TECHNISCHE UNIVERSITÄT WIEN
Institut für Computergraphik und Algorithmen

Branch-and-Price for a Survivable Network Design Problem

Markus Leitner and Günther R. Raidl and Ulrich Pferschy

Forschungsbericht / Technical Report

TR–186–1–10–02

May 27, 2010

Favoritenstraße 9-11 / E186, A-1040 Wien, Austria
Tel. +43 (1) 58801-18601, Fax +43 (1) 58801-18699
www.cg.tuwien.ac.at

Branch-and-Price for a Survivable Network Design Problem

Markus Leitner[1], Günther R. Raidl[1], and Ulrich Pferschy[2]

[1]Institute of Computer Graphics and Algorithms
Vienna University of Technology, Austria
{leitner|raidl}@ads.tuwien.ac.at

[2]Institute of Statistics and Operations Research
University of Graz, Austria
pferschy@uni-graz.at

Abstract

We consider a specific variant of the survivable network design problem suitable to model real world scenarios occurring in the extension of fiber optic networks. In this article, two mixed integer programming models, which can be solved by branch-and-price, are discussed and compared to existing approaches theoretically as well as by a computational study. We further discuss the usage of alternative dual-optimal solutions to stabilize our approaches and significantly reduce the computational times needed to solve the linear relaxations of our models. The obtained computational results show that both branch-and-price approaches are suitable for solving small to medium sized problem instances to proven optimality.

Keywords: Network Design, Column Generation, Mixed Integer Programming, Branch-and-Price

1 Introduction

The $b_{\max}$-Survivable Network Design Problem ($b_{\max}$-SNDP) is a real-world communication network design problem which arises for instance in the expansion of fiber optic networks. Recently, fiber-to-the-home has become economically feasible for individual households in urban areas. However, covering larger districts with such networks requires enormous financial resources from an operators point of view. Since customers are usually not willing to pay significantly more than for existing lower bandwidth connections, good algorithms for finding cost-efficient network layouts are crucial.

$b_{\max}$-SNDP considers the problem of augmenting an existing network infrastructure by additional links and switches in order to connect additional customer nodes. Here, we distinguish between standard (type-1) customer nodes for which a single link connection suffices and type-2 customer nodes representing business customers who require a more reliable connection, ensuring connectivity even when a single link or routing node fails. Since offering full redundancy to each type-2 customer often is too expensive and does not pay off from an economic point of view, we consider a problem variant where the redundancy condition for type-2 customers is relaxed in the sense that a connection is allowed via a final non-redundant branch line that does not exceed a certain length $b_{\max}$. Thus, we restrict the length of the non-redundant part of a connection taking a compromise between reliability and construction costs.

The remainder of this article is organized as follows. After formally introducing $b_{\max}$-SNDP in Section 2 and reviewing previous and related work in Section 3 we present two mixed integer programming approaches – a directed and an undirected one – for solving $b_{\max}$-SNDP to proven optimality. These are based on an exponential number of so-called connection variables and can be solved by branch-and-price. As one main contribution within this section, we show how to significantly speed up the solution of the linear relaxation of these models by using alternative dual-optimal solutions in the pricing subproblem. Theoretical comparisons of the corresponding polyhedra of those two as well two previously existing formulations are given in Section 6. Test instances for benchmarking the approaches and computational results are discussed in Sections 7 and 8, respectively, before we finally draw conclusions and outline potential future work in Section 9.

This article significantly extends our previous work [33, 32] in various ways. We additionally propose the usage of alternative dual-optimal solutions in the pricing problem also for the directed model in Section 5 and compare our models to existing ones theoretically by a polyhedral study. Furthermore, an additional variant for generating alternative dual-optimal solutions as well as a new transformation to the elementary shortest path problem with resource constraints are considered for the pricing subproblem of the directed model. Finally, both models are embedded in a branch-and-price framework, more computational results are given, and most parts are described in more detail.

2 Problem Definition

Formally, we are given a connected undirected graph $G^{\mathrm{o}} = (V^{\mathrm{o}}, E^{\mathrm{o}})$ representing the spatial topology of the surrounding area of potential customers. Each edge $e = (u, v) \in E^{\mathrm{o}}$ corresponding to a possible cable route between its end points $u, v \in V^{\mathrm{o}}$ is given with its length $l_e \geq 0$ and costs $c_e^{\mathrm{o}} \geq 0$ for installing the corresponding fiber optic link. The node set $V^{\mathrm{o}} = S \cup C \cup V_{\mathrm{I}}$ is the disjoint union of customer nodes C, spatial nodes S (switches, possible Steiner nodes) and nodes of the already existing network infrastructure V_{I}. The set of customers $C = C_1 \cup C_2$ is partitioned into type-1 customer nodes C_1 without specific redundancy requirements and type-2 customer nodes C_2 that need to be redundantly connected by means of two node disjoint paths to the existing infrastructure. Each customer node $k \in C$ further has associated a prize $p_k \geq 0$ modeling the expected return on investment when supplying customer k. Finally, the already existing network infrastructure is represented by the subgraph $(V_{\mathrm{I}}, E_{\mathrm{I}})$, $V_{\mathrm{I}} \subsetneq V^{\mathrm{o}}$, $E_{\mathrm{I}} \subsetneq E^{\mathrm{o}}$, see Figure 1.

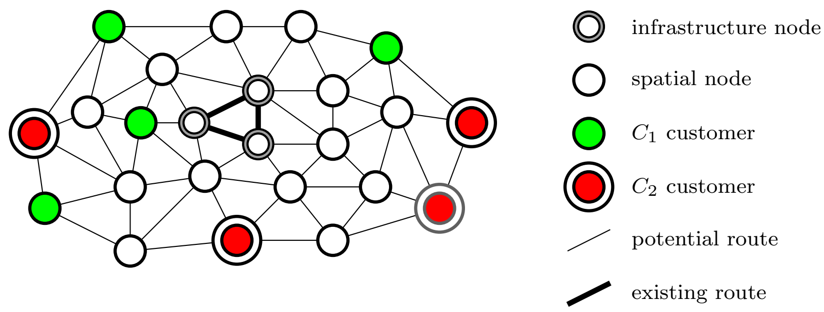


Figure 1: An instance of $b_{\max}$-SNDP.

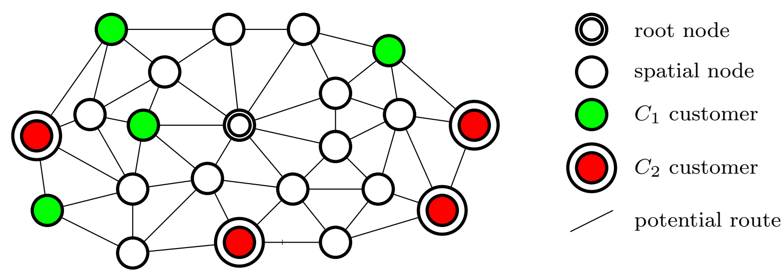


Figure 2: The instance of $b_{\max}$-SNDP from Figure 1 after shrinking the existing infrastructure.

In a first preprocessing step, we create a reduced graph $G = (V, E)$ by shrinking the whole existing network infrastructure into a single root node $r \in V$. From all edges $(u, v) \in E^{\mathrm{o}}$ connecting a node $u \in V^{\mathrm{o}} \setminus V_{\mathrm{I}}$ to the existing infrastructure – i.e. $v \in V_{\mathrm{I}}$ – only the cheapest edge (r, u) from the root node to u is included in E. Formally, $G = (V, E)$ is defined by its node set $V = \{r\} \cup S \cup C$, and its edge set $E = \{(u, v) \mid u, v \in V \wedge (u, v) \in E^{\mathrm{o}}\} \cup \{(r, v) \mid \exists (u, v) \in E^{\mathrm{o}} \wedge u \in V_{\mathrm{I}} \wedge v \in V^{\mathrm{o}} \setminus V_{\mathrm{I}}\}$, see Figure 2. Customers with associated prizes and edge lengths are adopted from the original graph $G^{\mathrm{o}} = (V^{\mathrm{o}}, E^{\mathrm{o}})$. Since we include one edge (r, v) for all original edges connecting v with some node of the existing infrastructure $w \in V_{\mathrm{I}}$, edge costs c_e, are defined as follows:

$$c_e = \begin{cases} c_e^{\mathrm{o}}, & \text{if } u, v \notin V_{\mathrm{I}} \\ \min\{c_f^{\mathrm{o}} \mid f = (w, v) \in E^{\mathrm{o}} : w \in V_{\mathrm{I}}\} & \text{otherwise} \end{cases}, \forall e = (u, v) \in E.$$

Let $G' = (V', E')$, $V' \subseteq V$, $E' \subseteq E$, represent a solution network to an instance of $b_{\max}$-SNDP. The following conditions specify how customer nodes are to be connected:

- *Simple connection:*

  A type-1 customer node $k \in C_1$ is feasibly connected iff there exists a path from node r to k.

- *Redundant connection:*

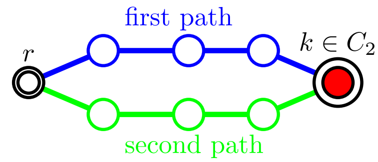


Figure 3: A feasible connection to $k \in C_2$ with $b_{\max}(k) = 0$.

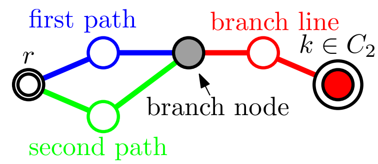


Figure 4: A feasible connection to $k \in C_2$ with $b_{\max}(k) > 0$.

A customer node $k \in C_2$ is feasibly connected iff there exist two node (and edge) disjoint paths from node r to k, see Figure 3.

- $b_{\max}$-*redundant connection:*

  Occasionally, the biconnectivity condition for the nodes in set C_2 is relaxed in the sense that such a node $k \in C_2$ may be connected to any biconnected (Steiner or customer) node $j \in V$ (the *branch node* of k) via a single path of maximum total length $b_{\max}(k) > 0$. This (optional) single path is called *branch line* and $b_{\max}(k)$ the *maximum branch line length* for customer k, see Figure 4. We denote the set of potential branch nodes for a customer $k \in C_2$, i.e. the nodes from which paths to k no longer than $b_{\max}(k)$ exist, by $\mathcal{B}(k) \subseteq V$.

  Since each type-2 customer is a potential branch node of itself whereas k is the only potential branch node if $b_{\max}(k) = 0$, $k \in \mathcal{B}(k)$ holds for all type-2 customers $k \in C_2$ independent of a concrete problem instance and a given maximum branch line length.

Note that we assume $r \notin \mathcal{B}(k)$, $\forall k \in C_2$, since above mentioned shrinking of the existing infrastructure into the root node r might influence the optimal solution value otherwise.

Regarding the objective, we distinguish between two alternative goals:

- In the *Operative Planning Task* (OPT) we focus on finding a minimum-cost subgraph G' feasibly connecting all customers C, with the total costs being

$$o_{\mathrm{OPT}}(G') = \sum_{e \in E'} c_e. \tag{1}$$

  This case can be considered a generalization of the classical Steiner tree problem on a graph (STP)

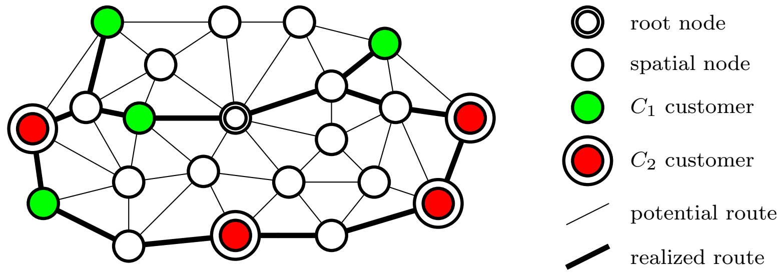


Figure 5: An exemplary solution to the OPT variant of $b_{\max}$-SNDP with $b_{\max}(k) = 0$, $\forall k \in C_2$.

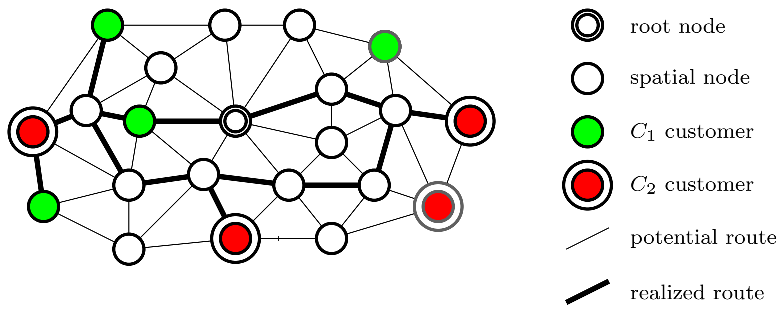


Figure 6: An exemplary solution to the SST variant of $b_{\max}$-SNDP with $b_{\max}(k) \neq 0$, $\forall k \in C_2$.

where a special form of redundancy is required for the nodes in C_2.

- In the *Strategic Simulation Task* (SST) customers' prizes are also considered, and the objective is to determine a subset $C' \subseteq C$ of customers which are connected so that the costs for building the network minus the earned prizes are minimized. In order to always have positive total costs, which eases some parts of our algorithms and notations, we perform a simple transformation by adding the constant $\sum_{k \in C} p_k$ to the objective function, yielding

$$o_{\mathrm{SST}}(G') = \sum_{e \in E'} c_e - \sum_{k \in C'} p_k + \sum_{k \in C} p_k = \sum_{e \in E'} c_e + \sum_{k \in C \setminus C'} p_k. \tag{2}$$

  This problem variant is a generalization of the prize-collecting Steiner tree problem (PCSTP).

Figure 5 depicts an exemplary solution to the OPT variant of $b_{\max}$-SNDP without considering $b_{\max}$-redundancy – i.e. $b_{\max}(k) = 0$, $\forall k \in C_2$ – while Figure 6 shows an exemplary solution to the SST variant including $b_{\max}$-redundancy.

As already the classical Steiner tree problem on a graph is $\mathcal{NP}$-hard [28], this obviously also holds for both of our problem variants. In the following presentation of our solution approaches, we primarily consider the more complex SST case if not explicitly stated and assume $p_k = \infty$, $\forall k \in C$, to include the OPT case.

3 Related Work

$b_{\max}$-SNDP has been introduced by Bachhiesl et al. [3]. Ljubić [34] introduced its name[1] and pointed out the relation to $\{0,1,2\}$-SNDP [29] which corresponds to $b_{\max}$-SNDP with $b_{\max}(k) = 0$, $\forall k \in C_2$.

Wagner et al. [45] presented mixed integer programming (MIP) approaches for $b_{\max}$-SNDP based on multi-commodity flows. With the general purpose ILP-solver CPLEX [25], instances with up to 190 total nodes, 377 edges but only 6 customer nodes could be solved to proven optimality, and instances up to 2804 nodes, 3082 edges and 12 customer nodes could be solved with a final LP gap of about 7%. Unfortunately, this approach turned out to be unsuitable for larger instances and/or in particular instances with larger number of customer nodes, as already solving the linear programming (LP) relaxation of the MIP requires too much time due to the huge number of variables involved. In [44], the same authors approached $b_{\max}$-SNDP with a different formulation based on connectivity constraints. While this formulation involves only a reasonable number of variables, the number of inequalities is exponentially large. By using a branch-and-cut algorithm, this model could be solved relatively well, and they were able to find proven optimal solutions for instances with up to 190 nodes, 377 edges, and 13 customer nodes. For larger, practical instances this approach unfortunately still is not applicable at all or finds quite poor solutions with huge LP-gaps only. The current authors heuristically approached medium-sized instances of $b_{\max}$-SNDP by means of Lagrangian decomposition (LD), variable neighborhood search, a greedy randomized adaptive search procedure (GRASP), as well as by hybrid methods combining LD with variable neighborhood descent (VND) [31].

