A Directed Cut Model for the Design of the Last Mile in Real-World Fiber Optic Networks *

Daniel Wagner, *Carinthia University of Applied Sciences, Austria*
Ulrich Pferschy, *University of Graz, Austria*
Petra Mutzel, *University of Dortmund, Germany*
Günther R. Raidl, *Vienna University of Technology, Austria*
Peter Bachhiesl, *Carinthia University of Applied Sciences, Austria*

Keywords: Network design, (Prize-Collecting) Steiner Problem, biconnectivity, ILP

1. Introduction

Fiber optic networks have recently become economically feasible for single households. Since the area-wide expansion of fiber optic networks requires enormous financial resources, exact algorithms for finding provably-optimal solutions of the arising network design problems are desirable. We consider the problem of augmenting an existing network infrastructure by additional links (and switches) in order to connect it to the given customer nodes. We distinguish between standard customers (type-1) for which a single link to the infrastructure suffices, and important customers (type-2) that require a redundant connection. We also consider a variant of the problem in which the condition for type-2 customers is relaxed in the sense that the non-redundant part of the link shall not exceed a certain length ($b_{\max}$-redundancy).

We distinguish between two different optimization tasks: In the *Operative Planning Task*, the goal is to find a cost-efficient network augmentation for all the customers, whereas in the *Strategic Simulation Task* each customer gives a certain prize, i.e., an intended return on investment, and the optimization goal is to choose those customers whose connection provides the optimal profit.

More precisely, the problems considered are the following: We are given a connected undirected graph $G = (V, E, c, l, p, b_{\max})$ describing the spatial topology of the surrounding area of specified customer nodes, in particularly, specifying possible cable routes. The set of nodes V is the disjoint union of the customer nodes C and spatial nodes S (switches, possible Steiner nodes) resulting from the underlying spatial topology. The set of customers C is the disjoint union of C_1 and C_2, whereby customers C_1 require a single connection (type-1) and customers C_2 need to be redundantly connected (type-2). Each edge $e = (i, j) \in E$ represents a straight segment of Euclidean length $l_e \geq 0$ where a fiber optics cable might be installed with construction costs $c_e \geq 0$. In the (PCS) problem, the prizes of the potential customers are given by $p_i \geq 0$ for all $i \in C$. The already existing network infrastructure is represented by the subgraph $I = (V_I, E_I)$ of G. For constructing our network, we need to consider the following constraints arising from several technical and practical requirements in real-world network design:

- *Junction constraints:* Customer nodes of set C have to be connected to the existing infrastructure. Attaching new connections to the infrastructure is only allowed at predefined junction nodes $J \subseteq V_I$.
- *Non-crossing constraint:* Cable routes are not allowed to cross each other in a geometric sense.
- *Biconnectivity constraints:* Customer nodes of set C_2 (type-2) need to be redundantly connected to the existing infrastructure by two node-disjoint paths.
- $b_{\max}$*-redundancy:* Occasionally, the biconnectivity constraints 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 v via a single path of maximum Euclidean length $b_{\max}(k)$.

*This work is supported by the Austrian Research Promotion Agency (FFG) under grant 811378 and the European Marie Curie RTN ADONET under grant 504438.

In the *Operative Planning Task* the objective is to find a minimum-cost connected subgraph $G' = (V_{G'}, E_{G'})$ of G, $V_{G'} \subseteq V$, $E_{G'} \subseteq E$, connecting all customers in such a way that the technical constraints and requirements are satisfied. This problem can be considered as a generalization of the Steiner tree problem with additional redundancies (since typically we have $|C_2| < |C_1|$) and is therefore denoted as (STR).

In the *Strategic Simulation Task* the objective is to find a subgraph G' of G and a subset of the potential customers C' of C minimizing

$$\sum_{e \in E_{G'}} c_e + \sum_{i \in C' \setminus V_{G'}} p_i$$

so that all the technical constraints and requirements are satisfied with respect to the chosen customer set $C' \subseteq C$. This problem can be considered as a generalization of the prize-collecting Steiner tree problem and is therefore denoted as (PCS).

2. A Formulation based on Directed Cuts

Previously we applied a multi-commodity flow approach [**?**] to our problem. Since cut formulations have shown to provide good descriptions for many network design problems, e.g. the prize-collecting Steiner tree problem [**?**] which has many similarities to our problem, we also developed such a model. The idea of the formulation is to enforce the validity of a solution with lower bounds on the capacity of all cuts between the root node and each customer node to be connected.

