the following, we present the merging process concerning commodities respective to two nodes $p, q \in V$. We merge all commodities from $\mathcal{K}_{p q} \cup \mathcal{K}_{q p}$ in a new commodity $k_{p q}$. This commodity is uniquely identified by the node pair since it is finally the only commodity with a relation to both nodes. All properties of the commodities are assigned to the aggregation $k_{p q}$, whereby we always pass on the strongest property. Afterwards we present how the respective variables of the commodities $k \in \mathcal{K}_{p q} \cup \mathcal{K}_{q p}$ can be merged into new variables for $k_{p q}$. Finally we dwell on the changes of the predeveloped model constraints becoming necessary as a result of the merging process.

Passing properties In the following we explain how the properties of the connected commodities are passed on to the newly generated commodity. Let $p, q \in V$.

Origin/Destination The new commodity $k_{p q}$ represents the communication demand between the nodes $p$ and $q$. It is still directed with the origin $p$ and the destination $q$, such that

$$
\begin{aligned}
o_{k_{p q}} & =p \\
d_{k_{p q}} & =q
\end{aligned}
$$

Value The value of $k_{p q}$ indicates the amount of data which has to be routed from $p$ to $q$, hence it is the sum of the values of all commodities $k \in \mathcal{K}_{p q}$.

Concerning the communication demand from $q$ to $p$ we introduce for each commodity $k \in \mathcal{K}$ a new parameter coupled value $\bar{U}_{k}$. This parameter indicates the amount of data that has to be routed from the destination node $d_{k}$ to the origin node $o_{k}$.

Considering the commodity $k_{p q}$, this means that the coupled value is the sum of the values of all commodities $k \in \mathcal{K}_{\text {q }}$. Now it follows that

$$
\begin{aligned}
& U_{k_{p q}}=\sum_{k \in \mathcal{K}_{p q}} U_{k}, \\
& \bar{U}_{k_{p q}}=\sum_{k \in \mathcal{K}_{q p}} U_{k} .
\end{aligned}
$$

Secure status All secure commodities have to be routed securely. So if at least one secure commodity $k \in \mathcal{K}_{p q} \cup \mathcal{K}_{q p}$ with $S_{k}=1$ exists, all commodities have to be routed securely, and as a consequence also $k_{p q}$. Because of this condition, the secure value of $k_{p q}$ is derived as followed:

$$
S_{k_{p q}}=\max _{k \in \mathcal{K}_{p q} \cup \mathcal{K}_{q p}} S_{k}
$$

Secure value The secure value of $k_{p q}$ takes the amount of the value of $k_{p q}$ if $k_{p q}$ has to be routed securely, otherwise 0 .
To cope with the reverse direction we introduce for each commodity $k \in \mathcal{K}$ the new parameter coupled secure value $\bar{U}_{k}^{S}$, equal to the coupled value $\bar{U}_{k}$ if $k$ has to be routed securely, otherwise 0 .
Therefore it follows that

$$
\begin{gathered}
U_{k_{p q}}^{S}=U_{k_{p q}} \cdot S_{k_{p q}}, \\
\bar{U}_{k_{p q}}^{S}=\bar{U}_{k_{p q}} \cdot S_{k_{p q}} .
\end{gathered}
$$

Hoplimit Since all commodities $k \in \mathcal{K}_{p q} \cup \mathcal{K}_{q p}$ have to use the same edges and since the path length limitation constraint has to be fulfilled for all $k$ at the same time, the used path must contain at most the same amount of edges as the minimal hoplimit of all commodities:

$$
M_{k_{p q}}=\min _{k \in \mathcal{K}_{p q} \cup \mathcal{K}_{q p}} M_{k}
$$

Example 3.16 The merging procedure concering three connected commodities into one is presented in Table 3.2.

Merging variables The commodity $k_{p q}$ has to be routed from $o_{k_{p q}}$ to $d_{k_{p q}}$ over the same path as all commodities $k \in \mathcal{K}_{p q}$. At the same time it has to use the same edges as all commodities $k \in \mathcal{K}_{q p}$, only in the opposite direction. This relation is formalized as follows:

$$
\begin{array}{ll}
f_{u v}^{k_{p q}}=f_{u v}^{k} & \forall\{u, v\} \in E, k \in \mathcal{K}_{p q} \\
f_{v u}^{k_{p q}}=f_{v u}^{k} & \forall\{u, v\} \in E, k \in \mathcal{K}_{p q} \\
f_{u v}^{k_{p q}}=f_{v u}^{k} & \forall\{u, v\} \in E, k \in \mathcal{K}_{q p} \tag{3.19}
\end{array}
$$

| commodity | $o_{k}$ | $d_{k}$ | $U_{k}$ | $\bar{U}_{k}$ | $S_{k}$ | $U_{k}^{S}$ | $\bar{U}_{k}^{S}$ | $M_{k}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $k_{1}$ | $p$ | $q$ | 12 | 0 | 0 | 0 | 0 | 6 |
| $k_{2}$ | $p$ | $q$ | 5 | 0 | 0 | 0 | 0 | 2 |
| $k_{3}$ | $q$ | $p$ | 13 | 0 | 1 | 13 | 0 | 4 |
| $k_{p q}$ | $p$ | $q$ | 17 | 13 | 1 | 17 | 13 | 2 |

Table 3.2: Merging of connected commodities. The three connected commodities $k_{1}, k_{2}$ and $k_{3}$ are merged in the new commodity $k_{p q}$

Effects of the transformation on the model inequalities: After this transformation, we consider which of our model constraints are affected by the changes. Starting with the original commodities $k \in \mathcal{K}_{\text {old }}$ and assuming that the symmetric routing constraints (3.14)-(3.16) are fulfilled, we argue why it is equivalent considering only the new merged commodities $k \in \mathcal{K}_{\text {new }}$ according to the fulfillment of the model constraints including minor modifications.

Let $p, q \in V$. If $k_{p q} \in \mathcal{K}_{\text {new }}$ fulfills the flow balance constraints (3.2), these are fulfilled by all commodities $k \in \mathcal{K}_{p q} \cup \mathcal{K}_{q p} \subseteq \mathcal{K}_{\text {old }}$, subject to being routed symmetrically. This also holds for the subtour elimination constraints (3.3) and by construction of $M_{k_{p q}}$ and $S_{k_{p q}}$ also for the path length restriction (3.13) and node secure capacity constraints (3.11), (3.12). We only have to concentrate on the node and edge capacity constraints (3.4), (3.5) and (3.8)-(3.10). Now also the couple-(secure)-value has to be integrated.

Adaptation of the edge capacity constraints: For the expansion of the edge capacity constraints we consider the original commodity set $\mathcal{K}_{\text {old }}$. The commodities $k \in \mathcal{K}_{\text {old }}$ accomplish the symmetric routing constraints (3.14)-(3.16). We show how the edge capacity constraints have to be expanded and modified such that we only have to consider the new commodities $k \in \mathcal{K}_{\text {new }}$ generated by the merging process.

Let $v_{i} \in V, i=1, \ldots,|V|$ and $\{u, v\} \in E$.

$$
\begin{aligned}
\sum_{k \in \mathcal{K}_{o l d}} U_{k} f_{u v}^{k} & =\sum_{i=1}^{|V|} \sum_{j>i}^{|V|}\left(\sum_{k \in \mathcal{K}_{v_{i} v_{j}}} U_{k} f_{u v}^{k}+\sum_{k \in \mathcal{K}_{v_{j} v_{i}}} U_{k} f_{u v}^{k}\right) \\
& =\sum_{i=1}^{|V|} \sum_{j>i}^{|V|}\left(U_{k_{v_{i} v_{j}}} f_{u v}^{k}+\bar{U}_{k_{v_{i} v_{j}}} f_{v u}^{k}\right) \\
& =\sum_{k \in \mathcal{K}_{\text {new }}} U_{k} f_{u v}^{k}+\sum_{k \in \mathcal{K}_{\text {new }}} \bar{U}_{k} f_{v u}^{k}
\end{aligned}
$$

Applying this analogously for the other direction, edges and also the secure edge capacity constraints, the modified capacity constraints for edges can be stated as fol-
lows:

$$
\begin{array}{ll}
\sum_{k \in \mathcal{K}} U_{k} f_{u v}^{k}+\sum_{k \in \mathcal{K}} \bar{U}_{k} f_{v u}^{k} \leq \sum_{l \in \mathcal{L}_{\{u, v\}}} C_{u v}^{l} x_{\{u, v\}}^{l} & \forall\{u, v\} \in E \\
\sum_{k \in \mathcal{K}} U_{k} f_{v u}^{k}+\sum_{k \in \mathcal{K}} \bar{U}_{k} f_{u v}^{k} \leq \sum_{l \in \mathcal{L}_{\{u, v\}}} C_{v u}^{l} x_{\{u, v\}}^{l} & \forall\{u, v\} \in E \\
\sum_{k \in \mathcal{K}} U_{k}^{S} f_{u v}^{k}+\sum_{k \in \mathcal{K}} \bar{U}_{k}^{S} f_{v u}^{k} \leq \sum_{l \in \mathcal{L}_{\{u, v\}}} S_{\{u, v\}}^{l} C_{u v}^{l} x_{\{u, v\}}^{l} & \forall\{u, v\} \in E \\
\sum_{k \in \mathcal{K}} U_{k}^{S} f_{v u}^{k}+\sum_{k \in \mathcal{K}} \bar{U}_{k}^{S} f_{u v}^{k} \leq \sum_{l \in \mathcal{L}_{\{u, v\}}} S_{\{u, v\}}^{l} C_{v u}^{l} x_{\{u, v\}}^{l} & \forall\{u, v\} \in E \tag{3.23}
\end{array}
$$

Adaptation of the node capacity constraints: Concerning the capacity of a node $v \in V$ we now also have to consider the coupled value of the commodities that pass through $v$. We consider once more the original commodities $k \in \mathcal{K}_{\text {old }}$ fulfilling the symmetric routing constraints (3.14)-(3.16). We demonstrate that by use of the coupled value these commodities can be replaced by the new commodities $k \in \mathcal{K}_{\text {new }}$ generated by the merging process.

Let $v, v_{i} \in V, i=1, \ldots,|V|$ with $v=v_{i^{*}}$, it follows that

$$
\begin{aligned}
& \sum_{k \in \mathcal{K}_{\text {old }}} \sum_{\{u, v\} \in \delta(v)} U_{k} f_{u v}^{k}+\sum_{\substack{k \in \mathcal{K}_{\text {old }} \\
v=o_{k}}} U_{k} \\
& =\sum_{i=1}^{|V|} \sum_{j>i}^{|V|} \sum_{\{u, v\} \in \delta(v)}\left(\sum_{k \in \mathcal{K}_{v_{i} v_{j}}} U_{k} f_{u v}^{k}+\sum_{k \in \mathcal{K}_{v_{j} v_{i}}} U_{k} f_{u v}^{k}\right) \\
& +\sum_{i>i^{*}}^{|V|} \sum_{\substack{k \in \mathcal{K}_{v} v_{i} \\
o_{k}=v_{i}}} U_{k}+\sum_{\substack{i<i^{*}}}^{|V|} \sum_{\substack{ \\
\\
c_{\in \mathcal{K}} \mathcal{K}_{v_{v}} v \\
k_{k}=v}} U_{k} \\
& =\sum_{i=1}^{|V|} \sum_{j>i}^{|V|} \sum_{\{u, v\} \in \delta(v)} U_{k_{v_{i} v_{j}}} f_{u v}^{k}+\sum_{i=1}^{|V|} \sum_{j>i}^{|V|} \sum_{\{u, v\} \in \delta(v)} \bar{U}_{k_{v_{i} v_{j}}} f_{v u}^{k} \\
& +\sum_{i>i^{*}}^{|V|} U_{k_{v v_{i}}}+\sum_{i<i^{*}}^{|V|} \bar{U}_{k_{v_{i} v}} \\
& =\sum_{k \in \mathcal{K}_{\text {new }}} \sum_{\{u, v\} \in \delta(v)} U_{k} f_{u v}^{k}+\sum_{k \in \mathcal{K}_{\text {new }}} \sum_{\{u, v\} \in \delta(v)} \bar{U}_{k} f_{v u}^{k} \\
& +\sum_{\substack{k \in \mathcal{K}_{\text {new }} \\
v=o_{k}}} U_{k}+\sum_{\substack{k \in \mathcal{K}_{\text {new }} \\
v=d_{k}}} \bar{U}_{k} .
\end{aligned}
$$

Therefore the modified node capacity constraints can be written:

$$
\begin{equation*}
\sum_{k \in \mathcal{K}} \sum_{\{u, v\} \in \delta(v)} U_{k} f_{u v}^{k}+\sum_{k \in \mathcal{K}: v=o_{k}} U_{k}+\sum_{k \in \mathcal{K}} \sum_{\{u, v\} \in \delta(v)} \bar{U}_{k} f_{v u}^{k}+\sum_{k \in \mathcal{K}: v=d_{k}} \bar{U}_{k} \leq C_{v} \quad \forall v \in V \tag{3.24}
\end{equation*}
$$

Taking into account the flow balance constraints (3.2) we can prove the following claim:
Claim 3.17 Let $v \in V$, it follows that

$$
\begin{equation*}
\sum_{k \in \mathcal{K}} \sum_{\{u, v\} \in \delta(v)} \bar{U}_{k} f_{v u}^{k}+\sum_{k \in \mathcal{K}: v=d_{k}} \bar{U}_{k}=\sum_{k \in \mathcal{K}} \sum_{\{u, v\} \in \delta(v)} \bar{U}_{k} f_{u v}^{k}+\sum_{k \in \mathcal{K}: v=o_{k}} \bar{U}_{k} \tag{3.25}
\end{equation*}
$$

Proof.

$$
\begin{aligned}
& \sum_{k \in \mathcal{K}} \sum_{\{u, v\} \in \delta(v)} \bar{U}_{k} f_{v u}^{k}+\sum_{k \in \mathcal{K}: v=d_{k}} \bar{U}_{k} \\
& =\sum_{\substack{k \in \mathcal{K} \\
v \neq o_{k}, v \neq d_{k}}} \bar{U}_{k} \sum_{\{u, v\} \in \delta(v)} f_{v u}^{k}+\sum_{\substack{k \in \mathcal{K} \\
v=o_{k}}} \bar{U}_{k} \sum_{\{u, v\} \in \delta(v)} f_{v u}^{k} \\
& +\sum_{\substack{k \in \mathcal{K} \\
v=d_{k}}} \bar{U}_{k} \sum_{\{u, v\} \in \delta(v)} f_{v u}^{k}+\sum_{\substack{k \in \mathcal{K} \\
v=d_{k}}} \bar{U}_{k} \sum_{\{u, v\} \in \delta(v)}\left(f_{u v}^{k}-f_{v u}^{k}\right) \\
& =\sum_{\substack{k \in \mathcal{K} \\
v \neq o_{k}, v \neq d_{k}}} \bar{U}_{k}\left(\sum_{\{u, v\} \in \delta(v)} f_{v u}^{k}+\sum_{\{u, v\} \in \delta(v)}\left(f_{u v}^{k}-f_{v u}^{k}\right)\right) \\
& +\sum_{\substack{k \in \mathcal{K} \\
v=d_{k}}} \bar{U}_{k}\left(\sum_{\{u, v\} \in \delta(v)} f_{v u}^{k}+\sum_{\{u, v\} \in \delta(v)}\left(f_{u v}^{k}-f_{v u}^{k}\right)\right) \\
& +\sum_{\substack{k \in \mathcal{K} \\
v=o_{k}}} \bar{U}_{k} \sum_{\{u, v\} \in \delta(v)}\left(f_{v u}^{k}-f_{u v}^{k}+f_{u v}^{k}\right) \\
& =\sum_{\substack{k \in \mathcal{C} \\
v \neq o_{k}, \neq \neq d_{k}}} \bar{U}_{k} \sum_{\substack{\{u, v\} \in \delta(v)}} f_{u v}^{k}+\sum_{\substack{k \in \mathcal{K} \\
v=d_{k}}} \bar{U}_{k} \sum_{\{u, v\} \in \delta(v)} f_{u v}^{k} \\
& +\sum_{\substack{k \in \mathcal{K} \\
v=o_{k}}} \bar{U}_{k} \sum_{\{u, v\} \in \delta(v)} f_{u v}^{k}+\sum_{\substack{k \in \mathcal{K} \\
v=o_{k}}} \bar{U}_{k} \sum_{\{u, v\} \in \delta(v)}\left(f_{v u}^{k}-f_{u v}^{k}\right) \\
& =\sum_{k \in \mathcal{K}} \sum_{\{u, v\} \in \delta(v)} \bar{U}_{k} f_{u v}^{k}+\sum_{k \in \mathcal{K}: v=o_{k}} \bar{U}_{k}
\end{aligned}
$$

After this short observation, we can finally formulate the new modified node capacity constraints. Combining (3.24) and (3.25) it follows:

$$
\begin{equation*}
\sum_{k \in \mathcal{K}} \sum_{\{u, v\} \in \delta(v)}\left(U_{k}+\bar{U}_{k}\right) f_{u v}^{k}+\sum_{k \in \mathcal{K}: v=o_{k}}\left(U_{k}+\bar{U}_{k}\right) \leq C_{v} \quad \forall v \in V . \tag{3.26}
\end{equation*}
$$

Remark 3.18 Note that, when the symdem-constraint is not considered, the couple-(secure)-values are all 0 , so that we have the same constraints as in (3.4),(3.5) and (3.8)-(3.10).

Remark 3.19 At first sight the symdem-constraints (3.14)-(3.16) seem to promise the possibility to halve the amount of directed edge flow variables by simply considering the undirected routing case. This is also proposed in the existing literature concerning the given problem [2]. This would only be reasonable in the case of symmetrical homogeneous data, or more precisely, if for all nodes $p, q \in V$, all edges $\{u, v\} \in E$ and linkdesigns $l \in \mathcal{L}_{\{u, v\}}$

- $\sum_{k \in \mathcal{K}_{p q}} U_{k}=\sum_{k \in \mathcal{K}_{q p}} U_{k}$
- $M_{k_{1}}=M_{k_{2}} \quad \forall k_{1}, k_{2} \in \mathcal{K}_{p q} \cup \mathcal{K}_{q p}$
- $S_{k_{1}}=S_{k_{2}} \quad \forall k_{1}, k_{2} \in \mathcal{K}_{p q} \cup \mathcal{K}_{q p}$
- $C_{u v}^{l}=C_{v u}^{l}$

For our model it is impossible to apply this simplification to the undirected case. We illustrate this by a short example:

Example 3.20 Let $p_{1}, p_{2}, q_{1}, q_{2} \in V, e=\{u, v\} \in E, \mathcal{L}_{e}=\{l\}$. The commodities $k_{1} \in \mathcal{K}_{p_{1} q_{1}}$ and $k_{2} \in \mathcal{K}_{p_{2} q_{2}}$ are isolated with values $U_{k_{1}}=U_{k_{2}}=\alpha$. The linkdesign $l$ provides capacity $\alpha$ for both directions ( $C_{u v}^{l}=C_{v u}^{l}=\alpha$ ). We assume that $e$ is equipped with $l$.
In our model the directions in which the commodities are routed over $e$ ascertain the feasibility of this routing regarding the respective edge capacity constraint. If $k_{1}$ and $k_{2}$ are routed in the same direction over $e$, it is not feasible, whereas if they are routed in oppositional directions, it would be feasible. We can not cope with this difficulty in an undirected model with edge flow variables, even though it would have meant an easier reduction in terms of the problem size.

### 3.2.4 Objective function

Since the overall cost for two associated arcs has been bundled in the linkdesign parameter $W_{e}^{l}$, the objective of the minimization of the overall costs can be formalized as follows:

$$
\min \sum_{e \in E} \sum_{l \in \mathcal{L}_{e}} W_{e}^{l} x_{e}^{l}
$$

### 3.3 The model

| $\sum_{l \in \mathcal{L}_{e}} x_{e}^{l}$ | $\leq 1$ | $\forall e \in E$ |
| :--- | :--- | :--- |
| $\sum_{\{u, v\} \in \delta(v)}\left(f_{u v}^{k}-f_{v u}^{k}\right)$ | $=\left\{\begin{array}{cc}-1 & \text { if } v=o_{k} \\ 1 & \text { if } v=d_{k} \\ 0 & \text { otherwise }\end{array}\right.$ | $\forall v \in V, k \in \mathcal{K}$ |
| $\sum_{\{u, v\} \in E(S)}\left(f_{u v}^{k}+f_{v u}^{k}\right)$ | $\leq\|S\|-1$ | $\forall S \subseteq V, k \in \mathcal{K}$ |
| $\sum_{k \in \mathcal{K}} U_{k} f_{u v}^{k}+\sum_{k \in \mathcal{K}} \bar{U}_{k} f_{v u}^{k}$ | $\leq \sum_{l \in \mathcal{L}_{\{u, v\}}} C_{v u}^{l} x_{\{u, v\}}^{l}$ | $\forall\{u, v\} \in E$ |
| $\sum_{k \in \mathcal{K}} U_{k} f_{v u}^{k}+\sum_{k \in \mathcal{K}} \bar{U}_{k} f_{u v}^{k}$ | $\leq \sum_{l \in \mathcal{L}_{\{u, v\}}} C_{u v}^{l} x_{\{u, v\}}^{l}$ | $\forall\{u, v\} \in E$ |
| $\sum_{k \in \mathcal{K}} U_{k}^{S} f_{u v}^{k}+\sum_{k \in \mathcal{K}} \bar{U}_{k}^{S} f_{v u}^{k}$ | $\leq \sum_{l \in \mathcal{L}_{\{u, v\}}} S_{\{u, v\}}^{l} C_{v u}^{l} x_{\{u, v\}}^{l}$ | $\forall\{u, v\} \in E$ |
| $\sum_{k \in \mathcal{K}} U_{k}^{S} f_{v u}^{k}+\sum_{k \in \mathcal{K}} \bar{U}_{k}^{S} f_{u v}^{k}$ | $\leq \sum_{l \in \mathcal{L}_{\{u, v\}}} S_{\{u, v\}}^{l} C_{u v}^{l} x_{\{u, v\}}^{l}$ | $\forall\{u, v\} \in E$ |
| $\sum_{k \in \mathcal{K}} \sum_{\{u, v\} \in \delta(v)}\left(U_{k}+\bar{U}_{k}\right) f_{u v}^{k}$ | $\leq C_{v}-\sum_{\substack{k \in \mathcal{K} \\ v=o_{k}}}\left(U_{k}+\bar{U}_{k}\right)$ | $\forall v \in V$ |
| $\sum_{\{u, v\} \in \delta(v)} \sum_{l \in \mathcal{L}_{\{u, v\}}} P_{u v}^{l} x_{\{u, v\}}^{l} \leq P_{v}^{-}$ | $\forall v \in V$ |  |
| $\sum_{\{u, v\} \in \delta(v)} \sum_{l \in \mathcal{L}_{\{u, v\}}} P_{v u}^{l} x_{\{u, v\}}^{l} \leq P_{v}^{+}$ | $\forall v \in V$ |  |
| $\sum_{e=\{u, v\} \in E}\left(f_{u v}^{k}+f_{v u}^{k}\right)$ | $\leq M_{k}$ | $\forall k \in \mathcal{K}$ |

$$
\begin{array}{ll}
f_{u v}^{k} \in\{0,1\} & \forall\{u, v\} \in E, k \in \mathcal{K} \\
f_{v u}^{k} \in\{0,1\} & \forall\{u, v\} \in E, k \in \mathcal{K} \\
x_{e}^{l} \in\{0,1\} & \forall e \in E, l \in \mathcal{L}_{e}
\end{array}
$$

We define the polyhedron associated with this integer linear program and its LPrelaxation as

$$
\begin{aligned}
& \mathcal{P}:=\operatorname{conv}\left\{(x, f) \in\{0,1\}^{\sum_{e \in E}\left|\mathcal{L}_{e}\right|} \times\{0,1\}^{2 \times|E| \times|\mathcal{K}|} \mid(x, f) \text { satisfies } \begin{array}{c}
(3.1)-(3.3),(3.6),(3.7), \\
(3.13),(3.20)-(3.26)
\end{array}\right\} \\
& \mathcal{L P}:=\operatorname{conv}\left\{(x, f) \in[0,1]^{\sum_{e \in E}\left|\mathcal{L}_{e}\right|} \times[0,1]^{2 \times|E| \times|\mathcal{K}|} \mid(x, f) \text { satisfies } \begin{array}{c}
(3.1)-(3.3),(3.6),(3.7), \\
(3.13),(3.20)-(3.26)
\end{array}\right\}
\end{aligned}
$$

## Chapter 4

## Classes of valid inequalities

In this chapter we present different types of valid inequalities. We give some examples to show that in special cases they really cut off some fractional solution points and which conditions have to be fulfilled such that some are even facet defining.

### 4.1 GUB cover inequalities

Anyone of our model inequalities can be represented as a knapsack inequality, which is an inequality of the form

$$
\begin{equation*}
a^{T} x=\sum_{j \in J} a_{j} x_{j} \leq b \text { with } x \in\{0,1\}^{|J|} \tag{4.1}
\end{equation*}
$$

In [9] it was shown that these inequalities can be used to derive valid inequalities. These so called cover inequalities are subsets of the variables in the inequality such that if all variables were set to 1 , the inequality would be violated, but if any one variable were excluded, the inequality would be satisfied. Inequalities of the form of the sum of the variables in the cover must be less than or equal to the size of the cover less one describe the relationship of the variables in the cover. These inequalities have been extended to knapsacks with generalized upper bounds [18]. These so callled GUB constraint for a set of binary variables is the sum of variables less than or equal to one. If the variables in a GUB constraint are also members of a knapsack constraint, then the minimal cover can be selected with the additional consideration that at most one of the members of the GUB constraint can be one in a solution. This additional restriction makes the GUB cover inequalities stronger (that is, more restrictive) than ordinary cover inequalities.

In our model arise three types of GUBs, one are the linkdesign installation constraints (3.1) which consider the linkdesign installation variables:

$$
\begin{equation*}
\sum_{l \in \mathcal{L}_{e}} x_{e}^{l} \leq 1 \quad \forall e \in E \tag{4.2}
\end{equation*}
$$

The other two classes of GUBs are deduced by the subtour elimination constraints (3.3) and the third also by the flow balance equalities (3.2). Both consider sets of commodity edge flow variables and we apply them alternately:

$$
\begin{align*}
& f_{u v}^{k}+f_{v u}^{k} \leq 1 \quad \forall k \in \mathcal{K},\{u, v\} \in E  \tag{4.3}\\
& \sum_{\{u, v\} \in \delta(v)} f_{u v}^{k} \leq 1 \forall k \in \mathcal{K}, v \in V \tag{4.4}
\end{align*}
$$

First we generally describe how the so called GUB cover inequalities are deduced.

### 4.1.1 Valid inequalities from knapsack and GUB

Consider the 0-1 knapsack set $H=\left\{x \in\{0,1\}^{|J|} \mid \sum_{j \in J} a_{j} x_{j} \leq b\right\}$. Assuming that all $a_{j} \neq 0$ (erasing previously all $x_{j}$ with $a_{j}=0$ ), we can split $J$ into two disjoint subsets $J_{P}$ and $J_{N}$, where $P$ stands for positive and $N$ for negative. $J_{P}$ consists of the indices with the corresponding coefficient being positive, and for $J_{N}$ accordingly negative. We can now rewrite $H=\left\{x \in\{0,1\}^{|J|} \mid \sum_{j \in J_{P}} \bar{a}_{j} x_{j}-\sum_{j \in J_{N}} \bar{a}_{j} x_{j} \leq b\right\}$ with all coefficients $\bar{a}_{j}>0\left(\bar{a}_{j}=\left|a_{j}\right| \forall j \in J=J_{P} \cup J_{N}\right)$. Using the presence of some GUBs, these subsets $J_{P}$ and $J_{N}$ can be further split into subsets $\mathbb{G}_{i}$ tying together all variables affected by one GUB. Note that we only apply GUBs that bind together exclusively the variables of the respective uppersets and that these GUBs are pairwise disjoint with each other. When some variable $x_{j}$ does not appear in a GUB we apply the trivial GUB $x_{j} \leq 1$ since we only consider binary variables. We denote by $I_{P}$ and $I_{N}$ the index sets for the used GUBs, such that:

$$
J_{P}=\bigcup_{i \in I_{P}} \mathbb{G}_{i}, \quad J_{N}=\bigcup_{i \in I_{N}} \mathbb{G}_{i} \quad \text { and } \quad \mathbb{G}_{i} \cap \mathbb{G}_{j}=\emptyset \quad \forall i, j \in I=I_{P} \cup I_{N}
$$

Now we consider the knapsack polytope with explicit GUB constraints
$X_{G U B}:=\left\{x \in\{0,1\}^{|J|} \mid \sum_{j \in J_{P}} \bar{a}_{j} x_{j}-\sum_{j \in J_{N}} \bar{a}_{j} x_{j} \leq b, \quad \sum_{j \in \mathbb{G}_{i}} x_{j} \leq 1 \quad \forall i \in I=I_{P} \cup I_{N}\right\}$

We say that $C:=C_{P} \cup C_{N}$ is a GUB cover if

1. $C_{P} \subseteq J_{P}$ and $C_{N} \subseteq J_{N}$
2. $\left|C_{P} \cap \mathbb{G}_{i}\right| \leq 1 \quad \forall i \in I_{P}$ and $\left|C_{N} \cap \mathbb{G}_{i}\right| \leq 1 \quad \forall i \in I_{N}$
3. $\sum_{j \in C_{P}} \bar{a}_{j}-\sum_{j \in C_{N}} \bar{a}_{j}>b$

With the GUB cover $C$, we associate the sets:

$$
\begin{align*}
I_{P}^{+} & :=\left\{i \in I_{P} \mid C_{P} \cap \mathbb{G}_{i} \neq \emptyset\right\}  \tag{4.5}\\
I_{N}^{+} & :=\left\{i \in I_{N} \mid C_{N} \cap \mathbb{G}_{i} \neq \emptyset\right\}  \tag{4.6}\\
\mathbb{G}_{i}^{+} & :=\left\{j \in \mathbb{G}_{i}: \bar{a}_{j} \geq \bar{a}_{l} \quad \text { for } l \in C_{P} \cap \mathbb{G}_{i}\right\} \quad \forall i \in I_{P}^{+}  \tag{4.7}\\
\mathbb{G}_{i}^{+} & :=\left\{j \in \mathbb{G}_{i}: \bar{a}_{j} \leq \bar{a}_{l} \quad \text { for } l \in C_{N} \cap \mathbb{G}_{i}\right\} \quad \forall i \in I_{N}^{+} \tag{4.8}
\end{align*}
$$

With these definitions, the following was derived in [18]:
Proposition 4.1 The inequality

$$
\begin{equation*}
\sum_{i \in I_{P}^{+}} \sum_{j \in \mathbb{G}_{i}^{+}} x_{j} \leq\left|C_{P}\right|-1+\sum_{i \in I_{N}^{+}} \sum_{j \notin \mathbb{G}_{i}^{+}} x_{j}+\sum_{i \in I_{N} \backslash I_{N}^{+}} \sum_{j \in \mathbb{G}_{i}} x_{j} \tag{4.9}
\end{equation*}
$$

is valid for $X_{G U B}$.