Modeling redundant connections by pairs of reversely oriented paths, Chimani et al. [13, 12] further came up with strong formulations for $\{0,1,2\}$-SNDP based on multi-commodity flows and directed connection cuts, theoretically dominating those of Wagner et al. [45, 44] for the case of $b_{\max}(k) = 0$, $\forall k \in C_2$. Their formulations were able to solve larger instances and to consider a greater number of customer nodes than the approaches of Wagner et al. However, their directed model cannot be easily adapted to consider $b_{\max}$-redundancy, too.

The classical Steiner tree problem (STP) on graphs has been considered by many authors, see e.g. [47] for a survey. Among the various authors that considered integer programming models for the STP, Koch and Martin [30] described an effective branch-and-cut method based on directed connectivity cuts. More recently, Bahiense et al. [4] presented a Lagrangian Further well known heuristic methods have e.g. been described by Takahashi and Matsuyama [41] and Duin and Voß [17].

The prize collecting Steiner tree problem (PCSTP) was introduced by Segev [38] who considered the node weighted STP, which is a special version of the PCSTP. The term "prize collecting" has first been used by Balas [5] for the prize collecting traveling salesman problem. Ljubić et al. [35] presented an exact

[1] Ljubić used the name $k_{\max}$-SNDP.

method for the PCSTP based on directed connection cuts. Other successful mathematical programming based approaches include a relax-and-cut by Cunha et al. [14] and a cutting plane method by Lucena et al. [37]. Canuto et al. [9] described an effective multi-start local search based on perturbation of the nodes' prizes, where path-relinking and variable neighborhood search are used to further improve the obtained solutions. Preprocessing conditions for reducing the number of nodes and edges of a PCSTP instance have been described by Uchoa [42], whereas Chapovska et al. [11] discuss the complexity of several special variants of the PCSTP and corresponding solution methods.

Other related problems are the various variants of the survivable network design problem (SNDP) [21]. Among these, especially the "low connectivity" variants such as above mentioned $\{0, 1, 2\}$-SNDP are relevant for $b_{\max}$-SNDP, see e.g. [29, 39] for surveys.

4 The Undirected Connection Formulation for $b_{\max}$-SNDP

To model $b_{\max}$-SNDP as a mixed integer program (MIP) we consider the set of all possible feasible connections $\mathcal{F}_k$ for each customer $k \in C$. For type-1 customers $k \in C_1$, $\mathcal{F}_k$ corresponds to the set of all paths from the root node r to k, i.e.

$$\mathcal{F}_k = \{p \subseteq E \mid p \text{ forms a path from } r \text{ to } k\},$$

while for type-2 customers $k \in C_2$, $\mathcal{F}_k$ can be expressed as follows:

$$\begin{aligned} \mathcal{F}_k = \{p \subseteq E \mid\ & p \text{ forms two node disjoint paths from } r \text{ to some node } j \in \mathcal{B}(k) \text{ and} \\ & \text{one path from } j \text{ to } k \text{ whose length does not exceed } b_{\max}(k)\}. \end{aligned}$$

We formulate the SST variant of $b_{\max}$-SNDP by the following *integer master problem* (Col) using variables $f_p^k \in \{0, 1\}$, $\forall k \in C$, $\forall p \in \mathcal{F}_k$, to indicate whether a corresponding connection $p \in \mathcal{F}_k$ is realized ($f_p^k = 1$) or not ($f_p^k = 0$), decision variables $x_e \in \{0, 1\}$, $\forall e \in E$, to specify whether an edge e is part of the solution ($x_e = 1$) or not ($x_e = 0$), and variables $y_k \in \{0, 1\}$, $\forall k \in C$, to denote whether a feasible route to customer k is installed ($y_k = 1$) or not ($y_k = 0$). Variables y_k are fixed to one in the OPT variant.

$$(3) \qquad (\text{Col}) \quad z = \min \sum_{e \in E} c_e x_e + \sum_{k \in C} p_k (1 - y_k)$$

$$(4) \qquad \text{s.t.} \quad \sum_{p \in \mathcal{F}_k} f_p^k - y_k \geq 0 \qquad \forall k \in C$$

$$(5) \qquad x_e - \sum_{p \in \mathcal{F}_k | e \in p} f_p^k \geq 0 \qquad \forall k \in C,\ \forall e \in E$$

$$(6) \qquad x_e \in \{0, 1\} \qquad \forall e \in E$$

$$(7) \qquad 0 \leq y_k \leq 1 \qquad \forall k \in C$$

$$(8) \qquad f_p^k \geq 0 \qquad \forall k \in C,\ \forall p \in \mathcal{F}_k$$

Constraints (4) ensure that a customer's prize can only be earned if it is feasibly connected to r, while constraints (5) link connection variables to edge variables. We define only lower and upper bounds for variables y_k and only lower bounds for variables f_p^k in inequalities (7) and (8). If all edge variables x_e, $\forall e \in E$, are integral, each set of potentially existing fractional connections to some customer $k \in C$ can be replaced by an integral connection without including additional edges and thus without modifying the solution's objective value. Since customer prizes reduce the objective value, variables y_k, $\forall k \in C$, we further conclude that they will automatically become integer.

The linear relaxation of (Col) – the *linear master problem* $(\text{Col})^{\text{LP}}$ – is given by substituting the integrality constraints (6) by

$$(9) \qquad x_e \geq 0 \qquad \forall e \in E$$

Let $\mu_k \geq 0$, $\forall k \in C$, be the dual variables associated to the convexity constraints (4) and $\pi_{k,e} \geq 0$, $\forall k \in C,\ \forall e \in E$, be the dual variables associated to the coupling constraints (5).

Furthermore, let $F = \{f_p^k \mid k \in C, p \in \mathcal{F}_k\}$ be the set of all f_p^k variables representing columns in $(\text{Col})^{\text{LP}}$. Since F consists of an exponential number of variables we cannot solve $(\text{Col})^{\text{LP}}$ directly, but use column generation [6, 15]. We define the *restricted master problem* $(\text{Col})^{\text{RMP}}$ using only a small subset of connection variables $\tilde{F} \subsetneq F$; otherwise $(\text{Col})^{\text{RMP}}$ corresponds to $(\text{Col})^{\text{LP}}$.

When solving $(\text{Col})^{\text{RMP}}$ we obtain optimal dual variable values μ_k^* and $\pi_{k,e}^*$, defining reduced prices $\bar{c}_{k,p}$ for variables $f_p^k \in F \setminus \tilde{F}$:

$$\bar{c}_{k,p} = -\mu_k^* + \sum_{e \in p} \pi_{k,e}^*$$

The pricing problem is now to find $(k^*, p^*) = \operatorname{argmin}_{k \in C, p \in F_k} \{\bar{c}_{k,p}\}$. If $\bar{c}_{k^*,p^*} \geq 0$ we have obtained an optimal solution to $(\mathrm{Col})^{\mathrm{LP}}$. Otherwise, we add at least one column with negative reduced costs and resolve $(\mathrm{Col})^{\mathrm{RMP}}$.

More generally speaking, in the pricing subproblem we have to find a feasible connection for some $k \in C$ yielding negative reduced costs $\bar{c}_{k,p} = -\mu_k^* + \sum_{e \in p} \pi_{k,e}^*$ or prove that no such connection exists. For this purpose we need to determine a cheapest feasible connection on graph $G = (V, E)$ with modified edge costs $\pi_{k,e} \geq 0$, $\forall e \in E$, for each customer node $k \in C$. When the costs of such a connection are less then μ_k, we have found an appropriate connection, i.e. the corresponding variable f_p^k can be added to $(\mathrm{Col})^{\mathrm{RMP}}$.

While for type-1 customers $k \in C_1$ finding the cheapest feasible connection is a simple shortest path calculation from r to k, we have to find a cheapest pair of node-disjoint paths from r to k for type-2 customers (without yet considering $b_{\max}$-redundancy). Suurballe and Tarjan [40] (see also [27]) presented an algorithm to efficiently compute a shortest arc-disjoint pair of paths between two nodes in time $O(|E| + |V| \log |V|)$. By applying this algorithm on the split graph of the original graph we can compute a shortest node-disjoint pair of paths. The split graph is obtained by replacing each node $v \in V$ by a pair of nodes v' and v''. For each such pair, we add an arc (v', v'') with zero costs. Each edge $e = (u, v)$ of G is replaced by two directed arcs (u'', v'), (v'', u') having costs c_e.

In case of $b_{\max}$-redundancy, the above algorithm must further be extended. We consider each node $v \in \mathcal{B}(k)$ in the $b_{\max}$-neighborhood of node $k \in C_2$ and determine a cheapest pair of paths to this node. Furthermore, a cheapest length constrained shortest path from node k to each potential branch node must be computed. The overall cheapest combination is the final result. Since, computing a (length) constrained cheapest path is $\mathcal{NP}$-hard [22] relaxing the biconnectivity constraints by means of $b_{\max}$-redundancy turns out to significantly increase the subproblem's complexity not only from a computational but also from a theoretic point of view. However, several pseudo-polynomial algorithms for solving constrained shortest path problems have been proposed, see e.g. [7, 18]. In our work, we use the approach described by Gouveia et al. [24] which solves this problem for a customer $k \in C_2$ in $O(b_{\max}(k)|\mathcal{E}(k)|)$, where $\mathcal{E}(k) = \{e = (u, v) \in E \mid u, v \in \mathcal{B}(k)\}$. Since $b_{\max}(k)$ and thus $|\mathcal{E}(k)|$ is typically rather small, we are able to solve this $\mathcal{NP}$-hard problem by above mentioned dynamic programming based approach without increasing the computational effort too much.

4.1 Analyzing the Restricted Dual Problem

It is well known that (simplex based) column generation approaches often suffer from inefficiency resulting in a large number of required pricing iterations as well as long computation times. Vanderbeck [43] describes five major efficiency issues of simplex based column generation.

Several stabilization techniques to reduce their effects have been proposed, see e.g. [16] or [36] for reviews on those methods. From the issues described by Vanderbeck preliminary tests showed that *primal degeneracy* as well as the *heading-in effect* are mainly relevant in our case. The occurrence of primal degeneracy is based on the fact that typically only very few connection and edge variables will have nonzero values in a solution of $(\text{Col})^{\text{RMP}}$.

Instead of using a problem-independent stabilization approach we analyze the dual of $(\text{Col})^{\text{RMP}}$ to take advantage of problem specific characteristics. Let $\lambda_k \leq 0$ denote the dual variables associated to inequalities (7). As mentioned before $\tilde{F} \subsetneq F$ denotes the set of variables representing connections to customers in $(\text{Col})^{\text{RMP}}$. The dual of the restricted master problem $(\text{Col})^{\text{RMP}}$ – i.e. the *restricted dual problem* – for the SST variant is given by model (10)–(16).

$$\text{(10)} \qquad \max \sum_{k \in C} \lambda_k + p_k$$

$$\text{(11)} \qquad \sum_{k \in C} \pi_{k,e} \leq c_e \qquad \forall e \in E$$

$$\text{(12)} \qquad \mu_k - \sum_{e \in p} \pi_{k,e} \leq 0 \qquad \forall k \in C,\ \forall p \in \mathcal{F}_k | \exists f_p^k \in \tilde{F}$$

$$\text{(13)} \qquad -\mu_k + \lambda_k \leq -p_k \qquad \forall k \in C$$

$$\text{(14)} \qquad \pi_{k,e} \geq 0 \qquad \forall k \in C,\ \forall e \in E$$

$$\text{(15)} \qquad \mu_k \geq 0 \qquad \forall k \in C$$

$$\text{(16)} \qquad \lambda_k \leq 0 \qquad \forall k \in C$$

Let $E'' \subseteq E$ denote the subset of edges which are not part of any so far included connection, i.e. $E'' = \left\{ e \in E \mid \nexists f_p^k \in \tilde{F} : e \in p \right\}$. For edges $e \in E''$, only inequalities (11) are relevant. Thus all values $\pi_{k,e} \geq 0$, $\forall k \in C$, $\forall e \in E''$, are dual optimal as long as $\sum_{k \in C} \pi_{k,e} \leq c_e$ holds.

Since almost the complete edge set E will not be in any included connection in the beginning of our column generation procedure, dual variable values $\pi_{k,e}$ used as edge costs in the pricing subproblem will not be meaningful. Furthermore, in order to be able to solve $(\text{Col})^{\text{LP}}$ efficiently, we aim at keeping the number of included connection variables as well as the set $E \setminus E''$ as small as possible.

Generally speaking, the structure of model (10)–(16) imposes the generation of many irrelevant columns having identical reduced prices. This observation explains the occurrence of the heading in effect. This effect is even intensified by the fact that CPLEX [25] – which we use for solving the linear relaxation of our model – generates minimal dual-optimal values for all dual variables, i.e. most of them will be zero.

4.2 Alternative Dual-Optimal Solutions

In the following, we detail our stabilization procedure for generating meaningful dual variable values in the pricing problem. Hereby, we exploit different dual-optimal solutions to improve the convergence properties of our column generation algorithm. This approach can be interpreted as a stabilization technique that "centers" an actual LP solution.

Let $D^* = (\lambda^*, \mu^*, \pi^*)$ be an optimal solution to the restricted dual problem (10)–(16). As shown in the previous section, for edges $e \in E''$ all values $\pi_{k,e} \geq 0$, $\forall k \in C$, are dual optimal as long as $\sum_{k \in C} \pi_{k,e} \leq c_e$. Furthermore, for edges $e \in E \setminus E''$, we may increase the sum of dual variable values $\sum_{k \in C} \pi_{k,e}$ by $\delta_e = c_e - \sum_{k \in C} \pi_{k,e}$.

As mentioned earlier CPLEX [25] generates minimal values for dual variables (i.e. $\pi_{k,e} = 0, \forall k \in C, \ \forall e \in E''$; usually $\delta_e > 0$ for some edges $e \in E \setminus E''$). For creating more meaningful dual variable values and thus keeping the set of edges and connection variables that will be finally included relatively small, we aim to increase variable values $\pi_{k,e}$, $\forall k \in C$, $\forall e \in E$, while maintaining dual optimality.

The probably simplest and most obvious strategy is to use the alternative dual optimal solution $D' = (\lambda^*, \mu^*, \pi')$ with $\pi'_{k,e} = \frac{c_e}{|C|}$, $\forall k \in C, \ \forall e \in E''$ and $\pi'_{k,e} = \pi^*_{k,e} + \frac{\delta_e}{|C|}$, $\forall k \in C, \ \forall e \in E \setminus E''$. However, as will be illustrated by our computational results we can do even better by initially using different dual-optimal solutions $D^{(k,d)} = \left(\lambda^*, \mu^*, \pi^{(k,d)}\right)$, for all $k \in C$ – controlled by parameter d ($1 \leq d \leq |C|$) – which finally converge to D' for $d = |C|$. When considering client $k \in C$ in the pricing problem, we use dual values $\pi^{(k,d)}_{k,e} = \frac{c_e}{d}$, $\forall e \in E''$, and $\pi^{(k,d)}_{k,e} = \pi^*_{k,e} + \frac{\delta_e}{d}$, $\forall e \in E \setminus E''$. Note that assuming $\pi^{(k,d)}_{k',e} = 0$, $\forall k' \neq k \in C$, $\forall e \in E''$, and $\pi^{(k,d)}_{k',e} = \pi^*_{k,e}$, $\forall k' \neq k \in C$, $\forall e \in E \setminus E''$ we again only use dual optimal solutions when solving the pricing problem. Parameter d is initially set to one and gradually incremented up to $|C|$ in case no column with negative reduced cost could be priced in and reset to one in case columns including new edges have been added to $(\mathrm{Col})^{\mathrm{RMP}}$. Since we essentially use D' if $d = |C|$ we can terminate the column generation process if no column with negative reduced costs could be found for $d = |C|$.