Our directed cut formulation for the described problem relies on a transformation to the problem of finding a minimum subgraph in a related, directed graph similar as it was previously proposed for the prize-collecting Steiner tree problem [**?**]. Hence, we shrink the subgraph I of the existing infrastructure in G into a single root node 0 (hyper-node transformation) and construct a directed support graph $D = (V_D, A_D)$ corresponding to the undirected graph G. Node set V_D contains the nodes of the input graph G resulting from the hyper-node transformation. Arc set A_D contains an arc (0,i) for each edge incident to the root node 0 and two reversely directed arcs (i, j) and (j, i) for each other edge $(i, j) \in E$.

With respect to the non-crossing constraint we define $a(e, e') = 1$ if the edges $e, e' \in E$ are non-crossing ones and 0 otherwise. Crossing pairs of edges can be determined efficiently in advance. In a further preprocessing step we determine relevant nodes and arcs with respect to the $b_{\max}$-redundancy constraint. Let $N(k) \subseteq V_D$, $\forall k \in C_2$, be the set of all nodes for which a path to k of length not exceeding $b_{\max}(k)$ exists and $B(k) \subseteq A_D$ be set of all arcs $a = (i, j)$, $i, j \in N(k)$. These sets can be determined efficiently, e.g. by a modification of Dijkstra's shortest path algorithm [**?**].

The following variables are used in the integer linear programming formulation.

- Variables $e_{ij} \in \{0, 1\}$, $\forall (i, j) \in E$, indicating whether edge (i, j) is used ($e_{ij} = 1$) or not ($e_{ij} = 0$).
- Variables $x_{ij} \in \{0, 1\}$, $\forall (i, j) \in A_D$, indicating whether arc (i, j) is used ($x_{ij} = 1$) or not ($x_{ij} = 0$).
- Variables $y_i \in \{0, 1\}$, $\forall i \in V_D$, indicating in the (PCS) variant whether node i is connected ($y_i = 1$) or not ($y_i = 0$). In the (STR) variant, these variables are fixed to 1 for nodes $i \in C$.
- Variables $z_i \in \{0, 1\}$, $\forall i \in V_D$, indicating whether node i has two node-disjoint paths to the root node ($z_i = 1$) or not ($z_i = 0$).
- Variables $a_j^k \in \{0, 1\}$, $\forall k \in C_2$, $\forall j \in N(k)$, indicating whether customer node k is connected to the node j with a simple path and j is biconnected ($a_j^k = 1$) or not ($a_j^k = 0$).
- Variables $h_{ij}^k \in \{0, 1\}$, $\forall k \in C_2$, $\forall (i, j) \in B(k)$, indicating whether arc (i, j) is on the path from customer node k to its assigned biconnected node ($h_{ij}^k = 1$) or not ($h_{ij}^k = 0$).

For convenience we introduce the following notation: A set of vertices $S \subset M$ and its complement $\overline{S} = M \backslash S$ induce two directed cuts: $\delta^+_M(S) = \{(i,j) \mid i \in S,\ j \in \overline{S}\}$ and $\delta^-_M(S) = \{(i,j) \mid i \in \overline{S},\ j \in S\}$. We also write $x(A) = \sum_{ij \in A} x_{ij}$ for any subset of arcs $A \subset A_D$.

We start with a basic integer linear programming formulation that does not consider $b_{\max}$-redundancy and only enforces node-biconnectivity for nodes in C_2:

Minimize

$$\sum_{(i,j) \in E} c_{ij} e_{ij} + \sum_{i \in C} p_i (1 - y_i) \tag{1}$$

subject to

$$x(\delta^-(S)) \geq y_i \qquad \forall i \in C,\ \forall S \subset V_D \setminus \{0\} \mid i \in S \tag{2}$$
$$x(\delta^-(S)) \geq 2z_i \qquad \forall i \in V_D \setminus \{0\},\ \forall S \subset V_D \setminus \{0\} \mid i \in S \tag{3}$$
$$x(\delta^-_{V_D \setminus \{v\}}(S)) \geq z_i \qquad \forall i \in V_D \setminus \{0\},\ \forall v \in V_D \setminus \{0, i\},\ \forall S \subset V_D \setminus \{0, v\} \mid i \in S \tag{4}$$
$$z_i \geq y_i \qquad \forall i \in C_2 \tag{5}$$
$$x_{ij} \leq e_{ij} \qquad \forall (i,j) \in A_D \tag{6}$$
$$x_{ji} \leq e_{ij} \qquad \forall (i,j) \in A_D \tag{7}$$
$$x_e + x_{e'} \leq 1 \qquad \forall (e, e') \in E^2 \mid a(e, e') = 1 \tag{8}$$