For the special case that there exist only positive coefficients in the knapsack inequality, stronger inequalities were introduced. For the GUB cover $C$, let

$$
\begin{equation*}
E(C):=\left\{k \in J_{P} \mid \bar{a}_{k} \geq \bar{a}_{j} \quad \forall j \in C\right\} \backslash \bigcup_{i \in I_{P}^{+}} \mathbb{G}_{i}^{+} \tag{4.10}
\end{equation*}
$$

Proposition 4.2 The inequality

$$
\begin{equation*}
\sum_{i \in I_{P}^{+}} \sum_{j \in \mathbb{G}_{i}^{+}} x_{j}+\sum_{j \in E(C)} x_{j} \leq\left|C_{P}\right|-1 \tag{4.11}
\end{equation*}
$$

is valid for $X_{G U B}$ if $J_{N}=\emptyset$.
We use this class of inequalities in the following variants:

### 4.1.2 Edge (secure) capacity constraints

Edge capacity constraints: For an edge $\{u, v\} \in E$ we first concern the direction $u$ to $v$. In this case the underlying knapsacks $a^{T} x \leq b$ are the inequalities (3.20),

$$
\begin{equation*}
\sum_{k \in \mathcal{K}} U_{k} f_{u v}^{k}+\sum_{k \in \mathcal{K}} \bar{U}_{k} f_{v u}^{k}-\sum_{l \in \mathcal{L}_{\{u, v\}}} C_{u v}^{l} x_{\{u, v\}}^{l} \leq 0 \quad \forall\{u, v\} \in E \tag{4.12}
\end{equation*}
$$

We apply the linkdesign GUB (4.2) and commodity edge flow variable GUB (4.3), such that for the above constraints we have one negative and $|\mathcal{K}|$ positive GUBs. The $\bar{a}_{j}$ correspond to the capacity of the installable linkdesigns and the value (coupled value) of a commodity routed over $\{u, v\}$ in direction $u$ to $v(v$ to $u)$, respectively.

A GUB cover corresponds to the election of two commodity subsets $K_{u v} \subseteq \mathcal{K}$ and $K_{v u} \subseteq \mathcal{K}$ (with $K_{u v} \cap K_{v u}=\emptyset$ ) and the choice of a particular linkdesign $\hat{l}$ for the edge $\{u, v\}$. The commodities from $K_{u v}\left(K_{v u}\right)$ are routed over $\{u, v\}$ in direction $u$ to $v(v$ to $u)$. Let

$$
\begin{align*}
\hat{K}_{u v} & =K_{u v} \cup\left\{k \in K_{v u} \mid U_{k} \geq \bar{U}_{k}\right\}  \tag{4.13}\\
\hat{K}_{v u} & =K_{v u} \cup\left\{k \in K_{u v} \mid \bar{U}_{k} \geq U_{k}\right\} \tag{4.14}
\end{align*}
$$

The resulting inequality,

$$
\begin{equation*}
\sum_{k \in \hat{K}_{u v}} f_{u v}^{k}+\sum_{k \in \hat{K}_{v u}} f_{v u}^{k}-\sum_{\substack{l \in \mathcal{L}_{\{u, v\}}: \\ C_{u v}^{l}>C_{u v}^{l}\{u, v\}}} x_{\{u, v\}}^{l} \leq\left|K_{u v}\right|+\left|K_{v u}\right|-1 \tag{4.15}
\end{equation*}
$$

states that, choosing the cover commodities to be routed over the edge $\{u, v\}$, a linkdesign with higher directed capacity than the cover capacity has to be installed.

An analogous inequality follows for the direction $v$ to $u$.
Example 4.3 Consider an instance with three nodes $V=\left\{v_{1}, v_{2}, v_{3}\right\}$, three edges $E=\left\{\left\{v_{1}, v_{2}\right\},\left\{v_{1}, v_{3}\right\},\left\{v_{2}, v_{3}\right\}\right\}$ and two commodities $\mathcal{K}=\left\{k_{1}, k_{2}\right\}$. The commodities have the following demand values: $U_{k_{1}}=U_{k_{2}}=3$ and $\bar{U}_{k_{1}}=\bar{U}_{k_{2}}=$ 0 respectively. For the edge $\left\{v_{2}, v_{3}\right\}$ the linkdesign set consists of two linkdesigns $\mathcal{L}_{\left\{v_{2}, v_{3}\right\}}=\left\{l_{1}, l_{2}\right\}$ providing capacity for the direction $v_{2}$ to $v_{3}$ in amount of $C_{l_{1}}^{v_{2} v_{3}}=5$ and $C_{l_{1}}^{v_{2} v_{3}}=10$ respectively.
After solving the LP-relaxation, the optimal solution results in $f_{v_{2} v_{3}}^{k_{1}}=f_{v_{2} v_{3}}^{k_{2}}=0.8$ and $x_{\left\{v_{2}, v_{3}\right\}}^{l_{1}}=1$, fulfilling the corresponding edge capacity inequality. Regarding the cover for the edge $\left\{v_{2}, v_{3}\right\}$, direction $v_{2}$ to $v_{3}$, represented by the commodities $k_{1}$ and $k_{2}$ and the linkdesign $l_{1}$, we obtain the inequality

$$
\begin{equation*}
f_{v_{2} v_{3}}^{k_{1}}+f_{v_{2} v_{3}}^{k_{2}}-x_{\left\{v_{2}, v_{3}\right\}}^{l_{2}} \leq 1 \tag{4.16}
\end{equation*}
$$

This GUB cover inequality cuts off the current non-integral solution.
Edge secure capacity constraints: In analogy to the edge capacity constraints above, we derive GUB cover inequalities from the secure constraints (3.22) and (3.23).

### 4.1.3 Port constraints

Inport constraints: In this case the underlying knapsacks $a^{T} x \leq b$ are the inequalities (3.6),

$$
\begin{equation*}
\sum_{\{u, v\} \in \delta(v)} \sum_{l \in \mathcal{L}_{\{u, v\}}} P_{u v}^{l} x_{\{u, v\}}^{l} \leq P_{v}^{-} \quad \forall v \in V \tag{4.17}
\end{equation*}
$$

Applying the linkdesign installation GUB (4.2) we obtain that we have $|\delta(v)|$ positive GUBs and no negative GUB for these constraints. The $\bar{a}_{j}$ correspond to the inport consumption of the installable linkdesigns.
A GUB cover corresponds to the choice of a particular linkdesigns $\hat{l}_{\{u, v\}}$ on each edge $\{u, v\}$ of a subset $E^{\prime} \subseteq \delta(v)$. We define the maximal inport consumption of all edges and linkdesigns in the cover:

$$
\begin{equation*}
P_{\max }(C):=\max _{\{u, v\} \in E^{\prime}} P_{u v}^{\hat{l}_{\{u, v\}}} \tag{4.18}
\end{equation*}
$$

The resulting inequality,

$$
\begin{equation*}
\sum_{\{u, v\} \in E^{\prime}} \sum_{\substack{l \in \mathcal{L}_{\{u, v\}}: \\
P_{u v}^{l} \geq P_{u v} l_{\{u, v\}}}} x_{\{u, v\}}^{l}+\sum_{\substack{ }} \sum_{\substack {l \in v\} \in \delta(v) \backslash E^{\prime} \\
\begin{subarray}{c}{l \in \mathcal{L}_{\{u, v\}}: \\
P_{u v}^{l} \geq P_{\max }(C){ l \in v \} \in \delta ( v ) \backslash E ^ { \prime } \\
\begin{subarray} { c } { l \in \mathcal { L } _ { \{ u , v \} } : \\
P _ { u v } ^ { l } \geq P _ { \operatorname { m a x } } ( C ) } }\end{subarray}} x_{\{u, v\}}^{l} \leq\left|E^{\prime}\right|-1 \tag{4.19}
\end{equation*}
$$

states that at least at one edge in $E^{\prime}$, a linkdesign with a lower inport consumption than in the cover has to be installed.

Example 4.4 Consider the instance from Example 4.3. For $v_{1}$ the number of inports is $P_{v_{1}}^{-}=5$. The set of linkdesigns for the edges $\left\{v_{1}, v_{2}\right\}$ and $\left\{v_{1}, v_{3}\right\}$ are the following:

$$
\begin{aligned}
\mathcal{L}_{\left\{v_{1}, v_{2}\right\}} & =\left\{l_{1}, l_{2}\right\} \text { with } P_{v_{2} v_{1}}^{l_{1}}=3 \text { and } P_{v_{2} v_{1}}^{l_{2}}=4 \\
\mathcal{L}_{\left\{v_{1}, v_{3}\right\}} & =\left\{l^{\prime}\right\} \text { with } P_{v_{3} v_{1}}^{l^{\prime}}=3 .
\end{aligned}
$$

Solving the LP-relaxation results in an optimal solution with $x_{\left\{v_{1}, v_{2}\right\}}^{l_{2}}=x_{\left\{v_{1}, v_{3}\right\}}^{l^{\prime}}=$ 0.8 fulfilling the corresponding inport inequality. The cover concerning $v_{1}$ consists of the edges $\left\{v_{1}, v_{2}\right\}$ and $\left\{v_{1}, v_{3}\right\}$ and the linkdesigns $l_{1}, l^{\prime}$ respectively. The implied inport GUB cover inequality

$$
\begin{equation*}
x_{\left\{v_{1}, v_{2}\right\}}^{l_{1}}+x_{\left\{v_{1}, v_{2}\right\}}^{l_{2}}+x_{\left\{v_{1}, v_{3}\right\}}^{l^{\prime}} \leq 1 \tag{4.20}
\end{equation*}
$$

cuts off the current fractional solution.

Outport constraints: Analogous inequalities follow from the outport inequalities (3.7).

### 4.1.4 Node capacity constraints

Node capacity constraints: In this case the underlying knapsacks $a^{T} x \leq b$ are the inequalities (3.8),

$$
\begin{equation*}
\sum_{k \in \mathcal{K}} \sum_{\substack{e \in \delta(v) \\ e=\{u, v\}}}\left(U_{k}+\bar{U}_{k}\right) f_{u v}^{k} \leq C_{v}-\sum_{k \in \mathcal{K}: v=o_{k}}\left(U_{k}+\bar{U}_{k}\right) \quad \forall v \in V \tag{4.21}
\end{equation*}
$$

Utilizing the commodity edge flow variable GUB (4.4) it results that for the above constraints we have no negative and $|\mathcal{K}|$ positive GUBs. The $\bar{a}_{j}$ correspond to the overall demand values $U_{k}+\bar{U}_{k}$.
A GUB cover corresponds to the election of a commodity subset $K \subseteq \mathcal{K}$ and the choice of a particular edge $\hat{e}_{k}$ of $\delta(v)$ for each commodity $k \in K$, over which it is routed into the node $v$. We define the maximal overall demand value for the commodities in the cover:

$$
\begin{equation*}
U_{\max }(C):=\max _{k \in K}\left(U_{k}+\bar{U}_{k}\right) \tag{4.22}
\end{equation*}
$$

The resulting inequality,

$$
\begin{equation*}
\sum_{k \in K} \sum_{\substack{\{u, v\} \in \delta(v)}} f_{u v}^{k}+\sum_{\substack{k \in \mathcal{K} \backslash K: \\ U_{k}+\overline{U_{k} \geq U_{\max }(C)}}} \sum_{\{u, v\} \in \delta(v)} f_{u v}^{k} \leq|K|-1 \tag{4.23}
\end{equation*}
$$

states that number of commodities of the cover (or commodities with even higher overall demand values) which are routed over the node $v$ must be less than the number of commodities in the cover.

Example 4.5 Consider the instance from Example 4.3 with the there presented commodities $k_{1}$ and $k_{2}$. The node $v_{1}$ has a capacity of $C_{v_{1}}=5$.
The optimal solution of the LP-relaxation contains the values $f_{v_{1} v_{3}}^{k_{1}}=f_{v_{1} v_{3}}^{k_{2}}=0.8$, fulfilling the node capacity constraint for $v_{1}$. The GUB cover consists of the commodity $k_{1}$ with the edge $\left\{v_{1}, v_{2}\right\}$ and $k_{2}$ with $\left\{v_{1}, v_{3}\right\}$ respectively. We obtain the GUB cover inequality

$$
\begin{equation*}
\sum_{\left\{u, v_{1}\right\} \in \delta\left(v_{1}\right)} f_{u v_{1}}^{k_{1}}+\sum_{\left\{u, v_{1}\right\} \in \delta\left(v_{1}\right)} f_{u v_{1}}^{k_{2}} \leq 1 \tag{4.24}
\end{equation*}
$$

cutting off the current fractional solution.

These GUB Cover inequalities are all valid for our feasibility space since the knapsack and GUB inequalities are all part of our model.

### 4.2 Cutset inequalities

In this section we present the types of inequalities which arise from the consideration of cuts.

Sfrag replacements
W
$V \backslash W$
$v_{1}$
$v_{2}$
$v_{3}$
$v_{4}$

| $\mathbf{c}$ |  |  |  |
| :---: | :---: | :---: | :---: |
| $k$ | $\bar{U}_{k}$ | $U_{k}^{S}$ | $\bar{U}_{k}^{S}$ |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
|  | 00 | 20 | 00 |
|  | 0 | 30 | 0 |

Figure 4.1: Cut traffic

Consider the node subset $W \subseteq V$. Commodities crossing the cut $\delta(W)$ impose certain requirements on the capacity provided by some linkdesigns installed on the edges of $\delta(W)$.
Using Figure 4.1, we present these conditions. The amount of traffic that has to be routed from $W$ to $V \backslash W$ is 120 . Thus, a overall capacity of at least 120 must be provided by the linkdesigns installed on the edges $\left\{v_{1}, v_{3}\right\}$, $\left\{v_{2}, v_{3}\right\}$, and $\left\{v_{2}, v_{4}\right\}$ with respect to the direction out of $W$.
Taking into account the single path condition, we state that the above commodities $k_{1}, k_{2}, k_{3}$, and $k_{4}$ must cross the cut using at least one edge $e \in \delta(W)$. Considering the commodities $k_{1}$ and $k_{2}$ with their relatively high data values, it follows that at least one edge $\left\{v_{1}, v_{3}\right\},\left\{v_{2}, v_{3}\right\}$, or $\left\{v_{2}, v_{4}\right\}$ must be equipped with a linkdesign providing at least a capacity of 100 w.r.t. the direction out of $W$, to route commodity $k_{1}$. Another edge $e \in \delta(W)$ must be equipped with a linkdesign providing at least a capacity of 200 w.r.t. the direction into the node subset $W$, such that commodity $k_{2}$ can be routed over the cut.

In the following we specify these requirements. At first we define useful denominations:

Definition 4.6 (Sourced commodities) Let $W \subseteq V$. A commodity is called $W$ sourced commodity for the node subset $W$, if its origin is in $W$ and its destination in $V \backslash W$. Define

$$
\mathcal{K}_{W^{+}}:=\left\{k \in \mathcal{K} \mid o_{k} \in W, d_{k} \in V \backslash W\right\}
$$

the set of $W$ sourced commodities.
Definition 4.7 (Targeted commodities) Let $W \subseteq V$. A commodity is called $W$ targeted commodity with respect to the node subset $W$, if its destination lays in $W$
and its origin in $V \backslash W$. Define

$$
\mathcal{K}_{W^{-}}:=\left\{k \in \mathcal{K} \mid d_{k} \in W, o_{k} \in V \backslash W\right\}
$$

the set of $W$ targeted commodities.

### 4.2.1 Cut traffic inequalities

This class of inequalities states that the directed (secure) capacity across a cut must be at least as large as the (secure) demand data across the cut. First we define this cut-crossing traffic in the following:

Definition 4.8 ((Secure) outbound traffic) Let $W \subseteq V$. We define

$$
\begin{aligned}
T^{+}(W) & :=\sum_{k \in \mathcal{K}_{W^{+}}} U_{k}+\sum_{k \in \mathcal{K}_{W^{-}}} \bar{U}_{k} \\
& \text { respectively } \\
T^{S+}(W) & :=\sum_{k \in \mathcal{K}_{W^{+}}} U_{k}^{S}+\sum_{k \in \mathcal{K}_{W^{-}}} \bar{U}_{k}^{S}
\end{aligned}
$$

to be the amount of (secure) outbound traffic of $W$.
Let $W \subseteq V$. Considering the normal data we obtain the following inequality

$$
\begin{equation*}
\sum_{\substack{\{u, v\} \in E: \\ u \in W, v \in V \backslash W}} \sum_{l \in \mathcal{L}_{\{u, v\}}} C_{u v}^{l} x_{\{u, v\}}^{l} \geq T^{+}(W) \tag{4.25}
\end{equation*}
$$

Likewise, we can generate inequalities concerning the outbound traffic which has to be routed securely. We obtain the inequality

$$
\begin{equation*}
\sum_{\substack{\{, v\} \in \in E \\ u \in W, v \in V \backslash W}} \sum_{l \in \mathcal{L}_{\{u, v\}}} S_{\{u, v\}}^{l} C_{u v}^{l} x_{\{u, v\}}^{l} \geq T^{S+}(W) \tag{4.26}
\end{equation*}
$$

Validity of (4.25) and (4.26) for the considered polyhedron $\mathcal{P}$ is obvious. We apply the above inequalities (4.25) and (4.26) in terms of base knapsack inequalities and employ the same procedures as in Section 4.1. As GUB we use the linkdesign installation constraints (3.1).

### 4.2.2 Particular commodity inequalities

Investigating the cut we can also deal with the single path routing constraint. All commodities crossing the cut have to use at least one edge entirely, since the flow must not be splitted. Considering remarkable commodities (in terms of (secure) amount of demand value) this fact imposes the requirement that at least one edge must be equipped with a linkdesign providing enough capacity for a remarkable commodity. To simplify the argumentation, we first define the following:

Definition 4.9 (Maximum (secure) outbound value) Concerning the node subset $W$ we define the largest (secure) single communication demand from $W$ to $V \backslash W$ Let $W \subseteq V$ :

$$
\begin{aligned}
U_{\max }^{W^{+}} & :=\max \left\{\max _{k \in K_{W^{+}}} U_{k}, \max _{k \in K_{W^{-}}} \bar{U}_{k}\right\} \\
& \text { respectively } \\
U_{\max }^{W_{S}^{+}} & :=\max \left\{\max _{k \in K_{W^{+}}} U_{k}^{S}, \max _{k \in K_{W^{-}}} \bar{U}_{k}^{S}\right\}
\end{aligned}
$$

the maximum (secure) outbound value of $W$.
Definition 4.10 (Maximum (secure) inbound value) Concerning the node subset $W$ we denote the biggest (secure) single communication demand from $V \backslash W$ to $W$. Let $W \subseteq V$. Define

$$
\begin{aligned}
U_{\max }^{W^{-}} & :=\max \left\{\max _{k \in K_{W^{-}}} U_{k}, \max _{k \in K_{W^{+}}} \bar{U}_{k}\right\} \\
& \text { respectively } \\
U_{\max }^{W_{S}^{-}} & :=\max \left\{\max _{k \in K_{W^{-}}} U_{k}^{S}, \max _{k \in K_{W^{+}}} \bar{U}_{k}^{S}\right\}
\end{aligned}
$$

the maximum (secure) inbound value of $W$.
Definition 4.11 ((Secure) outbound sufficient linkdesign) A linkdesign is called (secure) outbound sufficient with respect to a node subset $W$, if it provides as least as much (secure) capacity as the maximum (secure) outbound value of $W$. Let $W \subseteq V,\{u, v\} \in \delta(W)$ with $u \in W, v \in V \backslash W$, and let

$$
\begin{aligned}
\mathcal{L}_{\{u, v\}}^{W^{+}} & :=\left\{l \in \mathcal{L}_{\{u, v\}} \quad \mid \quad C_{u v}^{l} \geq U_{\max }^{W^{+}}\right\} \\
& \quad \text { respectively } \\
\mathcal{L}_{\{u, v\}}^{W_{S}^{+}} & :=\left\{l \in \mathcal{L}_{\{u, v\}} \mid S_{\{u, v\}}^{l} C_{u v}^{l} \geq U_{\max }^{W^{+}}\right\}
\end{aligned}
$$

be the set of (secure) outbound sufficient linkdesigns for the node subset $W$.
Definition 4.12 ((Secure) inbound sufficient linkdesign) A linkdesign is called (secure) inbound sufficient with respect to a node subset $W$, if it provides as least as much (secure) capacity as the maximum (secure) inbound value of $W$. Let $W \subseteq V,\{u, v\} \in \delta(W)$ with $u \in W, v \in V \backslash W$, and let

$$
\begin{aligned}
& \mathcal{L}_{\{u, v\}}^{W^{-}}:=\left\{l \in \mathcal{L}_{\{u, v\}} \mid C_{u v}^{l} \geq U_{\text {max }}^{W^{+}}\right\} \\
& \text {respectively } \\
& \mathcal{L}_{\{u, v\}}^{W_{S}^{-}}:=\left\{l \in \mathcal{L}_{\{u, v\}} \mid S_{\{u, v\}}^{l} C_{u v}^{l} \geq U_{\max }^{W^{+}}\right\}
\end{aligned}
$$

be the set of (secure) inbound sufficient linkdesigns for the node subset $W$.

We derive the inequalities

$$
\begin{align*}
& \sum_{\{u, v\} \in \delta(W)} \sum_{l \in \mathcal{L}_{\{u, v\}}^{W^{+}}} x_{\{u, v\}}^{l} \geq 1  \tag{4.27}\\
& \sum_{\{u, v\} \in \delta(W)} \sum_{l \in \mathcal{L}_{\{u, v\}}^{W_{S}^{+}}} x_{\{u, v\}}^{l} \geq 1  \tag{4.28}\\
& \sum_{\{u, v\} \in \delta(W)} \sum_{l \in \mathcal{L}_{\{u, v\}}^{W^{-}}} x_{\{u, v\}}^{l} \geq 1  \tag{4.29}\\
& \sum_{\{u, v\} \in \delta(W)} \sum_{l \in \mathcal{L}_{\{u, v\}}^{W_{S}^{-}}} x_{\{u, v\}}^{l} \geq 1 \tag{4.30}
\end{align*}
$$

Proposition 4.13 The above inequalities are valid for $\mathcal{P}$.
Proof. Assume that $\hat{p}=(\hat{x}, \hat{f})$ does not satisfy the inequality (4.27), i.e. all edges in $\delta(W)$ provide less capacity than $U_{\max }^{W^{+}}$. On the other hand the commodity corresponding to $U_{\max }^{W^{+}}$has to cross the cut unsplitted, i.e. has to use one single edge. Hence the edge capacity constraint is violated concerning this particular edge and therefore $\hat{p}$ is not feasible for $\mathcal{P}$. Likewise for the other inequalities.

The inequalities (4.27)-(4.30) can be combined. Consider the case that we have four different commodities, each remarkable in its own way. Due to the availability of linkdesigns, these four commodities may be forced to be routed over four different edges of the cut. Let $W \subseteq V$ and four commodities $k_{1}, k_{2}, k_{3}, k_{4} \in \mathcal{K}$, where $k_{1}$ is the corresponding commodity of $U_{\max }^{W^{+}}$, $k_{2}$ of $U_{\max }^{W_{S}^{+}}, k_{3}$ of $U_{\max }^{W^{-}}$, and $k_{4}$ of $U_{\max }^{W_{S}^{-}}$. When at the same time $\mathcal{L}_{e}^{W^{+}} \cap \mathcal{L}_{e}^{W_{S}^{+}} \cap \mathcal{L}_{e}^{W^{-}} \cap \mathcal{L}_{e}^{W_{S}^{-}}=\emptyset \quad \forall e \in \delta(W)$ then we obtain the inequality

$$
\begin{equation*}
\sum_{e \in \delta(W)} \sum_{l \in \mathcal{L}_{e}^{W^{+}} \cup \mathcal{L}_{e}^{W_{S}^{+}} \cup \mathcal{L}_{e}^{W^{-}} \cup \mathcal{L}_{e}^{W_{S}^{-}}} x_{e}^{l} \geq 4 \tag{4.31}
\end{equation*}
$$

Example 4.14 We consider the node subset $W \subseteq V$ from Figure 4.1 and suppose that $\mathcal{L}_{\left\{v_{1} v_{3}\right\}}=\mathcal{L}_{\left\{v_{2} v_{3}\right\}}=\mathcal{L}_{\left\{v_{2} v_{4}\right\}}=\left\{l_{1}, l_{2}, l_{3}, l_{4}\right\}$ with the properties presented in Table 4.1.

In this case all commodities $k_{1}, k_{2}, k_{3}$, and $k_{4}$ have to use different edges to cross the cut. Thus each edge $e \in \delta(W)$ must be equipped with an other linkdesign $l \in$ $\left\{l_{1}, l_{2}, l_{3}, l_{4}\right\}$.

|  | $C_{u v}^{l}$ | $C_{v u}^{l}$ | $S_{\{u, v\}}^{l}$ |
| :---: | :---: | :---: | :---: |
| $l_{1}$ | 200 | 0 | 0 |
| $l_{2}$ | 0 | 200 | 0 |
| $l_{3}$ | 50 | 0 | 1 |
| $l_{4}$ | 0 | 50 | 1 |

Table 4.1: Available linkdesigns for $e \in \delta(W)$

### 4.3 Edge capacity inequalities

We now present a class of inequalities that are valid for our considered polyhedron. First, we introduce some notation and definitions to simplify the discussion and prove validity. Then we consider a relaxation of our problem and identify the members of the class which define facets of the associated polyhedron.

### 4.3.1 Setup

Definition 4.15 (Arc-inclusive paths) Let $a \in A$ an arc of the overlaying digraph $D(G)$ of $G$ (as defined in Section 2.2) and $s, t \in V$. An $s$ - $t$-path is called $a$-inclusive if it contains $a$. Let

$$
\mathfrak{P}_{s t}^{a-i n c l}:=\{\mathfrak{p} \text { path from } s \text { to } t \mid a \in \mathfrak{p}\}
$$

be the set of $a$-inclusive $s$ - $t$-paths.
Definition 4.16 (Arc-exclusive paths) Let $a \in A$ be an arc of the overlaying digraph $D(G)$ of $G$ and $s, t \in V$. An $s$-t-path is called $a$-exclusive if it does not contain $a$. We define

$$
\mathfrak{P}_{s t}^{a-e x c l}:=\{\mathfrak{p} \text { path from } s \text { to } t \mid a \notin \mathfrak{p}\}
$$

as the set of $a$-exclusive $s$ - $t$-paths.
Definition 4.17 (Edge-exclusive cycle) Let $e \in E$. A cycle $C$ is called $e$-exclusive if it is simple and does not contain $e$. Let

$$
\mathfrak{C}^{e-e x c l}:=\{C \text { simple cycle in } G \mid e \notin C\}
$$

be the set of $e$-exclusive cycles.
Definition 4.18 (Oversized linkdesign) Let $\{u, v\} \in E$. A linkdesign is called oversized if it provides enough capacity such that it cannot be exceeded by any routing of the commodities $k \in \mathcal{K}$, even when the routing contains cycles. Let

$$
\begin{aligned}
\mathcal{L}_{\{u, v\}}^{\max }:=\left\{l \in \mathcal{L}_{\{u, v\}} \mid\right. & \sum_{k \in \mathcal{K}}\left(U_{k}+\bar{U}_{k}\right) \leq C_{u v}^{l}, \\
& \left.\sum_{k \in \mathcal{K}}\left(U_{k}+\bar{U}_{k}\right) \leq C_{v u}^{l}\right\}
\end{aligned}
$$

be the set of oversized linkdesigns for $\{u, v\} \in E$.
Definition 4.19 (Sufficient linkdesign) Let $\{u, v\} \in E, K_{u v}, K_{v u} \subseteq \mathcal{K}$. A linkde$\operatorname{sign} l \in \mathcal{L}_{\{u, v\}}$ is called sufficient with respect to the commodity subsets $K_{u v}$ and $K_{v u}$, if it provides enough capacity that all commodities $k \in K_{u v}$ can be routed from $u$ to $v$ and all commodities $k \in K_{v u}$ from $v$ to $u$ simultaneously. Accordingly we define

$$
\begin{aligned}
\mathcal{L}_{K_{u v} K_{v u}}^{+}:=\left\{l \in \mathcal{L}_{\{u, v\}} \mid\right. & \sum_{k \in K_{u v}} U_{k}+\sum_{k \in K_{v u}} \bar{U}_{k} \leq C_{u v}^{l}, \\
& \left.\sum_{k \in K_{u v}} \bar{U}_{k}+\sum_{k \in K_{v u}} U_{k} \leq C_{v u}^{l}\right\}
\end{aligned}
$$

to be the set of sufficient linkdesigns for the commodity subsets $K_{u v}$ and $K_{v u}$.
Definition 4.20 (Insufficient linkdesign) Let $\{u, v\} \in E, K_{u v}, K_{v u} \subseteq \mathcal{K}$. A linkdesign $l \in \mathcal{L}_{\{u, v\}}$ is called insufficient with respect to the commodity subsets $K_{u v}$ and $K_{v u}$, if it is not sufficient. Accordingly we define

$$
\mathcal{L}_{K_{u v} K_{v u}}^{-}:=\mathcal{L}_{\{u, v\}} \backslash \mathcal{L}_{K_{u v} K_{v u}}^{+}
$$

to be the set of insufficient linkdesigns for the commodity subsets $K_{u v}$ and $K_{v u}$.
Definition 4.21 (Large linkdesign) Let $\{u, v\} \in E, K_{u v}, K_{v u} \subseteq \mathcal{K}$. A linkdesign $l \in \mathcal{L}_{\{u, v\}}$ is called large with respect to the commodity subsets $K_{u v}$ and $K_{v u}$, if it is sufficient and furthermore provides enough capacity such that all commodities $k \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)$ can be routed over $\{u, v\}$ in arbitrary directions simultaneously. Accordingly we define

$$
\begin{aligned}
\mathcal{L}_{K_{u v} K_{v u}}^{\text {large }}:=\left\{l \in \mathcal{L}_{\{u, v\}} \mid\right. & \sum_{k \in K_{u v}} U_{k}+\sum_{k \in K_{v u}} \bar{U}_{k}+\sum_{k \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)} \max \left\{U_{k}, \bar{U}_{k}\right\} \leq C_{u v}^{l}, \\
& \left.\sum_{k \in K_{u v}} \bar{U}_{k}+\sum_{k \in K_{v u}} U_{k}+\sum_{k \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)} \max \left\{U_{k}, \bar{U}_{k}\right\} \leq C_{v u}^{l}\right\}
\end{aligned}
$$

the set of large linkdesigns for the commodity subsets $K_{u v}$ and $K_{v u}$.
Obviously, $\mathcal{L}_{\{u, v\}}^{\max } \subseteq \mathcal{L}_{K_{u v} K_{v u}}^{\text {large }} \subseteq \mathcal{L}_{K_{u v} K_{v u}}^{+} \subseteq \mathcal{L}_{\{u, v\}}$ holds for each $\{u, v\} \in E$, $K_{u v}, K_{v u} \subseteq \mathcal{K}$.