We further apply a simpler variant of $D^{(k,d)}$ where d is initially set to one and set to $|C|$ in case no connection yielding negative reduced costs could be identified. In this strategy – we refer to the resulting dual optimal solutions by $D^{(k,d')} - d$ will not be decreased any more.

While the above mentioned strategies are feasible for both the SST as well as the OPT variant of $b_{\max}$-SNDP, we further consider a fourth approach for the SST variant that also takes each customer's prize into consideration. For each edge $e \in E$, we add a value corresponding to its prize relative to the sum of all prizes, i.e. for all customers $k \in C$ we set $\pi^{(\mathrm{p})}_{k,e} = c_e \frac{p_k}{\sum_{l \in C} p_l}$ if $e \in E''$ and $\pi^{(\mathrm{p})}_{k,e} = \pi^*_{k,e} + \delta_e \frac{p_k}{\sum_{l \in C} p_l}$ otherwise. The resulting alternative dual optimal solution is denoted by $D^{(\mathrm{p})} = \left(\lambda^*, \mu^*, \pi^{(\mathrm{p})}\right)$.

5 The Directed Connection Formulation for $b_{\max}$-SNDP

Since directed formulations are in many cases theoretically stronger than undirected ones and frequently also outperform those from a computational point of view, it is natural to ask whether it is possible to replace the undirected formulation (Col) presented in the previous section by a directed one.

Chimani et al. [13] showed that any feasible solution to $\{0,1,2\}$-SNDP can be transformed into a directed graph with a simple path from r to each connected type-1 customer and two oppositely directed, internally node disjoint paths between r and any connected type-2 customer $k \in C_2$. Interpreting a feasible connection to some customer $k \in C_2$ with $b_{\max}(k) > 0$ as two independent connections – a non-redundant one from r to k and a fully redundant connection to its branching node $v \in \mathcal{B}(k)$ – the orientability of any solution to $b_{\max}$-SNDP follows from the result of Chimani et al.

In this section, we introduce model (dCol), resembling a directed variant of model (Col), which exploits the orientability of solutions to $b_{\max}$-SNDP. Let $A = \{(u,v),(v,u) \mid e = (u,v) \in E\}$ consist of two oppositely directed arcs for each original edge $e \in E$. To model $b_{\max}$-SNDP we utilize binary variables $a_{u,v} \in \{0,1\}$, $\forall (u,v) \in A$, indicating whether or not arc $(u,v) \in A$ is part of the (oriented) solution ($a_{u,v} = 1$) or not ($a_{u,v} = 0$). As for model (Col), variables $y_k \in \{0,1\}$, $\forall k \in C$, specify whether a customer is feasibly connected according to its redundancy requirements or not. We further use variables $h^k_p \in \{0,1\}$, $\forall k \in C$, $\forall p \in \mathcal{H}_k$, where $\mathcal{H}_k$ is the set of all feasible directed connections for customer $k \in C$, indicating whether the corresponding connection is realized ($h^k_p = 1$) or not ($h^k_p = 0$).

Analogously to (Col), for type-1 customers $k \in C_1$, $\mathcal{H}_k$ corresponds to the set of all directed paths from the root node r to k, i.e.

$$\mathcal{H}_k = \{p \subseteq A \mid p \text{ forms a directed path from } r \text{ to } k\},$$

while for type-2 customers $k \in C_2$, $\mathcal{H}_k$ can be expressed as follows:

$$\mathcal{H}_k = \{p \subseteq A \mid p \text{ forms two oppositely directed, internally node disjoint paths between } r \text{ and some node } j \in \mathcal{B}(k) \text{ and a directed path from } j \text{ to } k \text{ whose length does not exceed } b_{\max}(k)\}.$$

Using directed arc costs $c_{u,v} = c_e$, $\forall (u,v) \in A$, $e = (u,v) \in E$, we can express $b_{\max}$-SNDP by the following model (dCol).

$$\text{(dCol)} \quad z = \min \sum_{(u,v)\in A} c_{u,v} a_{u,v} + \sum_{k \in C} p_k (1 - y_k) \tag{17}$$

$$\text{s.t.} \quad \sum_{p \in \mathcal{H}_k} h_p^k - y_k \geq 0 \qquad \forall k \in C \tag{18}$$

$$a_{u,v} - \sum_{p \in \mathcal{H}_k \mid (u,v) \in p} h_p^k \geq 0 \qquad \forall k \in C,\ \forall (u,v) \in A \tag{19}$$

$$a_{u,v} + a_{v,u} \leq 1 \qquad \forall e = (u,v) \in E \tag{20}$$

$$a_{u,v} \in \{0,1\} \qquad \forall (u,v) \in A \tag{21}$$

$$0 \leq y_k \leq 1 \qquad \forall k \in C \tag{22}$$

$$h_p^k \geq 0 \qquad \forall k \in C,\ \forall p \in \mathcal{H}_k \tag{23}$$

Constraints (18) ensure that a customer's prize can only be earned if it is feasibly connected to r, while constraints (19) link connection variables to arc variables. Inequalities (20) guarantee that at most one out of each pair of oppositely directed arcs is used in a solution. Note that for variables y_k and h_p^k only bounds are defined in (22) and (23), as they will automatically become integer by the same arguments as for model (Col).

As in model (Col), there are exponentially many variables $H = \{h_p^k \mid k \in C \wedge p \in \mathcal{H}_k\}$ corresponding to feasible directed connections. Thus, we cannot directly solve the linear relaxation $\text{(dCol)}^{\text{LP}}$ of model (17)–(23) which is given by substituting inequalities (21) by

$$a_{u,v} \geq 0 \qquad \forall (u,v) \in A. \tag{24}$$

We apply column generation [6, 15] for solving $\text{(dCol)}^{\text{LP}}$ analogously to the undirected connection for-

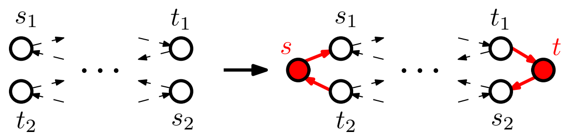


Figure 7: Transformation of 2DP on (s_1, t_1), (s_2, t_2) into ODP on (s, t).

mulation presented in the last section. Again, we start with a small subset of connection variables $\tilde{H} \subsetneq H$ considered in the restricted master problem $(\mathrm{dCol})^{\mathrm{RMP}}$ and dynamically add further variables $h \in H \setminus \tilde{H}$ by iteratively solving the pricing problem.

Let $\nu_k \geq 0$, $\forall k \in C$, be the dual variables associated to constraints (18) and $\omega_{k,u,v} \geq 0$, $\forall k \in C$, $\forall (u,v) \in A$, denote the dual variables associated to constraints (19). Then, when solving $(\mathrm{dCol})^{\mathrm{RMP}}$ reduced prices $\bar{c}_{k,p}$ for connection variables $h^k_p \in H \setminus \tilde{H}$ can be computed by

$$\bar{c}_{k,p} = -\nu_k + \sum_{(u,v)\in p} \omega_{k,u,v}.$$

In the pricing problem, we need to find $(k^*, p^*) = \operatorname{argmin}_{k \in C, p \in \mathcal{H}_k} \{\bar{c}_{k,p}\}$. As long as at least one variable with negative reduced costs does exist, we add it to $\tilde{H}$ and resolve $(\mathrm{dCol})^{\mathrm{RMP}}$.

In other words, in the pricing problem we need to determine a cheapest directed connection to each customer $k \in C$ in $D = (V, A)$ with arc costs $\omega_{k,u,v} \geq 0$, $\forall (u,v) \in A$. If the total costs of such a connection are smaller than ν_k, the corresponding connection variable has negative reduced costs and is included in $(\mathrm{dCol})^{\mathrm{RMP}}$. Since arc costs are non-negative we can efficiently solve the pricing problem for type-1 customers by simple cheapest path calculations. For customers $k \in C_2$ with $b_{\max}(k) = 0$ we need to compute the cheapest pair of oppositely directed, internally node disjoint paths (ODP) between the root node r and k.

As shown in Figure 7 any instance of the directed disjoint pair of paths problem (2DP) for two source-destination pairs (s_1, t_1), (s_2, t_2), which is known to be $\mathcal{NP}$-hard [20], can be transformed into an instance of ODP for s, t by adding nodes s, t and arcs $\{(s, s_1), (t_2, s), (t_1, t), (t, s_2)\}$. We conclude that ODP as well as the pricing problem for the more general case of customers $k \in C_2$ with $b_{\max}(k) > 0$ are $\mathcal{NP}$-hard.

While, several algorithms for solving the directed disjoint pair of paths problem have been proposed for special cases such as planar graphs or dual arc costs [23], its general case has gained surprisingly few consideration so far. In the following we present two alternatives for solving our pricing problem: First, by mixed integer programming and second by modeling it as an elementary shortest path problem with resource constraints.

5.1 Solving the Pricing Problem by Mixed Integer Programming

We solve the pricing problem for each customer $k \in C_2$ using the MIP (25)–(38), where $\mathcal{A}(k) = \{(u,v) \in A \mid u,v \in \mathcal{B}(k)\}$ denotes the set of potential edges in the customer's branch line. Each feasible connection is represented by a directed cycle containing r and at least one potential branching node $w \in \mathcal{B}(k)$ and a path from r to k using arcs not on this cycle for the branch line only. The directed cycle containing r and the finally selected branch node is described by variables $q_{u,v} \in \{0,1\}$, $\forall (u,v) \in A$. Variables $s_{u,v} \in \{0,1\}$, $\forall (u,v) \in A$, indicate whether an arc is part of the non-redundant path from the root to k, while variables $b_{u,v} \in \{0,1\}$, $\forall (u,v) \in \mathcal{A}(k)$, denote whether an arc is part of the connection's branch line, i.e. those arcs that are on the non-redundant path described by variables $s_{u,v}$ but not on the cycle described by variables $q_{u,v}$.

$$\min \sum_{(u,v)\in A} \omega_{k,u,v} q_{u,v} + \sum_{(u,v)\in \mathcal{A}(k)} \omega_{k,u,v} b_{u,v} \tag{25}$$

$$\text{s.t.} \sum_{(u,v)\in A} q_{u,v} - \sum_{(v,w)\in A} q_{v,w} = 0 \qquad \forall v \in V \tag{26}$$

$$\sum_{(r,v)\in A} q_{r,v} = 1 \tag{27}$$

$$q_{u,v} + q_{v,u} \le 1 \qquad \forall (u,v) \in E \tag{28}$$

$$\sum_{(u,v)\in A} q_{u,v} \le 1 \qquad \forall v \in V \setminus \{r\} \tag{29}$$

$$\sum_{v\in \mathcal{B}(k)} \sum_{(u,v)\in A} q_{u,v} \ge 1 \tag{30}$$

$$\sum_{(u,v)\in A} s_{u,v} - \sum_{(v,w)\in A} s_{v,w} = \begin{cases} -1 & \text{if } v = r \\ 1 & \text{if } v = k \\ 0 & \text{otherwise} \end{cases} \qquad \forall v \in V \tag{31}$$

$$s_{u,v} + s_{v,u} \le 1 \qquad \forall (u,v) \in E \tag{32}$$

$$s_{u,v} \le q_{u,v} \qquad \forall (u,v) \in A \setminus \mathcal{A}(k) \tag{33}$$

$$b_{u,v} \ge s_{u,v} - q_{u,v} \qquad \forall (u,v) \in \mathcal{A}(k) \tag{34}$$

$$\sum_{(u,v)\in \mathcal{A}(k)} l_{u,v} b_{u,v} \le b_{\max}(k) \tag{35}$$

$$(36) \qquad q_{u,v} \in \{0,1\} \qquad \forall (u,v) \in A$$

$$(37) \qquad s_{u,v} \in \{0,1\} \qquad \forall (u,v) \in A$$

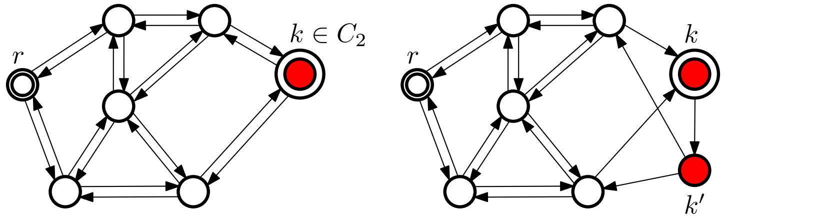
$$(38) \qquad 0 \leq b_{u,v} \leq 1 \qquad \forall (u,v) \in \mathcal{A}(k)$$

The flow conservation constraints (26) ensure that the arcs (u,v) on which $q_{u,v} = 1$ form a directed cycle. Constraints (28) avoid the simultaneous usage of two oppositely directed arcs and constraints (29) prevent the repetition of nodes on the cycle. These constraints, in conjunction with constraints (27) and (30) which force the cycle to contain r and at least one potential branch node, ensure that the final cycle corresponds to two oppositely directed, internally node disjoint paths between r and some branch node. Due to the flow conservation constraints (31) together with constraints (32), variables $s_{u,v}$, $\forall(u,v) \in A$, describe a directed path from r to k. Furthermore, constraints (33) force this path to use arcs part of the above mentioned cycle outside the $b_{\max}$-neighborhood of k. Finally, constraints (34) ensure that variables $b_{u,v}$, $\forall(u,v) \in \mathcal{A}(k)$, indicate the arcs forming the branch line, whereas constraints (35) restrict the branch line's length. For variables $b_{u,v}$, it suffices to define lower and upper bounds in (38) as they will automatically become integral in feasible solutions.

5.2 Modeling the Pricing Problem as an Elementary Shortest Path Problem with Resource Constraints

Without yet considering $b_{\max}$-redundancy, the pricing problem for a customer $k \in C_2$ can be interpreted as finding a cheapest cycle containing r and k. Finding negative cost cycles is a problem which frequently occurs as pricing problem in branch-and-price approaches from the context of vehicle routing and crew scheduling. There, algorithms for solving the (elementary) shortest path problem with resource constraints (ESPPRC) are frequently used for solving the pricing subproblem, see e.g. [26]. As a consequence, ESPPRC which is $\mathcal{NP}$-hard, has recently gained great attention and several methods for solving it have been proposed [19, 10].

We transform the pricing subproblem for a customer $k \in C_2$ into an instance of the ESPPRC on graph $G'_k = (V'_k, A'_k)$ with the root node r being the source and destination node. The transformed graph – see Figure 8 for an example – is defined by its node set $V'_k = V \cup \{k'\}$ and its arc set $A'_k = \{(u,v) \in A \mid u \neq k\} \cup \{(k',v) \mid \exists (k,v) \in A\} \cup \{(k,k')\}$. Here, we augment the node set by a duplicate of k, called k', and connect these two nodes by an arc (k,k'). Each arc $(k,v) \in A$ emanating from k, is replaced by an arc $(k',v) \in A'_k$ going out from k'. We call k' the *split node* of k while we refer to arc (k,k') as *split arc* of k. Since k has only one outgoing arc, each non-trivial path in G'_k containing k which does not end at node k must also contain the split arc (k,k').



a) Original graph: $G = (V, A)$ b) Transformed graph: $G' = (V', A')$

Figure 8: Transformation to ESPPRC for $k \in C_2$, with $b_{\max}(k) = 0$.

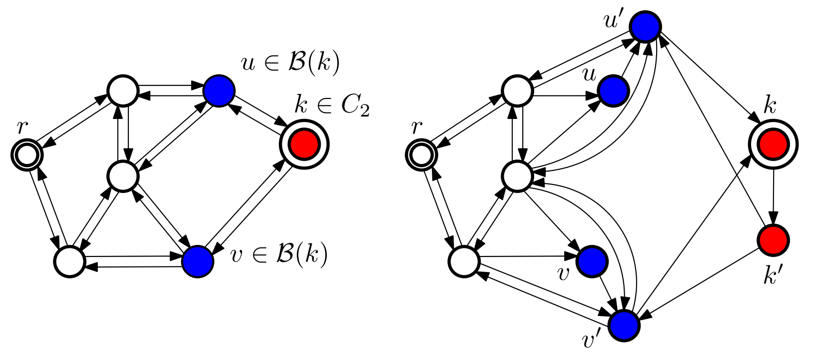
Arc costs $c'_{u,v}$, are defined as

$$c'_{u,v} = \begin{cases} -\nu_k & \text{if } u = k \text{ and } v = k' \\ \omega_{k,k,v} & \text{if } u = k' \\ \omega_{k,u,v} & \text{otherwise} \end{cases} \qquad \forall (u,v) \in A'_k.$$

As $(k, k') \in A'_k$ is the only arc with negative costs $c'_{k,k'} = -\nu_k$ in $G'_k = (V'_k, A'_k)$ and each pair of oppositely directed internally node-disjoint paths between r and k must have costs smaller than ν_k to price out favorably, we conclude that there is a one-to-one correspondence between the set of elementary shortest paths from r to itself with negative costs in $G'_k = (V'_k, A'_k)$ and the set of oppositely directed internally node-disjoint paths between r and k yielding negative reduced costs. As discussed by Boland et al. [8], node disjointness can be ensured by additionally adding one resource for each node $v \in V'_k$ with a maximum resource consumption of one for each individual node resource.

Next, we slightly adapt the above described transformation, in order to generalize it to the case of type-2 customer nodes $k \in C_2$ with $b_{\max}(k) > 0$. We split each potential branch node $v \in \mathcal{B}(k)$ into nodes v and v' and add an arc (v, v') between each of those pairs. In case a path in G'_k corresponding to a feasible connection between r and k uses an arc between some potential branch node v and its split node v', v will be the branch node of the resulting connection. Since each potential branch node $v \in \mathcal{B}(k)$ except k can be used either as a connection's branch node or as a standard node of a connection to k, G'_k contains arcs (u, v) and (u, v') for each arc $(u, v) \in A$, $v \in \mathcal{B}(k)$, where $v \neq k$. Arcs $(v, w) \in A$ going out from $v \in \mathcal{B}(k)$ are replaced by arcs $(v', w) \in A'_k$. Formally the transformed graph $G'_k = (V'_k, A'_k)$ is thus defined by its node set $V'_k = V \cup \{v' \mid \exists v \in \mathcal{B}(k)\}$ and its arc set $A'_k = \{(u,v) \in A \mid u \neq \mathcal{B}(k)\} \cup \{(u', v) \mid \exists (u,v) \in A \wedge u \in \mathcal{B}(k)\} \cup \{(u, u') \mid u \in \mathcal{B}(k)\} \cup \{(u, v') \mid \exists (u,v) \in A \wedge v \in \mathcal{B}(k) \wedge v \neq k\}$, see Figure 9.

Let $\hat{c}_u \geq 0$, $\forall u \in \mathcal{B}(k)$, denote the costs of the precomputed branch line between u and k when using node u as branch node of the connection between r and k with respect to arc costs $\omega_{k,u,w}$, $\forall (u,v) \in \mathcal{A}(k)$.



a) Original graph: $G = (V, A)$ b) Transformed graph: $G' = (V', A')$

Figure 9: Transformation to ESPPRC for $k \in C_2$, with $b_{\max}(k) > 0$.

Then arc costs $c'_{u,v}$ are defined as

$$c'_{u,v} = \begin{cases} -\nu_k + \hat{c}_u & \text{if } u \in \mathcal{B}(k) \\ \omega_{k,u,w} & \text{if } v \text{ is the split node of } w \\ \omega_{k,w,v} & \text{if } u \text{ is the split node of } w \\ \omega_{k,u,v} & \text{otherwise} \end{cases} \qquad \forall (u,v) \in A'_k.$$

Since only split arcs $(u, u') \in A'_k$, $\forall u \in \mathcal{B}(k)$, might eventually have negative costs $c'_{u,u'} = -\nu_k + \hat{c}_u$, and due to the above introduced transformation, there is a one-to-one correspondence between the set of feasible connections $p \in \mathcal{H}_k$ that price out favorably and the set of elementary shortest paths from r to itself with negative costs in $G'_k = (V'_k, A'_k)$ using exactly one split arc.

Thus, by associating a resource of value one to each split arc, we can model the pricing problem for customer $k \in C$ as an elementary shortest path problem with resource constraints (ESPPRC) with a maximum resource consumption of one. Furthermore, above mentioned node resources for ensuring node disjointness need to be additionally considered.

5.3 Analyzing the Restricted Dual Problem

In accordance with Section 4.1, we analyze the dual problem of $\text{(dCol)}^{\text{RMP}}$ to see whether we may expect the same issues as for model (Col) when solving the linear relaxation of (dCol). If this is the case, we are interested if we can pursue a similar stabilization approach as proposed for the undirected model in Section 4.2.

Let $\gamma_e \leq 0$, $\forall e \in E$, denote the dual variable values associated to constraints (20) and $\rho_k \leq 0$, $\forall k \in C$, denote the dual variable values associated to constraints (22). Then the restricted dual problem – i.e. the dual of the restricted master problem $\text{(dCol)}^{\text{RMP}}$ – for the SST variant of $b_{\max}$-SNDP is given by formulation

(39)–(46).

$$
\begin{aligned}
&(39) && \max \quad \sum_{k\in C} \rho_k + p_k + \sum_{e\in E} \gamma_e && \\
&(40) && \sum_{k\in C} \omega_{k,u,v} + \gamma_e \leq c_{u,v} && \forall (u,v)\in A,\ e=(u,v)\in E \\
&(41) && \nu_k - \sum_{(u,v)\in p} \omega_{k,u,v} \leq 0 && \forall k\in C,\ \forall p\in \mathcal{H}_k | \exists h_p^k \in \tilde{H} \\
&(42) && -\nu_k + \rho_k \leq -p_k && \forall k\in C \\
&(43) && \omega_{k,u,v} \geq 0 && \forall k\in C,\ \forall (u,v)\in A \\
&(44) && \nu_k \geq 0 && \forall k\in C \\
&(45) && \gamma_e \leq 0 && \forall e\in E \\
&(46) && \rho_k \leq 0 && \forall k\in C
\end{aligned}
$$

Let $A'' = \{(u,v)\in A \mid \nexists h_p^k \in \tilde{H} : (u,v)\in p\}$ denote the set of arcs not included in any connection of $(\mathrm{dCol})^{\mathrm{RMP}}$. As only inequalities (40) are relevant for arcs $(u,v)\in A''$ and $\gamma_e \leq 0$, $\forall e\in E$, any variable values $\omega_{k,u,v}\geq 0$, $\forall k\in C$, $\forall (u,v)\in A''$, $e=(u,v)\in E$, are optimal with respect to model (39)–(46) as long as $\sum_{k\in C}\omega_{k,u,v} \leq c_{u,v} - \gamma_e$. In particular, it is easy to see that if $(u,v),(v,u)\in A''$, an optimal solution with $\gamma_e = 0$, $e=(u,v)\in E$, and $\omega_{k,u,v} = \omega_{k,v,u} = 0$, $\forall k\in C$, does exist. Thus, next to the issue of degeneracy based upon the fact that only few arc and connection variables will be nonzero in any solution to $(\mathrm{dCol})^{\mathrm{RMP}}$, we observe that edge costs used in the pricing subproblems are in general not meaningful.

5.4 Alternative Dual-Optimal Solutions

Let $(\gamma^*, \rho^*, \nu^*, \omega^*)$ be an optimal solution to the restricted dual problem (39)–(46). As motivated for the undirected model in Section 4.2 we focus on increasing dual variable values ω^* used as arc costs in the pricing problem. Thus, the costs for individual connections will rise and we expect that less connections are finally included in $(\mathrm{dCol})^{\mathrm{RMP}}$. As inequalities (40) are the only constraints imposing upper bounds for dual variables ω, we only need to consider these constraints when increasing the values.

Let $\delta_{u,v} = c_{u,v} - \gamma_e - \sum_{k\in C}\omega_{k,u,v}$, $\forall (u,v)\in A$, $e=(u,v)\in E$, denote the total amount by which we may increase the sum of dual variable values ω on arc (u,v). It is easy to see that $\delta_{u,v} = c_{u,v}$, $\forall (u,v)\in A : (u,v),(v,u)\in A''$, where $A''\subseteq A$ denotes the subset of arcs which are not part of any so far included connection[2]. Generally, $\delta_{u,v}$ will also be greater than zero at least for some arcs $(u,v)\in A\setminus A''$.

[2]Since CPLEX [25] will compute dual variable values equal to zero in this case.

As in Section 4.2 we pursue four strategies for generating alternative dual optimal solutions. For $D' = (\gamma^*, \rho^*, \nu^*, \omega')$, we set $\omega'_{k,u,v} = \omega^*_{k,u,v} + \frac{\delta_{u,v}}{|C|}$, $\forall k \in C$, $\forall (u,v) \in A$. As $\sum_{k \in C} \omega'_{k,u,v} - \omega^*_{k,u,v} = \sum_{k \in C} \frac{\delta_{u,v}}{|C|} = \delta_{u,v}$, $\forall (u,v) \in A$, D' is feasible for the restricted dual problem in the OPT as well as in the SST case. Furthermore, since the objective value does not change due to our adaptation, D' is dual optimal.

We further apply the parameterized approach, where dual optimal solutions $D^{(k,d)} = (\gamma^*, \rho^*, \nu^*, \omega^{(k,d)})$ with $\omega^{(k,d)}_{k,u,v} = \omega^*_{k,u,v} + \frac{\delta_{u,v}}{d}$, $\forall k \in C$, $\forall (u,v) \in A$, are used. Here, we initialize d to be equal to one and gradually increment d up to $|C|$ if no column could be priced in and reset d to one in case a column including new arcs has been added. Also, we consider the simpler variant where d is immediately set to $|C|$ if no connection variable prices out favorably and d will not be decremented anymore. We refer to the corresponding dual optimal solutions by $D^{(k,d')}$. All above mentioned strategies are valid for both, the SST as well as the OPT variant of our problem. Finally in our last strategy which is feasible for the SST variant only we use dual optimal solutions $D^{(p)} = (\gamma^*, \rho^*, \nu^*, \omega^{(p)})$ with $\omega^{(p)}_{k,u,v} = \omega^*_{k,u,v} + \delta_{u,v} \frac{p_k}{\sum_{l \in C} p_l}$, $\forall k \in C$, $\forall (u,v) \in A$.

6 Polyhedral Comparison

In this section, we theoretically compare the undirected and directed connection formulation to each other as well as to previous formulations introduced by Wagner et al. [45, 44] based on multi-commodity flows [45] and connectivity cuts [44], respectively. Hereby, we denote by $\mathcal{P}_{\mathrm{col}}$ the polyhedron corresponding to the set of feasible solutions to the linear relaxation of model (Col). Similarly, $\mathcal{P}_{\mathrm{dcol}}$ denotes the polyhedron induced by the LP relaxation of model (dCol), $\mathcal{P}_{\mathrm{mcf}}$ those of the multi-commodity flow (MCF) formulation from [45], and $\mathcal{P}_{\mathrm{cut}}$ the polyhedron corresponding to the cut formulation from [44]. By $proj_{x,y}(\mathcal{P})$ we refer to the projection of a polyhedron $\mathcal{P}$ into the space of x and y variables only. As a prerequisite, we are also reviewing the MCF and cut formulations in this section.

In their MCF formulation, Wagner et al. [45] used arc set $A_r = \{(r,j) \in E \mid j \in S\}$ denoting all arcs connecting r with Steiner nodes, the set of edges $E_S(k) = \{(i,j) \mid i,j \in V \setminus \{r,k\}\}$ connecting two Steiner nodes with respect to customer $k \in C$, as well as the corresponding arc set $A_S(k) = \{(i,j),\ (j,i) \mid i,j \in V \setminus \{r,k\}\}$. Furthermore, $A(k) = \{(i,k) \mid (i,j) \in E\}$ denotes the set of arcs to customer $k \in C$, and $A'(k) = A_r \cup A_S(k) \cup A(k)$ the set of all arcs relevant for a customer k. Finally, $B(k)$ denotes the set of arcs $(i,j) \in A'(k)$, with $i,j \in \mathcal{B}(k)$, i.e. those arcs that are potentially used in a branch line of a connection to customer k. In formulation (47)–(65) introduced by Wagner et al. [45] variables $x_{i,j} \in \{0,1\}$, $\forall (i,j) \in E$, indicate whether edge (i,j) is used ($x_{i,j} = 1$) in a solution or not ($x_{i,j} = 0$). Flow variables $0 \leq m^k_{i,j} \leq 1$, $\forall k \in C$, $\forall (i,j) \in A'(k)$, and $0 \leq n^k_{i,j} \leq 1$, $\forall k \in C_2$, $(i,j) \in A'(k)$, model the connection to a customer node k. Here, the second set of flow variables is used to achieve redundancy for type-2 customers. Variables $0 \leq q^k_{i,j} \leq 1$, $\forall k \in C_2$, $\forall (i,j) \in B(k)$, indicate the edges used in the branch line to node k. Finally, variables

$y_k \in \{0,1\}$, $\forall k \in C$, indicate in the SST variant whether customer node k is connected ($y_k = 1$) or not ($y_k = 0$). In the OPT variant, these variables are fixed to one. Using these sets and variables, $b_{\max}$-SNDP is stated by the following MIP:

$$\text{(MCF)} \quad \min \sum_{(i,j)\in E} c_{i,j} x_{i,j} + \sum_{k\in C} p_k (1 - y_k) \tag{47}$$

$$\text{s.t.} \quad \sum_{(i,j)\in A'(k)} m^k_{i,j} - \sum_{(j,i)\in A'(k)} m^k_{j,i} = \begin{cases} -y_k & \text{if } j = r \\ y_k & \text{if } j = k \\ 0 & \text{otherwise} \end{cases} \qquad \forall k \in C,\ \forall j \in V \tag{48}$$

$$\sum_{(i,j)\in A'(k)} n^k_{i,j} - \sum_{(j,i)\in A'(k)} n^k_{j,i} = \begin{cases} -y_k & \text{if } j = r \\ y_k & \text{if } j = k \\ 0 & \text{otherwise} \end{cases} \qquad \forall k \in C_2,\ \forall j \in V \tag{49}$$

$$m^k_{i,j} \le x_{i,j} \qquad \forall k \in C,\ \forall (i,j) \in A_r \cup A(k) \tag{50}$$

$$m^k_{i,j} + m^k_{j,i} \le x_{i,j} \qquad \forall k \in C,\ \forall (i,j) \in E_S(k) \tag{51}$$

$$n^k_{i,j} \le x_{i,j} \qquad \forall k \in C_2,\ \forall (i,j) \in A_r \cup A(k) \tag{52}$$

$$n^k_{i,j} + n^k_{j,i} \le x_{i,j} \qquad \forall k \in C_2,\ \forall (i,j) \in E_S(k) \tag{53}$$

$$m^k_{i,j} + n^k_{j,i} \le x_{i,j} \qquad \forall k \in C_2,\ \forall (i,j) \in A_S(k) \tag{54}$$

$$m^k_{i,j} + n^k_{i,j} \le x_{i,j} \qquad \forall k \in C_2,\ \forall (i,j) \in A'(k) \setminus B(k) \tag{55}$$

$$m^k_{i,j} + n^k_{i,j} - q^k_{i,j} \le x_{i,j} \qquad \forall k \in C_2,\ \forall (i,j) \in B(k) \tag{56}$$

$$q^k_{i,j} \le m^k_{i,j} \qquad \forall k \in C_2,\ \forall (i,j) \in B(k) \tag{57}$$

$$q^k_{i,j} \le n^k_{i,j} \qquad \forall k \in C_2,\ \forall (i,j) \in B(k) \tag{58}$$

$$\sum_{(i,j)\in B(k)} (m^k_{i,j} + n^k_{i,j} - q^k_{i,j}) + {} + \sum_{(i,j)\in A'(k)\setminus B(k)} (m^k_{i,j} + n^k_{i,j}) \le 1 \qquad \forall k \in C_2,\ \forall i \in V \setminus \{r,k\}, \tag{59}$$

$$\sum_{(i,j)\in B(k)} l_{i,j} q^k_{i,j} \le b_{\max}(k) \qquad \forall k \in C_2 \tag{60}$$

$$x_{i,j} \in \{0,1\} \qquad \forall (i,j) \in E \tag{61}$$

$$y_k \in \{0,1\} \qquad \forall k \in C \tag{62}$$

$$0 \le m^k_{i,j} \le 1 \qquad \forall k \in C,\ \forall (i,j) \in A'(k) \tag{63}$$

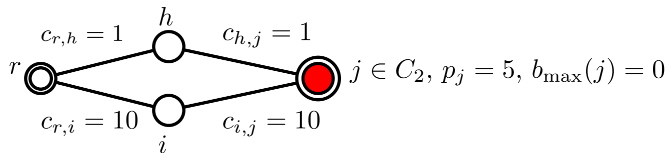


Figure 10: An exemplary instance of $b_{\max}$-SNDP with a single customer node.