Definition 4.22 (Close linkdesign) Let $\{u, v\} \in E, K_{u v}, K_{v u} \subseteq \mathcal{K}$. A linkdesign $l \in \mathcal{L}_{\{u, v\}}$ is called uv-close with respect to the commodity subsets $K_{u v}$ and $K_{v u}$, if it is insufficient, but provides enough capacity such that as soon as any commodity $k^{*} \in K_{u v}$ is not routed from $u$ to $v$, all other commodities $k \in K_{u v} \backslash\left\{k^{*}\right\}$ can be routed from $u$ to $v$, all commodities $k \in K_{v u}$ from $v$ to $u$, and in addition, all commodities
$k \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)$ can be route over $\{u, v\}$ in arbitrary directions simultaneously. Accordingly we define

$$
\begin{aligned}
\mathcal{L}_{K_{u v} K_{v u}}^{u v-c l o s e}:=\left\{l \in \mathcal{L}_{K_{u v} K_{v u}}^{-} \mid\right. & \sum_{k \in K_{u v} \backslash\left\{k^{*}\right\}} U_{k}+\sum_{k \in K_{v u} \cup\left\{k^{*}\right\}} \bar{U}_{k} \\
& +\sum_{k \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)} \max \left\{U_{k}, \bar{U}_{k}\right\} \leq C_{u v}^{l} \\
& \sum_{k \in K_{u v} \backslash\left\{k^{*}\right\}} \bar{U}_{k}+\sum_{k \in K_{v u} \cup\left\{k^{*}\right\}} U_{k} \\
& \left.+\sum_{k \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)} \max \left\{U_{k}, \bar{U}_{k}\right\} \leq C_{v u}^{l} \quad \forall k^{*} \in K_{u v}\right\}
\end{aligned}
$$

to be the set of $u v$-close linkdesigns for the commodity subsets $K_{u v}$ and $K_{v u}$.
Analogously we define $\mathcal{L}_{K_{u v} K_{v u}}^{v u-c l o s e}$, the set of vu-close linkdesigns for the commodity subsets $K_{u v}$ and $K_{v u}$. These linkdesigns are insufficient, but provide enough capacity such that as soon as any commodity $k^{*} \in K_{v u}$ is not routed from $v$ to $u$ all other commodities can be routed accordingly.

Example 4.23 It seems un-apparent that these uv-close linkdesigns can exist. Therefore we present an example. Let $G=(V, E)$ with $\{u, v\} \in E$ and $\mathcal{K}=\left\{k_{1}, k_{2}, k_{3}, k_{4}\right\}$ with the properties presented in Table 4.2. We choose $K_{u v}=\left\{k_{1}, k_{2}\right\}$ and $K_{v u}=\left\{k_{3}\right\}$. It results that the linkdesign $l^{*}$ with the capacity values $C_{u v}^{l^{*}}=C_{v u}^{l^{*}}=80$ is uv-close and vu-close for the commodity subsets $K_{u v}$ and $K_{v u}$.

|  | $U_{k}$ | $\bar{U}_{k}$ |
| :---: | :---: | :---: |
| $k_{1}$ | 30 | 0 |
| $k_{2}$ | 30 | 0 |
| $k_{3}$ | 10 | 30 |
| $k_{4}$ | 10 | 10 |

Table 4.2: Existing commodities

### 4.3.2 Inequality

We now introduce a class of inequalities that are valid for our considered polyhedron. The inequalities were motivated by the fact that the routing of a set of commodities over an edge $e \in E$ forbids the installation of certain linkdesigns on $e$.

Proposition 4.24 Let $\{u, v\} \in E, K_{u v}, K_{v u} \subseteq \mathcal{K}$ with $K_{u v} \cap K_{v u}=\emptyset$, then

$$
\begin{equation*}
\sum_{k \in K_{u v}} f_{u v}^{k}+\sum_{k \in K_{v u}} f_{v u}^{k}+\sum_{l \in \mathcal{L}_{K_{u v} K v u}^{-}} x_{\{u, v\}}^{l} \leq\left|K_{u v}\right|+\left|K_{v u}\right| \tag{4.32}
\end{equation*}
$$

is a valid inequality for $\mathcal{P}$, stating that when we want to use an edge for the routing of some commodities, it must be equipped with an adequate linkdesign.

Proof. Assume that an integer point $\hat{p}=(\hat{x}, \hat{f}) \in \mathcal{P}$ does not satisfy (4.32) for an edge $\{u, v\} \in E$ and the commodity subsets $K_{u v}$ and $K_{v u} \in \mathcal{K}$ where $K_{u v} \cap K_{v u}=\emptyset$. The only way to violate the inequality is to route all commodities $k \in K_{u v}$ over $\{u, v\}$ in direction $u$ to $v$, all commodities $k \in K_{v u}$ over $\{u, v\}$ in direction $u$ to $v$, and equip at the same time $\{u, v\}$ with a linkdesign $\hat{l} \in \mathcal{L}_{K_{u v}}^{-} K_{v u}$, which is insufficient with respect to $K_{u v}$ and $K_{v u}$. This implies

$$
\begin{aligned}
\hat{f}_{u v}^{k}=1 \quad \forall k \in K_{u v} \\
\hat{f}_{v u}^{k}=1 \quad \forall k \in K_{v u} \\
\hat{x}_{\{u, v\}}^{\hat{l}}=1 .
\end{aligned}
$$

In this case, $\hat{p}$ does not satisfy one of the edge capacity constraints (3.20) or (3.21) concerning $\{u, v\}$ :

$$
\sum_{k \in \mathcal{K}} U_{k} \hat{f}_{u v}^{k}+\sum_{k \in \mathcal{K}} \bar{U}_{k} \hat{f}_{v u}^{k} \geq \sum_{k \in K_{u v}} U_{k}+\sum_{k \in K_{v u}} \bar{U}_{k}>C_{u v}^{\hat{l}}=\sum_{l \in \mathcal{L}_{\{u, v\}}} C_{u v}^{l} \hat{x}_{\{u, v\}}^{l}
$$

or

$$
\sum_{k \in \mathcal{K}} U_{k} \hat{f}_{v u}^{k}+\sum_{k \in \mathcal{K}} \bar{U}_{k} \hat{f}_{u v}^{k} \geq \sum_{k \in K_{v u}} U_{k}+\sum_{k \in K_{u v}} \bar{U}_{k}>C_{v u}^{\hat{l}}=\sum_{l \in \mathcal{L}_{\{u, v\}}} C_{v u}^{l} \hat{x}_{\{u, v\}}^{l}
$$

so that $\hat{p} \notin \mathcal{P}$, which is a contradiction.

### 4.3.3 Relaxation

Due to the fact that our problem with its high number of different constraints is quite complicated, the determination of facets resulted to be a bold venture. Therefore, we concentrate on a relaxation of our problem. We restrict ourselves to the problem of finding a feasible routing for our commodity set $\mathcal{K}$ and a feasible linkdesign installation. We do not consider any port, hoplimit, secure or node capacity constraints and do allow cycles. We define the polyhedra associated with this integer linear program by
$\mathcal{P}_{b a s e}:=\operatorname{conv}\left\{(x, f) \in\{0,1\}^{\sum_{e \in E}\left|\mathcal{L}_{e}\right|} \times\{0,1\}^{2 \times|E| \times|\mathcal{K}|} \mid(x, f)\right.$ satisfies $\left.\begin{array}{c}(3.1),(3.2), \\ (3.20),(3.21)\end{array}\right\}$

Among the inequalities (4.32), some are facet-defining for this polyhedron. The following proposition identifies some of these:

Proposition 4.25 Let $\{u, v\} \in E, K_{u v}, K_{v u} \subseteq \mathcal{K}$. If

1. $G=(V, E)$ is 3-connected (for every two nodes $i, j \in V$, there exist three node disjoint paths from i to $j$ )
2. $\mathcal{L}_{e}^{\max } \neq \emptyset \quad \forall e \in E$
3. $\mathcal{L}_{\{u, v\}} \subseteq\left(\mathcal{L}_{K_{u v} K_{v u}}^{u v-\text { close }} \cup \mathcal{L}_{K_{u v} K_{v u}}^{v u-\text { close }} \cup \mathcal{L}_{K_{u v} K_{v u}}^{\text {large }}\right)$
4. $\mathcal{L}_{K_{u v} K_{v u}}^{u v-\text {-close }} \cap \mathcal{L}_{K_{u v} K_{v u}}^{v u-c \text { close }} \neq \emptyset$,

## then (4.32) defines a facet of $\mathcal{P}_{\text {base }}$.

Proof. Validity of (4.32) for $\mathcal{P}_{\text {base }}$ follows from Proposition 4.24 (since each point in $\mathcal{P}_{\text {base }}$ has to fulfill the edge capacity constraints). Let

$$
\begin{equation*}
\mathcal{F}:=\left\{(x, f) \in \mathcal{P}_{\text {base }}\left|\sum_{k \in K_{u v}} f_{u v}^{k}+\sum_{k \in K_{v u}} f_{v u}^{k}+\sum_{l \in \mathcal{L}_{K_{u v} K_{v u}}^{-}} x_{\{u, v\}}^{l}=\left|K_{u v}\right|+\left|K_{v u}\right|\right\}\right. \tag{4.34}
\end{equation*}
$$

be the face of $\mathcal{P}_{\text {base }}$ induced by (4.32). We prove that $\mathcal{F}$ is a facet of $\mathcal{P}_{\text {base }}$ in three steps: First, $\mathcal{F} \neq \emptyset$ is shown by representing points in $\mathcal{F}$. Second, we show that any face containing $\mathcal{F}$ is either $\mathcal{P}_{\text {base }}$ or $\mathcal{F}$ itself. Third, we show that $\mathcal{P}_{\text {base }} \neq \mathcal{F}$ by representing a point in $\mathcal{P}_{\text {base }} \backslash \mathcal{F}$.

Claim 4.26 The face $\mathcal{F}$ is not empty
For each two nodes $v_{1}, v_{2} \in V$ there exits a $v_{1}-v_{2}$-path in the overlaying directed graph $D$ of $G$ (as defined in Section 2.2). We construct a point $\tilde{p}$ by considering for each commodity $k \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)$ a $o_{k}-d_{k}$-path and setting the respective commodity edge flow variables to 1 . For each commodity $k \in K_{u v}$ we consider an $(u, v)$-inclusive $o_{k}{ }^{-} d_{k^{-}}$ path and for all $k \in K_{v u}$ an ( $v, u$ )-inclusive $o_{k}-d_{k}$-path, and fixate the corresponding commodity edge flow variables to 1 . Finally we equip all edges $e \in E$ with an oversized linkdesign $l \in \mathcal{L}_{e}^{\max }$, so that we fulfill all edge capacity constraints and derive that $\tilde{p} \in \mathcal{F}$.

Claim 4.27 Any inequality valid for $\mathcal{P}_{\text {base }}$ representing $\mathcal{F}$ is a positive multiple of (4.32) plus a linear combination of (3.2)

At first we group the points in $\mathcal{F}$ and access the similarities within these groups later. Then we proof the claim by representing points in $\mathcal{F}$ and combining the flow balance constraints.

There exist three different types of points in $\mathcal{F}$ :

- All commodities $k \in K_{u v}$ are routed over the edge $\{u, v\}$ in the direction from $u$ to $v$ and all commodities $k \in K_{v u}$ from $v$ to $u$. In this case $\{u, v\}$ is equipped with a linkdesign $l \in \mathcal{L}_{K_{u v} K_{v u}}^{\text {large }}=\mathcal{L}_{\{u, v\}} \backslash \mathcal{L}_{K_{u v} K_{v u}}^{-}$.
- Exactly one commodity $k^{*} \in K_{u v}$ is not routed from $u$ to $v$, while the rest of the commodities $k \in K_{u v} \backslash\left\{k^{*}\right\}$ are routed over the edge $\{u, v\}$ in the direction from $u$ to $v$ and all commodities $k \in K_{v u}$ from $v$ to $u$. In this case $\{u, v\}$ is equipped with a $u v$-close linkdesign $l \in \mathcal{L}_{K_{u v} K_{v u}}^{u v-\text { close }} \subseteq \mathcal{L}_{K_{u v} K_{v u}}^{-}$.
- Exactly one commodity $k^{*} \in K_{v u}$ is not routed from $v$ to $u$, while the rest of the commodities $k \in K_{v u} \backslash\left\{k^{*}\right\}$ are routed over the edge $\{u, v\}$ in the direction from $v$ to $u$ and all commodities $k \in K_{u v}$ from $u$ to $v$. In this case $\{u, v\}$ is equipped with a $v u$-close linkdesign $l \in \mathcal{L}_{K_{u v} K_{v u}}^{v u-\text { close }} \subseteq \mathcal{L}_{K_{u v} K_{v u}}^{-}$.

We notice that the routing of any commodity $k \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)$ does not influence the capacity on the edge $\{u, v\}$. In all three types of points of $\mathcal{F}$ any routing of the commodities of $\mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)$ neither consumes nor releases a notable amount of capacity on $\{u, v\}$ such that the routing of the commodities $k \in K_{u v} \cup K_{v u}$ is affected. Thus the routing of a commodity $k \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)$ is totally independent of the capacity installed on the edge $\{u, v\}$ as well as from the routing of any other commodity over $\{u, v\}$.

Suppose there exists an equation of the form

$$
\begin{equation*}
\alpha x+\beta f=\pi \tag{4.35}
\end{equation*}
$$

satisfied by all points of $\mathcal{F}$, where $\alpha$ and $\beta$ are vectors of an appropriate dimension and $\pi$ is a real number. We show that (4.35) is a positive multiple of (4.32) plus a linear combination of the equal constraints of our polyhedral (i.e. the flow balance constraints (3.2)), partially using a similar proof as presented in [15].

From the point $\tilde{p} \in \mathcal{F}$ constructed above we can derive new points. Let $e^{*} \in$ $E \backslash\{u, v\}$, we consider the set of commodities $K_{\text {over_e }} e^{*}$ which are routed over $e^{*}$ in the routing employed in $\tilde{p}$. Since $G$ is 3 -connected there exist for each two nodes $o, d \in V$ two edge-disjoint $o-d$-paths in $D$, so that we can reroute all commodities $k \in K_{\text {over_e }^{*}}$ in the way that they do not use $e^{*}$ anymore. Since we use the same linkdesign installation as in $\tilde{p}$ there exists enough capacity and we get a new point of $\mathcal{F}$ with a routing where none of the commodities $k \in \mathcal{K}$ is using $e^{*}$. We can now generate new points of $\mathcal{F}$ by equipping $e^{*}$ with none or an arbitrary linkdesign $l \in \mathcal{L}_{e^{*}}$. Since the latter points all satisfy (4.35) with equality it follows that $\alpha_{e^{*}}^{l}=0 \quad \forall l \in \mathcal{L}_{e^{*}}$ and since the edge was chosen arbitrarily we can conclude that

$$
\alpha_{e}^{l}=0 \quad \forall e \in E \backslash\{u, v\}, l \in \mathcal{L}_{e}
$$

Now we consider the remaining coefficients corresponding to the linkdesign installation variables of the edge $\{u, v\}$. As mentioned above, the points of $\mathcal{F}$ can be categorized in three types. All points of a certain type have in common that the edge $\{u, v\}$ is equipped with a linkdesign of the corresponding linkdesign set $\left(\mathcal{L}_{K_{u v} K_{v u}}^{\text {large }}, \mathcal{L}_{K_{u v} K_{v u}}^{u v-c l o s e}\right.$ or $\left.\mathcal{L}_{K_{u v} K_{v u}}^{v u-c l o s e}\right)$. Taking a point from $\mathcal{F}$ with a certain type, we can construct another point
of the same type by equipping $\{u, v\}$ with another linkdesign from the corresponding linkdesign set. Since these two points fulfill (4.35) with equality and exploiting the fact that $\mathcal{L}_{K_{u v} K_{v u}}^{u v-c l o s e} \cap \mathcal{L}_{K_{u v} K_{v u}}^{v u-\text { close }} \neq \emptyset$, it follows that

$$
\begin{array}{ll}
\alpha_{\{u, v\}}^{l_{1}}=\alpha_{\{u, v\}}^{l_{2}} & \forall l_{1}, l_{2} \in \mathcal{L}_{K_{u v} K_{v u}}^{\text {large }} \\
\alpha_{\{u, v\}}^{l_{1}}=\alpha_{\{u, v\}}^{l_{2}} & \forall l_{1}, l_{2} \in \mathcal{L}_{K_{u v} K_{v u}}^{u v-\text { close }} \cup \mathcal{L}_{K_{u v} K_{v u}}^{v u-\text { close }}
\end{array}
$$

We now deal with the coefficients corresponding to the commodity edge flow variables of an arbitrary commodity $\check{k} \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)$.

We start once more from the point $\tilde{p} \in \mathcal{F}$ from the beginning and an edge $\{i, j\} \in E$. If the commodity $\check{k}$ is routed over $\{i, j\}$ we modify $\tilde{p}$ by rerouting $\check{k}$ but employing the same routing for the other commodities as well as the same link design installation. Since $G$ is 3 -connected we can find a $o_{\breve{k}}-d_{\breve{k}}$-path which is not using $\{i, j\}$, such that we obtain the new point $p^{\prime}$. A new point $p^{\prime \prime}$ of $\mathcal{F}$ is generated employing exactly the same routing strategy for all commodities $k \in \mathcal{K}$ as in $p^{\prime}$, except that commodity $\check{k}$ is additionally routed over the edge $\{i, j\}$ in the directions $i$ to $j$ and $j$ to $i$. Since both points $p^{\prime}$ and $p^{\prime \prime}$ satisfy the equality, $\alpha_{e}^{l}=0 \quad \forall e \in E \backslash\{u, v\}$ and routing of commodities $k \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)$ does not notably affect the capacity of $\{u, v\}$ it follows that $\beta_{i j}^{\check{k}}+\beta_{j i}^{\check{k}}=0 \quad \forall\{i, j\} \in E$.

Next we show that for all cycles $C$ in the overlaying digraph $D$ of $G$ it holds that $\sum_{(i, j) \in C} \beta_{i j}^{\breve{k}}=0$. Since any cycle in the graph can be decomposed into a collection of simple cycles (i.e. cycles that visit each node at most once) it follows that we only have to prove this claim for simple cycles. If $C$ is a 2 -cycle the claim is already shown. Let $\mathfrak{p}$ be a simple path from $o_{\check{k}}$ to $d_{\breve{k}}$ in the digraph $D$. If the number of nodes on the path $\mathfrak{p}$ that are also on the cycle $C$ is less than or equal to one, then we use similar arguments as before to show that $\sum_{(i, j) \in C} \beta_{i j}^{\check{k}}=0$. We consider a point in $\mathcal{F}$ using path $\mathfrak{p}$ for the routing of commodity $\check{k}$ and simply create a new point by conducting $\check{k}$ additionally over the cycle $C$.
If the number of nodes on the path $\mathfrak{p}$ that are also on the cycle $C$ is greater than or equal to 2 , then define $v_{1}$ as the first, and $v_{2}$ to be the last node on the path that is also on the cycle. As a result, path $\mathfrak{p}$ can be decomposed into three parts $\mathfrak{p}_{1}, \mathfrak{p}_{2}$ and $\mathfrak{p}_{3}$, where $\mathfrak{p}_{1}$ is a path from $o_{\check{k}}$ to $v_{1}, \mathfrak{p}_{2}$ is a path from $v_{1}$ to $v_{2}$, and $\mathfrak{p}_{3}$ is a path from $v_{2}$ to $d_{\check{k}}$. Similarly, the cycle $C$ can be decomposed into a path $C_{1}$ from $v_{1}$ to $v_{2}$ and a path $C_{2}$ from $v_{2}$ to $v_{1}$. Given these definitions, we can construct two new paths from $o_{\check{k}}$ to $d_{\check{k}}$ in the graph. The first path can be represented as $\mathfrak{p}_{1}, C_{1}, \mathfrak{p}_{3}$ and the second path as $\mathfrak{p}_{1}, C_{2}^{r}, \mathfrak{p}_{3}$, where $C_{2}^{r}$ is the reversed path of $C_{2}$. Consider a point in $\mathcal{F}$ using the first path for the routing of commodity $\check{k}$. We can construct now another point in $\mathcal{F}$ by employing the same routing strategy for all commodities $k \in \mathcal{K} \backslash\{\check{k}\}$, but using the second path for commodity $\check{k}$. Since both points satisfy the equality it follows that $\sum_{(i, j) \in C_{1}} \beta_{i j}^{\check{k}}-\sum_{(i, j) \in C_{2}^{r}} \beta_{i j}^{\check{k}}=0$. Exploiting the fact that $\beta_{i j}^{\check{k}}=-\beta_{j i}^{\check{k}} \quad \forall\{i, j\} \in E$, it follows that $\sum_{(i, j) \in C} \beta_{i j}^{\breve{k}}=\sum_{(i, j) \in C_{1}} \beta_{i j}^{\breve{k}}+\sum_{(i, j) \in C_{2}} \beta_{i j}^{\breve{k}}=0$,
which proves our intermediate claim. Since the commodity $\check{k}$ was chosen arbitrarily from $\mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)$, we can conclude that

$$
\sum_{(i, j) \in C} \beta_{i j}^{k}=0 \quad \forall k \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right),\{i, j\} \in E, C \text { cycle }
$$

Next, for a commodity $\check{k} \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)$, for all $q \in V$ and a path $\mathfrak{p}$ from $o_{\check{k}}$ to $q$ in the graph, let $\mu_{q}^{\check{k}}=\sum_{(i, j) \in \mathfrak{p}} \beta_{i j}^{\check{k}}$. We claim that the value of $\mu_{q}^{\check{k}}$ is independent of the selected path $\mathfrak{p}$. To verify this claim, let $\mathfrak{p}_{1}, \mathfrak{p}_{2}$ be two paths from $o_{\check{k}}$ to $q$ in the graph, and let $\mathfrak{p}_{1}^{r}, \mathfrak{p}_{2}^{r}$ be the reversed paths. Then $\mathfrak{p}_{1} \cup \mathfrak{p}_{2}^{r}$ forms a cycle, hence $\sum_{(i, j) \in \mathfrak{p}_{1} \cup \mathfrak{p}_{2}^{r}} \beta_{i j}^{\check{k}}=0$. Using $\beta_{i j}^{\check{k}}=-\beta_{j i}^{\check{k}}$ it then follows that $\sum_{(i, j) \in \mathfrak{p}_{1}} \beta_{i j}^{\check{k}}=\sum_{(i, j) \in \mathfrak{p}_{2}} \beta_{i j}^{\check{k}}$, thus indeed, the value of $\mu_{q}^{\grave{k}}$ is independent of the selected path from $o_{\breve{k}}$ to $q$. If we multiply the flow balance equalities corresponding to the commodities of $\mathcal{K} \backslash\left(K_{u v} \cup\right.$ $K_{v u}$ ) by these multipliers and add them all up, we obtain the following expression:

$$
\begin{aligned}
& \sum_{k \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)} \sum_{i \in V} \mu_{i}^{k}\left(\sum_{\{i, j\} \in \delta(i)}\left(f_{j i}^{k}-f_{i j}^{k}\right)\right)= \\
& \sum_{k \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)} \sum_{\{i, j\} \in E}\left(\left(\mu_{i}^{k}-\mu_{j}^{k}\right) f_{j i}^{k}+\left(\mu_{j}^{k}-\mu_{i}^{k}\right) f_{i j}^{k}\right)= \\
& \sum_{k \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)} \sum_{\{i, j\} \in E}\left(\beta_{i j}^{k} f_{i j}^{k}+\beta_{j i}^{k} f_{j i}^{k}\right)
\end{aligned}
$$

This implies that the coefficients $\beta_{i j}^{k}$ corresponding to commodities $k \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)$ are derived from a linear combination of the model equalities (3.2).

Now we address ourselves to the remaining coefficients of the commodity edge flow variables of the commodities $k \in K_{u v} \cup K_{v u}$. Starting from the point $\tilde{p} \in \mathcal{F}$ from the beginning and an edge $\left\{v_{1}, v_{2}\right\} \in E \backslash\{u, v\}$. Concerning a commodity $\check{k} \in K_{u v}$ and exploiting the fact that the graph is 3 -connected, we can find an $(u, v)$-inclusive $o_{\check{k}}-d_{\check{k}}$-path $\mathfrak{p}$ which does not use the edge $\{i, j\}$. We modify $\tilde{p}$ by rerouting $\check{k}$ using the prementioned path $\mathfrak{p}$. We employ the same routing for the other commodities as well as the same link design installation. A new point of $\mathcal{F}$ is generated employing exactly the same routing strategy for all commodities $k \in \mathcal{K}$, except that commodity $\check{k}$ is additionally routed over the edge $\{i, j\}$ in the directions $i$ to $j$ and $j$ to $i$. Since both points satisfy the equality and $\alpha_{e}^{l}=0 \quad \forall e \in E \backslash\{u, v\}$ it follows that $\beta_{i j}^{\check{k}}+\beta_{j i}^{\breve{k}}=$ $0 \forall\{i, j\} \in E \backslash\{u, v\}, l \in \mathcal{L}_{\{u, v\}}$. Since $\check{k}$ was chosen arbitrarily, and applying an analogue argumentation for the commodities $k \in K_{v u}$, we can conclude :

$$
\beta_{i j}^{k}+\beta_{j i}^{k}=0 \quad \forall k \in K_{u v} \cup K_{v u},\{i, j\} \in E \backslash\{u, v\}
$$

Using a similar argumentation as before and exploiting the fact that $G$ is 3 connected, we can show that

$$
\sum_{(i, j) \in C} \beta_{i j}^{k}=0 \quad \forall k \in K_{u v} \cup K_{v u},\{i, j\} \in E, C \in \mathfrak{C}^{\{u, v\}-e x c l} .
$$

Now we consider a spanning tree $T=\left(V, E^{\prime}\right)$ of $G$ using edges in $E \backslash\{u, v\}$. We choose a node $v_{r} \in V$ and direct the edges of $T$ suitably such that we obtain the directed tree $T^{\prime}=\left(V, A^{\prime}\right)$ with the root node $v_{r}$. For an $\operatorname{arc}\left(v_{r}, v_{c}\right) \in \delta^{+}\left(v_{r}\right) \cap A^{\prime}$ and a commodity $k \in K_{u v} \cup K_{v u}$ we consider the respective variable $f_{v_{r} v_{c}}^{k}$. Since this variable appears in the flow balance equality of $v_{c}$ and $k$, we can subtract a multiple of this equality from (4.35) and assume for the respective coefficient $\beta_{v_{r} v_{c}}^{k}=0$. We can do this analogously traversing the tree from the root $v_{r}$ to its leaves, subtracting successively for every arc $a \in A^{\prime}$ multiples of the flow balance equalities respective to its target and the commodity $k$.
For $k \in K_{u v} \cup K_{v u}$ appears each variable $f_{i j}^{k}$ with $(i, j) \in A^{\prime}$ in two flow balance equalities, one of the node $i$ and the other of $j$. On our way from $v_{r}$ to the leaves of $T^{\prime}$ we subtract at first a multiple of the flow balance equality of $i$. In the next step we subtract the flow balance equality of $j$ multiplied with (if necessary a negation of) the current coefficient $\beta_{i j}^{k}$ of the variable $f_{i j}^{k}$. Since we never pass the nodes $i$ or $j$ again on the path to the leaves, the variable $f_{i j}^{k}$ will not be considered anymore. So we derive $\beta_{i j}^{k}=0 \quad \forall(i, j) \in A^{\prime}$. Since $\beta_{i j}^{k}=-\beta_{j i}^{k}$ for any $\{i, j\} \in E \backslash\{u, v\}$, this implies that $\beta_{i j}^{k}=0 \quad \forall(i, j) \in E^{\prime}$ and $k \in K_{u v} \cup K_{v u}$.
For $\left\{v_{1}, v_{2}\right\} \in(E \backslash\{u, v\}) \backslash E^{\prime}$ we can find the unique cycle $C$ formed by $\left\{v_{1}, v_{2}\right\}$ and the edges in $E^{\prime}$. Since $C$ does not contain $\{u, v\}$ it follows that $\sum_{(i, j) \in C} \beta_{i j}^{k}=0$ and therefore $\beta_{v_{1}, v_{2}}^{k}=0$. We can finally conclude that

$$
\beta_{i j}^{k}=\beta_{j i}^{k}=0 \quad \forall k \in K_{u v} \cup K_{v u},\{i, j\} \in E \backslash\{u, v\} .
$$

Once more making use of the point $\tilde{p} \in \mathcal{F}$ presented in the beginning of the proof we consider the commodity $k^{*} \in K_{u v}$. Exploiting the fact that the graph is 3-connected, we can find a $(u, v)$-inclusive $o_{k^{*}}-d_{k^{*}}$-path $\mathfrak{p}$ and a $(u, v)$-inclusive cycle $C$, such that both do not use the same edges in the same direction ( $\mathfrak{p} \cap C=\emptyset$ ). We can now generate two new points in $\mathcal{F}$ from $\tilde{p}$ by employing the same linkdesign installation and commodity routing, except that $k^{*}$ is routed in both cases over the path $\mathfrak{p}$ and in one of the new points additionally on the cycle $C$. Since these two new points fulfill (4.35) with equality and $\beta_{i j}^{k^{*}}=0 \quad \forall(i, j) \notin\{(u, v),(v, u)\}$ it follows that $\beta_{v u}^{k^{*}}=0$. Since $k^{*}$ was chosen arbitrarily, and applying the argumentation analogously to the commodities in $K_{v u}$ we conclude that

$$
\begin{array}{ll}
\beta_{v u}^{k}=0 & \forall k \in K_{u v} \\
\beta_{u v}^{k}=0 & \forall k \in K_{v u} .
\end{array}
$$

We now address ourselves to the other types of points in $\mathcal{F}$ presented in the beginning of the proof. We construct a point $\tilde{p}_{u v}$ of the second group by installing on the edge $\{u, v\}$ a $u v$-close linkdesign $l \in \mathcal{L}_{K_{u v} K_{v u}}^{u v-\text { close }} \cap \mathcal{L}_{K_{u v} K_{v u}}^{v u-\text { close }}$ and equipping the other
edges with a maximal linkdesign. One commodity $k^{*} \in K_{u v}$ will be routed over an $(u, v)$-exclusive $o_{k^{*}}-d_{k^{*}}$-path, while the commodities $k \in K_{u v} \backslash\left\{k^{*}\right\}$ are routed over $(u, v)$-inclusive $o_{k}-d_{k}$-paths and all commodities $k \in K_{v u}$ over $(v, u)$-inclusive $o_{k}-d_{k^{-}}$ paths. The other commodities $k \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)$ can be routed arbitrarily since there exists enough capacity in respect of every edge and direction.
We can now construct a new point in $\mathcal{F}$ by employing the same linkdesign installation and routing of commodities, except that we replace $k^{*}$ by another commodity $k^{\prime} \in K_{u v}$. Commodity $k^{\prime}$ is now routed over an $(u, v)$-inclusive $o_{k^{*}}-d_{k^{*}}$-path and $k^{\prime}$ over an $(u, v)$-exclusive $o_{k^{\prime}}-d_{k^{\prime}}$-path. Since these two points fulfill (4.35) with equality we can state $\beta_{u v}^{k^{*}}=\beta_{u v}^{k^{\prime}}$.
A point in the third group can be generated from $\tilde{p}_{u v}$ by routing $k^{*}$ over an $(u, v)$ inclusive $o_{k^{*}}-d_{k^{*}}$-path and choosing one commodity $k^{\prime \prime} \in K_{v u}$ not to be routed over $\{u, v\}$ in direction $v$ to $u$. Since we employ in both points the same linkdesign installation and both fulfill (4.35) with equality it follows that $\beta_{u v}^{k^{*}}=\beta_{u v}^{k^{\prime \prime}}$.
We can now conclude that

$$
\beta_{u v}^{k_{1}}=\beta_{v u}^{k_{2}} \quad \forall k_{1} \in K_{u v}, k_{2} \in K_{v u} .
$$

Since $\tilde{p}_{u v}$ and $\tilde{p}$ are both points in $\mathcal{F}$ they both fulfill (4.35) with equality. The difference between them is the fact that $k^{*}$ is or is not routed over $\{u, v\}$ in direction $u$ to $v$ and $\{u, v\}$ is equipped with different linkdesigns $l^{\prime} \in \mathcal{L}_{K_{u v} K_{v u}}^{u v-c l o s e}$ or $l^{\prime \prime} \in \mathcal{L}_{K_{u v} K_{v u}}^{\text {large }}$. Since all other $\beta_{i j}^{k^{*}}$ are 0 we obtain that $\alpha_{\{u, v\}}^{l^{\prime}}=\beta_{u v}^{k^{*}}+\alpha_{\{u, v\}}^{l^{\prime \prime}}$. Considering other commodities in $K_{u v}$ analogously we can conclude the following:

$$
\alpha_{\{u, v\}}^{l_{1}}=\beta_{u v}^{k}+\alpha_{\{u, v\}}^{l_{2}} \quad \forall k \in K_{u v}, l_{1} \in \mathcal{L}_{K_{u v} K_{v u}}^{u v-c l o s e}, l_{2} \in \mathcal{L}_{K_{u v} K_{v u}}^{\text {large }}
$$

Defining the following auxiliary variables

$$
\begin{array}{rlrl}
\beta & :=\beta_{u v}^{k} & & \forall k \in K_{u v} \\
\alpha & :=\alpha_{\{u, v\}}^{l} & \forall l \in \mathcal{L}_{K_{u v} K_{v u}}^{u v-\text { close }} \\
\alpha^{\max } & :=\alpha_{\{u, v\}}^{l} & \forall l \in \mathcal{L}_{K_{u v} K_{v u}}^{l a r g e},
\end{array}
$$

we can state the intermediate result as follows:

$$
\begin{gather*}
\beta\left(\sum_{k \in K_{u v}} f_{u v}^{k}+\sum_{k \in K_{v u}} f_{v u}^{k}\right)+\alpha\left(\sum_{l \in \mathcal{L}_{K_{u v} K_{v u}}^{-}} x_{\{u, v\}}^{l}\right)+\alpha^{\max }\left(\sum_{l \in \mathcal{L}_{K_{u v} K_{v u}}^{+}} x_{\{u, v\}}^{l}\right) \leq \pi,  \tag{4.36}\\
\beta=\alpha-\alpha^{\max } . \tag{4.37}
\end{gather*}
$$

We multiply the initial inequality (4.32) with $\beta$ and add the linkdesign installation inequality (3.1) multiplied with $\alpha^{\text {max }}$. We obtain:

$$
\begin{align*}
& \beta\left(\sum_{k \in K_{u v}} f_{u v}^{k}+\sum_{k \in K_{v u}} f_{v u}^{k}\right) \\
& +\left(\alpha-\alpha^{\max }\right) \sum_{l \in \mathcal{L}_{K_{u v}}^{-}} x_{v u}^{l} x_{\{u, v\}}^{l} \leq \beta\left(\left|K_{u v}\right|+\left|K_{v u}\right|\right)  \tag{I}\\
& \alpha^{\max } \sum_{l \in \mathcal{L}_{\{u, v\}}^{+}} x_{\{u, v\}}^{l} \\
& +\alpha^{\max } \sum_{l \in \mathcal{L}_{\{u, v\}}^{-}} x_{\{u, v\}}^{l} \leq \alpha^{\max }  \tag{II}\\
& \beta\left(\sum_{k \in K_{u v}} f_{u v}^{k}+\sum_{k \in K_{v u}} f_{v u}^{k}\right) \\
& \quad+\alpha^{\max } \sum_{l \in \mathcal{L}_{K_{u v} K_{v u}}^{+} x_{\{u, v\}}^{l}} \begin{array}{l}
\quad+\alpha \sum_{l \in \mathcal{L}_{K_{u v} K_{v u}}^{-}} x_{\{u, v\}}^{l}
\end{array} \quad \leq \beta\left(\left|K_{u v}\right|+\left|K_{v u}\right|\right)+\alpha^{\max }
\end{align*}
$$

If $\alpha^{\max }>0$, the inequality (4.38) (and thus (4.36)) would be weaker than the initial inequality (4.32). Therefore $\alpha^{\max }$ has to be 0 and

$$
\alpha_{\{u, v\}}^{l}=0 \quad \forall l \in \mathcal{L}_{K_{u v} K_{v u}}^{\text {large }} .
$$

The above paragraph covers all relevant coefficients such that the Claim 4.27 is proven.

## Claim $4.28 \mathcal{F} \neq \mathcal{P}_{\text {base }}$

We can generate a point in $\mathcal{P}_{\text {base }} \backslash \mathcal{F}$ easily by equipping all edges $e \in E$ with a maximal linkdesign $l \in \mathcal{L}_{K_{u v} K_{v u}}^{\text {large }}$, routing all commodities $k \in K_{u v}$ over $(u, v)$-exclusive $o_{k}-d_{k}$ paths, all commodities $k \in K_{v u}$ over ( $v, u$ )-exclusive $o_{k}$ - $d_{k}$-paths and all $k \in \mathcal{K} \backslash\left(K_{u v} \cup\right.$ $K_{v u}$ ) arbitrarily.

This completes the proof that (4.32) is in fact facet-defining.

Remark 4.29 The assumption of 3-connectedness seems to be a very restrictive condition since telecommunication networks are typically sparse. This is not the case in the given data. All the networks of all considered problem instances are extremely dense and the great majority even consist of complete graphs.

Remark 4.30 The condition $\mathcal{L}_{\{u, v\}} \subseteq\left(\mathcal{L}_{K_{u v} K_{v u}}^{u v-c l o s e} \cup \mathcal{L}_{K_{u v} K_{v u}}^{v u-c l o s e} \cup \mathcal{L}_{K_{u v} K_{v u}}^{\text {large }}\right)$ is indeed very restrictive; we apply a lifting approach to include linkdesigns which are not in this union.

Remark 4.31 The assumption $\mathcal{L}_{K_{u v} K_{v u}}^{u v-\text { close }} \cap \mathcal{L}_{K_{u v} K_{v u}}^{v u-c \text { cose }} \neq \emptyset$ also results to be very restrictive, since it requires a certain composition for the commodity subsets $K_{u v}$ and $K_{v u}$. Defining $\Delta:=\sum_{k \in \mathcal{K} \backslash\left(K_{u v} \cup K_{v u}\right)} \max \left\{U_{k}, \bar{U}_{k}\right\}$, the subsets must fulfill the following: Either

$$
\Delta<U_{k_{1}}-\bar{U}_{k_{1}} \text { and } \Delta<\bar{U}_{k_{2}}-U_{k_{2}} \quad \forall k_{1} \in K_{u v}, k_{2} \in K_{v u}
$$

or

$$
\Delta<\bar{U}_{k_{1}}-U_{k_{1}} \text { and } \Delta<U_{k_{2}}-\bar{U}_{k_{2}} \quad \forall k_{1} \in K_{u v}, k_{2} \in K_{v u} .
$$

### 4.3.4 Lifting

In the latter section we have shown that the inequalities of the type (4.32) are facetdefining for a relaxation $\mathcal{P}_{\text {Base }}$ of our problem, in the case that some conditions are fulfilled. Since these requirements are quite restrictive, we lift the inequality (4.32). The next proposition from [11] is given here for readers unfamiliar with the lifting procedure.

Proposition 4.32 Suppose $S=\left\{x \in\{0,1\}^{n} \mid A x \leq b\right\}$, $S^{\delta}=\left\{x \in S \mid x_{1}=\right.$ $\delta$ for $\delta \in\{0,1\}\}$, and $\sum_{j=2}^{n} \pi_{j} x_{j} \leq \pi_{0}$ is valid for $S^{0}$. If $S^{1}=\emptyset$, then $x_{1} \leq 0$ is valid for $S$. If $S^{1} \neq \emptyset$, then

$$
\begin{equation*}
\alpha_{1} x_{1}+\sum_{j=2}^{n} \pi_{j} x_{j} \leq \pi_{0} \tag{4.39}
\end{equation*}
$$

is valid for any $\alpha_{1} \leq \pi_{0}-\xi$ where

$$
\begin{equation*}
\xi=\max \sum_{j=2}^{n} \pi_{j} x_{j} \quad x \in S^{1} \tag{4.40}
\end{equation*}
$$

Moreover, if $\alpha_{1}=\pi_{0}-\xi$ and $\sum_{j=2}^{n} \pi_{j} x_{j} \leq \pi_{0}$ is facet-defining for $\operatorname{conv}\left(S^{0}\right)$, then (4.39) is facet-defining for $\operatorname{conv}(S)$.

Proposition 4.32 represents a lifting with respect to the variable $x_{1}$. We call $\alpha_{1}$ the lifting coefficient for $x_{1}$.

In the following we show how the inequality (4.32) concerning an edge and the two commodity subsets, has to be lifted, such that it is facet-defining even when the corresponding set of linkdesigns contains also linkdesigns which are neither large nor
uv- or vu-close. Let $\{u, v\} \in E, K_{u v}, K_{v u} \subseteq \mathcal{K}$, we define:

$$
\begin{align*}
& \mathcal{L}_{K_{u v} K_{v u}}^{\text {res }}:=\left\{l \in \mathcal{L}_{\{u, v\}} \mid l \notin \mathcal{L}_{K_{u v} K_{v u}}^{u v-\text { close }} \cup \mathcal{L}_{K_{u v} K_{v u}}^{v u-\text { close }} \cup \mathcal{L}_{K_{u v} K_{v u}}^{\text {large }}\right\}  \tag{4.41}\\
& S_{\text {base }}^{\text {res }}:=\left\{(x, f) \in\{0,1\}^{\sum_{e \in E}\left|\mathcal{L}_{e}\right|} \times\{0,1\}^{2 \times|E| \times|\mathcal{K}|} \mid(x, f)\right. \text { satisfies }  \tag{3.1}\\
& \left.\sum_{l \in \mathcal{L}_{K u v K v u}^{r e s}} x_{\{u, v\}^{l}} \leq 0\right\} \text {. } \tag{3.20}
\end{align*}
$$

With assistance of the fixing of the variables corresponding to the linkdesigns $l \in$ $\mathcal{L}_{K_{u v} K_{v u}}^{r e s}$, Proposition 4.25 implies that (4.32) is facet-defining for $\operatorname{conv}\left(S_{\text {base }}^{r e s}\right)$, if

1. $G=(V, E)$ is 3-connected
2. $\mathcal{L}_{e}^{\max } \neq \emptyset \quad \forall e \in E$
3. $\mathcal{L}_{K_{u v} K_{v u}}^{u v-\text { close }} \cap \mathcal{L}_{K_{u v} K_{v u}}^{v u-c \text { close }} \neq \emptyset$.

We consider lifting the variable $x_{\{u, v\}}^{\grave{l}}$ corresponding to a linkdesign $\check{l} \in \mathcal{L}_{K_{u v} K_{v u}}^{r e s}$ and define:

$$
\begin{align*}
& S_{\text {base }}^{\text {res }}(\check{l}):=\left\{(x, f) \in\{0,1\}^{\sum_{e \in E}\left|\mathcal{L}_{e}\right|} \times\{0,1\}^{2 \times|E| \times|\mathcal{K}|} \mid(x, f)\right. \text { satisfies } \tag{3.1}
\end{align*}
$$

The lifting coefficient is $\alpha_{\check{l}}=\left|K_{u v}\right|+\left|K_{v u}\right|-\xi_{\check{l}}$, where $\xi_{\check{l}}$ can be obtained as follows:

$$
\begin{align*}
\xi_{\grave{l}} & =\max _{(x, f) \in S_{b a s e}^{e s e}(l)}\left\{\sum_{k \in K_{u v}} f_{u v}^{k}+\sum_{k \in K_{v u}} f_{v u}^{k}+\sum_{l \in \mathcal{L}_{K_{u v} K_{v u}}^{-}} x_{\{u, v\}}^{l} \mid x_{\{u, v\}}^{\check{l}}=1\right\} \\
& =\max _{(x, f) \in S_{b a s e}^{r e s}(\check{l})}\left\{\sum_{k \in K_{u v}} f_{u v}^{k}+\sum_{k \in K_{v u}} f_{v u}^{k} \mid x_{\{u, v\}}^{\check{l}}=1\right\} \tag{4.44}
\end{align*}
$$

The value $\xi_{\check{l}}$ is the amount of commodities $k \in K_{u v}$ and $K_{v u}$ which can be routed simultaneously over the edge $\{u, v\}$ in direction $u$ to $v, v$ to $u$ respectively, without exceeding the directed capacities provided by the linkdesign $l$. More precisely:

$$
\begin{align*}
\xi_{\check{l}}=\max \left\{\left|\check{K}_{u v}\right|+\left|\check{K}_{v u}\right|\right. & \mid \check{K}_{u v} \subseteq K_{u v}, \check{K}_{v u} \subseteq K_{v u}, \\
& \sum_{k \in \breve{K}_{u v}} U_{k}+\sum_{k \in \check{K}_{v u}} \bar{U}_{k} \leq C_{u v}^{\check{l}}, \\
& \left.\sum_{k \in \check{K}_{u v}} \bar{U}_{k}+\sum_{k \in \check{K}_{v u}} U_{k} \leq C_{v u}^{\check{l}}\right\} . \tag{4.45}
\end{align*}
$$

Since all linkdesign installation variables $\mathcal{L}_{\{u, v\}}$ are contained in one GUB (represented by the corresponding linkdesign constraints (3.1)), we can calculate all lifting coefficients independent of the order. Hence (4.32) can be lifted by solving (4.45) for each $l \in \mathcal{L}_{K_{u v} K_{v u}}^{\text {res }}$, such that
is facet-defining for $\mathcal{P}_{\text {Base }}$, if

1. $G=(V, E)$ is 3-connected
2. $\mathcal{L}_{e}^{\max } \neq \emptyset \quad \forall e \in E$
3. $\mathcal{L}_{K_{u v} K_{v u}}^{u v-c l o s e} \cap \mathcal{L}_{K_{u v} K_{v u}}^{v u-c \text { cose }} \neq \emptyset$.

## Chapter 5

## Algorithmic approach and implementational aspects

This chapter describes the algorithmic procedure used to solve the problems of the given benchmark suite. In the first part of this chapter we give an outline of the employed branch-and-cut approach. Later, we present separation approaches for inequalities, various preprocessing mechanisms and introduce a heuristic for finding feasible integral solutions.

### 5.1 Branch-and-Cut

### 5.1.1 Overview

We may find a solution by examining all possible combinations exhaustively. However, the number of combinations becomes explosively large. Thus a simple enumerative method is not applicable to our problem. In order to cope with the vast solution space, we have employed a branch-and-cut technique for effective exploration. This technique is a generalization of branch-and-bound approach with application of cutting planes.

At first we give a short survey of the generic branch-and-bound algorithm: Let $N$ be a variable index set and $Z \subseteq N$. Let $c \in \mathbb{R}^{N}$ be an objective vector, and $P \subseteq \mathbb{R}^{N}$ be a polyhedron, e.g. a relaxation of some MIP. Let $\underline{x}, \bar{x} \in \mathbb{R}^{N}$. Consider the optimization problem

$$
\begin{align*}
\min \left\{c^{T} x \mid\right. & x \in P, \\
& \underline{x} \leq x \leq \bar{x},  \tag{5.1}\\
& \left.x_{i} \text { integer } \forall i \in Z\right\} .
\end{align*}
$$

The branch-and-bound algorithm solves this problem using a modified divide-andconquer approach. The following algorithm summarizes the whole procedure:

## Algorithm 5.1 (Branch-and-bound)

Input : Bounded problem (5.1): $\min \left\{c^{T} x \mid x \in P, \underline{x} \leq x \leq \bar{x}, x \in \mathbb{R}^{N}, x_{i} \in \mathbb{Z} \forall i \in Z\right\}$
Output: Optimal solution for (5.1)
(2) $\quad$ while $\mathcal{A} \neq \emptyset$ do

```
```

(1) $\mathcal{T} \leftarrow\left\{\left(\underline{x}^{\prime}, \bar{x}^{\prime}\right)\right\}, \mathcal{A} \leftarrow\left\{\left(\underline{x}^{\prime}, \bar{x}^{\prime}\right)\right\},\left\{\left(\underline{x}^{\prime}, \bar{x}^{\prime}\right)\right\}, z^{*} \leftarrow+\infty$

```
```

(1) $\mathcal{T} \leftarrow\left\{\left(\underline{x}^{\prime}, \bar{x}^{\prime}\right)\right\}, \mathcal{A} \leftarrow\left\{\left(\underline{x}^{\prime}, \bar{x}^{\prime}\right)\right\},\left\{\left(\underline{x}^{\prime}, \bar{x}^{\prime}\right)\right\}, z^{*} \leftarrow+\infty$
(2.4.2.1) $\quad x^{\prime}$ is valid solution for the original problem (5.1).
(2.4.2.1) $\quad x^{\prime}$ is valid solution for the original problem (5.1).
(2.4.2.2) if $c^{T} x^{\prime}<z^{*}$ then
(2.4.2.2) if $c^{T} x^{\prime}<z^{*}$ then
(2.4.2.2.1) $\quad z^{*} \leftarrow c^{T} x^{\prime}, x^{*} \leftarrow x^{\prime}$
(2.4.2.2.1) $\quad z^{*} \leftarrow c^{T} x^{\prime}, x^{*} \leftarrow x^{\prime}$
if $z^{*}=\infty$ then
if $z^{*}=\infty$ then

```
        Select a \(\left(\underline{x}^{\prime}, \bar{x}^{\prime} \in \mathcal{A}\right.\)
```