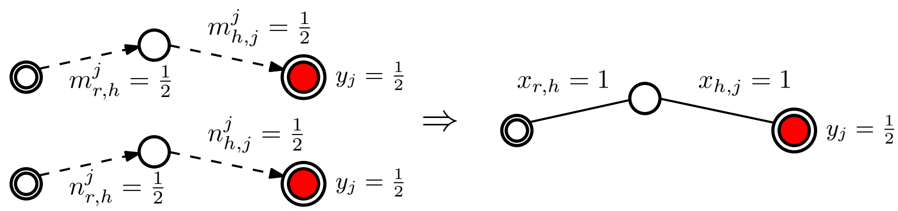


Figure 11: A feasible solution of $\mathcal{P}_{\text{mcf}}$ for the instance given in Figure 10.

$$0 \leq n^k_{i,j} \leq 1 \qquad \forall k \in C_2,\ \forall (i,j) \in A'(k) \tag{64}$$

$$0 \leq q^k_{i,j} \leq 1 \qquad \forall k \in C_2,\ \forall (i,j) \in B(k) \tag{65}$$

Lemma 1 *The multi-commodity flow formulation (47)–(65) from [45] does not dominate* (Col), *i.e.* $proj_{x,y}(\mathcal{P}_{\text{mcf}}) \not\subseteq proj_{x,y}(\mathcal{P}_{\text{col}})$.

Proof Consider the instance of $b_{\max}$-SNDP illustrated in Figure 10. Obviously, the optimal solution to $(\text{Col})^{\text{LP}}$ does not connect customer $j \in C_2$ since it does not pay off, i.e. all variables will be set to zero and thus the objective value is equal to five. $\mathcal{P}_{\text{mcf}}$, however, does contain the solution depicted in Figure 11, where both types of flows – i.e. m and n – to $j \in C_2$ each of which of value 0.5 are routed over the same arcs. Thus, by setting $y_j = 0.5$ and the resulting edge variables $x_{r,h}$ and $x_{h,j}$ to one, the costs for connecting customer j in such a way are lower than the resulting profit. The objective value of the solution depicted in Figure 11 is equal to 4.5.

Lemma 2 *Let k be an arbitrary customer $k \in C$ connected in some – potentially fractional – solution $G' \in \mathcal{P}_{\text{col}}$ and y_k denote its variable value in G'. Furthermore, let $x_e \geq \sum_{p \in \mathcal{F}_k | e \in p} f^k_p$, $\forall e \in E$, denote the values of all edge variables induced by the (fractional) connections to k due to constraints (5).*
Then, variable values x_e, $\forall e \in E$, allow for describing a feasible connection to customer k of value y_k in $\mathcal{P}_{\text{mcf}}$.

Proof Let $f_p^k \in F_k$ be an arbitrary connection variable corresponding to connection $p \subseteq \mathcal{F}_k$. Since f_k contains only feasible connections to customer k, we can derive a feasible flow $(m^k,\, n^k)$ of value f_p^k by orienting each edge $e \in p$ towards k. Furthermore, as the length constraints with respect to the branch line are met by the definition of f_k, we conclude that each undirected connection $p \in f_k$ can be represented as a set of flow variables corresponding to a feasible connection of value f_p^k in model (47)–(65).
As $\sum_{p \in \mathcal{F}_k | e \in p} f_p^k \leq x_e$, $\forall e \in E$, holds due to constraints (5), we can simply use above mentioned equivalence between connections and feasible flows for each connection individually, yielding feasible flow variable values $(m^k,\, n^k)$ that allow for setting the customer variable y_k to $\sum_{p \in \mathcal{F}_k} f_p^k \geq y_k$ in model (47)–(65).

Theorem 3 *The undirected connection formulation* (Col) *strictly dominates the multi-commodity flow formulation (47)–(65) from [45], i.e.* $proj_{x,y}(\mathcal{P}_{\mathrm{col}}) \subsetneq proj_{x,y}(\mathcal{P}_{\mathrm{mcf}})$.

Proof Due to Lemma 1 it is enough to show that any feasible solution to $(\mathrm{Col})^{\mathrm{LP}}$, can be projected into a feasible solution of formulation (47)–(65) with identical objective value, i.e. with identical values for x_e, $\forall e \in E$, and y_k, $\forall k \in C$. Since no constraint of formulation (47)–(65) considers multiple customers simultaneously, we can take into account each customer individually. Thus, Theorem 3 follows due to Lemma 2.

In their second formulation based on connectivity cuts, Wagner et al. [44] used variables $x_{i,j} \in \{0, 1\}$, $\forall (i, j) \in E$, indicating edges being part of a solution. Variables $a_{i,j} \in \{0, 1\}$, $\forall (i, j) \in A_D$, are defined on the arc set A_D, containing one arc going out of r and two oppositely directed arcs for the remaining edges. Variables $y_k \in \{0, 1\}$, $\forall k \in C$, which are fixed to one in the OPT variant, specify whether a customer k is connected or not. Binary variables $z_i \in \{0, 1\}$, $\forall i \in V$, indicate whether a node i has two node-disjoint paths to r and variables $b_j^k \in \{0, 1\}$, $\forall k \in C_2$, $\forall j \in \mathcal{B}(k)$, denote whether j is the branching node of customer k. Finally, variables $q_{i,j}^k \in \{0, 1\}$, $\forall k \in C_2$, $\forall (i, j) \in B(k)$, describe the branch line of the connection to customer k.

$$\text{(66)} \qquad (\mathrm{Cut}) \quad \min \sum_{(i,j) \in E} c_{i,j} x_{i,j} + \sum_{k \in C} p_k (1 - y_k)$$

$$\text{(67)} \qquad \text{s.t.} \quad a(\delta^-(S)) \geq y_k \qquad \forall k \in C,\ \forall S \subseteq V \setminus \{r\} \mid k \in S$$

$$\text{(68)} \qquad a(\delta^-(S)) \geq 2 z_i \qquad \forall i \in V \setminus \{r\},\ \forall S \subseteq V \setminus \{r\} \mid i \in S$$

$$\text{(69)} \qquad a(\delta^-_{V \setminus \{v\}}(S)) \geq z_i \qquad \forall i \in V \setminus \{r\},\ \forall v \in V \setminus \{r, i\},\ \forall S \subseteq V \setminus \{r, v\} \mid i \in S$$

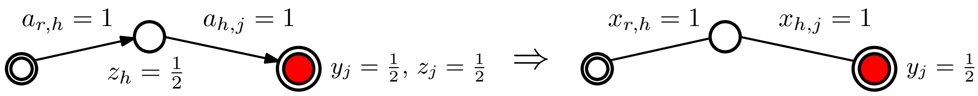


Figure 12: A feasible solution of $\mathcal{P}_{\text{cut}}$ for the instance given in Figure 10.

$$\sum_{j \in \mathcal{B}(k)} b_j^k = y_k \qquad k \in C_2 \tag{70}$$

$$b_j^k \leq z_j \qquad \forall k \in C_2, \forall j \in \mathcal{B}(k) \tag{71}$$

$$q_{i,j}^k \leq a_{i,j} \qquad \forall k \in C_2,\ \forall (i,j) \in B(k) \tag{72}$$

$$q^k(\delta^-_{\mathcal{B}(k)}(S)) \geq b_j^k \qquad \forall k \in C_2,\ \forall j \in \mathcal{B}(k) \setminus \{k\}, \quad \forall S \subseteq \mathcal{B}(k) \setminus \{j\} \mid k \in S \tag{73}$$

$$\sum_{(i,j) \in B(k)} l_{i,j} q_{i,j}^k \leq b_{\max}(k) \qquad \forall k \in C_2 \tag{74}$$

$$a_{i,j} \leq x_{i,j} \qquad \forall (i,j) \in A_D \tag{75}$$

$$a_{j,i} \leq x_{i,j} \qquad \forall (i,j) \in A_D \tag{76}$$

$$x_{i,j} \in \{0,1\} \qquad \forall (i,j) \in E \tag{77}$$

$$a_{i,j} \in \{0,1\} \qquad \forall (i,j) \in A_D \tag{78}$$

$$y_k \in \{0,1\} \qquad \forall k \in C \tag{79}$$

$$z_i \in \{0,1\} \qquad \forall i \in V \tag{80}$$

$$b_j^k \in \{0,1\} \qquad \forall k \in C_2,\ \forall j \in \mathcal{B}(k) \tag{81}$$

$$q_{i,j}^k \in \{0,1\} \qquad \forall k \in C_2,\ \forall (i,j) \in B(k) \tag{82}$$

Lemma 4 *The cut formulation (66)–(82) from [44] does not dominate* (Col)*, i.e. $proj_{x,y}(\mathcal{P}_{\text{cut}}) \not\subseteq proj_{x,y}(\mathcal{P}_{\text{col}})$.*

Proof Consider the instance given in Figure 10 with an optimal solution value to the LP relaxation of (Col) equal to five, where all variable values are set to zero. As the multi-commodity flow formulation, the cut model (66)–(82) does allow for "half-connecting" customer $j \in C_2$ via a single path where the corresponding arc and edge variables are set to one, see Figure 12.

Lemma 5 *Let k be an arbitrary customer $k \in C$ connected in some – potentially fractional – solution $G' \in \mathcal{P}_{\text{col}}$ and y_k denote its variable value in G'. Furthermore, let $x_e \geq \sum_{p \in \mathcal{F}_k | e \in p} f_p^k$, $\forall e \in E$, denote the* *variable values of all edge variables induced by the (fractional) connections to k due to constraints (5). Then, variables values x_e, $\forall e \in E$, allow for describing a feasible connection of value y_k in $\mathcal{P}_{\text{cut}}$.*

Proof As already observed by Chimani et al. [13] model (66)–(82) uses directed variables $a_{i,j}$, $\forall (i,j) \in A_D$, but is equivalent to an undirected model since constraints (75) and (76) allow for simultaneously using oppositely directed arcs corresponding to a single edge without increasing the cost function, i.e. $a_{i,j} = a_{j,i} = x_{i,j}$, $\forall (i,j) \in E$, $i \neq r$, $j \neq r$.

Let $p \in \mathcal{F}_k$ be an arbitrary connection to customer k. Due to constraints (5), p induces variable values $x_e^{(\mathrm{p})} \geq f_p^k$. By definition of $\mathcal{F}_k$, p is a feasible connection to k and thus setting $a_{i,j} = a_{j,i} = x_e^{(\mathrm{p})}$, $\forall e \in p$, allows for supplying customer k with a value of f_p^k in model (66)–(82). Due to constraints (4), $y_k \leq \sum_{p \in F_k} f_p^k$ holds and thus Lemma 5 follows.

Theorem 6 *The undirected connection formulation* (Col) *strictly dominates the cut formulation (66)–(82) from [44], i.e. $proj_{x,y}(\mathcal{P}_{\text{col}}) \subsetneq proj_{x,y}(\mathcal{P}_{\text{cut}})$.*

Proof Since model (66)–(82) considers each customer individually, Theorem 6 follows due to Lemmas 4 and 5.

Theorem 7 *The directed connection formulation* (dCol) *strictly dominates the undirected variant* (Col)*, i.e. $proj_{x,y}(\mathcal{P}_{\text{dcol}}) \subsetneq proj_{x,y}(\mathcal{P}_{\text{col}})$.*

Proof It is easy to see that $proj_{x,y}(\mathcal{P}_{\text{dcol}}) \subseteq proj_{x,y}(\mathcal{P}_{\text{col}})$ holds, if $proj_{x,y}(\mathcal{P}_{\text{dcol}})$ denotes the obvious projection of $\mathcal{P}_{\text{dcol}}$ into the space of $\mathcal{P}_{\text{col}}$, i.e. $x_e = a_{i,j} + a_{j,i}$, $\forall e = (i,j) \in E$.

Consider the instance given in Figure 13 and the optimal solution G'_{col} of $(\text{Col})^{\text{LP}}$ to this instance as shown in Figure 14. Here, each type-1 customer is connected via two connections. The corresponding edges are also used for connecting the type-2 customer $j \in C_2$. Thus, it is possible to set $y_h = y_i = 1$ and $y_j = 0.5$, while all edge variables are set to 0.5. The objective value of the shown solution is $o(G'_{\text{col}}) = 6.5$. On the other hand, the optimal oriented solution G'_{dcol} of $(\text{dCol})^{\text{LP}}$ does not connect any customers, i.e. $y_k = 0$, $\forall k \in \{h, i, j\}$, and $a_{u,v} = 0$, $\forall (u,v) \in A$. We conclude that (dCol) strictly dominates its undirected variant, i.e. $proj_{x,y}(\mathcal{P}_{\text{dcol}}) \subsetneq proj_{x,y}(\mathcal{P}_{\text{col}})$.

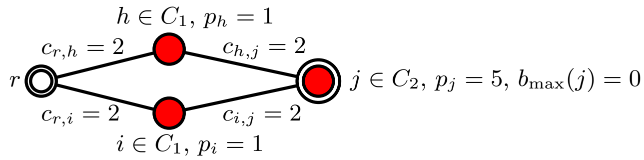


Figure 13: Another exemplary instance of $b_{\max}$-SNDP.