        Select a \(\left(\underline{x}^{\prime}, \bar{x}^{\prime} \in \mathcal{A}\right.\)
        \(\mathcal{A} \leftarrow \mathcal{A} \backslash\left\{\left(\underline{x}^{\prime}, \bar{x}^{\prime}\right)\right\}\)
        \(\mathcal{A} \leftarrow \mathcal{A} \backslash\left\{\left(\underline{x}^{\prime}, \bar{x}^{\prime}\right)\right\}\)
        Solve relaxation \(\min \left\{c^{T} x \mid x \in P, \underline{x}^{\prime} \leq x \leq \bar{x}^{\prime}, x \in \mathbb{R}^{N}\right\}\).
        Solve relaxation \(\min \left\{c^{T} x \mid x \in P, \underline{x}^{\prime} \leq x \leq \bar{x}^{\prime}, x \in \mathbb{R}^{N}\right\}\).
        if there exists an optimal solution \(x^{\prime}\) with \(c^{T} x<z^{*}\) then
        if there exists an optimal solution \(x^{\prime}\) with \(c^{T} x<z^{*}\) then
            if there exists an \(i \in Z\) such that \(x_{i}^{\prime}\) is fractional then
            if there exists an \(i \in Z\) such that \(x_{i}^{\prime}\) is fractional then
                            Branch on \(i\) : Create two subproblems ( \(\underline{x}^{\downarrow}, \bar{x}^{\downarrow}\) ) and \(\left(\underline{x}^{\uparrow}, \bar{x}^{\uparrow}\right)\) that equal
                            Branch on \(i\) : Create two subproblems ( \(\underline{x}^{\downarrow}, \bar{x}^{\downarrow}\) ) and \(\left(\underline{x}^{\uparrow}, \bar{x}^{\uparrow}\right)\) that equal
                            \((\underline{x}, \bar{x})\) with the exception that \(\bar{x}_{i}^{\downarrow}=\left\lfloor x_{i}^{\prime}\right\rfloor\) and \(\underline{x}_{i}^{\uparrow}=\left\lceil x_{i}^{\prime}\right\rceil\).
                            \((\underline{x}, \bar{x})\) with the exception that \(\bar{x}_{i}^{\downarrow}=\left\lfloor x_{i}^{\prime}\right\rfloor\) and \(\underline{x}_{i}^{\uparrow}=\left\lceil x_{i}^{\prime}\right\rceil\).
                            \(\stackrel{\mathcal{T}}{\leftarrow} \leftarrow \mathcal{T} \cup\left\{\left(\underline{x}^{\downarrow}, \bar{x}^{\downarrow}\right),\left(\underline{x}^{\uparrow}, \bar{x}^{\uparrow}\right)\right\}\)
                            \(\stackrel{\mathcal{T}}{\leftarrow} \leftarrow \mathcal{T} \cup\left\{\left(\underline{x}^{\downarrow}, \bar{x}^{\downarrow}\right),\left(\underline{x}^{\uparrow}, \bar{x}^{\uparrow}\right)\right\}\)
                        \(\mathcal{A} \leftarrow \mathcal{A} \cup\left\{\left(\underline{x}^{\downarrow}, \bar{x}^{\downarrow}\right),\left(\underline{x}^{\uparrow}, \bar{x}^{\uparrow}\right)\right\}\)
                        \(\mathcal{A} \leftarrow \mathcal{A} \cup\left\{\left(\underline{x}^{\downarrow}, \bar{x}^{\downarrow}\right),\left(\underline{x}^{\uparrow}, \bar{x}^{\uparrow}\right)\right\}\)
            else
            else
        return 'problem (5.1) has no solution'
        return 'problem (5.1) has no solution'
        else
        else
        return 'optimal solution \(x^{*}\) with objective value \(z^{*}\),
    ```
        return 'optimal solution \(x^{*}\) with objective value \(z^{*}\),
```

Algorithm 5.1 solves the relaxation of (5.1) obtained by dropping the integrality constraints. If the solution contains variables that are fractional, albeit they are required to be integral in the original problem, it creates two subproblems such that the current point is feasible for neither of the two, but any solution of the original problem is applicable to one of them.
Since this technique is widely known, we do not go into further details. Rather some facts which become important later in this work are discussed now:

- The problems $\left(\underline{x}^{\prime}, \bar{x}^{\prime}\right) \in \mathcal{T}$ have a tree-like structure, because each problem, except for the initial one, has a unique parent problem from which it was created. It is common to refer to elements of $\mathcal{T}$ as node in a tree, and to call the initial problem the root node.
- The optimal solution value of a node's relaxation is a lower bound for the according MIP. The offspring problems of a node contain all integral (w.r.t. Z) points of that node's MIP. Hence, the greater of the two lower bounds can be used as lower bound for the node itself. This way, bounds get propagated up to the root node, whose lower bound is valid for the original problem and all nodes.
- If the optimal solution to a node's problem relaxation is integer w.r.t. $Z$, (2.4.2) gets executed and no child problems are generated, even if ( $\underline{x}^{\prime} \neq \bar{x}^{\prime}$ ). This is correct because no child problem can have a fractional solution of better objective, let alone integral ones. This reduces the size of $\mathcal{T}$.
- The if-clause (2.4) could be omitted without breaking algorithm correctness. It exploits the fact that $c^{T} x^{\prime}$ is a lower bound for all integral (w.r.t. $Z$ ) solutions of the node's local problem. Hence if $c^{T} x^{\prime}<z^{*}$, the node can only contain an optimal solution to (5.1) if the already known $x^{*}$ is optimal as well. This method of removing (pruning) sub-trees is called bounding.

Branch-and-cut is a generalization of branch-and-bound where, after solving the LP-relaxation and having not been successful in pruning the node on basis of the LP solution, we try to find a violated inequality of a set of valid inequalities. The problem of finding such an inequality is called the separation problem. Sometimes, the separation problem is restricted to a certain class of inequalities, in which case we are looking for a violated inequality of this class. If we are able to find such a inequality, we can strengthen the relaxation and the LP is reoptimized. In that way we cut off fractional solutions such that local and global lower bounds rise faster and integral solutions found more quickly.

Figure 5.1 gives a survey on the used algorithm, where $Z^{*}$ denotes the best known upper bound and $Z_{L P}$ the lower bound, obtained from the LP-relaxation.

There are two steps in the branch-and-bound part that leave some choices. In step (2.1) of Algorithm 5.1, we have to select the next (sub)-problem from the list of unsolved problems to work on next, denoted as node selection. In step (2.4.1.1) we must decide how to split the problem into subproblems, referred to as variable selection or branching. In the next two sections we describe the applied branch-andbound strategies.

### 5.1.2 Node selection

In the first usage of the node selection, we look for the subproblem with the worst lower bound. We select this problem with the intent of improving the global lower bound. After a certain number of branch-and-bound iterations, we change the strategy and dive in the branch-and-bound tree, i.e. the "up"-node of the previous iteration is chosen until no child nodes are generated for the chosen nodes. This phase is equivalent to subsequently rounding up variables until the local problem is either feasible, or a solution is found. This changeover is applied to find integer solution and hence improve the upper bound.

### 5.1.3 Variable selection

In our implementation, we used three different strategies of variable selection, a generic one and two newly developed ones, which utilize information about the relation of certain variables.

Figure 5.1: Flowchart of the applied branch-and-cut algorithm

The generic branching rule is the fixation of the most fractional variable to 1 or 0 for the following two branches. The other two branching rules are branching over the commodity edge flow variables primarily, considering the paths manifested in the current solution. If no fractional commodity edge flow variables exist, we branch over the linkdesign installation variables using the above generic branching rule. Given a fractional solution $(\hat{x}, \hat{f})$, these rules can be applied as follows:
most fractional Branch on a variable $\hat{x}_{e}^{l}$ or $\hat{f}_{u v}^{k}$ that is closest to 0.5 .
most splitted path We compute the node-commodity-pair where the flow is most splitted. This means that we look for a node $v \in V$ and a commodity $k \in \mathcal{K}$ where the maximal number of edges are used to route the flow of $k$ out of $v$ :

$$
\max _{v \in V, k \in \mathcal{K}}\left|\left\{\{u, v\} \in \delta(v) \mid \hat{f}_{v u}^{k}>0\right\}\right|
$$

From the set of outgoing variables of $v$ and $k$ we pick the greatest and fix it to 0 respectively 1 . If none of the paths is splitted, which means that there exist no fractional commodity edge flow variables, we branch over the linkdesign installation variables using the above most fractional branching rule.
least splitted path The only difference to the above most splitted path branching rule is the choice of the node-commodity-pair. We look for a node $v \in V$ and a commodity $k \in \mathcal{K}$ where the splitted flow of $k$ is routed out of $v$ using the least number of edges. We only consider commodity edge flow variables with values less than 1 because we still want to detect splitted flows.

$$
\min _{v \in V, k \in \mathcal{K}}\left|\left\{\{u, v\} \in \delta(v) \mid 0<\hat{f}_{v u}^{k}<1\right\}\right|
$$

### 5.2 Separation

The separation problem can be stated as follows. Given a solution to the LP-relaxation of a problem, decide whether the solution satisfies a given set of inequalities, and if not, find one or more inequalities violated by the solution.
We separated the inequalities presented in Chapter 3. Note that if one of these inequalities is violated, but is not identified, this slows down the solution process (the bound could have been tightened), but the algorithm still produces an optimal solution. Furthermore we did not include all of the model inequalities, in order to decrease the size of the formulation (and thus the solving time of the LP-relaxation). These inequalities are also separated.

### 5.2.1 Separation of the GUB cover inequalities

For the separation of the GUB cover inequalities presented in section 4.1 we apply the separators proposed in [16].

### 5.2.2 Separation of the hoplimit constraints

Since the size of the problem formulation resulted in a major difficulty for large problem instances, we tried leaving out the hoplimit constraints (3.13) and separated the respective inequalities. Two different techniques were applied:

Trivial separation We can find violated length restriction inequalities easily by simply summing up the current values of all commodity edge flow variables of a commodity $k \in \mathcal{K}$. If the sum is greater than the hoplimit $M_{k}$ we just add the associated inequality (3.13) to the formulation.

Separation using shortest paths We noticed that the support (the amount of nonzero coefficients) of added inequalities influences the solving time of the resulting LP-relaxation. So we identify for each commodity $k \in \mathcal{K}$ the path $\mathfrak{p}$ over which it is mostly routed, check the length of $\mathfrak{p}$ and (in case of violation) add the inequality $\sum_{(u, v) \in \mathfrak{p}} f_{u v}^{k} \leq M_{k}$. These inequalities are weaker than the model path length constraints (3.13) used in the above paragraph, but at the same time the support is smaller and they reliably exclude solutions with violated paths when the flow for a commodity is determined to a large extent.

### 5.2.3 Separation of the subtour elimination constraints

Since the amount of subtour elimination constraints (3.3) is exponential in the number of nodes, we are separating these constraints. In this subsection we present two approaches in order to handle this. The first separation procedure, according to an idea presented in [17] in the context of the traveling salesman problem, utilizes the consideration of a linear program. For the second approach we introduce the weaker subtour elimination constraint, which can be separated applying simple combinatorial techniques.

Let $(\tilde{x}, \tilde{f})$ be the optimal solution of the LP-relaxation without the subtour elimination constraints. For each $k \in \mathcal{K}$ we look for the most violated subtour elimination inequality (3.3), which is equivalent to the maximization problem

$$
\begin{equation*}
\zeta_{k}=\max _{W \subseteq V}\left\{\sum_{\{u, v\} \in E(W)}\left(\tilde{f}_{u v}^{k}+\tilde{f}_{v u}^{k}\right)-|W|\right\} . \tag{5.2}
\end{equation*}
$$

If $\zeta_{k} \leq-1$ for every commodity $k \in \mathcal{K}$, then all subtour elimination inequalities (3.3) are fulfilled and $(\tilde{x}, \tilde{f}) \in \mathcal{L P}$, the LP-relaxation of our problem defined in Section 3.3. We first convert the maximization problem (5.2) into a linear integer program, and then we observe that the LP-relaxation is integral. We represent the unknown set $W \subseteq V$ by defining variables for all $v \in V$ as follows:

$$
z_{v}= \begin{cases}1 & \text { if } v \in W  \tag{5.3}\\ 0 & \text { otherwise }\end{cases}
$$

With these variables, (5.2) is equivalent to

$$
\begin{equation*}
\zeta_{k}=\max \left\{\sum_{\{u, v\} \in E}\left(\left(\tilde{f}_{u v}^{k}+\tilde{f}_{v u}^{k}\right) z_{u} z_{v}\right)-\sum_{v \in V} z_{v} \mid z_{v} \in\{0,1\}^{|V|}\right\} . \tag{5.4}
\end{equation*}
$$

We can eliminate the quadratic terms by introducing additional variables :

$$
w_{\{u, v\}}=\left\{\begin{array}{ll}
1 & \text { if } z_{u}=z_{v}=1  \tag{5.5}\\
0 & \text { otherwise }
\end{array} \quad \forall\{u, v\} \in E\right.
$$

Hence we can reformulate (5.4) as the following integer program:

## Formulation 5.2 (Subtour-IP)

$$
\begin{align*}
\max \sum_{\{u, v\} \in E}\left(\left(\tilde{f}_{u v}^{k}+\tilde{f}_{v u}^{k}\right) w_{\{u, v\}}\right)- & \sum_{v \in V} z_{v} \\
w_{\{u, v\}} \leq z_{u} & \forall\{u, v\} \in E  \tag{5.6}\\
w_{\{u, v\}} \leq z_{v} & \forall\{u, v\} \in E  \tag{5.7}\\
w_{\{u, v\}} \geq z_{v}+z_{u}-1 & \forall\{u, v\} \in E  \tag{5.8}\\
& \\
w_{\{u, v\}} \in\{0,1\} & \forall\{u, v\} \in E \\
z_{v} \in\{0,1\} & \forall v \in V
\end{align*}
$$

Proposition 5.3 If $\tilde{f}_{u v}^{k}, \tilde{f}_{v u}^{k} \geq 0$ for all $\{u, v\} \in E$, then the optimal solution of the $L P$-relaxation of the Subtour-IP is integral.

Proof. From (5.6) and (5.7) it follows that $w_{\{u, v\}} \leq \min \left\{z_{u}, z_{v}\right\} \forall\{u, v\} \in E$. The Subtour-IP is a maximization problem and $\tilde{f}_{u v}^{k}$ and $\tilde{f}_{v u}^{k} \geq 0$ for all $\{u, v\} \in E$. Thus there exists an optimal solution for the Subtour-IP with $w_{\{u, v\}}$ as large as possible such that (with the assistance of (5.6) and (5.7)) $w_{\{u, v\}}=\min \left\{z_{u}, z_{v}\right\}$. Since $z_{v} \leq 1$, for all $v \in V$, it holds that $\min \left\{z_{u}, z_{v}\right\} \geq z_{v}+z_{u}-1$ such that the inequalities (5.8) are fulfilled and can be omitted.
In each constraint of (5.6) and (5.7) there exist exactly two nonzero coefficients: one with +1 and one with -1 . From this it follows that the constraint matrix without the conditions (5.8) is totally unimodular [14]. Hence solving the LP-relaxation of the Subtour-IP without the constraints (5.8) will result in an integer optimal solution.

From Proposition 5.3 it follows that it is possible to separate the subtour elimination constraints (3.3) by solving a linear program. Actually, these inequalities can be separated in polynomial time. In spite of that it is too time consuming to set up a linear program and solve it in every separation step. On the other hand it is possible
to separate weaker subtour elimination inequalities with simple combinatorial methods. The subtour elimination inequalities (3.3) exclude cycles on a node subset. The inequalities

$$
\begin{equation*}
\sum_{a \in C} f_{a}^{k} \leq|C|-1 \quad \forall k \in \mathcal{K}, C \subseteq A: C \text { is a cycle } \tag{5.9}
\end{equation*}
$$

exclude cycles from the solutions as well, but they are weaker than (3.3).
For any commodity $k \in \mathcal{K}$, consider the overlaying directed graph $D_{k}=(V, A)$ of $G\left(\right.$ as defined in Section 2.2) with the arc weights $x_{(u, v)}=f_{u v}^{k}$ and $x_{(v, u)}=f_{v u}^{k} \quad \forall e=$ $\{u, v\} \in E$. In order to prohibit solutions with cycles we introduce the weaker subtour elimination inequality:

$$
\begin{equation*}
\sum_{a \in C} x_{a} \leq|C|-1 \quad \forall C \subseteq A: C \text { is a cycle } \tag{5.10}
\end{equation*}
$$

When the above inequality is fulfilled, then there exist no cycles in the routing for the commodity $k \in \mathcal{K}$.

## Separation of the weaker subtour elimination inequality:

Finding the most violated inequality is equivalent to the maximation problem

$$
\begin{equation*}
\zeta=\max _{C \text { cycle }}\left\{\sum_{a \in C} x_{a}-|C|\right\} \tag{5.11}
\end{equation*}
$$

All inequalities are fullfilled if $\zeta \leq-1$. Defining a new arc weight function $y$ with $y_{a}=1-x_{a} \quad \forall a \in A$, we can reformulate:

$$
\max _{C \text { cycle }}\left\{\sum_{a \in C} x_{a}-|C|\right\}=\max _{C \text { cycle }}\left\{\sum_{a \in C}\left(1-y_{a}\right)-|C|\right\}=\max _{C \text { cycle }} \sum_{a \in C}\left(-y_{a}\right)
$$

Therefore (5.11) is equivalent to

$$
\begin{equation*}
\min _{C \text { cycle }} \sum_{a \in C} y_{a}=-\zeta \tag{5.12}
\end{equation*}
$$

If $\min _{C}$ cycle $\sum_{a \in C} y_{a} \geq 1(\zeta \leq-1)$ is fullfilled, then no violated inequality exists. This can be verified by searching for a shortest cycle in $D_{k}$ with arc weights $y_{a}=1-x_{a}$. When we detect such a cycle $C$ for a commodity $k \in \mathcal{K}$, we add the inequality

$$
\begin{equation*}
\sum_{\{u, v\} \in E(C)}\left(f_{u v}^{k}+f_{v u}^{k}\right) \leq|C|-1 \tag{5.13}
\end{equation*}
$$

to the formulation.
Remark 5.4 Since $x_{a} \in[0,1] \quad \forall a \in A$, none of the $y_{a}$ is negative, and we can apply a modified Floyd-Warshall algorithm in order to find a shortest cycle. This All-Pairs-Shortest-Path algorithm can be extended to find the shortest $v$ - $v$-path for all nodes $v \in V$. We simply iterate in all three nested loops over all nodes.

### 5.3 Preprocessing

We apply a series of preprocessing steps in order to reduce our problem size. We take advantage of natural observations of the problem as well as particular characteristics of the structure of the used data. Our aim is, concerning particular problem instances, to keep out irrelevant constraints and to prevent the consideration of unnecessary variables. We state the main ideas of these reductions and specify the relevancy concerning the given data. To comprehend the assumptions the reader is referred to Chapter 6 and especially to Table 6.1 which contains the main information about the problem instances of the whole benchmark suite.

### 5.3.1 Constraints reduction

Particularities of the given problem instances might result in excluding some constraints.

Nodes with many ports If the port limits for a node $v \in V$ are very large we do not have to include the corresponding pmax constraints (3.6) and (3.7) into the formulation. These constraints can be totally ignored for a certain node $v \in V$ if the amount of its ports is very big, or more precisely if

$$
\min \left\{P_{v}^{-}, P_{v}^{+}\right\} \geq|\delta(v)| \sum_{\{u, v\} \in \delta(v)} \max _{l \in \mathcal{L}_{\{u, v\}}}\left\{P_{u v}^{l}, P_{v u}^{l}\right\}
$$

This case, in which the port amount of a node could never be exceeded, appears for example in the problem instance A04 or A05.

Only secure commodities In problem instances where all commodities are secure, all edge capacity constraints (3.20) and (3.21) are dominated by the corresponding stronger secure edge capacity constraints (3.22) and (3.23). This is due to the equality of the secure and insecure value of each commodity $k \in \mathcal{K}\left(U_{k}=U_{k}^{S}\right.$ and $\bar{U}_{k}=\bar{U}_{k}^{S}$ ) which yields

$$
\begin{align*}
\sum_{k \in \mathcal{K}} U_{k} f_{u v}^{k} & +\sum_{k \in \mathcal{K}} \bar{U}_{k} f_{v u}^{k}=\sum_{k \in \mathcal{K}} U_{k}^{S} f_{u v}^{k}+\sum_{k \in \mathcal{K}} \bar{U}_{k}^{S} f_{v u}^{k} \\
& \leq \sum_{l \in \mathcal{\mathcal { L } _ { \{ u , v \} }}} S_{\{u, v\}}^{l} C_{u v}^{l} x_{\{u, v\}}^{l} \leq \sum_{l \in \mathcal{\mathcal { L } _ { \{ u , v \} }}} C_{u v}^{l} x_{\{u, v\}}^{l} \quad \forall\{u, v\} \in E \tag{5.14}
\end{align*}
$$

respectively

$$
\begin{align*}
\sum_{k \in \mathcal{K}} U_{k} f_{v u}^{k} & +\sum_{k \in \mathcal{K}} \bar{U}_{k} f_{u v}^{k}=\sum_{k \in \mathcal{K}} U_{k}^{S} f_{v u}^{k}+\sum_{k \in \mathcal{K}} \bar{U}_{k}^{S} f_{u v}^{k} \\
& \leq \sum_{l \in \mathcal{\mathcal { L } _ { \{ u , v \} }}} S_{\{u, v\}}^{l} C_{v u}^{l} x_{\{u, v\}}^{l} \leq \sum_{l \in \mathcal{\mathcal { L } _ { \{ u , v \} }}} C_{v u}^{l} x_{\{u, v\}}^{l} \quad \forall\{u, v\} \in E \tag{5.15}
\end{align*}
$$

The special case that all commodities have to be routed securely appears in the instances C10 and C12.

No secure commodities If, on the other side, none of the commodities have to be routed securely we do not consider the secure constraints, as well as in the case where only secure linkdesigns exist.

High hoplimits If the hoplimit for a commodity $k \in \mathcal{K}$ is very large $\left(M_{k} \geq|V|-1\right)$, we do not consider the corresponding constraint (3.13).

High node capacity When the traffic limit of a node $v \in V$ is very large ( $C_{v} \geq$ $\sum_{k \in \mathcal{K}}\left(U_{k}+\bar{U}_{k}\right)$ ), we do not consider the respective node capacity constraint (3.8).

Nomult As a result of the generation of linkdesigns, we do not need any restrictive conditions concerning an edge for the nomult case.

### 5.3.2 Reduction of the commodity edge flow variables

Some commodity edge flow variables can be erased or fixed in the preprocessing phase if the data holds certain characteristics.

No cycles Since we do not want to allow cycles in our routing it is always prohibited to route a commodity into its origin or out of its destination node. Hence for each commodity $k \in \mathcal{K}$ and appropriate node $v \in V$ the variables $f_{u o_{k}}^{k}$ and $f_{d_{k} v}^{k}$ do not have to be generated.

Risky node exclusion Concerning the security constraint sec, various commodity edge flow variables can be fixed or erased. Since a secure commodity must not be routed over a risky node, certain variables can be set to 0 as described in (3.11) and (3.12). We can not profit from this reduction in the larger instance since all instances of the series B and $C$ (except for C 16 ) contain only secure nodes.

Small hoplimits We can also apply variable reduction exploiting the existence of hoplimits. Actually, the hoplimit constraint should be perfectly suited for preprocessing as a number of the commodity edge flow variables might be fixed to 0 . Regarding a commodity $k \in \mathcal{K}$ with the hoplimit $M_{k}$ we calculate all $o_{k}-d_{k^{-}}$ paths with a length lower or equal to $M_{k}$ (with respect to the overlaying directed graph). Now all variables whose corresponding arc does not appear in any of these paths can be fixed to 0 .
Unfortunately, the graphs of the ROCOCO benchmark suite are very dense (there are indeed only three instances (C10, C12 and C16) which are not complete), such that in general two nodes are connected by an edge. Hence, as soon as the hoplimit of a commodity $k$ is greater 2 nearly each edge and direction (or respectively arc) lies on a potential $o_{k}-d_{k}$-path. In the case when $M_{k}=2$, all
variables $f_{v u}^{k}$ and $f_{v u}^{k}$ with $u, v \notin\left\{o_{k}, d_{k}\right\}$ can be erased or fixed to 0 . This very strict hop limit appears, for example, in the instances A06 and C20.

Symmetrical routing As described in Chapter 3, commodities that have to be routed symmetrically can be merged together. In general the number of commodities and thus the number of commodity edge flow variables is halved. In the problem instance C20, for example, the 404 commodities could be reduced to 190.

Low node capacity The observation of the node capacities may result in the elimination of some variables. If for a commodity $k \in \mathcal{K}$ the amount of data $\max \left\{U_{k}, \bar{U}_{k}\right\}$ is greater than the node capacity $C_{v}$ of some node $v \in V$, the commodity edge flow variables corresponding to $k$ and the edges $e \in \delta(v)$ in both directions can be eliminated. This preprocessing approach has no effect on the given data since the node capacity is at least as large as the commodity with the highest amount of data.

### 5.3.3 Reduction of the linkdesign installation variables

When the data fulfills certain conditions, it is possible that the set $\mathcal{L}_{e}$ of linkdesigns of an edge $e \in E$ and hence the number of linkdesign installation variables can be reduced significantly.

High port consumption Concerning the port constraints pmax, all linkdesigns with an extraordinary high port consumption can be excluded from the set of linkdesigns. A linkdesign is dispensable if its minimal port consumption is greater than the maximum of in and out ports of all nodes. Let $\{u, v\}, l \in \mathcal{L}_{\{u, v\}}$; we can exclude $l$ if

$$
\min \left\{P_{u v}^{l}, P_{v u}^{l}\right\}>\max _{v \in V}\left\{P_{v}^{-}, P_{v}^{+}\right\} .
$$

This simplification has been applied successfully for example in the problem instance C11

No multiplier Restrictions of the base capacity multiplier, denoted by nomult, reduce the set of linkdesigns (as described in Chapter 3 concerning the data transformation and generation of the linkdesigns) so that the corresponding linkdesign installation variables are not even created.

Only secure commodities Using the secure parameters, it is also possible to reduce the number of linkdesigns. When all commodities have to be routed securely there is no use for the application of risky linkdesigns. Concerning for example the instance C12 this yields to halve the amount of linkdesigns corresponding to an edge $e \in E$.

### 5.4 Heuristics

We applied a heuristic to find feasible integral solutions which, at the same time, provide upper bounds in the branch-and-bound tree in order to enable fathoming of branches.
The main idea of our heuristic approach is to determine a feasible routing which imposes certain capacity requirements for each edge. Based on these values, we solve an IP on the linkdesign installation variables and achieve a feasible linkdesign configuration. This IP can be solved with a MIP-solver in reasonable time, since it contains a relative small number of variables and constraints.

Figure 5.2 gives a survey on the used algorithm. For each commodity $k \in \mathcal{K}$ we determine a $o_{k}$ - $d_{k}$-path $\mathfrak{p}_{k}$ in $D=(V, A)$, the overlaying digraph of $G$ (as defined in Section 2.2), fulfilling certain restrictions. Each arc in $A$ has a weight appointed by the current commodity and $\operatorname{len}\left(\mathfrak{p}_{k}\right)$ denotes the length of the path respective to these weights, while $\left|\mathfrak{p}_{k}\right|$ denotes the number of arcs in $\mathfrak{p}_{k}$. The set of paths (one for each commodity) represents a routing, for which we calculate a feasible linkdesign installation.

### 5.4.1 Determination of a feasible routing

In order to compute a feasible routing we primarily have to consider the flow balance constraints (3.2), the subtour elimination constraints (3.3), the path length constraints (3.13), the node capacity constraints (3.26), and the node secure capacity constraints (3.11), (3.12). The charge concerning the fulfillment of the edge (secure) capacity constraints (3.20)-(3.23) will be passed over to the problem of finding a feasible linkdesign installation for this routing.
Based on the (fractional) values of the commodity edge variable $\hat{f}$ we generate a feasible integer routing fulfilling the above constraints described as follows:

Determine a single path For each commodity $k \in \mathcal{K}$ we determine an $o_{k}-d_{k}$-path $\mathfrak{p}_{k}$ in the overlaying digraph $D=(V, A)$. This path can be found by searching for a shortest $o_{k}-d_{k}$-path with respect to preassigned nonnegative arc weights. These weights may be influenced by the current fractional routing denoted by $\hat{f}$. Since $\mathfrak{p}_{k}$ is simple, the routing is feasible with respect to the flow balance and the subtour elimination constraints.

Attending hoplimits Calculating the individual/specific paths concerning a commodity $k \in \mathcal{K}$, we can easily verify whether the amount of the used arcs is greater than the limitation $M_{k}$. If this happens we can try to find a shorter $o_{k}-d_{k}$-path applying another weight function or finally terminate this heuristic iteration unsuccessfully. If we continue we know that the routing is feasible concerning the path length constraints.
frag replacements
mmodities $k \in \mathcal{K}$ in list L
Take $k$ from L
Is L empty?
ts for the arcs of the over-
$D$, prohibiting certain arcs ortest $o_{k}-d_{k}$-path $\mathfrak{p}_{k}$ in $D$
$\left|\mathfrak{p}_{k}\right| \leq M_{k} ?$
Node capacity exceeded? $\operatorname{len}\left(\mathfrak{p}_{k}\right)<\infty$ ?

Update weights?
Update weights Find a feasible
linkdesign installation for the calculated routing (represented by the $\mathfrak{p}_{k}$ ) by solving a MIP

Figure 5.2: Flowchart of the applied heuristic algorithm

Attending node capacity Whenever we have calculated a feasible path for a commodity $k \in \mathcal{K}$, we add, for each node $v \in V$ on the path, the value of $k$ to the sum of the commodities which we already dealt with and which were also routed over $v$. If the sum exceeds the node capacity $C_{v}$, we try to find another path for $k$ prohibiting the arcs incident to $v$ (by setting the corresponding arc weights to the maximal number). If this approach is not successful we can apply another weight function or finally terminate the heuristic unsuccessfully. When we continue we know that the node capacity constraints are fullfilled by the found routing.

Attending secure node capacity We now assert that the routing does not violate the node secure capacity constraints. For a secure commodity $k \in \mathcal{K}$, we prohibit nodes that are not secure (and neither origin nor destination of $k$ ) by setting the weight of the incident arcs to the maximal number). So when we determine a shortest $o_{k}-d_{k}$-path with a length less than the maximal number, we know that $k$ is not routed over risky nodes, such that the routing is feasible concerning the node secure capacity constraints.

After calculation of the feasible routing, we have for each commodity $k \in \mathcal{K}$ a set of $\operatorname{arcs} A_{k} \subseteq A$ representing a cycle-free $o_{k}-d_{k}$-path. This corresponds to the fixed variable assignment $\tilde{f}$ with

$$
\tilde{f}_{u v}^{k}=\left\{\begin{array}{ll}
1 & \text { if }(u, v) \in A_{k} \\
0 & \text { otherwise }
\end{array} \quad \forall(u, v) \in A .\right.
$$

### 5.4.2 Determination of a feasible linkdesign installation

The mechanism described above provides us with a feasible routing represented by the commodity edge variable assignment $\tilde{f}$. In order to get a feasible linkdesign installation for this routing, we have to consider the (secure) capacity constraints (3.20)-(3.23), the linkdesign installation constraints (3.1) as well as the port constraints (3.6) and (3.7). The optimal solution $\tilde{x}$ of the following IP provides us with an optimal linkdesign installation concerning the given routing.

## Formulation 5.5 (Linkdesign installation IP)

$$
\min \sum_{e \in E} \sum_{l \in \mathcal{L}_{e}} W_{e}^{l} x_{e}^{l}
$$

$$
\begin{array}{cc}
\sum_{k \in \mathcal{K}} U_{k} \tilde{f}_{u v}^{k}+\sum_{k \in \mathcal{K}} \bar{U}_{k} \tilde{f}_{v u}^{k} \leq \sum_{l \in \mathcal{L}_{\{u, v\}}} C_{u v}^{l} x_{\{u, v\}}^{l} & \forall\{u, v\} \in E \\
\sum_{k \in \mathcal{K}} U_{k} \tilde{f}_{u v}^{k}+\sum_{k \in \mathcal{K}} \bar{U}_{k} \tilde{f}_{v u}^{k} \leq \sum_{l \in \mathcal{L}_{\{u, v\}}} C_{v u}^{l} x_{\{u, v\}}^{l} & \forall\{u, v\} \in E \\
\sum_{k \in \mathcal{K}} U_{k}^{S} \tilde{f}_{u v}^{k}+\sum_{k \in \mathcal{K}} \bar{U}_{k}^{S} \tilde{f}_{v u}^{k} \leq \sum_{l \in \mathcal{L}_{\{u, v\}}} S_{\{u, v\}}^{l} C_{u v}^{l} x_{\{u, v\}}^{l} & \forall\{u, v\} \in E \\
\sum_{k \in \mathcal{K}} U_{k}^{S} \tilde{f}_{u v}^{k}+\sum_{k \in \mathcal{K}} \bar{U}_{k}^{S} \tilde{f}_{v u}^{k} \leq \sum_{l \in \mathcal{L}_{\{u, v\}}} S_{\{u, v\}}^{l} C_{v u}^{l} x_{\{u, v\}}^{l} & \forall\{u, v\} \in E \\
\sum_{\{u, v\} \in \delta(v)} \sum_{l \in \mathcal{L}_{\{u, v\}}} P_{u v}^{l} x_{\{u, v\}}^{l} \leq P_{v}^{-} & \forall v \in V \\
\sum_{\{u, v\} \in \delta(v)} \sum_{l \in \mathcal{L}_{\{u, v\}}} P_{v u}^{l} x_{\{u, v\}}^{l} \leq P_{v}^{+} & \forall v \in V \\
\sum_{l \in \mathcal{L}_{e}} x_{e}^{l} \leq 1 \quad \forall e \in E  \tag{5.22}\\
x_{e}^{l} \in\{0,1\} & \forall e \in E, l \in \mathcal{L}_{e}
\end{array}
$$

Combining both partial solutions $\tilde{f}$ and $\tilde{x}$, the heuristic yields a feasible integral solution $\tilde{p}=(\tilde{x}, \tilde{f})$ to our problem.

### 5.4.3 Additional edge exclusion

Since our approach (due to the utilization of shortest path calculation) results in a dense network, we run into some major problems when we are confronted with strict port limits. The number of ports of a node imposes an upper bound of the number of edges which can be equipped by some linkdesigns and therefore can be used by some routing. Small port limits enforce sparse networks while the resulting routings of our heuristic approach require dense networks. We coped with this problem by adapting the weights of arcs and commodity after every iteration. (Arcs of) edges which are already used by some commodities get very cheap as long as there is still enough capacity left. Virgin edges (edges were nothing is routed until now) are quite expensive. Every time we use an edge the first time we remove a port of its incident nodes. If the ports of a node are exceeded we forbid the incident arcs (by setting their weight to the maximal limit).

### 5.4.4 Heuristic parameters

We have two options to change the attitudes of our heuristic while determinating a feasible routing. Different commodity sequences are imaginable as well as various arc weight functions.

## Order of commodities

The importance of the order of the commodities follows from the idea that the first considered commodities can already establish the outline of the required linkdesign installation. It seems reasonable that commodities with high needs (in terms of large (secure) demand value or strict hoplimits) should first have the possibility to fulfill their requirements.
demand value We order the commodities in a way such that the routing of the commodities with a high demand value $\left(U_{k}+\bar{U}_{k}\right)$ is fixed first.
secure data amount The commodities are ordered such that at first the routing of commodities with a big secure demand value $\left(U_{k}^{S}+\bar{U}_{k}^{S}\right)$ is determined.
hoplimit The order of the commodities is associated with the hoplimits, so that first the commodities with a very small hoplimit are concerned.

## Weights of arcs

To each commodity $k \in \mathcal{K}$ and each arc $(u, v) \in A$ we can apply various arc weights $w^{k}: A \rightarrow \mathbb{R}_{+}$. We can distinguish between two types of arc weight functions; static functions which always provide the same weights and flexible functions influenced by the current fractional solution. Since the static weight functions always provide the same routing, we only apply them once in the root node of the branch-and-bound tree. The heuristic implementation using the flexible weight functions is employed several times. In any case we disable the arcs corresponding to the commodity edge flow variables eliminated or fixed to zero before, by setting their weights to the maximal number.
trivial distance We search a shortest path with respect to the number of arcs by setting

$$
w_{(u, v)}^{k}=1 \quad \forall(u, v) \in A .
$$

trivial costs The average installation costs of linkdesigns on an edge is integrated in the weight generation of the respective arcs:

$$
w_{(u, v)}^{k}=\left(\sum_{l \in \mathcal{L}_{\{u, v\}}} W_{\{u, v\}}^{l}\right) /\left|\mathcal{L}_{\{u, v\}}\right| \quad \forall(u, v) \in A .
$$

current flow We use the current solution values of the commodity edge flow variables by setting

$$
w_{(u, v)}^{k}=\left(1-\hat{f}_{u v}^{k}\right) \quad \forall(u, v) \in A .
$$

current pricy flow The current flow and the average costs of an edge are integrated in the weight value of the respective arc:

$$
w_{(u, v)}^{k}=\left(\left(\sum_{l \in \mathcal{L}_{\{u, v\}}} W_{\{u, v\}}^{l}\right) /\left|\mathcal{L}_{\{u, v\}}\right|\right) \cdot\left(1-\hat{f}_{u v}^{k}\right) \quad \forall(u, v) \in A
$$

Remark 5.6 Since the weights of the arcs will always be nonnegative, the shortest paths in (5.4.1) can be determined efficiently using the DIJKSTRA algorithm.
We could have applied a hop-limited DIJKSTRA, so that we would not have had to attend the corresponding constraints afterwards.

## Chapter 6

## Data and computational results

In this chapter we describe the problem instances contained in the ROCOCO benchmark suite and expose some structural characteristics of certain instances. Afterwards, we present the impact of preprocessing and data transformation and the results applying certain algorithm options obtained by extensive tests using the given data. The chapter concludes with a presentation of our computational results for every problem instance and combination and a comparison with the the corresponding best published results.
Concerning the smaller problem instances, our implementation provided exactly the same optimal solution values as the ones published, which indicates that our approach models the given problem description properly. For some instances and combinations we were even able to obtain better upper bounds than the ones found until now. But in general the gap between the obtained upper and lower bounds resulted to be quite large due to the restrictive time limit.

### 6.1 Data set

The applied benchmark suite consists of three series, each including 7 instances whereupon 6 parameters are included in any combination. This yields 1344 problem instances. Series A contains the smallest instances, from 4 to 10 nodes, while the series $B$ and C include larger graphs up to 25 nodes. What seems remarkable is the fact that the graphs of all problem instances are extremely dense; nearly all (except C10, C12 and C16) are even complete. There exist instances with multiple commodities having the same origin and destination (in Section 3.2 referred to as parallel), e.g. in the instance C16. There are as well instances with commodities which are isolated, as defined in Section 3.2. In every instance there appears at least one commodity which has to be routed securely and there also always exist both, secure and risky linkdesigns. Table 6.1 shows also the occurrence of unidirected linkdesigns in instance C11 (i.e. linkdesigns providing a capacity greater 0 for one direction and no capacity for the other). It can also be noticed that the number of ports of the nodes in the instances of
the series B and C are quite small, and that therefore the port constraint proves to be quite restrictive. In C10, for instance, all nodes have two inports and two outports so that at each node, at most two incident edges can be equipped with a linkdesign. This imposes that the resulting capacity network is either a chain or a single loop. Except for C20, in every instance all commodities have the same hoplimit.

### 6.2 Computational results

In this section we present a listing of our results on the ROCOCO benchmark suite. We present the influence of the preprocessing and data generation endeavors in order to minimize the size of the problem. Additionally, we used the data sets for further analysis of certain settings for algorithm and heuristic. Our implementation was run on all 27 instances of the benchmark suite, and each combination of the additional constraints was considered. For upper bounds, we made a comparison with the best solutions published until now.

The possible 64 combinations are identified by a six-bit vector, whereby the first digit indicates if the security constraint sec is considered. The second bit corresponds to the consideration of the nomult constraint, while the third states if the symmetrical routing constraint symdem was taken into account. The forth and fifth bit indicate the consideration of the hoplimit constraints bmax and the port constraints pmax respectively. The last digit corresponds to the consideration of the node capacity constraints tmax. For example, " 011000 " indicates that only the second constraint nomult and the third constraint symdem were taken into account. The order is presented in Table 6.2.

We implemented our algorithm on top of the network dimensioning tool DISCNET [4]. The data structures and algorithms are implemented in $\mathrm{C}++$, using the $\mathrm{C}++$ library and partially LEDA [3]. We used CPLEX 8.0 [10] to solve the LPs in the branch-and-bound tree and the MIPs in the heuristic. All results were obtained within a 10 -minute time limit on an $\operatorname{Intel}(\mathrm{R}) \mathrm{Xeon}(\mathrm{TM}) \mathrm{CPU} 3.06 \mathrm{GHz}$ running the Linux operating system.

### 6.2.1 Downsizing results

In this subsection we present the results towards downsizing the problems. We consider the consequences of our endeavors concerning the reduction of variables as well as the constraint reduction. Considering the combination where all additional constraints are taken into account (i.e. the combination 111111), we start with the initial problem size corresponding to a naive formulation of the problem. Then we document the impact of our more sophisticated model and the consequential data transformation on the number of variables/constraints and finally the influence of preprocessing.

 number of unidirected linkdesigns



$\left.\begin{array}{r}\mid \text {｜47suo，} \mid \\ \mid \text { s．sp } \\ \hline\end{array} \right\rvert\,$



 ：


 for all combinations

 s

| L890ZI－09さてI | 009082 | 9－I | 0ャて0さ－ъて0さ | 0 | 9 | 01 | † | ¢9－91 | 901 | ¢88 | 0 | 808 | 29才 | 008 | 9 | Z618 | 92 | gz | 9\％． |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| セサセ¢0โ－0998 | 0t9get | g－I | 0ъて0さ－๖て0¢ | 0 | 9 | 01 | 9－\％ | 82I－¢ |  | 061 | 92 | 88 | ¢0ヶ | 061 | 9 | 乙618 | 07 | 02 | 0zo |
| 06zgz－tze\％ | 9898t | $\varepsilon-\mathrm{I}$ | 000z0โ－8も0z | 0 | $\varepsilon$ | 9 | g | 0991L－8もし | 0 | $\varepsilon 6$ | 0 | 99 | 981 | ¢ıI | 9 | 00098 | 01 | 91 | 910 |
| LZ665－G208 | 96268 | g－I |  | 0 | 9 | 01 | † | 8Zt－0z | ¡ぇ | п01 | 0 | 86 | ¢81 | got | 9 | 960¢ | gI | $9 \pm$ | 910 |
| 8080t－02LT | 0902I | g－I | 00002さ－8ち0て | 0 | 9 | 01 | $\pm$ | くてฑ6โ－て¢くL | 0 | 99 | 0 | z81 | z81 | z9 | 9 | 000zLz | zI | zı | zio |
| 68tL－E9ET | 980\＆t | g－I |  | $0{ }^{1}$ | gi | 02 | $\dagger$ | 26I－0T | 乙 | 99 | 0 | 68 | 801 | 99 | $\dagger$ | 960t | II | LI | It， |
| 8028－88 | 6699 | $\varepsilon-\mathrm{I}$ | 000z0t－8ち0z | 0 | $\varepsilon$ | 9 | g | 0991t－8t ${ }^{\text {d }}$ | 0 | \＆ 8 | 0 | 99 | 99 | Lt | 乙 | 00098 | 01 | 01 | 01， |
| 2890てI－09才てI | 000988 | ¢－I | 0096－8てI | 0 | 02 | 98 | $\dagger$ | ャ9－91 | 901 | $\dagger 8$ ¢ | 0 | ع0¢ | 29ヵ | 008 | 9 | 2618 | 98 | 9 c | 9zq |
| 9929L－068 | 0z0ゅ币 | g－I | 0096－8ZI | 0 | $0 z$ | ¢ | $\dagger$ | 82I－g | ゅI | 061 | 0 | LIZ | 998 | 061 | 9 | 2618 | 0 O | 02 | 0zя |
| gst | 09289 | g－I | 0096－8てI | 0 | $0 z$ | $9 \%$ | $\dagger$ | 00t－01 | Iz | 9 IL | 0 | 28 | LIz | 0zt | 9 | 9607 | 91 | 91 | 919 |
| 6Lも0z－g20¢ | 028Lt | g－I | 0096－8zI | 0 | $0 z$ | gz | $\dagger$ | 8て！－0̌ | ゅて | ゅ01 | 0 | 86 | ¢81 | got | 9 | 960ヶ | ¢1 | ¢1 | ¢19 |
| 94LL－も6さT | zL69 | g－I | 0096－8てI | 0 | $0 z$ | 98 | $\dagger$ | 6もて－ヵ9 | 81 | $\varepsilon 9$ | 0 | 69 | 801 | 99 | 9 | 960t | ZI | zI | zIG |
| 68t－898 | 018\＆ | g－I | 0096－8ZI | 0 | $0 z$ | gz | † | 26I－01 | 乙 | g9 | 0 | 68 | 801 | 99 | † | 960ヶ | LI | ${ }_{\text {LI }}$ | LIG |
| 2809－9001 | 0006 | ¢－I | 0096－8ZI | 0 | 02 | ¢ | † | 86I－ヤL | $\varepsilon$ | git | 0 | 甲¢ | 28 | 9t | ¢－z | 960t | 01 | 01 | 019 |
| ZLLE－992 | 9689 | $\varepsilon-\mathrm{I}$ | てLG－も9 | 0 | 9 | 9 | $\varepsilon$ | $99-$－ | 0 | เ\＆ | 0 | 9 | z9 | $9 \pm$ | 99z－z | ZIG | 2 | 01 | 0tV |
| \＆も¢z－899 | z988 | $\varepsilon-\mathrm{I}$ | てL¢－t9 | 0 | 9 | 9 | $\varepsilon$ | $99-$－ | 0 | $9 \%$ | 0 | $\dagger$ | OS | 98 | 99z－z | ZIg | 9 | 6 | 60 V |
| てL9T－セ0ヶ | 9\＆ゅて | $\varepsilon-\mathrm{I}$ | てLS－ゅ9 | 0 | 9 | 9 | $\varepsilon$ | $99-$－ | 0 | $0{ }^{2}$ | 0 | † | Ot | 87 | 99z－z | ZIS | g | 8 | 80 V |
| L6ti－gic | 6991 | \＆－I | てL¢－t9 | 0 | 9 | 9 | $\varepsilon$ | 99－ヤ | 0 | 81 | 0 | 乙 | 98 | 18 | 99z－z | zig | † | 2 | 20 V |
| 089－toz | 988 | $\varepsilon-I$ | てLS－ヶ9 | 0 | g | 9 | z | 99－ヵ | 0 | ¢I | 0 | z | 92 | ¢1 | 99z－z | zig | $\varepsilon$ | 9 | 90 V |
| 968－0¢ | 097 | \＆－I | 99z－ヶ9 | 0 | $\dagger$ | 9 | z | 99－8 | 0 | 01 | 0 | z | 02 | 01 | 9¢z－z | $99 \%$ | $\varepsilon$ | 9 | 90 V |
| 08t－99 | 085 | $\varepsilon$ | 99\％－†9 | 0 | † | 9 | 乙 | 99－ヵ | 0 | 9 | 0 | 乙 | zI | 9 | 99z－z | 99\％ | \％ | † | ¢0V |
| ｜47suo，${ }^{\text {a }}$ | ${ }^{\mid s, u n} \Lambda \mid$ | $7_{\text {J }}$ | ${ }^{7}$ ， | $\left.\right\|^{n}$ J $\mid$ | $\left.\right\|_{S} \boldsymbol{J} \mid$ | $\|7\|$ | ${ }^{x_{I N}}$ | $x^{\prime}$ | $\left.\right\|^{2} x \mid$ | $\left.\right\|^{5} \boldsymbol{x} \mid$ | $\left.\right\|^{d} \boldsymbol{\chi} \mid$ | $\left\|{ }_{S} x\right\|$ | $\|x\|$ | ｜$\|c\|^{\text {｜}}$ | ${ }^{\Lambda_{d}}$ | ${ }^{1}$ 万 | $\left.\right\|_{S} \Lambda \mid$ | $\|\Lambda\|$ | ${ }^{\circ} \mathrm{N}$ |
| sapqumu dT |  | sus！！səpyu！ |  |  |  |  | sश！？！poumos |  |  |  |  |  |  | ¢оомұә |  |  |  |  |  |


| sec | nomult | symdem | bmax | pmax | tmax |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\{0,1\}$ | $\{0,1\}$ | $\{0,1\}$ | $\{0,1\}$ | $\{0,1\}$ | $\{0,1\}$ |

Table 6.2: Combination of additional constraints

Downsizing was a really important ambition for us. At first the LP-formulation of the problem was so large, that for the big problem instances (B15-B25 and C15-C25) no solutions could be obtained since the solving time of the LP-relaxation overstepped the time limit. Not until we applied preprocessing and elaborate data transformation did we achieve a breakthrough.

## Reduction of variables

We present the average reduction of variables per series, which arise due to the data transformation and certain preprocessing procedures as introduced in Section 5.3.2 and Section 5.3.3. The first two reduction steps correspond to the transformation of the problem from a naive formulation to our more sophisticated one. In Table 6.1 it can be seen that the merging process for the symmetrical routing presented in the end of Section 3.2.3 resulted for all instances in the greatest reduction. Also the restriction of the linkdesign sets as a result of the nomult constraint had major reduction effects. The greatest impact of the corresponding linkdesign elimination was achieved on the instances of series B where for all edges the set of linkdesigns were reduced from 25 to 5 linkdesigns. The average effect of the latter preprocessing mechanisms on the instances of the series B and C was (except for the nocycle procedure) quite poor. In the instances of these two series there exist only commodities with a hoplimit greater than 2 (except for C20) and only secure nodes (except for C16), such that the corresponding preprocessing procedures (small hoplimits respectively risky node exclusion) hardly influence the number of variables. The only secure commodities mechanism effects only the number of variables for the instances C10 and C12 since they are the only instances in which all commodities have to be routed securely.
For the instances of series A we were most successful. After the transformation and preprocessing, we were only confronted with $34 \%$ of the initial variables. But also the influence on the number of variables for the instances of series B and C was worthwhile since we were only confronted with $46 \%$, or $43 \%$ respectively of the initial variables. Note that the order of the applied variable reducing strategies influences the respective impact. The mechanisms may effect/delete the same variables.

## Reduction of constraints

As in the latter subsection the first two reduction steps correspond to the transformation of the problem from a naive formulation to our more sophisticated one. In Table

Figure 6.1: Reduction of variables. Influence on the average number of variables per series concerning the combination 111111.
6.2 we can see the effects of the exclusion of constraints thru separation (see Section 5.2 .2 ), as well as the further preprocessing on the number of constraints.

Like above the symmetrical routing resulted in the major reduction. As a consequence of the merging mechanism the corresponding inequalities (3.14)-(3.16) became irrelevant, and at the same time the number of hoplimit (3.13) and flow balance inequalities (3.2) were reduced, since their amount depends on the number of commodities. Compared to this major effect the impact of the remaining presented preprocessing mechanisms (presented in Section 5.3.1) seems marginal. Only in the instances of series A there exist nodes with an extremely high number of ports and C10 and C12 are the only instances with only secure commodities. The results of the separation of the hoplimit constraints, thus their omission, were relatively slight, even though there were always excluded as many inequalities as merged commodities were present. Concluding, the number of constraints could be significantly reduced, such that after the application of our mechanisms in general only $10 \%$ of the initial constraints for the instances of the series B and C, or $23 \%$ for the series A respectively, were present.

### 6.2.2 Branching rule comparison

We tested the branching rules introduced in Section 5.1.3 using four problem instances of series A and six of the series B and C. Our implementation ran on all possible combinations $000000-111111$ for each of these 16 instances.
Table 6.3 gives an overview of the results for the different branching rules. We summarized the results of all combinations for each problem instance consisting of the

Figure 6.2: Reduction of constraints. Influence on the average number of constraints per series concerning the combination 111111.
average gap $\oslash$ GAP (the value 100 (upper bound - lower bound)/(lower bound)) and the number of combinations $\# \mathrm{nU}$ where our implementation did not find any feasible integer solution within the time limit of 10 minutes. We applied the heuristic presented in Section 5.4 (with the additional edge exclusion, the highest secure amount as sort criteria and the pricy flow as arc weights) and the trivial separation to include the hoplimit constraints (see Section 5.2.2). Testing the generic most fractional branching rule and the two edge variable brancher developed, most splitted and least splitted, the following observations seem appropriate:

- In the small and medium problem instances (except for A10) our edge variable brancher resulted in better solutions than the generic most fractional branching rule.
- No great difference between edge variable brancher concerning most/least splitted flow can be noticed. In general the least splitted flow strategy tends to be slightly more efficient.
- For the large instances B16, B20, C16 and C20, we could obtain better results applying the most fractional branching rule than by using the other variable selection strategies. It seems that the overhead caused by the more sophisticated edge variable brancher consumes too much time, such that it is impossible to regain the lost time taking advantage of the benefit of these techniques.

|  | most fractional |  | most splitted |  | least splitted |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | $\oslash$ GAP | $\# \mathrm{nU}$ | $\oslash$ GAP | $\# \mathrm{nU}$ | $\oslash$ GAP | \# nU |
| $\begin{gathered} \mathrm{A} 07 \\ 000000-111111 \end{gathered}$ | 7.15 | 0 | 3.07 | 0 | 2.88 | 0 |
| $\begin{gathered} \mathrm{A} 08 \\ 000000-111111 \end{gathered}$ | 19.39 | 0 | 12.41 | 0 | 11.8 | 0 |
| $\begin{gathered} \text { A09 } \\ 000000-111111 \end{gathered}$ | 37.59 | 0 | 25.16 | 0 | 23.91 | 1 |
| $\begin{gathered} \mathrm{A} 10 \\ 000000-111111 \end{gathered}$ | 43.92 | 0 | 32.2 | 7 | 31.0 | 5 |
| $\begin{gathered} \text { B10 } \\ 000000-111111 \end{gathered}$ | 81.19 | 21 | 57.48 | 20 | 61.26 | 19 |
| $\begin{gathered} \text { B11 } \\ 000000-111111 \end{gathered}$ | 102.08 | 18 | 71.14 | 16 | 70.86 | 16 |
| $\begin{gathered} \text { B12 } \\ 000000-111111 \end{gathered}$ | 91.78 | 18 | 80.06 | 14 | 74.23 | 15 |
| $\begin{gathered} \mathrm{B} 15 \\ 000000-111111 \end{gathered}$ | 230.9 | 1 | 239.45 | 0 | 247.3 | 0 |
| $\begin{gathered} \text { B16 } \\ 000000-111111 \end{gathered}$ | 308.63 | 3 | 349.06 | 1 | 351.74 | 1 |
| $\begin{gathered} \text { B20 } \\ 000000-111111 \end{gathered}$ | 287.74 | 9 | 314.92 | 8 | 315.17 | 8 |
| $\begin{gathered} \mathrm{C} 10 \\ 000000-111111 \end{gathered}$ | 50.59 | 26 | 34.07 | 21 | 33.73 | 20 |
| $\begin{gathered} \text { C11 } \\ 000000-111111 \end{gathered}$ | 153.2 | 12 | 109.2 | 12 | 105.95 | 11 |
| $\begin{gathered} \mathrm{C} 12 \\ 000000-111111 \end{gathered}$ | 47.57 | 26 | 40.17 | 24 | 39.42 | 24 |
| $\begin{gathered} \mathrm{C} 15 \\ 000000-111111 \end{gathered}$ | 430.89 | 0 | 506.55 | 1 | 507.82 | 1 |
| $\begin{gathered} \mathrm{C} 16 \\ 000000-111111 \end{gathered}$ | 151.55 | 1 | 166.99 | 5 | 192.69 | 4 |
| $\begin{gathered} \mathrm{C} 20 \\ 000000-111111 \end{gathered}$ | 1067.97 | 5 | 1474.17 | 5 | 1482.06 | 5 |

Table 6.3: Branching rule comparison. Results of running three branching rules from Section 5.1.3 on $16 \cdot 64$ data sets.

### 6.2.3 Hoplimit inclusion

We tested three different strategies taking into account the hoplimit constraints: The two separation approaches presented in Section 5.2 .2 were run against the inclusion of the corresponding inequalities into the LP-formulation. For each series we have chosen three problem instances and ran our implementation on all combinations taking into account the hoplimit constraints. We applied the generic most fractional variable selection strategy and the same heuristic settings as in the branching rule test runs.

As it can be seen in Table 6.4, the inclusion of inequalities into the LP-formulation performed best for the most tested instance. Only for the instance C10 more solutions could be found applying the hoplimit inequality separators. In this instance all commodities have the same, relatively large hoplimit of 5 . For comparison: In the similar instances A10 and B10 the commodities have hoplimits of 3 or 4, respectively. So we propose an application of the separation approaches when large hoplimits are present. For problem instances with small hoplimits (like the ones in the given benchmark), the inclusion of the hoplimit constraints resulted more appropriate. It appears that the advantage of a faster solving of the LP-relaxation is exhausted by the separation routines.

### 6.2.4 Heuristic modes

In Section 5.4 .4 we presented two different possibilities to vary the attitudes of the applied heuristic. Various settings were tested together with the application of the additional edge exclusion. Our implementation was applied on the secure combinations $100000-111111$ for 3 problem instances per problem series.

We only implemented and tested the first two sort criteria (highest amount vs. highest secure amount) for ordering the set of commodities. The third criterion would only have made sense for exploiting the existence of different hoplimits. Since this occurs only in one instance we concentrated on the other two sort criteria.
Two ways of generating edge weights were compared. The current flow which uses the current solution values of the commodity edge flow variables, and the current pricy flow which is additionally influenced by the standardized costs of the edge. We also analyzed the influence of the edge exclusion, which is applied in the case of low port limits for the nodes.
Table 6.5 gives an overview of the results for the different heuristic settings. As in Table 6.3 the results consists of the average gap $\oslash$ GAP and $\# \mathrm{nU}$. We tested all secure combinations applying trivial separation to include the hoplimit constraints and the generic most fractional variable selection strategy.

The following observations seem appropriate:

- The edge exclusion was indispensable since it leads to more feasible integral solutions and, in general, to smaller gaps

|  | Inclusion into formulation |  | Separation trivial |  | Separation using shortest path |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | $\oslash$ GAP | \#nU | $\oslash$ GAP | \#nU | $\oslash$ GAP | \# nU |
| $\begin{gathered} \hline \text { A07 } \\ 000100-111111 \end{gathered}$ | 2.99 | 0 | 3.19 | 0 | 2.96 | 0 |
| $\begin{gathered} \text { A09 } \\ 000100-111111 \end{gathered}$ | 22.71 | 1 | 24.87 | 2 | 28.13 | 2 |
| $\begin{gathered} \text { A10 } \\ 000100-111111 \end{gathered}$ | 27.72 | 4 | 37.26 | 4 | 34.82 | 7 |
| $\begin{gathered} \hline \text { B10 } \\ 000100-111111 \end{gathered}$ | 59.43 | 10 | 66.92 | 12 | 63.15 | 11 |
| $\begin{gathered} \text { B15 } \\ 000100-111111 \end{gathered}$ | 202.17 | 0 | 269.82 | 1 | 268.75 | 1 |