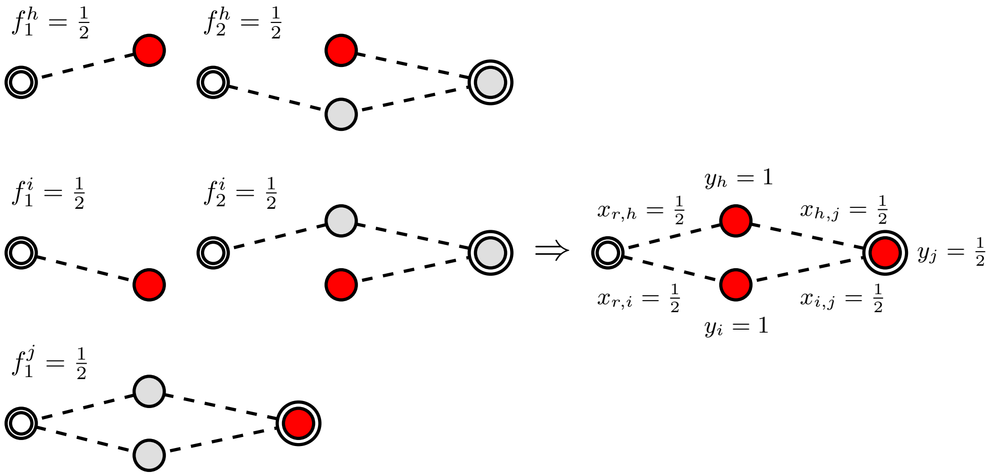


Figure 14: A feasible solution of $\mathcal{P}_{\mathrm{col}}$ for the instance given in Figure 13.

7 Test Instances and Environment

Real world instances from a German city [2] have been used to test our approaches. Table 1 details the characteristics of the five instance sets, listing the number of instances (#), the graph size, numbers of customers, and for all considered values of $b_{\max}$ the number of potential branching nodes and edges. Note that in each experiment the value of $b_{\max}$ is identical for each type-2 customer node and we write $b_{\max} = 30$ instead of $b_{\max}(k) = 30$, $\forall k \in C_2$, in the following.

All computational experiments have been performed on a single core of an Intel Xeon E5540 with 2.53GHz. IBM CPLEX 12.1 [25] has been used to solve the multi-commodity flow model (MCF) from Wagner et al. [45] as well as its linear relaxation $(\mathrm{MCF})^{\mathrm{LP}}$. We used SCIP 1.2.0 [1, 48] with IBM CPLEX 12.1 [25] as embedded LP solver for solving (Col) and (dCol) as well as their linear relaxations $(\mathrm{Col})^{\mathrm{LP}}$ and $(\mathrm{dCol})^{\mathrm{LP}}$, respectively.

Table 1: Instance set characteristics.

										$b_{\max} = 30$		$b_{\max} = 50$		$b_{\max} = 100$	
Set	#	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert C\rvert$	$\overline{\lvert C\rvert}$	$\lvert C_1\rvert$	$\overline{\lvert C_1\rvert}$	$\lvert C_2\rvert$	$\overline{\lvert C_2\rvert}$	$\overline{\mathcal{B}(k)}$	$\overline{\mathcal{E}(k)}$	$\overline{\mathcal{B}(k)}$	$\overline{\mathcal{E}(k)}$	$\overline{\mathcal{B}(k)}$	$\overline{\mathcal{E}(k)}$
ClgSE-I1	25	190	377	5-8	5.9	3-5	3.8	2-3	2.1	12.93	17.07	26.23	41.43	86.54	144.30
ClgSE-I2	15	190	377	11-17	13.8	7-12	8.9	4-7	4.9	10.23	12.58	29.47	42.33	131.01	235.36
ClgSE-I3	15	190	377	8-12	9.6	5-8	6.0	3-6	3.6	10.62	13.29	32.84	47.51	119.54	210.67
ClgN1B-I1	20	2804	3082	11-14	11.8	8-11	8.5	3-4	3.3	10.47	9.82	23.20	24.10	68.89	77.59
ClgN1B-I2	19	2804	3082	7-11	9.0	3-6	4.1	4-6	5.0	7.92	6.94	15.04	14.10	40.53	40.24

Note that CPLEX 12.1 is roughly three times faster than SCIP 1.2.0 with CPLEX 12.1 as an embedded LP solver[3]. An absolute time limit of 7200 CPU-seconds has been used for all experiments.

8 Computational Results

In the following we summarize all obtained computational results. First, a detailed comparison of the proposed exact models (Col), (dCol), and the MCF formulation of Wagner et al. [45] is given.

Since preliminary tests indicated identical trends for the OPT and SST variants of $b_{\max}$-SNDP, we primarily concentrate on the more complex SST in the following. Furthermore, to analyze the influence of the size of the $b_{\max}$-neighborhood, we consider different values for $b_{\max}$ for each instance set.

When solving (Col) we initialize $\tilde{F}$ by all variables corresponding to connections obtained by applying the minimum spanning tree augmentation heuristic (MSTAH) [31] plus connections obtained from a single run of the variable neighborhood descent (VND) from [31]. For (dCol) we pursue the same strategy, but additionally need to orient each of the obtained connections. Using the method described by Chimani et al. [13] we initially orient the solutions obtained by MSTAH and VND, respectively, and afterwards adopt the oriented connections obtained in this way.

Solving (Col) and (dCol) has been further configured as follows. For (Col) we add the cheapest connection to each customer $k \in C$ to the restricted master problem in each pricing iteration if it has negative reduced costs. Unfortunately, preliminary tests showed that solving the pricing subproblem for (dCol) by algorithms for the elementary shortest path problem with resource constraints – as discussed in Section 5.2 – is too time consuming already for relatively small instances. Too many labels need to be considered for each node and thus, this approach turned out to perform much worse than the MIP based approach discussed in Section 5.1. Hence, we only consider the MIP based approach in the following. To speed-up the pricing for type-2 customers, we return the first found solution that prices out favorably instead of trying to find a proven optimal solution in each execution. Thus, as for (Col) we add at most one connection for every customer in each pricing iteration.

As opposed to our problem definition in Section 2, we allow for the root node r being a potential branching node of some type-2 customer $k \in C_2$. Otherwise, we would restrict ourself to a too small set of feasible values of $b_{\max}$. Since the MIP for solving the pricing subproblem of (dCol) does not allow this case, we additionally apply a directed variant of the length constrained shortest path algorithm from [24] in this case.

[3]http://scip.zib.de

Table 2: Relative LP relaxation values and corresponding standard deviations in % for (MCF), (Col), and (dCol).

Variant	Set	$\frac{(\text{Col})^{\text{LP}}-(\text{MCF})^{\text{LP}}}{(\text{MCF})^{\text{LP}}}$ [%]		$\frac{(\text{dCol})^{\text{LP}}-(\text{MCF})^{\text{LP}}}{(\text{MCF})^{\text{LP}}}$ [%]		$\frac{(\text{dCol})^{\text{LP}}-(\text{Col})^{\text{LP}}}{(\text{Col})^{\text{LP}}}$ [%]	
OPT, $b_{\max} = 0$	ClgSE-I1	0.00	(0.00)	1.63	(2.38)	1.63	(2.38)
	ClgSE-I2	0.00	(0.00)	8.84	(4.08)	8.84	(4.08)
	ClgSE-I3	0.00	(0.00)	5.53	(4.55)	5.53	(4.55)
	ClgN1B-I1	0.00	(0.00)	2.78	(2.32)	2.78	(2.32)
	ClgN1B-I2	0.00	(0.00)	0.95	(0.89)	0.95	(0.89)
SST, $b_{\max} = 0$	ClgSE-I1	0.05	(0.23)	1.68	(2.35)	1.63	(2.38)
	ClgSE-I2	0.14	(0.55)	9.30	(5.07)	9.13	(4.65)
	ClgSE-I3	0.88	(2.47)	8.02	(5.31)	7.09	(4.84)
	ClgN1B-I1	3.07	(6.67)	5.29	(7.1)	2.58	(2.29)
	ClgN1B-I2	2.12	(5.05)	3.09	(4.7)	1.36	(1.47)
SST, $b_{\max} = 30$	ClgSE-I1	7.06	(5.07)	8.81	(5.65)	1.75	(2.36)
	ClgSE-I2	5.66	(2.63)	19.39	(5.95)	12.99	(4.66)
	ClgSE-I3	4.80	(2.89)	12.2	(5.07)	7.07	(4.01)
	ClgN1B-I1	5.88	(7.08)	9.07	(7.68)	2.72	(1.42)
	ClgN1B-I2	4.03	(5.52)	5.76	(5.44)	1.58	(1.89)
SST, $b_{\max} = 50$	ClgSE-I1	9.61	(8.98)	11.88	(10.14)	2.17	(3.05)
	ClgSE-I2	5.85	(3.48)	24.17	(6.81)	17.32	(5.31)
	ClgSE-I3	6.53	(3.78)	13.04	(6.42)	6.08	(3.95)
	ClgN1B-I1	2.45	(3.16)	5.53	(3.35)	2.97	(2.00)
	ClgN1B-I2	4.21	(6.16)	5.72	(6.36)	1.65	(1.91)
SST, $b_{\max} = 100$	ClgSE-I1	8.10	(11.94)	10.77	(13.70)	2.07	(2.58)
	ClgSE-I2	3.39	(2.57)	23.24	(7.27)	19.14	(4.88)
	ClgSE-I3	2.75	(2.56)	13.10	(8.16)	10.29	(6.57)
	ClgN1B-I1	2.37	(3.95)	6.07	(4.95)	3.55	(2.28)
	ClgN1B-I2	1.06	(2.03)	1.87	(1.01)	1.49	(1.18)

8.1 Linear Programming Relaxations

Table 2 depicts the average improvement and corresponding standard deviations in percent of the LP relaxation values of (Col) and (dCol) over (MCF). Furthermore, these values are additionally given for (dCol) compared to (Col).

The results from Table 2 confirm the results of our theoretical comparison from Section 6. While the LP relaxation values of (MCF) and (Col) are – for the considered instances – equal for the OPT variant without considering $b_{\max}$-redundancy – i.e. $b_{\max}(k) = 0$, $\forall k \in C_2$ – the values obtained from solving $(\text{Col})^{\text{LP}}$ are significantly better for all other configurations and instance sets. Furthermore, the LP relaxation values of (dCol) clearly dominate those of $(\text{Col})^{\text{LP}}$.

Tables 3 and 4 analyze the efficiency of the various approaches for using alternative dual-optimal solutions in the pricing subproblems of (Col) as proposed in Section 4.2. As previously described, D^* simply uses the obtained dual variable values without any modification, while D' equally splits the potential increase for each edge over all $|C|$ subproblems. $D^{(k,d)}$ refers to the fine-grained variant controlled by parameter d, while $D^{(k,d')}$ is the compromise between $D^{(k,d)}$ and D' where d is never decreased. Finally, $D^{(\text{p})}$ which is valid for the SST variant only, denotes the strategy considering each customer's prize.

From Table 3, we conclude that all variants are able to solve the linear relaxations of the smaller ClgS

Table 3: Median CPU-times for solving the LP relaxation of (MCF) and the various variants of (Col). Best values are marked bold.

Variant	Set	$(\text{MCF})^{\text{LP}}$	$(\text{Col})^{\text{LP}}$ D^*	D'	$D^{(k,d')}$	$D^{(k,d)}$	$D^{(\text{p})}$
OPT, $b_{\max}=0$	ClgSE-I1	**0.09**	0.55	0.16	0.11	0.13	-
	ClgSE-I2	**0.34**	5.26	2.83	2.10	1.03	-
	ClgSE-I3	**0.20**	3.30	0.40	0.34	0.40	-
	ClgN1B-I1	43.55	94.48	21.57	**11.98**	13.48	-
	ClgN1B-I2	58.27	203.53	41.52	14.05	**12.19**	-
SST, $b_{\max}=0$	ClgSE-I1	**0.10**	0.58	0.20	0.12	0.13	0.24
	ClgSE-I2	**0.35**	5.99	4.15	1.12	1.11	2.79
	ClgSE-I3	**0.19**	1.15	0.41	0.22	0.36	0.40
	ClgN1B-I1	42.82	116.25	19.26	**10.90**	16.04	25.32
	ClgN1B-I2	79.55	137.68	66.10	**13.32**	15.24	51.76
SST, $b_{\max}=30$	ClgSE-I1	**0.15**	0.86	0.51	0.30	0.38	0.45
	ClgSE-I2	**0.86**	6.45	4.34	2.62	2.38	3.79
	ClgSE-I3	**0.33**	2.48	1.00	0.58	1.03	1.11
	ClgN1B-I1	190.48	124.61	32.63	**20.85**	31.28	41.04
	ClgN1B-I2	1070.66	291.45	76.64	**30.09**	34.72	93.97
SST, $b_{\max}=50$	ClgSE-I1	**0.18**	1.11	0.49	0.40	0.53	0.39
	ClgSE-I2	**0.82**	6.26	4.20	3.13	4.31	4.18
	ClgSE-I3	**0.41**	3.60	1.28	1.11	2.01	1.42
	ClgN1B-I1	212.07	220.80	39.01	**24.70**	54.66	39.99
	ClgN1B-I2	1144.86	391.44	103.83	**40.02**	55.76	136.04
SST, $b_{\max}=100$	ClgSE-I1	**0.15**	3.04	0.95	0.74	1.28	1.21
	ClgSE-I2	**0.58**	23.80	11.29	8.63	15.78	10.80
	ClgSE-I3	**0.37**	9.40	2.97	1.97	4.94	3.48
	ClgN1B-I1	214.67	540.45	98.94	**59.61**	125.14	105.93
	ClgN1B-I2	1281.95	652.77	296.17	**78.53**	104.47	338.19

instances quite efficiently. On the one hand, $(\text{MCF})^{\text{LP}}$ usually can be solved slightly faster than $(\text{Col})^{\text{LP}}$ for these instances. On the other hand the obtained bounds due to $(\text{Col})^{\text{LP}}$ are better than those of $(\text{MCF})^{\text{LP}}$. For larger instances, $(\text{Col})^{\text{LP}}$ can be additionally solved more efficiently than $(\text{MCF})^{\text{LP}}$, especially when using alternative dual-optimal solutions according to D', $D^{(k,d')}$, $D^{(k,d)}$, or $D^{(\text{p})}$. Among these, $D^{(k,d')}$ performs better than the other three. Furthermore, we conclude that considering $b_{\max}$-redundancy yields an enormous increase in terms of necessary CPU-time for $(\text{MCF})^{\text{LP}}$, while the overhead in $(\text{Col})^{\text{LP}}$ is only moderate.

Table 4 compares the relative number of needed pricing iterations to solve $(\text{Col})^{\text{LP}}$, i.e. the relative number of times the restricted master problem needs to be solved, using D' as a basis. In consistency with the median CPU-times from Table 3, we conclude that using D', $D^{(k,d')}$, $D^{(k,d)}$, or $D^{(\text{p})}$ significantly reduces the number of needed pricing iterations. As for the CPU-times, slight advantages of $D^{(k,d')}$ over the other approaches can be observed. Note that already applying D' instead of simply using the standard dual-optimal variable values – i.e. using D^* – yields a major improvement. We conclude that D', $D^{(k,d')}$, $D^{(k,d)}$, and $D^{(\text{p})}$ are able to find meaningful connections already at the beginning of the column generation process and thus allow for efficiently solving the linear relaxation of (Col).

Tables 5 and 6 analyze the efficiency of the various approaches using alternative dual-optimal solutions for the directed connection formulation (dCol). As described in Section 5.4, the interpretations of D', $D^{(k,d')}$,

Table 4: Absolute and average relative number of pricing iterations and corresponding standard deviations for solving the LP relaxation of (Col) with various variants of alternative dual-optimal solutions. Best values are marked bold.