| $\begin{gathered} \text { B20 } \\ 000100-111111 \\ \hline \end{gathered}$ | 296.27 | 4 | 296.67 | 4 | 298.0 | 5 |
| $\begin{gathered} \hline \text { C10 } \\ 000100-111111 \end{gathered}$ | 33.86 | 12 | 33.4 | 10 | 35.09 | 10 |
| $\begin{gathered} \text { C12 } \\ 000100-111111 \end{gathered}$ | 31.72 | 14 | 34.4 | 14 | 34.38 | 15 |
| $\begin{gathered} \text { C16 } \\ 000100-111111 \end{gathered}$ | 165.38 | 4 | 210.13 | 5 | 213.55 | 4 |

Table 6.4: Hoplimit integration comparison. Results of running three different hoplimit constraint integration approaches from Section 5.2.2 on all combinations taking into account the hoplimit constraints, each time for three instances per series.

- The differences between the obtained gaps for the two sort criteria and also the two arc weight functions seemed marginal. Hence we uprated the number of integer solutions found. Under this criterion, the combination of the current flow as arc weight and the highest amount as sort criteria resulted to be the most advisable together with the application of the additional edge exclusion.


### 6.2.5 Results

Our implementation was run on all 27 instances of the ROCOCO benchmark suite. All 64 combinations were tested employing the 10-minutes timelimit. Using the results of the previous test series, we applied the most auspicious heuristic settings (with the additional edge exclusion, the highest amount as sort criteria and the current flow as arc weights) and included the hoplimit inequalities into the LP-formulation. For the small and medium instances we employed the developed edge variable brancher (least splitted) and for the large instances the generic most fractional branching strategy. The following tables Table 6.6 - Table 6.12 show the achieved results. For every problem instance and constraint combination, we present the obtained lower bound $\mathbf{L B}$ and the best feasible integer solutions UB found. We compare these values with the best so far published upper bounds Ilog-UB listed aside.
Considering the results, the following conclusions can be made:

- The given external upper bounds exactly meet our optimal solution in the instances A04-A06 (and in some combinations for the instance A07). In 19 combinations, applied together with the problem instance C16, we achieved better upper bounds (even though the gap is still high). Also in a combination with the instance B15 and in one with B16 we could identify better solutions. In all other cases the external upper bounds lay in the range imposed by our obtained lower and upper bounds.
- The problem instances $\mathrm{B} 15, \mathrm{~B} 16, \mathrm{~B} 20, \mathrm{C} 15, \mathrm{C} 16, \mathrm{C} 20$ and C20 are very large. Since the resulting LP is out of scale, the solution time for the LP-relaxation critically approximates the $10-$ minute time limit. Therefore the remaining time frame is too tight, so that only poor benefit can be drawn from our extensive endeavors. This fact is revealed in the relatively large gaps between the obtained lower and upper bounds.
- For the instances B25 and C25 we obtained no upper bounds since the solvingtime of the LP-relaxation overstepped the time limit, such that the heuristic could not be applied.
- In many combinations for the instances B10 and C10 no integer solution could be found. Since in both instances appear nodes with only 2 inports and 2 outports, our heuristic failed in the combinations considering the pmax constraint.

Table 6.5: Heuristic approach comparison. Results of running 4 heuristic parameter settings with/without an advanced edge

- In the instance C12 we had problems finding integer solutions for the combinations taking into account the nomult constraint. In this instance the mentioned constraint cause a reduction of the linkdesign sets, such that for all edges only two linkdesigns are available. One of these linkdesigns provides only an extremely small amount of capacity (it exists only one commodity out of 132 with a data value less its capacity). The other linkdesign provides much more capacity, but at the same time, it is also much more expensive. The heuristic had problems dealing with this special characteristic. Hence in many combinations no integer solutions could be found for C12.

| Instance: | A04 |  |  | A05 |  |  | A06 |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Comb: | LB | UB | Ilog-UB | LB | UB | Ilog-UB | LB | UB | Ilog-UB |
| 000000 | 22267.0 | 22267 | 22267 | 30744.0 | 30744 | 30744 | 37716.0 | 37716 | 37716 |
| 000001 | 25300.0 | 25300 | 25300 | 32758.0 | 32758 | 32758 | 37716.0 | 37716 | 37716 |
| 000010 | 22267.0 | 22267 | 22267 | 31386.0 | 31386 | 31386 | 37970.0 | 37970 | 37970 |
| 000011 | 25300.0 | 25300 | 25300 | 32758.0 | 32758 | 32758 | 37970.0 | 37970 | 37970 |
| 000100 | 23286.0 | 23286 | 23286 | 31353.0 | 31353 | 31353 | 40789.0 | 40789 | 40789 |
| 000101 | 25300.0 | 25300 | 25300 | 34105.0 | 34105 | 34105 | 40804.0 | 40804 | 40804 |
| 000110 | 25300.0 | 25300 | 25300 | 31386.0 | 31386 | 31386 | 40789.0 | 40789 | 40789 |
| 000111 | 25300.0 | 25300 | 25300 | 39136.0 | 39136 | 39136 | 41349.0 | 41349 | 41349 |
| 001000 | 22267.0 | 22267 | 22267 | 30744.0 | 30744 | 30744 | 37716.0 | 37716 | 37716 |
| 001001 | 25300.0 | 25300 | 25300 | 32758.0 | 32758 | 32758 | 37716.0 | 37716 | 37716 |
| 001010 | 22267.0 | 22267 | 22267 | 31386.0 | 31386 | 31386 | 37970.0 | 37970 | 37970 |
| 001011 | 25300.0 | 25300 | 25300 | 32758.0 | 32758 | 32758 | 37970.0 | 37970 | 37970 |
| 001100 | 23286.0 | 23286 | 23286 | 31353.0 | 31353 | 31353 | 40789.0 | 40789 | 40789 |
| 001101 | 25300.0 | 25300 | 25300 | 34105.0 | 34105 | 34105 | 40804.0 | 40804 | 40804 |
| 001110 | 25300.0 | 25300 | 25300 | 31386.0 | 31386 | 31386 | 40789.0 | 40789 | 40789 |
| 001111 | 25300.0 | 25300 | 25300 | 39136.0 | 39136 | 39136 | 41349.0 | 41349 | 41349 |
| 010000 | 22267.0 | 22267 | 22267 | 30744.0 | 30744 | 30744 | 37716.0 | 37716 | 37716 |
| 010001 | 25300.0 | 25300 | 25300 | 32758.0 | 32758 | 32758 | 37716.0 | 37716 | 37716 |
| 010010 | 22267.0 | 22267 | 22267 | 31386.0 | 31386 | 31386 | 37970.0 | 37970 | 37970 |
| 010011 | 25300.0 | 25300 | 25300 | 32758.0 | 32758 | 32758 | 37970.0 | 37970 | 37970 |
| 010100 | 23286.0 | 23286 | 23286 | 31353.0 | 31353 | 31353 | 40789.0 | 40789 | 40789 |
| 010101 | 25300.0 | 25300 | 25300 | 34105.0 | 34105 | 34105 | 40804.0 | 40804 | 40804 |
| 010110 | 25300.0 | 25300 | 25300 | 31386.0 | 31386 | 31386 | 40789.0 | 40789 | 40789 |
| 010111 | 25300.0 | 25300 | 25300 | 39136.0 | 39136 | 39136 | 41349.0 | 41349 | 41349 |
| 011000 | 22267.0 | 22267 | 22267 | 30744.0 | 30744 | 30744 | 37716.0 | 37716 | 37716 |
| 011001 | 25300.0 | 25300 | 25300 | 32758.0 | 32758 | 32758 | 37716.0 | 37716 | 37716 |
| 011010 | 22267.0 | 22267 | 22267 | 31386.0 | 31386 | 31386 | 37970.0 | 37970 | 37970 |
| 011011 | 25300.0 | 25300 | 25300 | 32758.0 | 32758 | 32758 | 37970.0 | 37970 | 37970 |
| 011100 | 23286.0 | 23286 | 23286 | 31353.0 | 31353 | 31353 | 40789.0 | 40789 | 40789 |
| 011101 | 25300.0 | 25300 | 25300 | 34105.0 | 34105 | 34105 | 40804.0 | 40804 | 40804 |
| 011110 | 25300.0 | 25300 | 25300 | 31386.0 | 31386 | 31386 | 40789.0 | 40789 | 40789 |
| 011111 | 25300.0 | 25300 | 25300 | 39136.0 | 39136 | 39136 | 41349.0 | 41349 | 41349 |
| 100000 | 25114.0 | 25114 | 25114 | 33295.0 | 33295 | 33295 | 38963.0 | 38963 | 38963 |
| 100001 | 26293.0 | 26293 | 26293 | 33751.0 | 33751 | 33751 | 38963.0 | 38963 | 38963 |
| 100010 | 26133.0 | 26133 | 26133 | 33372.0 | 33372 | 33372 | 39343.0 | 39343 | 39343 |
| 100011 | 28988.0 | 28988 | 28988 | 43684.0 | 43684 | 43684 | 39343.0 | 39343 | 39343 |
| 100100 | 26293.0 | 26293 | 26293 | 33372.0 | 33372 | 33372 | 41782.0 | 41782 | 41782 |
| 100101 | 26293.0 | 26293 | 26293 | 35098.0 | 35098 | 35098 | 41797.0 | 41797 | 41797 |
| 100110 | 27087.0 | 27087 | 27087 | 33372.0 | 33372 | 33372 | 41782.0 | 41782 | 41782 |
| 100111 | 28988.0 | 28988 | 28988 | 46300.0 | 46300 | 46300 | 43184.0 | 43184 | 43184 |
| 101000 | 25114.0 | 25114 | 25114 | 33295.0 | 33295 | 33295 | 38963.0 | 38963 | 38963 |
| 101001 | 26293.0 | 26293 | 26293 | 33751.0 | 33751 | 33751 | 38963.0 | 38963 | 38963 |
| 101010 | 26133.0 | 26133 | 26133 | 33372.0 | 33372 | 33372 | 40258.0 | 40258 | 40258 |
| 101011 | 28988.0 | 28988 | 28988 | 43684.0 | 43684 | 43684 | 40258.0 | 40258 | 40258 |
| 101100 | 26293.0 | 26293 | 26293 | 33372.0 | 33372 | 33372 | 41782.0 | 41782 | 41782 |
| 101101 | 26293.0 | 26293 | 26293 | 35098.0 | 35098 | 35098 | 41797.0 | 41797 | 41797 |
| 101110 | 27087.0 | 27087 | 27087 | 33372.0 | 33372 | 33372 | 41782.0 | 41782 | 41782 |
| 101111 | 28988.0 | 28988 | 28988 | 46300.0 | 46300 | 46300 | 43184.0 | 43184 | 43184 |
| 110000 | 35047.0 | 35047 | 35047 | 43114.0 | 43114 | 43114 | 47615.0 | 47615 | 47615 |
| 110001 | 36226.0 | 36226 | 36226 | 43684.0 | 43684 | 43684 | 47615.0 | 47615 | 47615 |
| 110010 | 35047.0 | 35047 | 35047 | 43114.0 | 43114 | 43114 | 48193.0 | 48193 | 48193 |
| 110011 | 36226.0 | 36226 | 36226 | 43684.0 | 43684 | 43684 | 48193.0 | 48193 | 48193 |
| 110100 | 36226.0 | 36226 | 36226 | 43852.0 | 43852 | 43852 | 50980.0 | 50980 | 50980 |
| 110101 | 36226.0 | 36226 | 36226 | 45031.0 | 45031 | 45031 | 50980.0 | 50980 | 50980 |
| 110110 | 36226.0 | 36226 | 36226 | 43852.0 | 43852 | 43852 | 51715.0 | 51715 | 51715 |
| 110111 | 36226.0 | 36226 | 36226 | 50062.0 | 50062 | 50062 | 52563.0 | 52563 | 52563 |
| 111000 | 35047.0 | 35047 | 35047 | 43114.0 | 43114 | 43114 | 47615.0 | 47615 | 47615 |
| 111001 | 36226.0 | 36226 | 36226 | 43684.0 | 43684 | 43684 | 47615.0 | 47615 | 47615 |
| 111010 | 35047.0 | 35047 | 35047 | 43114.0 | 43114 | 43114 | 48193.0 | 48193 | 48193 |
| 111011 | 36226.0 | 36226 | 36226 | 43684.0 | 43684 | 43684 | 48193.0 | 48193 | 48193 |
| 111100 | 36226.0 | 36226 | 36226 | 43852.0 | 43852 | 43852 | 50980.0 | 50980 | 50980 |
| 111101 | 36226.0 | 36226 | 36226 | 45031.0 | 45031 | 45031 | 50980.0 | 50980 | 50980 |
| 111110 | 36226.0 | 36226 | 36226 | 43852.0 | 43852 | 43852 | 51715.0 | 51715 | 51715 |
| 111111 | 36226.0 | 36226 | 36226 | 50062.0 | 50062 | 50062 | 52563.0 | 52563 | 52563 |
| TOTAL | 1782558.0 | 1782558 | 1782558 | 2351778.0 | 2351778 | 2351778 | 2708264.0 | 2708264 | 2708264 |

Table 6.6: Results for instances A04, A05 and A06. The implementation was run on all 64 combinations for every instance. In every instance the optimal solution could be found. These values are identical to the external given upper bounds.

| Instance: | A07 |  |  | A08 |  |  | A09 |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Comb: | LB | UB | Ilog-UB | LB | UB | Ilog-UB | LB | UB | Ilog-UB |
| 000000 | 46429.548 | 49093 | 47728 | 53175.965 | 62713 | 56576 | 65150.363 | 76956 | 70885 |
| 000001 | 46418.496 | 48500 | 47728 | 53652.759 | 62482 | 58195 | 65216.398 | 81746 | 71377 |
| 000010 | 47070.224 | 48643 | 48643 | 53635.708 | 64218 | 58025 | 66471.1 | 96757 | 73839 |
| 000011 | 47092.263 | 48643 | 48643 | 53961.491 | 61608 | 60804 | 66518.871 | 97778 | 75347 |
| 000100 | 46541.233 | 50450 | 48643 | 53371.244 | 60455 | 57185 | 65127.555 | 85318 | 71635 |
| 000101 | 46533.076 | 49795 | 48643 | 53480.096 | 60863 | 58195 | 65084.734 | 75747 | 71635 |
| 000110 | 47149.181 | 48643 | 48643 | 53739.592 | 63933 | 58372 | 66364.732 | 105964 | 75083 |
| 000111 | 47137.797 | 49876 | 48643 | 53882.898 | 63562 | 61608 | 66433.873 | 107199 | 76744 |
| 001000 | 47044.652 | 47728 | 47728 | 53704.389 | 56576 | 56576 | 66112.323 | 75133 | 70885 |
| 001001 | 47116.025 | 47728 | 47728 | 54145.794 | 60907 | 58195 | 66333.272 | 73953 | 71635 |
| 001010 | 47850.902 | 48818 | 48818 | 53973.551 | 60991 | 58025 | 67524.218 | 81917 | 73839 |
| 001011 | 47836.44 | 49376 | 48818 | 54248.265 | 61608 | 61172 | 67758.388 | 82786 | 75347 |
| 001100 | 47084.411 | 49430 | 48818 | 53866.751 | 58430 | 57185 | 66003.021 | 78545 | 71635 |
| 001101 | 47056.703 | 48818 | 48818 | 54066.047 | 61240 | 58195 | 66212.364 | 79336 | 71635 |
| 001110 | 47877.653 | 48818 | 48818 | 54262.525 | 59365 | 58372 | 67531.649 | 85369 | 75083 |
| 001111 | 47821.011 | 48818 | 48818 | 54412.656 | 62086 | 61608 | 67621.808 | 86223 | 76943 |
| 010000 | 46597.772 | 47728 | 47728 | 53741.181 | 60391 | 56576 | 65550.337 | 77679 | 70885 |
| 010001 | 46588.212 | 48500 | 47728 | 54142.405 | 61614 | 58195 | 65457.155 | 78043 | 71377 |
| 010010 | 47245.143 | 50051 | 48643 | 54067.474 | 58372 | 58025 | 66815.065 | 77713 | 73839 |
| 010011 | 47307.23 | 48643 | 48643 | 54411.455 | 67480 | 60804 | 66966.342 | 98507 | 75347 |
| 010100 | 46753.005 | 48643 | 48643 | 53556.313 | 58372 | 57185 | 65212.596 | 88651 | 71635 |
| 010101 | 46533.853 | 49051 | 48643 | 53694.875 | 63916 | 58195 | 65218.981 | 86253 | 71635 |
| 010110 | 47505.646 | 48818 | 48643 | 53834.679 | 60433 | 58372 | 66600.525 | 120266 | 75083 |
| 010111 | 47523.767 | 49898 | 48643 | 54164.867 | 68847 | 61608 | 66806.631 | 114553 | 76744 |
| 011000 | 47440.083 | 47728 | 47728 | 54587.931 | 58930 | 56576 | 66474.797 | 75425 | 70885 |
| 011001 | 47728.0 | 47728 | 47728 | 55115.224 | 60254 | 58195 | 66896.139 | 75046 | 71635 |
| 011010 | 48683.759 | 48818 | 48818 | 55386.571 | 59916 | 58025 | 68053.934 | 87663 | 73839 |
| 011011 | 48262.383 | 48818 | 48818 | 55569.692 | 63783 | 61172 | 68440.204 | 87956 | 75347 |
| 011100 | 47721.319 | 48818 | 48818 | 54979.189 | 57185 | 57185 | 66425.431 | 78476 | 71635 |
| 011101 | 47737.487 | 48818 | 48818 | 55202.016 | 60622 | 58195 | 66727.598 | 76442 | 71635 |
| 011110 | 48551.967 | 48818 | 48818 | 55222.397 | 58372 | 58372 | 68033.191 | 86921 | 75083 |
| 011111 | 48818.0 | 48818 | 48818 | 55954.423 | 62921 | 61608 | 68315.914 | 82224 | 76943 |
| 100000 | 47832.533 | 49811 | 49636 | 56277.72 | 58562 | 58562 | 67830.38 | 77672 | 72871 |
| 100001 | 47818.451 | 50392 | 49636 | 56816.659 | 62786 | 61106 | 68359.203 | 84725 | 73363 |
| 100010 | 49663.613 | 51268 | 50853 | 60949.655 | 68630 | 65504 | 70192.213 | 94073 | 79549 |
| 100011 | 49458.478 | 52399 | 50853 | 61012.843 | 72544 | 69678 | 70381.4 | 102149 | 82197 |
| 100100 | 47868.643 | 51417 | 49636 | 56146.023 | 62991 | 59171 | 67919.422 | 85270 | 73621 |
| 100101 | 47852.739 | 51589 | 49636 | 56580.439 | 62608 | 61106 | 68341.467 | 84481 | 73621 |
| 100110 | 49781.69 | 51146 | 50853 | 60541.807 | 73309 | 65504 | 70157.744 | 93742 | 79838 |
| 100111 | 49818.874 | 51146 | 50853 | 60399.954 | 73290 | 70694 | 70345.077 | 99693 | 82208 |
| 101000 | 48464.167 | 49811 | 49811 | 56556.23 | 58562 | 58562 | 68631.128 | 76509 | 72871 |
| 101001 | 48390.278 | 50450 | 49811 | 57145.426 | 63286 | 61106 | 69394.0 | 78121 | 73363 |
| 101010 | 50284.08 | 51268 | 50853 | 61837.503 | 68630 | 65504 | 71683.936 | 84238 | 79549 |
| 101011 | 50276.277 | 51268 | 50853 | 61675.826 | 72392 | 69678 | 72212.998 | 88623 | 82197 |
| 101100 | 48519.799 | 49811 | 49811 | 56659.907 | 60425 | 59171 | 68754.593 | 79208 | 73621 |
| 101101 | 48429.288 | 50821 | 49811 | 57043.515 | 62608 | 61106 | 69374.211 | 77798 | 73621 |
| 101110 | 50525.882 | 51146 | 50853 | 61419.02 | 71729 | 65504 | 71781.321 | 90438 | 79838 |
| 101111 | 50662.131 | 51146 | 50853 | 61806.09 | 73905 | 70694 | 72147.872 | 93137 | 82208 |
| 110000 | 57095.837 | 59308 | 58101 | 65940.393 | 69349 | 68630 | 77464.489 | 92331 | 81931 |
| 110001 | 57138.809 | 59592 | 58101 | 68378.911 | 76240 | 72686 | 80461.413 | 86514 | 85623 |
| 110010 | 57970.2 | 58413 | 58413 | 66261.017 | 68630 | 68630 | 77756.91 | 86734 | 82351 |
| 110011 | 58380.069 | 59106 | 58413 | 69581.133 | 77495 | 74942 | 80357.568 | 110518 | 91065 |
| 110100 | 57187.121 | 58588 | 58101 | 66213.139 | 73401 | 70245 | 77432.98 | 93597 | 82890 |
| 110101 | 57090.535 | 60497 | 58101 | 68162.978 | 78884 | 74476 | 79932.857 | 98992 | 85623 |
| 110110 | 58588.0 | 58588 | 58588 | 66297.039 | 72424 | 70245 | 77819.968 | 96410 | 83108 |
| 110111 | 58429.554 | 59467 | 59106 | 69237.526 | 78559 | 75303 | 81647.747 | 112118 | 92648 |
| 111000 | 58036.375 | 58311 | 58101 | 66865.026 | 68630 | 68630 | 78187.663 | 82901 | 81931 |
| 111001 | 58043.607 | 58899 | 58101 | 69329.842 | 74150 | 72686 | 81575.848 | 88167 | 85623 |
| 111010 | 58588.0 | 58588 | 58588 | 67068.82 | 68630 | 68630 | 78696.698 | 84342 | 82351 |
| 111011 | 59012.0 | 59012 | 59012 | 70865.48 | 74972 | 74942 | 83860.812 | 97815 | 91065 |
| 111100 | 58013.834 | 58101 | 58101 | 67099.492 | 70747 | 70245 | 78380.696 | 85126 | 82890 |
| 111101 | 58059.166 | 58101 | 58101 | 69439.228 | 76392 | 74476 | 81592.06 | 90522 | 86043 |
| 111110 | 58588.0 | 58588 | 58588 | 67197.01 | 70605 | 70245 | 78757.373 | 85371 | 83108 |
| 111111 | 59106.0 | 59106 | 59106 | 71105.311 | 78047 | 75303 | 84160.308 | 103287 | 93062 |
| TOTAL | 3230031.271 | 3320027 | 3290590 | 3767121.385 | 4205857 | 4049540 | 4525051.864 | 5647167 | 4952942 |

Table 6.7: Results for instances A07, A08 and A09. The implementation was run on all 64 combinations for every instance. An optimal solution could be found for some combinations concerning the instance A07. In the rest the external given upper bounds are situated in our lower bound upper bound range.

| Instance: | A10 |  |  | B10 |  |  | B11 |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Comb: | LB | UB | Ilog-UB | LB | UB | Ilog-UB | LB | UB | Ilog-UB |
| 000000 | 75092.999 | 104341 | 82306 | 14919.499 | 20570 | 19390 | 24732.746 | 41369 | 32246 |
| 000001 | 75095.785 | 109118 | 83112 | 14873.945 | 21265 | 21012 | 24509.971 | 42362 | 39522 |
| 000010 | 75477.077 | 108678 | 82511 | 15011.515 | 22473 | 19390 | 24689.588 | 36331 | 32943 |
| 000011 | 75371.488 | 108824 | 86324 | 14992.953 | 27549 | 21090 | 24584.978 | 44438 | 39522 |
| 000100 | 75079.363 | 100340 | 83136 | 14794.282 | 22914 | 19390 | 24043.417 | 47205 | 32246 |
| 000101 | 75000.507 | 111523 | 84433 | 14715.403 | 26906 | 22163 | 22767.652 | 48952 | 39522 |
| 000110 | 75409.931 | 104397 | 83695 | 14964.038 | - | 19390 | 23550.701 | 39964 | 33169 |
| 000111 | 75327.88 | 143311 | 86324 | 14383.967 | - | 22163 | 24360.083 | 50974 | 39522 |
| 001000 | 75374.001 | 98844 | 82306 | 15359.208 | 23844 | 19413 | 25417.289 | 38473 | 32312 |
| 001001 | 75382.241 | 105641 | 83112 | 15358.72 | 23521 | 21012 | 25480.462 | 43030 | 39682 |
| 001010 | 75698.364 | 106869 | 82511 | 15486.751 | 21850 | 19707 | 25458.984 | 40847 | 32943 |
| 001011 | 75716.752 | 104741 | 86342 | 15471.158 | 24686 | 21265 | 25436.231 | 44026 | 39682 |
| 001100 | 75375.286 | 97800 | 83136 | 15343.621 | 23736 | 19413 | 25412.736 | 46826 | 32312 |
| 001101 | 75216.06 | 99949 | 84433 | 15338.225 | 27716 | 22306 | 25387.551 | 54222 | 39948 |
| 001110 | 75703.552 | 108576 | 83695 | 15466.342 | - | 19787 | 25492.434 | - | 33472 |
| 001111 | 75660.072 | 111791 | 86384 | 15430.624 | 28228 | 22345 | 25479.081 | 52968 | 39948 |
| 010000 | 75430.939 | 103586 | 82306 | 14680.924 | 21614 | 19390 | 25180.483 | 38133 | 32246 |
| 010001 | 75198.703 | 104998 | 83112 | 14830.036 | 23438 | 21090 | 25082.069 | 44806 | 39522 |
| 010010 | 75775.952 | 111766 | 82511 | 14813.237 | 22101 | 19390 | 25162.602 | 35062 | 32943 |
| 010011 | 75714.233 | 108657 | 86324 | 14641.769 | 24209 | 21090 | 25083.586 | 43925 | 39522 |
| 010100 | 75321.055 | 109023 | 83695 | 14243.003 | 23178 | 19390 | 24705.849 | 40859 | 32246 |
| 010101 | 75241.669 | 109876 | 84757 | 14026.965 | 55523 | 22163 | 23170.568 | 51392 | 39522 |
| 010110 | 75715.213 | 118644 | 83695 | 14213.514 | 21465 | 19390 | 24132.872 | 33894 | 33169 |
| 010111 | 75656.973 | 115252 | 86324 | 15030.723 | - | 22163 | 23066.577 | - | 39522 |
| 011000 | 76291.939 | 91436 | 82306 | 15194.494 | 20400 | 19449 | 25631.551 | 38120 | 32312 |
| 011001 | 76279.746 | 99343 | 83112 | 15127.377 | 22932 | 21265 | 25624.628 | 40545 | 39948 |
| 011010 | 76740.61 | 110459 | 82511 | 15250.135 | 20570 | 19707 | 25631.985 | 36078 | 32943 |
| 011011 | 76816.051 | 100743 | 86342 | 15072.923 | 22519 | 21265 | 25541.145 | 42636 | 39948 |
| 011100 | 76076.563 | 99228 | 83695 | 15263.683 | 24701 | 19449 | 25606.287 | 38918 | 32855 |
| 011101 | 76111.249 | 98838 | 84757 | 15307.041 | 23781 | 22580 | 25505.738 | 56546 | 39948 |
| 011110 | 76668.214 | 103596 | 83695 | 15287.353 | 22843 | 19787 | 25501.532 | - | 33472 |
| 011111 | 76695.987 | 100124 | 86384 | 15204.949 | 24501 | 22580 | 25049.222 | - | 39948 |
| 100000 | 78311.77 | 104821 | 85435 | 19392.013 | 31193 | 22938 | 32599.196 | 53182 | 41461 |
| 100001 | 78187.344 | 106554 | 86582 | 19277.975 | 33259 | 23356 | 32522.741 | 57917 | 42444 |
| 100010 | 79731.629 | 112599 | 92428 | 19549.898 | - | 22938 | 32637.312 | - | 41461 |
| 100011 | 79527.273 | 111431 | 96305 | 19545.958 | - | 23411 | 32540.445 | - | 42444 |
| 100100 | 78260.864 | 107693 | 87027 | 19263.262 | 31478 | 23330 | 32573.515 | 55278 | 41627 |
| 100101 | 78218.15 | 110314 | 87905 | 19279.075 | 31891 | 23356 | 32105.132 | 59847 | 42444 |
| 100110 | 79782.111 | 105720 | 93016 | 19533.62 | - | 23330 | 32508.909 | - | 41627 |
| 100111 | 79446.195 | 264894 | 97396 | 19537.043 | - | 23411 | 32530.949 | - | 42444 |
| 101000 | 78604.598 | 101867 | 85435 | 20006.302 | 28163 | 23385 | 33991.69 | 51719 | 41461 |
| 101001 | 78444.024 | 112443 | 87436 | 20020.522 | 29470 | 24113 | 34075.557 | 51699 | 43012 |
| 101010 | 80008.428 | 111415 | 92554 | 20181.676 | - | 23658 | 34060.345 | - | 41461 |
| 101011 | 79872.463 | 118243 | 96305 | 20224.661 | - | 24223 | 34120.289 | - | 43012 |
| 101100 | 78515.695 | 108291 | 87027 | 20020.783 | 28111 | 23813 | 33994.969 | 52699 | 42247 |
| 101101 | 78502.763 | 101896 | 87905 | 19968.225 | 29934 | 24372 | 33845.724 | 50662 | 43580 |
| 101110 | 80005.751 | 102606 | 93016 | 20227.271 | - | 23813 | 33658.191 | - | 43317 |
| 101111 | 79782.374 | 131923 | 97396 | 20242.006 | - | 24412 | 34144.502 | - | 43580 |
| 110000 | 90978.401 | 115175 | 100141 | 19638.346 | 30997 | 22938 | 33012.411 | 49187 | 41461 |
| 110001 | 91151.892 | 135085 | 105144 | 19607.944 | 27351 | 23356 | 32876.626 | 59241 | 42444 |
| 110010 | 91068.875 | 116609 | 100141 | 19689.414 | - | 22938 | 33005.595 | 52338 | 41461 |
| 110011 | 91074.009 | 150149 | 109073 | 19658.136 | - | 23902 | 32928.27 | - | 42444 |
| 110100 | 91072.596 | 124144 | 101289 | 19617.554 | 39510 | 23330 | 32893.258 | 73366 | 41627 |
| 110101 | 91321.228 | 132835 | 105659 | 19623.004 | 26696 | 23356 | 32611.364 | 87950 | 42444 |
| 110110 | 90996.641 | 148596 | 101289 | 19646.317 | - | 23330 | 32900.51 | - | 41627 |
| 110111 | 90679.513 | 165083 | 109754 | 19553.963 | - | 24068 | 32841.095 | - | 42444 |
| 111000 | 91880.662 | 113153 | 100141 | 20314.36 | 28277 | 23658 | 34218.916 | 49771 | 41461 |
| 111001 | 92335.252 | 121701 | 105144 | 20350.128 | 29485 | 24113 | 34098.757 | 49796 | 43012 |
| 111010 | 91893.055 | 107851 | 100141 | 20396.878 | - | 23658 | 34149.769 | 49249 | 41461 |
| 111011 | 92261.176 | 119701 | 109754 | 20387.78 | 27662 | 24223 | 34267.307 | 47650 | 43012 |
| 111100 | 91898.819 | 122699 | 101289 | 20215.142 | 34354 | 23813 | 34245.832 | 53734 | 42247 |
| 111101 | 92215.761 | 127116 | 105659 | 19866.979 | 36800 | 24372 | 33861.359 | 51745 | 43580 |
| 111110 | 91790.349 | 131172 | 101289 | 20333.134 | - | 23813 | 34064.428 | - | 43317 |
| 111111 | 92150.723 | 161077 | 109754 | 20332.922 | - | 24472 | 34199.273 | 51978 | 43580 |
| TOTAL | 5151686.838 | 7415168 | 5808115 | 1115600.668 | - | 1416583 | 1864564.834 | - | 2514301 |

Table 6.8: Results for instances A10, B10 and B11. The implementation was run on all 64 combinations for every instance. In all instances the external given upper bounds are situated in our lower bound upper bound range.

| Instance: | B12 |  |  | B15 |  |  | B16 |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Comb: | LB | UB | Ilog-UB | LB | UB | Ilog-UB | LB | UB | Ilog-UB |
| 000000 | 19995.476 | 29168 | 24681 | 12313.854 | 33248 | 26126 | 11206.359 | 65402 | 28335 |
| 000001 | 19799.097 | 39219 | 36644 | 12296.277 | 39617 | 33966 | 11095.792 | 65427 | 30193 |
| 000010 | 19847.58 | 28493 | 24878 | 12313.486 | 33807 | 26337 | 11194.959 | 65898 | 28335 |
| 000011 | 19877.379 | - | 36708 | 12171.657 | 52466 | 33966 | 11055.97 | 71155 | 31962 |
| 000100 | 19688.324 | 30955 | 24970 | 12310.746 | 55484 | 26939 | 11207.665 | 60574 | 28496 |
| 000101 | 20092.506 | 39506 | 36936 | 12296.277 | 57343 | 35775 | 11129.273 | 66357 | 35682 |
| 000110 | 19801.004 | 37504 | 24970 | 12317.144 | 64885 | 27800 | 11194.959 | 70344 | 30193 |
| 000111 | 20276.998 | 40753 | 36936 | 12152.607 | 67451 | 35775 | 11051.98 | 68607 | 35682 |
| 001000 | 20424.501 | 27958 | 25036 | 12874.965 | 28542 | 26126 | 11579.317 | 37261 | 28335 |
| 001001 | 21149.994 | 39959 | 37094 | 12806.18 | 39172 | 33966 | 11565.579 | 46951 | 31962 |
| 001010 | 20431.125 | 26362 | 25036 | 12873.46 | 38864 | 26978 | 11571.995 | 47303 | 28335 |
| 001011 | 21128.72 | 38241 | 37094 | 12703.729 | 37151 | 33966 | 11551.524 | 59362 | 31962 |
| 001100 | 20402.485 | 33134 | 25036 | 13372.93 | 55070 | 28866 | 11575.226 | 56416 | 29308 |
| 001101 | 21140.131 | 41023 | 37212 | 12710.553 | 51162 | 36003 | 11571.995 | 58458 | 35682 |
| 001110 | 20440.497 | 28415 | 25036 | 12851.572 | 56106 | 29259 | 11573.624 | 42065 | 30193 |
| 001111 | 21156.741 | 41129 | 37212 | 12720.574 | 56112 | 36003 | 11460.492 | 67648 | 35682 |
| 010000 | 20091.456 | 28665 | 24731 | 12407.929 | 51432 | 26126 | 11564.828 | 70250 | 28586 |
| 010001 | 20427.207 | 38859 | 36644 | 12324.267 | 51944 | 33966 | 11129.273 | 73328 | 31872 |
| 010010 | 20250.174 | 27961 | 24878 | 12510.544 | 28341 | 27518 | 11564.828 | 63202 | 28656 |
| 010011 | 20200.458 | 39106 | 36708 | 12205.793 | 54176 | 33966 | 11181.508 | 65898 | 31962 |
| 010100 | 20159.993 | 53168 | 24970 | 12386.631 | 64687 | 27800 | 11514.623 | 70250 | 29483 |
| 010101 | 20216.426 | 39047 | 37212 | 12324.267 | 60602 | 36522 | 11099.234 | 76718 | 36074 |
| 010110 | 20128.987 | - | 24970 | 12407.929 | 62185 | 27800 | 11564.828 | 71172 | 31428 |
| 010111 | 20054.891 | 38201 | 37212 | 12324.267 | 67290 | 36664 | 11129.273 | 70828 | 36074 |
| 011000 | 20620.755 | 28810 | 25036 | 15171.659 | 28644 | 26126 | 12574.999 | 30891 | 28586 |
| 011001 | 21125.664 | 39471 | 37094 | 14844.029 | 35850 | 33966 | 12497.765 | 31701 | 31962 |
| 011010 | 20620.784 | 26527 | 25036 | 15189.089 | 27158 | 28809 | 12536.092 | 30169 | 28656 |
| 011011 | 21097.479 | 40882 | 37094 | 13324.785 | 35593 | 33966 | 12472.294 | 32832 | 31962 |
| 011100 | 20558.417 | 36023 | 25036 | 13118.238 | 39071 | 28866 | 12233.743 | 49494 | 31212 |
| 011101 | 20951.661 | 39711 | 37212 | 13103.235 | 66757 | 36701 | 11935.82 | 65650 | 36074 |
| 011110 | 20583.003 | - | 25036 | 14818.459 | 55859 | 32458 | 11935.82 | 71179 | 31428 |
| 011111 | 21197.704 | 37892 | 37212 | 14354.903 | 54236 | 36701 | 12233.743 | 71179 | 36074 |
| 100000 | 26778.641 | 46321 | 34697 | 18618.146 | 52893 | 32797 | 16601.026 | 69422 | 34401 |
| 100001 | 26960.427 | 49320 | 37438 | 18587.063 | 51342 | 35823 | 16326.25 | 67473 | 35528 |
| 100010 | 27096.288 | - | 34729 | 18622.79 | 64303 | 33067 | 16588.115 | 69316 | 34401 |
| 100011 | 26746.013 | - | 37815 | 18586.142 | 65263 | 36110 | 16326.25 | 68150 | 35528 |
| 100100 | 26746.857 | 46944 | 34697 | 18691.753 | 46039 | 32797 | 16597.637 | 69422 | 35889 |
| 100101 | 26909.985 | 48124 | 38017 | 18429.519 | 58051 | 36522 | 16468.466 | 69290 | 36498 |
| 100110 | 26826.713 | - | 35202 | 18618.146 | 65964 | 33067 | 16579.91 | 68957 | 36544 |
| 100111 | 26085.096 | 54223 | 38168 | 18581.498 | 61511 | 36664 | 16509.74 | - | 36544 |
| 101000 | 29089.965 | 40306 | 35202 | 19606.127 | 45747 | 34387 | 17671.42 | 47927 | 35207 |
| 101001 | 29264.052 | 42711 | 38017 | 19600.994 | 43857 | 35823 | 16090.77 | 45504 | 35528 |
| 101010 | 29053.473 |  | 35202 | 19671.425 | 57350 | 34662 | 16154.413 | 58470 | 35207 |
| 101011 | 29309.091 | 47929 | 38147 | 19599.59 | 54721 | 36247 | 16154.413 | 72915 | 35528 |
| 101100 | 29067.414 | 41954 | 35202 | 19603.244 | 48095 | 34387 | 16090.77 | 55250 | 35972 |
| 101101 | 29268.588 | 41359 | 38017 | 19543.275 | 46714 | 37403 | 17071.002 | 50691 | 36498 |
| 101110 | 29007.843 | 49905 | 35202 | 19606.127 | 61398 | 34927 | 16230.814 | 70264 | 36544 |
| 101111 | 29106.371 | 42871 | 38168 | 19590.656 | 60299 | 37403 | 16154.413 | 71580 | 36544 |
| 110000 | 27011.974 | 67449 | 34729 | 18714.368 | 69370 | 33023 | 16739.927 | 79436 | 34401 |
| 110001 | 26869.815 | 73124 | 37815 | 18752.797 | 75042 | 35823 | 16518.44 | 75790 | 35528 |
| 110010 | 26881.4 | - | 34729 | 18787.694 | 58272 | 33067 | 16739.927 | 66040 | 34401 |
| 110011 | 26591.289 | 45477 | 37815 | 18737.293 | 65623 | 36110 | 16549.812 | 68439 | 35528 |
| 110100 | 26689.91 | 69136 | 34868 | 18786.112 | 69202 | 33023 | 16661.023 | 79436 | 35889 |
| 110101 | 26550.489 | 72144 | 38911 | 18733.822 | 71098 | 36522 | 16488.629 | 78772 | 36544 |
| 110110 | 26465.435 | - | 35202 | 18758.3 | 65964 | 33067 | 16694.255 | 68957 | 36544 |
| 110111 | 26614.56 | - | 38911 | 18775.073 | 67785 | 36664 | 16549.812 | 68249 | 36544 |
| 111000 | 29203.323 | 68660 | 35202 | 19943.633 | 38168 | 34642 | 17834.97 | 40366 | 35510 |
| 111001 | 29381.657 | 41855 | 38147 | 20815.554 | 62244 | 35823 | 18025.851 | 37408 | 35528 |
| 111010 | 29223.01 | - | 35202 | 21238.38 | 38096 | 34662 | 17834.97 | 59387 | 35510 |
| 111011 | 29355.476 | - | 38147 | 20672.169 | 37742 | 36247 | 17245.072 | 39236 | 35528 |
| 111100 | 29217.623 | 64716 | 35202 | 19872.551 | 64149 | 34927 | 17890.953 | 64221 | 36544 |
| 111101 | 29439.263 | 43302 | 39720 | 20891.536 | 69553 | 38007 | 17536.851 | 61643 | 36544 |
| 111110 | 29187.214 | - | 35202 | 21179.091 | 58428 | 34927 | 17834.97 | 67671 | 36544 |
| 111111 | 29315.671 | - | 40441 | 20954.028 | 62040 | 38007 | 17713.758 | 71000 | 36544 |
| TOTAL | 1549252.543 | - | 2165723 | 1035070.931 | 3406628 | 2132433 | 909230.009 | - | 2156418 |

Table 6.9: Results for instances $\mathrm{B} 12, \mathrm{~B} 15$ and B 16 . The implementation was run on all 64 combinations for every instance.In the instance B15 we achieved a better upper bound concerning the combination 011010. In the instance B16 we could identify a better upper bound for the combination 011001. In the other instances and combinations, the external given upper bounds are situated in our lower bound upper bound range.

| Instance: | B20 |  |  | B25 |  |  | C10 |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Comb: | LB | UB | Ilog-UB | LB | UB | Ilog-UB | LB | UB | Ilog-UB |
| 000000 | 21245.583 | 106600 | 47180 | 24865.103 | - | 74051 | 10398.147 | 11675 | 11472 |
| 000001 | 21245.583 | 106600 | 47180 | 24865.103 | - | 74051 | 10982.051 | 14485 | 13388 |
| 000010 | 21245.583 | 120541 | 47465 | 24865.103 | - | 74051 | 12376.412 | 16334 | 16194 |
| 000011 | 21245.583 | 118326 | 47639 | 24865.103 | - | 74051 | 13010.948 | - | 18448 |
| 000100 | 21245.583 | 106600 | 49770 | 24865.103 | - | 80281 | 10293.747 | 12270 | 11472 |
| 000101 | 21245.583 | 106600 | 51625 | 24865.103 | - | 85721 | 10874.154 | 15263 | 13388 |
| 000110 | 21245.583 | - | 49770 | 24865.103 | - | 90916 | 12017.998 | - | 16194 |
| 000111 | 21245.583 | - | 51625 | 24865.103 | - | 90916 | 12335.761 | - | 18925 |
| 001000 | 21855.158 | 103948 | 47180 | 25082.308 | - | 74051 | 10428.978 | 11912 | 11465 |
| 001001 | 21855.158 | 103948 | 47180 | 25082.308 | - | 74051 | 11200.251 | 14098 | 13388 |
| 001010 | 21855.158 | 115086 | 47639 | 25082.308 | - | 74051 | 12909.566 | 16587 | 16194 |
| 001011 | 21855.158 | 113348 | 47639 | 25082.308 | - | 74051 | 13825.834 | - | 18925 |
| 001100 | 21855.158 | 103441 | 50317 | 25082.308 | - | 87244 | 10447.256 | 11645 | 11483 |
| 001101 | 21855.158 | 103441 | 53374 | 25082.308 | - | 87244 | 11263.987 | 15059 | 13388 |
| 001110 | 21855.158 | 110078 | 50514 | 25082.308 | - | 92278 | 12940.713 | 20449 | 16194 |
| 001111 | 21855.158 | 110078 | 53602 | 25082.308 | - | 92278 | 13846.861 | - | 21560 |
| 010000 | 21245.583 | 111184 | 47180 | 24865.103 | - | 74051 | 11465.957 | 17373 | 14141 |
| 010001 | 21245.583 | 111184 | 47180 | 24865.103 | - | 74051 | 12096.498 | 19228 | 16043 |
| 010010 | 21245.583 | 117225 | 47465 | 24865.103 | - | 74051 | 13210.033 | - | 16194 |
| 010011 | 21245.583 | 116695 | 47639 | 24865.103 | - | 74051 | 13970.733 | - | 18448 |
| 010100 | 21245.583 | 111184 | 49770 | 24865.103 | - | 80281 | 11501.616 | 14973 | 14141 |
| 010101 | 21245.583 | 111184 | 51625 | 24865.103 | - | 85721 | 12013.898 | 17953 | 16043 |
| 010110 | 21245.583 | - | 49770 | 24865.103 | - | 92124 | 13341.438 | - | 16194 |
| 010111 | 21245.583 | - | 51625 | 24865.103 | - | 92124 | 14031.767 | - | 18925 |
| 011000 | 22057.543 | 120466 | 47180 | 25148.987 | - | 74051 | 11706.79 | 15139 | 14141 |
| 011001 | 21855.158 | 120466 | 47180 | 25148.987 | - | 74051 | 12510.421 | 17212 | 16085 |
| 011010 | 22081.431 | 115086 | 47639 | 25107.781 | - | 74051 | 14193.801 | 17314 | 16194 |
| 011011 | 21855.158 | 113348 | 47639 | 25107.781 | - | 74051 | 14466.228 | - | 18925 |
| 011100 | 21937.53 | 119959 | 50376 | 25107.781 | - | 87244 | 11722.196 | 15264 | 14141 |
| 011101 | 21855.158 | 120466 | 53374 | 25148.987 | - | 87244 | 12527.024 | 17491 | 16164 |
| 011110 | 22057.543 | 110078 | 51125 | 25107.781 | - | 93840 | 14325.729 | 19339 | 16194 |
| 011111 | 21855.158 | 110078 | 53602 | 25107.781 | - | 93840 | 14602.132 | - | 21560 |
| 100000 | 35631.138 | 109583 | 67370 | 41017.034 | - | 90297 | 12227.56 | 16647 | 15157 |
| 100001 | 35631.138 | 109132 | 67370 | 41017.034 | - | 90297 | 12185.061 | 17926 | 16664 |
| 100010 | 35631.138 | - | 68233 | 41017.034 | - | 92077 | 12666.657 | 17894 | 16194 |
| 100011 | 35631.138 | - | 68233 | 41017.034 | - | 92077 | 13283.854 | - | 18448 |
| 100100 | 35631.138 | 109132 | 67574 | 41017.034 | - | 98439 | 12084.579 | 15494 | 15157 |
| 100101 | 35631.138 | 109132 | 67574 | 41017.034 | - | 103893 | 11651.743 | 20216 | 16664 |
| 100110 | 35631.138 | 116721 | 71881 | 41017.034 | - | 98439 | 12561.78 | - | 16194 |
| 100111 | 35631.138 | 116721 | 72884 | 41017.034 | - | 105268 | 12950.89 | - | 18925 |
| 101000 | 36993.179 | 103056 | 67574 | 41933.815 | - | 90297 | 12549.585 | 16396 | 15157 |
| 101001 | 36993.179 | 103056 | 67574 | 41933.815 | - | 90297 | 12613.21 | 18304 | 16664 |
| 101010 | 36993.179 | 113768 | 69119 | 41933.815 | - | 92077 | 13452.852 | 16628 | 16194 |
| 101011 | 36993.179 | 113768 | 69119 | 41933.815 | - | 92077 | 14037.853 | 20114 | 18925 |
| 101100 | 36993.179 | 103056 | 67574 | 41933.815 | - | 98439 | 12590.487 | 16322 | 15157 |
| 101101 | 36993.179 | 103056 | 67574 | 41933.815 | - | 103893 | 12677.734 | 18171 | 17015 |
| 101110 | 36993.179 | 110436 | 73982 | 41933.815 | - | 98439 | 13432.278 | 16556 | 16194 |
| 101111 | 36993.179 | 110436 | 74372 | 41933.815 | - | 106396 | 13868.179 | - | 21560 |
| 110000 | 35631.138 | 120988 | 67370 | 41017.034 | - | 90297 | 12276.721 | 15919 | 15157 |
| 110001 | 35631.138 | 120988 | 67370 | 41017.034 | - | 90297 | 12721.835 | 18952 | 16664 |
| 110010 | 35631.138 | - | 68233 | 41017.034 | - | 92077 | 13483.561 | 16194 | 16194 |
| 110011 | 35631.138 | - | 68233 | 41017.034 | - | 92077 | 14411.624 | - | 18448 |
| 110100 | 35631.138 | 120988 | 67574 | 41017.034 | - | 98439 | 12275.212 | 17507 | 15157 |
| 110101 | 35631.138 | 120988 | 67574 | 41017.034 | - | 103893 | 12779.126 | 18420 | 16664 |
| 110110 | 35631.138 | 116721 | 71881 | 41017.034 | - | 98439 | 13652.857 | - | 16194 |
| 110111 | 35631.138 | 116721 | 72884 | 41017.034 | - | 105268 | 13879.94 | - | 18925 |
| 111000 | 36993.179 | 119101 | 67574 | 41962.425 | - | 90297 | 12718.958 | 15919 | 15157 |
| 111001 | 36993.179 | 119101 | 67574 | 41962.425 | - | 90297 | 13479.674 | 18956 | 16664 |
| 111010 | 36993.179 | 113768 | 69119 | 41962.425 | - | 92077 | 14500.453 | 16194 | 16194 |
| 111011 | 36993.179 | 113768 | 69119 | 41962.425 | - | 92077 | 15634.18 | - | 18925 |
| 111100 | 36993.179 | 119101 | 67574 | 41962.425 | - | 98439 | 12779.648 | 15499 | 15157 |
| 111101 | 36993.179 | 119101 | 67574 | 41962.425 | - | 103893 | 13418.02 | 17531 | 17015 |
| 111110 | 36993.179 | 110436 | 73982 | 41962.425 | - | 98439 | 14682.532 | 16536 | 16194 |
| 111111 | 36993.179 | 110436 | 76937 | 41962.425 | - | 106396 | 15779.27 | - | 21560 |
| TOTAL | 1852314.343 | - | 3798647 | 2126928.442 | - | 5689516 | 820146.838 | - | 1046149 |

Table 6.10: Results for instances B20, B25 and C10. The implementation was run on all 64 combinations for every instance. In all instances the external given upper bounds are situated in our lower bound upper bound range. In the instance B25 we obtained no upper bounds since the solving-time of the LP-relaxation overstepped our time limit.

| Instance: | C11 |  |  | C12 |  |  | C15 |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Comb: | LB | UB | Ilog-UB | LB | UB | Ilog-UB | LB | UB | Ilog-UB |
| 000000 | 15766.287 | 23913 | 20889 | 30711.879 | 43582 | 35334 | 17599.665 | 56850 | 39347 |
| 000001 | 15649.138 | 28634 | 25641 | 30729.697 | 42964 | 35373 | 17428.125 | 111678 | 50491 |
| 000010 | 15721.777 | 24848 | 21428 | 30740.302 | 52806 | 35480 | 17599.713 | 114951 | 39347 |
| 000011 | 15609.039 | 31026 | 25641 | 30730.012 | - | 35480 | 17451.851 | 139597 | 50491 |
| 000100 | 14600.493 | 24395 | 20889 | 30701.603 | 41210 | 35334 | 17597.224 | 60059 | 42086 |
| 000101 | 14127.848 | 27850 | 25641 | 30526.886 | 44970 | 35373 | 17533.874 | 182879 | 54042 |
| 000110 | 15136.679 | 25435 | 21685 | 30706.013 | - | 35480 | 17589.153 | 158968 | 42378 |
| 000111 | 14548.701 | 34504 | 25641 | 30679.124 | - | 35480 | 17430.969 | 152975 | 54654 |
| 001000 | 16035.232 | 22655 | 20889 | 31110.932 | 41250 | 35334 | 18361.316 | 61884 | 39347 |
| 001001 | 16402.051 | 29565 | 25991 | 31111.363 | 40778 | 35835 | 18212.493 | 99194 | 51108 |
| 001010 | 16271.685 | 21727 | 21428 | 31125.565 | 49280 | 35880 | 18401.786 | 57528 | 39347 |
| 001011 | 16157.905 | 32383 | 26085 | 31129.743 | 49267 | 35880 | 18361.029 | 128510 | 51108 |
| 001100 | 16261.012 | 28343 | 20889 | 31108.933 | 41280 | 35334 | 18361.316 | 113151 | 42631 |
| 001101 | 16280.557 | 30011 | 26341 | 31101.255 | 42201 | 35880 | 18340.855 | 137915 | 54042 |
| 001110 | 16303.241 | 25338 | 21685 | 31111.35 | 45008 | 35880 | 18380.286 | 144495 | 43328 |
| 001111 | 16256.317 | 32831 | 26341 | 31100.704 | - | 35880 | 18251.522 | 142216 | 55796 |
| 010000 | 15852.657 | 24059 | 20889 | 30947.689 | - | 35480 | 17812.716 | 47436 | 39347 |
| 010001 | 15643.608 | 29227 | 25641 | 30909.855 | - | 35480 | 17595.214 | 73415 | 50491 |
| 010010 | 15651.183 | 23002 | 21428 | 30963.198 | 42516 | 35480 | 17783.952 | 48450 | 39347 |
| 010011 | 15671.104 | 30278 | 25641 | 30892.693 | 41769 | 35480 | 17579.968 | 72975 | 50491 |
| 010100 | 15179.538 | 23512 | 20889 | 30676.854 | - | 35480 | 17639.586 | 273805 | 42086 |
| 010101 | 14463.395 | 29432 | 25641 | 30469.436 | - | 35480 | 17619.336 | 71647 | 54042 |
| 010110 | 14930.167 | 25630 | 21685 | 30717.193 | 39046 | 35480 | 17639.586 | 124847 | 42378 |
| 010111 | 14363.316 | 32445 | 25641 | 30396.015 | 40454 | 35480 | 17671.723 | 143608 | 54654 |
| 011000 | 16104.933 | 23136 | 20889 | 31319.902 | - | 35835 | 21401.021 | 46493 | 39347 |
| 011001 | 16328.097 | 30665 | 25991 | 31298.706 | - | 35835 | 20635.778 | 68654 | 51108 |
| 011010 | 16121.225 | 22425 | 21428 | 31283.627 | 40985 | 35909 | 21293.72 | 47692 | 39347 |
| 011011 | 16303.618 | 29376 | 26085 | 31307.243 | 39302 | 36742 | 20491.941 | 71331 | 51108 |
| 011100 | 16069.693 | 25336 | 20889 | 31314.226 | - | 35909 | 20987.133 | 60603 | 42631 |
| 011101 | 16403.271 | 31925 | 26528 | 31290.481 | - | 36238 | 20251.616 | 154987 | 54042 |
| 011110 | 15994.096 | 28468 | 21685 | 31299.84 | 40649 | 35909 | 20367.978 | 98284 | 43679 |
| 011111 | 16377.795 | 31703 | 26528 | 31293.523 | 42292 | 36857 | 20373.53 | 106285 | 55796 |
| 100000 | 21524.023 | 41945 | 29468 | 30908.82 | 43756 | 35512 | 30217.923 | 97310 | 54638 |
| 100001 | 21065.776 | 42693 | 29530 | 30914.458 | 44465 | 35880 | 29991.403 | 169694 | 60531 |
| 100010 | 21509.88 | - | 29577 | 30915.447 | - | 35880 | 30262.853 | 144023 | 56688 |
| 100011 | 21346.636 | - | 29577 | 30912.288 | - | 35880 | 30014.558 | 143231 | 60531 |
| 100100 | 21683.772 | 62490 | 29530 | 30745.884 | 43085 | 35593 | 30196.512 | 174186 | 54638 |
| 100101 | 21372.017 | 53039 | 29530 | 30719.075 | 47248 | 35880 | 29767.427 | 189087 | 60531 |
| 100110 | 21583.586 | - | 29577 | 30740.461 | - | 35880 | 30261.988 | 154356 | 56688 |
| 100111 | 21060.62 | - | 29577 | 30719.476 | - | 35880 | 30030.079 | 147944 | 60531 |
| 101000 | 22997.951 | 42772 | 29468 | 31260.105 | 42313 | 35880 | 32086.238 | 115732 | 55022 |
| 101001 | 22735.643 | 42662 | 29577 | 31265.268 | 40874 | 35880 | 31751.85 | 158515 | 60531 |
| 101010 | 23146.494 | - | 29577 | 31252.294 | 52948 | 35880 | 31765.444 | 121698 | 57090 |
| 101011 | 23141.455 | - | 29577 | 31249.265 | 49448 | 35880 | 32094.208 | 126964 | 60531 |
| 101100 | 23016.172 | 43524 | 29577 | 31239.125 | 42191 | 35880 | 31751.882 | 178806 | 55022 |
| 101101 | 23233.525 | 43932 | 29577 | 31235.081 | 44117 | 35880 | 31839.74 | 157858 | 60531 |
| 101110 | 23164.923 | - | 29577 | 31225.315 | 42380 | 35880 | 32186.267 | 146161 | 57090 |
| 101111 | 23105.816 | 36884 | 29577 | 31251.797 | 48464 | 35880 | 32164.584 | 140618 | 60531 |
| 110000 | 21711.565 | 101300 | 29577 | 30954.871 | - | 35909 | 30309.81 | 93993 | 54638 |
| 110001 | 21131.305 | 44665 | 29577 | 30913.109 | - | 35933 | 30309.81 | 94705 | 60531 |
| 110010 | 21385.067 | 39383 | 29577 | 30947.249 | 39059 | 35909 | 30309.81 | 88716 | 56688 |
| 110011 | 20991.058 | 49459 | 29577 | 30892.06 | 42653 | 36604 | 30309.81 | 94705 | 60531 |
| 110100 | 21477.922 | 45067 | 29577 | 30663.005 | - | 35909 | 30248.063 | 308164 | 54638 |
| 110101 | 20751.852 | 114975 | 29577 | 30423.093 | - | 36238 | 30210.972 | 308461 | 60531 |
| 110110 | 21553.612 | - | 29577 | 30671.944 | 41202 | 35909 | 30244.933 | 154356 | 56688 |
| 110111 | 21125.161 | 47113 | 29577 | 30408.018 | 41264 | 36857 | 30248.063 | 154356 | 60531 |
| 111000 | 23087.41 | 41610 | 29577 | 31319.936 | - | 35909 | 34029.275 | 81210 | 55022 |
| 111001 | 22733.92 | 99347 | 29577 | 31347.77 | - | 35933 | 33414.061 | 96785 | 60531 |
| 111010 | 23063.378 | 37683 | 29577 | 31318.681 | 40147 | 35909 | 34029.275 | 86651 | 57480 |
| 111011 | 22947.672 | - | 29577 | 31333.254 | 40350 | 36857 | 33539.322 | 82046 | 60531 |
| 111100 | 23182.129 | 92149 | 29577 | 31318.329 | - | 35909 | 34002.6 | 271354 | 55022 |
| 111101 | 23156.913 | 38324 | 29577 | 31274.67 | - | 36238 | 33414.061 | 291057 | 60531 |
| 111110 | 23023.702 | - | 29577 | 31294.798 | 42175 | 35909 | 33920.081 | 162267 | 57480 |
| 111111 | 23030.076 | - | 29577 | 31287.386 | 40082 | 36857 | 33367.292 | 142846 | 60531 |
| TOTAL | 1212626.689 | - | 1700687 | 1983728.174 | - | 2294525 | 1602336.169 | 8251217 | 3362465 |

Table 6.11: Results for instances C11, C12 and C15. The implementation was run on all 64 combinations for every instance. In all instances the external given upper bounds are situated in our lower bound upper bound range.

| Instance: | C16 |  |  | C20 |  |  | C25 |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Comb: | LB | UB | Ilog-UB | LB | UB | Ilog-UB | LB | UB | Ilog-UB |
| 000000 | 12195.029 | 28313 | 26289 | 30264.927 | 650420 | 68831 | 23247.521 | - | 70994 |
| 000001 | 12170.665 | 25744 | 26289 | 30264.927 | 669547 | 69564 | 23247.521 | - | 70994 |
| 000010 | 12176.177 | 40550 | 26482 | 30264.927 | 238627 | 68831 | 23247.521 | - | 72271 |
| 000011 | 12174.139 | 40550 | 26754 | 30264.927 | 234785 | 70526 | 23247.521 | - | 72271 |
| 000100 | 12175.682 | 28117 | 26289 | 30264.927 | 660834 | 71705 | 23247.521 | - | 81267 |
| 000101 | 12170.733 | 25744 | 26289 | 30264.927 | 660834 | 76999 | 23247.521 | - | 85278 |
| 000110 | 12174.776 | 49162 | 26482 | 30264.927 | 241290 | 73658 | 23247.521 | - | 87710 |
| 000111 | 12174.776 | 47913 | 26754 | 30264.927 | 243122 | 78376 | 23247.521 | - | 88232 |
| 001000 | 13182.539 | 25449 | 26289 | 31332.591 | 118870 | 68831 | 23451.233 | - | 73161 |
| 001001 | 13015.877 | 25918 | 26289 | 31315.902 | 234947 | 70526 | 23451.233 | - | 73508 |
| 001010 | 13174.672 | 28874 | 26545 | 31603.236 | 238833 | 68831 | 23451.233 | - | 73508 |
| 001011 | 13003.574 | 30552 | 26769 | 31603.236 | 241513 | 70526 | 23451.233 | - | 73508 |
| 001100 | 13187.133 | 25449 | 26289 | 31499.549 | 148581 | 71968 | 23451.233 | - | 86598 |
| 001101 | 13015.877 | 25918 | 26289 | 31315.902 | 299554 | 76999 | 23451.233 | - | 95008 |
| 001110 | 13173.811 | 26721 | 26566 | 31603.236 | 243609 | 73658 | 23451.233 | - | 96768 |
| 001111 | 12994.547 | 30988 | 26769 | 31603.236 | 238833 | 78851 | 23451.233 | - | 96768 |
| 010000 | 12231.569 | 25622 | 27054 | 30465.041 | 700156 | 68831 | 23268.697 | - | 73161 |
| 010001 | 12175.293 | 35317 | 27191 | 30465.041 | 700156 | 69564 | 23268.697 | - | 75486 |
| 010010 | 12231.569 | 25016 | 27243 | 30465.041 | 234635 | 68831 | 23247.521 | - | 73990 |
| 010011 | 12171.137 | 27019 | 27267 | 30465.041 | 236990 | 70593 | 23247.521 | - | 75486 |
| 010100 | 12171.137 | 62524 | 27098 | 30465.041 | 700156 | 71722 | 23247.521 | - | 83268 |
| 010101 | 12171.137 | 34746 | 27265 | 30465.041 | 700156 | 77338 | 23268.697 | - | 88016 |
| 010110 | 12171.137 | 31010 | 27258 | 30441.054 | 244260 | 73900 | 23247.521 | - | 87710 |
| 010111 | 12171.137 | 31038 | 27267 | 30465.041 | 248278 | 78851 | 23247.521 | - | 88232 |
| 011000 | 13223.411 | 28028 | 27081 | 31587.979 | 587420 | 68831 | 23552.609 | - | 73161 |
| 011001 | 13114.439 | 27532 | 27191 | 31587.979 | 89320 | 70593 | 23509.59 | - | 75605 |
| 011010 | 13240.523 | 24775 | 27243 | 31587.979 | 95617 | 68831 | 23473.686 | - | 75332 |
| 011011 | 13071.18 | 28852 | 27267 | 31587.979 | 129679 | 70593 | 23473.686 | - | 75605 |
| 011100 | 13197.372 | - | 27098 | 31587.979 | 307957 | 72764 | 23509.59 | - | 88688 |
| 011101 | 13092.1 | 28485 | 27265 | 31587.979 | 253468 | 78851 | 23509.59 | - | 95008 |
| 011110 | 13213.766 | 30164 | 27258 | 31587.979 | 243609 | 74253 | 23473.686 | - | 96768 |
| 011111 | 13059.283 | 30894 | 27267 | 31587.979 | 115818 | 78851 | 23473.686 | - | 96768 |
| 100000 | 12588.934 | 30047 | 26445 | 30986.232 | 665844 | 95842 | 43438.097 | - | 100569 |
| 100001 | 12527.841 | 26483 | 26670 | 30986.232 | 665844 | 96948 | 43438.097 | - | 100569 |
| 100010 | 12601.262 | 56445 | 26872 | 30977.462 | - | 95842 | 43438.097 | - | 110945 |
| 100011 | 12527.841 | 27678 | 27010 | 30977.462 | - | 96948 | 43438.097 | - | 112726 |
| 100100 | 12583.611 | 28961 | 26670 | 30986.232 | 665844 | 97979 | 43438.097 | - | 108517 |
| 100101 | 12517.96 | 27522 | 26670 | 30986.232 | 665844 | 99098 | 43438.097 | - | 113323 |
| 100110 | 12586.717 | 54024 | 26888 | 30986.232 | 249775 | 97979 | 43438.097 | - | 115888 |
| 100111 | 12549.338 | 60518 | 27010 | 30986.232 | 251448 | 100055 | 43438.097 | - | 115888 |
| 101000 | 14508.66 | 24710 | 26445 | 34605.267 | 179541 | 95842 | 44662.061 | - | 100569 |
| 101001 | 14337.572 | 27024 | 26670 | 34605.267 | 233636 | 98173 | 44662.061 | - | 100569 |
| 101010 | 14509.414 | 26702 | 26872 | 34784.757 | 238622 | 95842 | 44662.061 | - | 111147 |
| 101011 | 14232.443 | 26353 | 27010 | 34784.757 | 239346 | 99414 | 44662.061 | - | 114915 |
| 101100 | 14486.785 | 29333 | 26670 | 34784.757 | 189567 | 97979 | 44662.061 | - | 113323 |
| 101101 | 14337.572 | 27960 | 26670 | 34621.956 | 204223 | 99098 | 44662.061 | - | 113323 |
| 101110 | 14499.008 | 44848 | 26888 | 34890.751 | 240478 | 97979 | 44662.061 | - | 115888 |
| 101111 | 14204.111 | 27914 | 27010 | 34890.007 | 240478 | 100976 | 44662.061 | - | 115888 |
| 110000 | 12580.537 | 59029 | 27135 | 30986.691 | 784186 | 95842 | 43438.097 | - | 100569 |
| 110001 | 12549.182 | 30002 | 27254 | 31175.761 | 669755 | 96948 | 43438.097 | - | 100569 |
| 110010 | 12672.413 | 22148 | 27243 | 30986.691 | - | 95842 | 43438.097 | - | 111147 |
| 110011 | 12549.182 | 34245 | 27267 | 31175.761 | - | 96948 | 43438.097 | - | 112726 |
| 110100 | 12580.537 | 59029 | 27267 | 31175.761 | 669755 | 100323 | 43497.628 | - | 108517 |
| 110101 | 12549.182 | 39707 | 27267 | 31175.761 | 669755 | 100323 | 43497.628 | - | 113323 |
| 110110 | 12580.537 | 33098 | 27267 | 31175.761 | 251221 | 101313 | 43497.628 | - | 115888 |
| 110111 | 12549.182 | 35247 | 27267 | 31175.761 | - | 101313 | 43497.628 | - | 115888 |
| 111000 | 14654.615 | 22729 | 27243 | 34873.187 | 583326 | 95842 | 44704.017 | - | 100569 |
| 111001 | 14508.116 | 27557 | 27254 | 34873.187 | 420250 | 98173 | 44704.017 | - | 100569 |
| 111010 | 14745.154 | 25082 | 27243 | 34873.187 | 202346 | 95842 | 44704.017 | - | 111147 |
| 111011 | 14542.634 | 26971 | 27267 | 34873.187 | 239346 | 101313 | 44704.017 | - | 114915 |
| 111100 | 14747.503 | 25416 | 27267 | 34873.187 | 583326 | 100323 | 44704.017 | - | 113323 |
| 111101 | 14534.614 | 32509 | 27267 | 34873.187 | 415263 | 100323 | 44704.017 | - | 113323 |
| 111110 | 14637.785 | 22969 | 27267 | 34873.187 | 205734 | 101313 | 44704.017 | - | 115888 |
| 111111 | 14488.57 | 31900 | 27267 | 34873.187 | 212660 | 101313 | 44704.017 | - | 115888 |
| TOTAL | 837935.009 | - | 1723258 | 2044249.756 | - | 5471165 | 2157786.151 | - | 6148424 |

Table 6.12: Results for instances C16, C20 and C25. The implementation was run on all 64 combinations for every instance. In 19 combinations of the instance C16 we are able to identify better upper bounds. In all the other instances the external given upper bounds are situated in our lower bound upper bound range. In the instance C25 we obtained no upper bounds since the solving-time of the LP-relaxation overstepped our time limit.

## Chapter 7

## Conclusion

In this thesis, a sophisticated model for the problem considered in the ROCOCO project has been developed. It is the first model published which includes all additional constraints such that any combination can be handled by parameterization. We have examined the associated polyhedra and have introduced a number of valid inequalities. We have been able to show that some of these are even facet-defining for a relaxation of the problem. These inequalities have been used as cutting planes in the applied branch-and-cut algorithm. The algorithm has been briefly explained and some strategies for obtaining better solutions have been introduced together with separation approaches. A detailed description to the applied preprocessing steps has been given which resulted to be very effective. We have presented a newly developed and implemented heuristic which reliably identifies integer solutions for the problem. This heuristic splits the network design problem into two subproblems (Determination of a feasible routing / Calculation of a appropriate linkdesign installation) and solves them successively. We have evaluated different algorithm strategies and have measured the effects of the applied preprocessing and elaborate data transformation using the given benchmark suite.

We have found optimal solutions for the small instances with exactly the same values as the best published results. This indicates that the developed model properly represents the initial problem description. For the larger problem instances the gap between the upper and lower bound turned out to be quite large, but for some instances we have been even able to identify better integer results than the best published ones.

Altogether, we have proposed a framework that successfully provides good solutions for the tested small and medium problem instances. The main problem has turned out to be the fact that the networks of the given problem instances are extremely dense. For the largest instances, the number of variables is much larger than what can be handled by any state of the art solver within the assumed 10 -minutes time limit. Hence more preprocessing or ultimately a path based model with column generation approach may be worth considering. Further improvement could be achieved by increasing the benefit of the given grouping of variables (e.g. GUB-branching) and also more tests for the enhancement of the branch-and-cut algorithm seem appropriate.

## List of Tables

3.1 Generated linkdesigns ..... 16
3.2 Merging of connected commodities ..... 24
4.1 Available linkdesigns for $e \in \delta(W)$ ..... 39
4.2 Existing commodities ..... 41
6.1 The ROCOCO data ..... 72
6.2 Combination of additional constraints ..... 73
6.3 Branching rule comparison ..... 76
6.4 Comparison of hoplimit integration ..... 78
6.5 Heuristic approach comparison ..... 80
6.6 Results for instances A04, A05 and A06 ..... 82
6.7 Results for instances A07, A08 and A09 ..... 83
6.8 Results for instances A10, B10 and B11 ..... 84
6.9 Results for instances B12, B15 and B16 ..... 85
6.10 Results for instances B20, B25 and C10 ..... 86
6.11 Results for instances C11, C12 and C15 ..... 87
6.12 Results for instances C16, C20 and C25 ..... 88

## List of Figures

3.1 Example of network with capacity installation. ..... 11
4.1 Cut traffic ..... 35
5.1 Flowchart of the applied branch-and-cut algorithm ..... 56
5.2 Flowchart of the applied heuristic algorithm ..... 65
6.1 Reduction of variables ..... 74
6.2 Reduction of constraints ..... 75

## Bibliography

[1] Network Time Protocol (NTP) project. Information available at http://www. ntp.org.
[2] E. Danna C. LePape. A. Chabrier. Solving a network design problem. Technical Report 02-005, ILOG, 2002.
[3] Algorithmic Solutions Software GmbH, Schützenstr. 3-5, D- 66123 Saarbrücken, Germany. Leda - Library of Efficient Data types and Algorithms, 1998-2003. Information available at http://www.algorithmic-solutions.com.
[4] atesio GmbH, Rubensstr. 126, D-12157 Berlin, Germany. Discnet, 2000-2003. Information available at http://www.atesio.de.
[5] D. Bienstock and O. Gunluk. Capacitated network design - polyhedral structure and computation.
[6] J.C. Régin C. Le Pape, L. Perron and Paul Shaw. Robust and parallel solving of a network design problem, 2002.
[7] Alain Chabrier. Heuristic branch-and-price-and-cut to solve a network design problem, 2003.
[8] V. Chvátal. Linear programming. A series of books in the mathematical sciences. New York - San Francisco: W. H. Freeman and Company, 1983.
[9] H. Crowder, E. Johnson, and M. Padberg. Solving large-scale 0-1 linear programming problems, 1983.
[10] ILOG CPLEX Division, 889 Alder Avenue, Suite 200, Incline Village, NV 89451, USA. ILOG CPLEX 8.0 Reference Manual, 2002. Information available at http: //www.cplex.com.
[11] G. L. Nemhauser and L. A. Wolsey. Integer and Combinatorial Optimization. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley \& Sons, Inc., 1988.
[12] S. Orlowski. Local and global restoration of node and link failures in telecommunication networks. Master's thesis, TU Berlin, 2003.
[13] Claude Le Pape Laurent Perron Raphal Bernhard, Jaques Chambon and Jean Charles Régin. Résolution d'un problème de conception de réseau avec parallel solver, 2002.
[14] A. Schrijver. Theory of linear and integer programming. Wiley-Interscience Series in Discrete Mathematics. A Wiley-Interscience Publication. Chichester: John Wiley \& Sons Ltd., 1986.
[15] C. P. M. van Hoesel, A. M. C. A. Koster, R. L. M. J. van de Leensel, and M. W. P. Savelsbergh. Polyhedral results for the edge capacity polytope. Mathematical Programming, series A, 92(2):335-358, 2002.
[16] R. Wessäly. Dimensioning survivable capacitated networks. PhD thesis, TU Berlin, 2000.
[17] L. A. Wolsey. Integer Programming. Wiley-Interscience Series in Discrete Mathematics and Optimization. New York: WileyInterscience, 1998.
[18] L.A. Wolsey. Valid inequalities for 0-1 knapsack and MIPs with generalized upper bound constraints. Discrete Applied Mathematics, 29:251-261, 1990.

## German Summary, Zusammenfassung

In der vorliegenden Diplomarbeit untersuchen wir die Optimierung von ausfallsicheren Telekommunikationsnetzen. Motiviert wurde diese Arbeit durch eine Problemstellung, welche im Kontext des ROCOCO-Projekts bearbeitet wurde und zum Ziel hatte robuste Algorithmen zu entwickeln. Als robust werden Algorithmen bezeichnet, welche nicht nur gute Ergebnisse für Probleminstanzen verschiedener Grösse und Charakteristik liefern, sondern auch weiterhin gut arbeiten wenn Bedingungen hinzugefügt oder entfernt werden.
Ausgehend von einer textuellen Problembeschreibung entwickeln wir ein umfassendes Modell. Es handelt sich um das nach unserer Kenntnis erste Modell, durch das sich jede beliebige Kombination der betrachteten Bedingungen abbilden lässt. Das durch das Model induzierte Polyeder wird untersucht und gültige Ungleichungen werden vorgestellt. Wir zeigen, dass einige dieser Ungleichungen facettendefinierend für eine Relaxierung unseres Problems sind. Die ermittelten Ungleichungen werden als Schnittebenen in einem Branch-and-Cut-Algorithmus verwendet, welcher als zentraler Algorithmus eingesetzt wird. Auf diesen Algorithmus wird genauer eingegegangen, und es werden Strategien eingeführt, durch die die Lösbarkeit verbessert werden kann. Weiterhin werden umfangreiche Preprocessing-Prozeduren und eine neu entwickelte und implementierte Heuristik vorgestellt.
Wir haben unsere Implementation an der für das ROCOCO-Project entwickelten Benchmarksuite getestet, was ausführlich beschrieben wird. Desweiteren werden für alle Datensätze und Kombinationen von Bedingung die erreichten Ergebnisse mit den besten verfügbaren Lösungen verglichen. Es wird gezeigt, dass wir in den kleineren Probleminstanzen exakt dieselben Optimallösungen erreichen und für einige Instanzen sogar ganzzahlige Lösungen erhalten, die besser als die bis jetzt veröffentlichten Lösungen sind.

# Statutory Declaration, Eidesstattliche Erklärung 

Hiermit versichere ich an Eides statt die selbständige und eigenhändige Anfertigung dieser Diplomarbeit.
(Ulrich Menne)