Variant	Set	# of pricing iterations							
		D'		$\frac{D^*}{D'}$		$\frac{D^{(k,d')}}{D'}$		$\frac{D^{(k,d)}}{D'}$		$\frac{D^{(p)}}{D'}$	
OPT, $b_{\max}=0$	ClgSE-I1	85.76	(77.39)	3.45	(1.53)	**0.99**	(0.35)	1.02	(0.57)	-	(-)
	ClgSE-I2	307.87	(230.09)	1.97	(1.02)	0.99	(0.34)	**0.69**	(0.41)	-	(-)
	ClgSE-I3	232.8	(333.95)	3.73	(1.74)	1.01	(0.36)	**0.85**	(0.52)	-	(-)
	ClgN1B-I1	381.65	(472.32)	5.83	(2.97)	0.85	(0.56)	**0.76**	(0.46)	-	(-)
	ClgN1B-I2	250.16	(212.13)	4.56	(2.25)	**0.81**	(0.38)	**0.81**	(0.61)	-	(-)
SST, $b_{\max}=0$	ClgSE-I1	101.24	(78.41)	2.31	(1.02)	**0.79**	(0.31)	0.84	(0.46)	1.25	(0.53)
	ClgSE-I2	327.53	(227.09)	1.63	(0.83)	**0.61**	(0.24)	0.62	(0.38)	1.13	(0.29)
	ClgSE-I3	280.93	(412.87)	3.12	(1.30)	**0.77**	(0.36)	0.81	(0.46)	0.98	(0.29)
	ClgN1B-I1	332.4	(397.04)	7.61	(5.33)	**0.73**	(0.32)	0.80	(0.45)	1.49	(0.98)
	ClgN1B-I2	254.37	(174.53)	3.87	(1.87)	**0.64**	(0.37)	0.67	(0.29)	1.08	(0.35)
SST, $b_{\max}=30$	ClgSE-I1	91.88	(84.91)	2.34	(0.70)	**0.81**	(0.34)	0.93	(0.37)	1.08	(0.3)
	ClgSE-I2	266.53	(133.01)	1.46	(0.37)	0.78	(0.32)	**0.65**	(0.21)	1.00	(0.32)
	ClgSE-I3	137.27	(177.27)	3.20	(1.43)	**0.83**	(0.33)	0.98	(0.46)	1.29	(0.42)
	ClgN1B-I1	622.65	(1015.9)	8.34	(9.49)	**0.66**	(0.44)	0.77	(0.48)	1.26	(0.53)
	ClgN1B-I2	232.79	(115.87)	4.68	(1.96)	**0.62**	(0.24)	0.66	(0.23)	1.15	(0.31)
SST, $b_{\max}=50$	ClgSE-I1	67.56	(58.34)	2.62	(1.09)	**0.89**	(0.31)	1.02	(0.39)	0.99	(0.27)
	ClgSE-I2	191.6	(96.62)	1.50	(0.55)	0.81	(0.34)	**0.79**	(0.26)	0.96	(0.22)
	ClgSE-I3	81.8	(62.1)	3.77	(1.70)	**0.92**	(0.32)	1.15	(0.41)	1.21	(0.43)
	ClgN1B-I1	361.8	(503.84)	5.82	(4.53)	**0.70**	(0.45)	0.94	(0.69)	1.15	(0.71)
	ClgN1B-I2	239.26	(113.94)	4.43	(2.59)	**0.59**	(0.19)	0.62	(0.23)	1.24	(0.52)
SST, $b_{\max}=100$	ClgSE-I1	49.04	(21.06)	2.95	(1.45)	**0.99**	(0.31)	1.23	(0.32)	1.16	(0.35)
	ClgSE-I2	119.20	(49.23)	2.24	(0.92)	**0.99**	(0.27)	1.16	(0.33)	1.04	(0.27)
	ClgSE-I3	72.53	(45.00)	3.96	(1.54)	**0.88**	(0.27)	1.24	(0.54)	1.24	(0.45)
	ClgN1B-I1	546.05	(853.58)	6.61	(6.63)	**0.75**	(0.46)	0.92	(0.61)	1.06	(0.45)
	ClgN1B-I2	280.58	(103.76)	3.61	(1.58)	0.55	(0.18)	**0.53**	(0.19)	1.10	(0.35)

$D^{(k,d)}$, and $D^{(\mathrm{p})}$ correspond to the undirected case, although some calculations are slightly different.

As expected the CPU-time overhead for solving $(\mathrm{dCol})^{\mathrm{LP}}$ due the $\mathcal{NP}$-hard pricing subproblems for type-2 customers $k \in C_2$ is not negligible. However, similar to the previous discussion for (Col) we can observe that D', $D^{(k,d')}$, $D^{(k,d)}$, and $D^{(\mathrm{p})}$ substantially speed-up the solution of $(\mathrm{dCol})^{\mathrm{LP}}$. Furthermore, the relative additional effort for solving $(\mathrm{dCol})^{\mathrm{LP}}$ compared to $(\mathrm{MCF})^{\mathrm{LP}}$ decreases when considering larger instances and $b_{\max}$-redundancy, i.e. if $b_{\max}(k) \neq 0, \forall k \in C_2$. Since the LP relaxation values of (dCol) are much tighter than those of the other models, (dCol) might nevertheless outperform them due to a significantly smaller number of nodes that need to be considered in the branch-and-bound tree.

Table 6 details the relative number of pricing iterations needed to solve $(\mathrm{dCol})^{\mathrm{LP}}$ for $D^{(k,d')}$, $D^{(k,d)}$, and $D^{(\mathrm{p})}$ in comparison to D'. Here, only those instances are considered where $(\mathrm{dCol})^{\mathrm{LP}}$ could be solved within the given time limit of 7200 CPU-seconds when using D'. We do not report on D^*, since it could solve $(\mathrm{dCol})^{\mathrm{LP}}$ for very few instances only. As for the undirected model, we conclude that the advanced adaptation strategies often significantly reduce the number of needed pricing iterations, and $D^{(k,d')}$ is the best option for solving $(\mathrm{dCol})^{\mathrm{LP}}$, too.

Table 5: Median CPU-times for solving the LP relaxation of (MCF) and the diverse variants of (dCol). Best values are marked bold.

Variant	Set	$(\mathrm{MCF})^{\mathrm{LP}}$	$(\mathrm{dCol})^{\mathrm{LP}}$				
			D^{*}	D'	$D^{(k,d')}$	$D^{(k,d)}$	$D^{(\mathrm{p})}$
OPT, $b_{\max}=0$	ClgSE-I1	**0.09**	28.44	5.66	5.74	6.36	-
	ClgSE-I2	**0.34**	92.30	50.30	57.87	62.16	-
	ClgSE-I3	**0.20**	70.45	9.79	8.64	19.43	-
	ClgN1B-I1	**43.55**	7200.00	3677.30	1805.73	2838.3	-
	ClgN1B-I2	**58.27**	7200.00	7200.00	7200.00	7200.00	-
SST, $b_{\max}=0$	ClgSE-I1	**0.10**	23.31	6.36	4.12	7.10	7.46
	ClgSE-I2	**0.35**	114.82	125.52	61.06	63.80	82.06
	ClgSE-I3	**0.19**	61.93	8.51	7.95	25.98	9.37
	ClgN1B-I1	**42.82**	7200.00	1342.96	800.41	2795.7	4410.18
	ClgN1B-I2	**79.55**	7200.00	6968.09	2884.42	6499.98	7200.00
SST, $b_{\max}=30$	ClgSE-I1	**0.15**	49.61	10.81	7.11	14.03	11.49
	ClgSE-I2	**0.86**	174.55	69.01	52.68	95.22	59.10
	ClgSE-I3	**0.33**	111.91	27.69	13.05	35.06	28.10
	ClgN1B-I1	**190.48**	7200.00	1457.13	791.07	3715.07	3055.81
	ClgN1B-I2	**1070.66**	7200.00	6821.66	3331.36	7200.00	7200.00
SST, $b_{\max}=50$	ClgSE-I1	**0.18**	38.77	8.90	7.73	16.11	9.19
	ClgSE-I2	**0.82**	179.41	39.57	36.78	113.35	83.53
	ClgSE-I3	**0.41**	98.76	12.39	11.53	35.63	13.71
	ClgN1B-I1	**212.07**	7200.00	1171.84	842.75	4493.58	1568.81
	ClgN1B-I2	**1144.86**	7200.00	7200.00	4782.36	6739.93	7200.00
SST, $b_{\max}=100$	ClgSE-I1	**0.15**	50.03	4.72	4.48	17.64	6.17
	ClgSE-I2	**0.58**	950.35	36.38	41.88	117.71	23.65
	ClgSE-I3	**0.37**	589.65	10.66	18.40	81.27	10.90
	ClgN1B-I1	**214.67**	7200.00	802.08	726.16	2841.21	1132.12
	ClgN1B-I2	**1281.95**	7200.00	7200.00	4463.8	7200.00	7200.00

Table 6: Absolute and average relative number of pricing iterations and corresponding standard deviations for solving the LP relaxation of (dCol) with various variants of alternative dual-optimal solutions. Best values are marked bold.

Variant	Set	# of pricing iterations							
		D'		$\frac{D^{(k,d')}}{D'}$		$\frac{D^{(k,d)}}{D'}$		$\frac{D^{(\mathrm{p})}}{D'}$	
OPT, $b_{\max}=0$	ClgSE-I1	135.13	(107.25)	0.91	(0.32)	**0.86**	(0.46)	-	(-)
	ClgSE-I2	417.6	(287.62)	1.04	(0.42)	**0.69**	(0.38)	-	(-)
	ClgSE-I3	122.07	(72.18)	**0.83**	(0.15)	1.06	(0.35)	-	(-)
	ClgN1B-I1	160.69	(69.88)	**0.88**	(0.22)	1.59	(0.51)	-	(-)
	ClgN1B-I2	185.89	(88.07)	**1.16**	(0.38)	1.25	(0.49)	-	(-)
SST, $b_{\max}=0$	ClgSE-I1	126.64	(85.87)	0.89	(0.37)	**0.87**	(0.49)	1.05	(0.30)
	ClgSE-I2	496.57	(283.48)	0.73	(0.39)	**0.56**	(0.36)	1.08	(0.54)
	ClgSE-I3	137.93	(104.35)	**0.94**	(0.19)	1.04	(0.46)	1.25	(0.61)
	ClgN1B-I1	153.21	(43.33)	**0.84**	(0.14)	1.37	(0.28)	1.48	(0.38)
	ClgN1B-I2	258.5	(117.52)	**0.67**	(0.35)	1.27	(0.92)	1.39	(0.47)
SST, $b_{\max}=30$	ClgSE-I1	205.68	(286.75)	**0.77**	(0.3)	0.83	(0.43)	1.06	(0.44)
	ClgSE-I2	423.67	(564.89)	0.80	(0.34)	**0.73**	(0.27)	1.13	(0.44)
	ClgSE-I3	159.40	(103.11)	**0.77**	(0.36)	0.86	(0.45)	1.22	(0.34)
	ClgN1B-I1	153.60	(91.40)	**0.96**	(0.24)	1.67	(0.55)	1.34	(0.44)
	ClgN1B-I2	267.20	(91.16)	**0.54**	(0.25)	0.83	(0.39)	0.66	(0.28)
SST, $b_{\max}=50$	ClgSE-I1	88.72	(54.75)	**0.90**	(0.39)	1.00	(0.37)	1.06	(0.29)
	ClgSE-I2	264.00	(226.09)	**0.89**	(0.54)	0.96	(0.64)	1.26	(0.45)
	ClgSE-I3	104.20	(58.63)	**0.82**	(0.28)	0.92	(0.38)	0.99	(0.56)
	ClgN1B-I1	145.89	(71.02)	**0.82**	(0.22)	1.43	(0.48)	1.15	(0.53)
	ClgN1B-I2	238.00	(100.27)	**0.66**	(0.20)	0.90	(0.39)	1.12	(0.25)
SST, $b_{\max}=100$	ClgSE-I1	50.83	(30.24)	**0.96**	(0.29)	1.16	(0.39)	1.19	(0.59)
	ClgSE-I2	91.20	(43.69)	**1.08**	(0.38)	1.33	(0.44)	**1.08**	(0.48)
	ClgSE-I3	113.43	(185.62)	1.46	(1.19)	**1.29**	(0.62)	1.36	(0.59)
	ClgN1B-I1	112.44	(51.13)	**0.94**	(0.27)	1.72	(0.47)	1.29	(0.45)
	ClgN1B-I2	232.00	(107.13)	**0.68**	(0.25)	0.97	(0.37)	0.81	(0.25)

Table 7: Average optimality gaps and corresponding standard deviations after 7200 CPU-seconds for instances where $(\text{dCol})^{\text{LP}}$ could be solved when using $D^{(k,d')}$. Best values are marked bold.

Variant	Set	#	(MCF)		(Col)		(dCol)	
OPT, $b_{\max} = 0$	ClgSE-I1	25	**0.00**	(0.00)	**0.00**	(0.00)	**0.00**	(0.00)
	ClgSE-I2	14	**0.00**	(0.00)	**0.00**	(0.00)	**0.00**	(0.00)
	ClgSE-I3	15	**0.00**	(0.00)	0.10	(0.39)	**0.00**	(0.00)
	ClgN1B-I1	13	**0.00**	(0.00)	0.99	(0.94)	**0.00**	(0.00)
	ClgN1B-I2	9	**0.00**	(0.00)	**0.00**	(0.00)	**0.00**	(0.00)
SST, $b_{\max} = 0$	ClgSE-I1	25	**0.00**	(0.00)	**0.00**	(0.00)	**0.00**	(0.00)
	ClgSE-I2	15	**0.00**	(0.00)	**0.00**	(0.00)	**0.00**	(0.00)
	ClgSE-I3	15	**0.00**	(0.00)	0.08	(0.31)	**0.00**	(0.00)
	ClgN1B-I1	16	0.02	(0.06)	0.95	(0.96)	**0.00**	(0.00)
	ClgN1B-I2	17	**0.00**	(0.00)	0.09	(0.25)	**0.00**	(0.00)
SST, $b_{\max} = 30$	ClgSE-I1	25	**0.00**	(0.00)	**0.00**	(0.00)	**0.00**	(0.00)
	ClgSE-I2	15	**0.00**	(0.00)	0.31	(1.21)	**0.00**	(0.00)
	ClgSE-I3	15	**0.00**	(0.00)	0.12	(0.46)	**0.00**	(0.00)
	ClgN1B-I1	15	0.32	(0.68)	1.11	(1.20)	**0.00**	(0.00)
	ClgN1B-I2	17	0.41	(1.10)	0.14	(0.32)	**0.00**	(0.00)
SST, $b_{\max} = 50$	ClgSE-I1	25	**0.00**	(0.00)	**0.00**	(0.00)	**0.00**	(0.00)
	ClgSE-I2	15	**0.00**	(0.00)	0.64	(1.50)	**0.00**	(0.00)
	ClgSE-I3	15	**0.00**	(0.00)	**0.00**	(0.00)	**0.00**	(0.00)
	ClgN1B-I1	18	0.20	(0.58)	1.41	(1.66)	**0.00**	(0.00)
	ClgN1B-I2	13	0.26	(0.95)	0.16	(0.38)	**0.00**	(0.00)
SST, $b_{\max} = 100$	ClgSE-I1	24	**0.00**	(0.00)	**0.00**	(0.00)	**0.00**	(0.00)
	ClgSE-I2	15	**0.00**	(0.00)	0.76	(1.61)	**0.00**	(0.00)
	ClgSE-I3	14	**0.00**	(0.00)	0.13	(0.40)	**0.00**	(0.00)
	ClgN1B-I1	18	1.14	(1.48)	1.78	(2.05)	**0.00**	(0.00)
	ClgN1B-I2	14	0.14	(0.54)	0.17	(0.45)	**0.00**	(0.00)

8.2 Solutions and Optimality Gaps

In the following, computational results for solving (Col) and (dCol) by branch-and-price are presented. Branching is performed on edge variables for (Col) and on arc variables for (dCol), respectively. We do not use any problem specific branching rules, but trust on the branching decisions as performed by SCIP. All results for (Col) and (dCol) have been computed using $D^{(k,d')}$ for adapting dual variable values, which has been shown to outperform the other variants. To allow for a meaningful comparison, we only report on those instances where the LP relaxation of (dCol) could be solved within the given time limit of 7200 CPU-seconds when using $D^{(k,d')}$. The corresponding number of considered instances is additionally stated in each table.

Table 7 shows average gaps as well as corresponding standard deviations in percent for each considered instance set and setting. We conclude that (dCol) could be solved to proven optimality whenever its linear relaxation was solved. The undirected connection formulation (Col), however, failed to find a proven optimal solution within two hours for some instances and performs slightly worse than the multi-commodity flow formulation of Wagner et al. [45] with respect to this criterion. Although the LP relaxation values of (Col) are better than those of model (MCF) and the root relaxation gaps are already quite small, a too large number of nodes needs to be considered in the branch-and-bound tree for further improving the obtained lower bound in order to proof optimality of a solution.

Table 8: Median CPU-times for instances where $(\mathrm{dCol})^{\mathrm{LP}}$ could be solved when using $D^{(k,d')}$. Best values are marked bold.

Variant	Set	#	(MCF)	(Col)	(dCol)
OPT, $b_{\max} = 0$	ClgSE-I1	25	**0.25**	0.28	5.74
	ClgSE-I2	14	**3.92**	177.88	54.46
	ClgSE-I3	15	**0.87**	2.25	8.64
	ClgN1B-I1	13	**643.28**	7200.01	1109.79
	ClgN1B-I2	9	**136.03**	241.72	3191.45
SST, $b_{\max} = 0$	ClgSE-I1	25	**0.30**	0.32	4.12
	ClgSE-I2	15	**3.63**	81.82	61.06
	ClgSE-I3	15	**0.76**	2.25	7.95
	ClgN1B-I1	16	**519.06**	7200.00	728.73
	ClgN1B-I2	17	237.52	**211.20**	2500.98
SST, $b_{\max} = 30$	ClgSE-I1	25	1.43	**0.48**	7.11
	ClgSE-I2	15	**23.87**	329.06	52.68
	ClgSE-I3	15	**2.06**	4.41	13.05
	ClgN1B-I1	15	1524.24	7200.00	**752.98**
	ClgN1B-I2	17	2322.39	**261.72**	3185.55
SST, $b_{\max} = 50$	ClgSE-I1	25	1.37	**0.70**	7.73
	ClgSE-I2	15	191.69	585.37	**36.78**
	ClgSE-I3	15	**2.78**	6.18	11.53
	ClgN1B-I1	18	2788.23	7200.00	**818.78**
	ClgN1B-I2	13	2210.25	**339.64**	3151.63
SST, $b_{\max} = 100$	ClgSE-I1	24	1.70	**1.65**	4.34
	ClgSE-I2	15	46.93	4000.79	**41.88**
	ClgSE-I3	14	**4.00**	19.41	16.25
	ClgN1B-I1	18	7156.75	7200.00	**672.97**
	ClgN1B-I2	14	2419.23	**557.36**	2445.72

Table 8 reports median CPU-times for solving (MCF), (Col), and (dCol), respectively. We conclude that the performance of both connection based formulations improves in comparison to the MCF formulation when considering $b_{\max}$-redundancy. When taking into account that SCIP 1.2.0 with CPLEX 12.1 is slower than CPLEX 12.1 alone roughly by a factor of three[4], (dCol) can be considered the most effective method on larger instances when $b_{\max}(k) \neq 0$, $\forall k \in C_2$. The undirected formulation (Col) is, however, typically the fastest approach for those larger instances where the branch-and-bound tree only has a moderate number of nodes, e.g. on the set ClgN1B-I2.

Since, we observed from Table 7 that (dCol) could be solved to proven optimality whenever its linear relaxation $(\mathrm{dCol})^{\mathrm{LP}}$ was solved, we further analyzed for how many instances the solution to its linear relaxation is integral, i.e. is an optimal solution to the corresponding instance. As detailed in Table 9, solving $(\mathrm{dCol})^{\mathrm{LP}}$ yields a proven optimal solution to (dCol) for almost all considered instances and settings. On the contrary, most solutions of $(\mathrm{MCF})^{\mathrm{LP}}$ and $(\mathrm{Col})^{\mathrm{LP}}$ contain fractional variables.

Overall, we conclude that both connection based formulations have their individual advantages. While the LP relaxation of (Col) is tighter than the one of (MCF) and can be solved efficiently, sometimes a too large number of nodes in the branch-and-bound tree needs to be considered. Thus (Col) sometimes fails to prove optimality of a solution within reasonable time. The resulting gaps are, however, relatively tight

[4]http://scip.zib.de

Table 9: Number of instances per set where the solution to the linear relaxation is integral. Only those instances are considered where $\text{(dCol)}^{\text{LP}}$ could be solved withing 7200 CPU-seconds using $D^{(k,d')}$.

Variant	Set	#	(MCF)^{LP}	(Col)^{LP}	$\text{(dCol)}^{\text{LP}}$
OPT, $b_{\max} = 0$	ClgSE-I1	25	3	3	25
	ClgSE-I2	14	0	0	14
	ClgSE-I3	15	0	0	15
	ClgN1B-I1	13	0	0	11
	ClgN1B-I2	9	3	3	9
SST, $b_{\max} = 0$	ClgSE-I1	25	2	3	25
	ClgSE-I2	15	0	0	15
	ClgSE-I3	15	0	0	15
	ClgN1B-I1	16	0	0	15
	ClgN1B-I2	17	1	3	17
SST, $b_{\max} = 30$	ClgSE-I1	25	0	1	25
	ClgSE-I2	15	0	0	14
	ClgSE-I3	15	0	0	14
	ClgN1B-I1	15	0	0	14
	ClgN1B-I2	17	0	2	17
SST, $b_{\max} = 50$	ClgSE-I1	25	0	2	25
	ClgSE-I2	15	0	0	14
	ClgSE-I3	15	0	0	14
	ClgN1B-I1	18	0	0	18
	ClgN1B-I2	13	1	2	13
SST, $b_{\max} = 100$	ClgSE-I1	24	0	0	24
	ClgSE-I2	15	0	0	14
	ClgSE-I3	14	0	0	13
	ClgN1B-I1	18	0	0	18
	ClgN1B-I2	14	0	1	14

already after solving the root node. With respect to model (dCol), we conclude that its LP relaxation is extremely tight and in particular turned out to be integral for almost all used test instances and settings. While the computational effort for solving it is not negligible, it nevertheless outperforms the other methods on medium sized instances. Both models perform bad, when simply using the dual variable values obtained by the used LP solver. Above computational results clearly show that the usage of alternative dual-optimal solutions as described in Sections 4.2 and 5.4, respectively, substantially reduces the time necessary for solving (Col) and (dCol). We further conclude that the performance of (MCF) heavily decreases then considering $b_{\max}$-redundancy. For (Col) and (dCol) the additional computational effort increases only moderately.

9 Conclusions and Future Work

In this article, two mixed integer programming approaches for solving the $b_{\max}$-Survivable Network Design Problem ($b_{\max}$-SNDP) have been considered. These are based on an exponential number of so-called connection variables and can be solved by branch-and-price. We showed how to significantly speed up the solution of both models by using alternative dual-optimal solutions in the pricing subproblems. Using a polyhedral comparison we further showed that both proposed models theoretically dominate existing ones and that the second model, which is a directed variant of the first one, dominates its undirected counterpart.

Computational results show that both branch-and-price approaches perform reasonably well on medium sized instances. While, the undirected model yields tight optimality gaps already after relatively short time, it sometimes has problems to further raise the obtained lower bounds in order to prove optimality of a solution. For solving the linear relaxation of its directed counterpart much more computational effort is needed. The obtained solutions are, however, already integral and thus proven optimal solutions in the majority of test cases.

Interesting areas for further research include the development of methods based on the multilevel approach; see e.g. [46] for a survey. These might use the methods proposed in this article for solving smaller subproblems and can be used to tackle very large scale instances of $b_{\max}$-SNDP. Furthermore, considering additional algorithms and methods for solving the $\mathcal{NP}$-hard pricing subproblems of the directed connection formulation might allow for solving even larger instances to proven optimality.

References

[1] T. Achterberg. *Constraint Integer Programming*. PhD thesis, Technische Universität Berlin, 2007.

[2] P. Bachhiesl. The OPT- and the SST-problems for real world access network design – basic definitions and test instances. Working Report 01/2005, Carinthia Tech Institute, Department of Telematics and Network Engineering, Klagenfurt, Austria, 2005.

[3] P. Bachhiesl, M. Prossegger, G. Paulus, J. Werner, and H. Stögner. Simulation and optimization of the implementation costs for the last mile of fibre optic networks. *Networks and Spatial Economics*, 3(4):467–482, 2003.

[4] L. Bahiense, F. Barahona, and O. Porto. Solving Steiner tree problems in graphs with Lagrangian relaxation. *Journal of Combinatorial Optimization*, 7(3):259–282, 2003.

[5] E. Balas. The prize collecting traveling salesman problem. *Networks*, 19(6):621–636, 1989.

[6] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

[7] J. E. Beasley and N. Christofides. An algorithm for the resource constrained shortest path problem. *Networks*, 19(4):379–394, 1989.

[8] N. Boland, J. Dethridge, and I. Dumitrescu. Accelerated label setting algorithms for the elementary resource constrained shortest path problem. *Operations Research Letters*, 34(1):58–68, 2006.

[9] S. A. Canuto, M. G. C. Resende, and C. C. Ribeiro. Local search with perturbations for the prize-collecting Steiner tree problem in graphs. *Networks*, 38:50–58, 2001.

[10] A. Chabrier. Vehicle routing problem with elementary shortest path based column generation. *Computers & Operations Research*, 33(10):2972–2990, 2006.

[11] O. Chapovska and A. P. Punnen. Variations of the prize-collecting Steiner tree problem. *Networks*, 47(4):199–205, 2006.

[12] M. Chimani, M. Kandyba, I. Ljubic, and P. Mutzel. Orientation-based models for {0,1,2}-survivable network design: Theory and practice. *Mathematical Programming, Series B*, 2009. accepted.

[13] M. Chimani, M. Kandyba, and P. Mutzel. A new ILP formulation for 2-root-connected prize-collecting Steiner networks. In *15th Annual European Symposium on Algorithms (ESA'07)*, volume 4698 of *LNCS*, pages 681–692. Springer, 2007.

[14] A. S. da Cunha, A. Lucena, N. Maculan, and M. G. C. Resende. A relax-and-cut algorithm for the prize-collecting Steiner problem in graph. *Discrete Applied Mathematics*, 157(6):1198–1217, 2009.

[15] G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors. *Column Generation*. Springer, 2005.

[16] O. du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilized column generation. *Discrete Mathematics*, 194(1-3):229–237, 1999.

[17] C. W. Duin and S. Voß. Efficient path and vertex exchange in Steiner tree algorithms. *Networks*, 29:89–105, 1997.

[18] I. Dumitrescu and N. Boland. Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem. *Networks*, 42(3):135–153, 2003.

[19] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44(3):216–229, 2004.

[20] S. Fortune, J. Hopcroft, and H. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10:111–121, 1980.

[21] B. Fortz and M. Labbé. Polyhedral approaches to the design of survivable networks. In M. G. C. Resende and P. M. Pardolas, editors, *Handbook of Optimization in Telecommunications*, pages 367–389. Springer, 2006.

[22] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

[23] T. Gomes, J. Craveirinha, and L. Jorge. An effective algorithm for obtaining the minimal cost pair of disjoint paths with dual arc costs. *Computers & Operations Research*, 36:1670–1682, 2009.

[24] L. Gouveia, A. Paias, and D. Sharma. Modeling and solving the rooted distance-constrained minimum spanning tree problem. *Computers & Operations Research*, 35(2):600–613, 2008.

[25] IBM. CPLEX 12.1. http://www-01.ibm.com/software/integration/optimization/cplex-optimizer.

[26] S. Irnich and G. Desaulniers. Shortest path problems with resource constraints. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, pages 33–65. Springer, 2005.

[27] K. Kar, M. Kodialam, and T. V. Lakshman. Routing restorable bandwidth guaranteed connections using maximum 2-route flows. *IEEE/ACM Transactions on Networking*, 11(5):772–781, 2003.

[28] R. M. Karp. Reducibility among combinatorial problems. In E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[29] H. Kerivin and A. R. Mahjoub. Design of survivable networks: A survey. *Networks*, 46(1):1–21, 2005.

[30] T. Koch and A. Martin. Solving Steiner tree problems in graphs to optimality. *Networks*, 32(3):207–232, 1998.

[31] M. Leitner and G. R. Raidl. Lagrangian decomposition, metaheuristics, and hybrid approaches for the design of the last mile in fiber optic networks. In M. J. Blesa et al., editors, *Hybrid Metaheuristics 2008*, volume 5296 of *LNCS*, pages 158–174. Springer, 2008.

[32] M. Leitner and G. R. Raidl. Strong lower bounds for a survivable network design problem. In *International Symposium on Combinatorial Optimization (ISCO 2010)*, Hammamet, Tunisia, March 2010.

[33] M. Leitner, G. R. Raidl, and U. Pferschy. Accelerating column generation for a survivable network design problem. In M. G. Scutellà et al., editors, *Proceedings of the International Network Optimization Conference 2009*, 2009.

[34] I. Ljubić. *Exact and Memetic Algorithms for Two Network Design Problems*. PhD thesis, Vienna University of Technology, 2004.

[35] I. Ljubić, R. Weiskircher, U. Pferschy, G. Klau, P. Mutzel, and M. Fischetti. An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem. *Mathematical Programming, Series B*, 105(2–3):427–449, 2006.

[36] M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.

[37] A. Lucena and M. G. C. Resende. Strong lower bounds for the prize collecting Steiner problem in graphs. *Discrete Applied Mathematics*, 141(1–3):277–294, 2004.

[38] A. Segev. The node-weighted Steiner tree problem. *Networks*, 17(1):1–17, 1987.

[39] M. Stoer. *Design of Survivable Networks*, volume 1531 of *LNCS*. Springer, 1992.

[40] J. W. Suurballe and R. E. Tarjan. A quick method for finding shortest pairs of disjoint paths. *Networks*, 14:325–335, 1984.

[41] H. Takahashi and A. Matsuyama. An approximated solution for the Steiner tree problem in graphs. *Math. Japonica*, 24(6):573–577, 1980.

[42] E. Uchoa. Reduction tests for the prize-collecting Steiner problem. *Operations Research Letters*, 34(4):437–444, 2006.

[43] F. Vanderbeck. Implementing mixed integer column generation. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, pages 331–358. Springer US, 2005.

[44] D. Wagner, U. Pferschy, P. Mutzel, G. R. Raidl, and P. Bachhiesl. A directed cut model for the design of the last mile in real-world fiber optic networks. In B. Fortz, editor, *Proceedings of the International Network Optimization Conference 2007*, pages 103/1–6, Spa, Belgium, 2007.

[45] D. Wagner, G. R. Raidl, U. Pferschy, P. Mutzel, and P. Bachhiesl. A multi-commodity flow approach for the design of the last mile in real-world fiber optic networks. In K.-H. Waldmann and U. M. Stocker, editors, *Operations Research Proceedings 2006*, pages 197–202. Springer, 2007.

[46] C. Walshaw. Multilevel refinement for combinatorial optimisation: Boosting metaheuristic performance. In C. Blum, M. J. B. Aquilera, A. Roli, and M. Sampels, editors, *Hybrid Metaheuristics: An Emerging Approach to Optimization*, volume 114 of *Studies in Computational Intelligence (SCI)*, pages 261–289. Springer, 2008.

[47] P. Winter. Steiner problem in networks: a survey. *Networks*, 17(2):129–167, 1987.

[48] ZIB. SCIP 1.2.0. http://scip.zib.de.