Cut inequalities (2) are also called *connectivity constraints* and inequalities (3) *edge biconnectivity constraints* because they require a capacity of one for each cut between the root node and customer nodes with $y_i = 1$ or a capacity of two for $z_i = 1$, respectively. Inequalities (4) *node biconnectivity constraints*, respectively. They enforce a capacity of one for each cut between the root node and customer nodes with $z_i = 1$ whereby each node $v \in V_D \setminus \{0, i\}$ may be removed from the graph. Inequalities (5) relate variables y_i and z_i and guarantee biconnectivity for customer nodes in set C_2 for which $y_i = 1$. Inequalities (6) and (7) relate the arcs of the directed support graph D with the edges of graph G. Finally, inequalities (8) guarantee the non-crossing constraint.

The number of constraints (2), (3) and (4) is exponential in the size of the graph. Therefore we dynamically separate this constraints within a cutting plane framework. Regarding the required node-disjointness in a first step we determine for each node $i \in V_D \setminus \{0\}$ with $z_i > 0$ a set of candidate nodes which may violate the node-disjointness constraint. In a second step for each candidate node v violated cut inequalities (4) are determined and added. In our cutting plane framework we utilize Cherkassky and Goldbergs implementation of the push-relabel method for the maximum flow problem [**?**] to perform the required minimum cut computations.

To also consider $b_{\max}$-redundancy, we assume that every customer node in $k \in C_2$ must be connected to a biconnected node by a path of length less than or equal to $b_{\max}(k)$. For customer nodes that require strict biconnectedness we set $b_{\max}(k) = 0$. For $b_{\max}$-redundancy the whole branch-line from the customer node k to its biconnected attachment node j has to be modeled. This requires the use of variables a^k_j and h^k_{ij}, which however can be restricted locally to the $b_{\max}$-neighborhood of each customer node k. In analogy to $x(A)$ we use notation $h^k(A) = \sum_{ij \in A} h^k_{ij}$ for any subset of arcs $A \subset B(k)$.

The previous formulation is adapted as follows. Inequalities (5), which force strict biconnectedness for all customer nodes, are replaced by the following constraints:

$$\sum_{j \in N(k)} a_j^k = y_k \qquad k \in C_2 \qquad (9)$$

$$a_j^k \leq z_j \qquad \forall k \in C_2, \forall j \in N(k) \qquad (10)$$

$$h_{ij}^k \leq x_{ij} \qquad \forall k \in C_2,\ \forall (i,j) \in B(k) \qquad (11)$$

$$h^k(\delta^-_{N(k)}(S)) \geq a_j^k \qquad \forall k \in C_2,\ \forall j \in N(k) \setminus \{k\},\ \forall S \subset N(K) \setminus \{j\} \mid k \in S \qquad (12)$$

$$\sum_{(i,j) \in B(k)} l_{ij} h_{ij}^k \leq b_{\max}(k) \qquad \forall k \in C_2 \qquad (13)$$

Equations (9) force that each selected customer node of set C_2 is assigned exactly one attachment node. Inequalities (10) ensure that any assigned attachment node is biconnected. Inequalities (11) relate variables h_{ij}^k of the branch-line for customer node k with the corresponding variables x_{ij}, and cut inequalities (12) guarantee the connectedness of customer node k and its biconnected attachment node j. These constraints are separated dynamically in our framework, too. Finally, inequalities (13) limit for each node $k \in C_2$ the total length of all edges for which $h_{ij}^k = 1$ to $b_{\max}(k)$, i.e. the maximum branch lengths are enforced. Obviously inequalities (9) and (10) imply inequalities (5) for customer nodes k with $b_{\max}(k) = 0$, since $N(k) = \{k\}$ in this case. Therefore a special handling of these customer nodes is not necessary.

3. Preliminary Experimental Results

We tested our approach on Intel Xeon 3.6GHz machines with 4GB memory using the general purpose ILP solver CPLEX 10.0 from ILOG. Each individual run was given a time limit of 7200 seconds. The experiments were performed using artificial grid-graph instances and real-world based instances that were constructed from GIS-data of a German city [**?**].

Table 1 gives a detailed overview about characteristics of the used instance sets. For each set it shows the number of individual instances (# Inst.), the size of graph, average number of customers per set, the dimension of the underlying Euclidean area and the average size of the $b_{\max}$-neighborhoods. Since our currently available test instances did not contain customer specific values for $b_{\max}$, we set $b_{\max} = 20$ for customers of the G0100-I{1,2,3} instances and $b_{\max} = 30$ for customers of the ClgSE-I{1,2,3} instances in the tests with $b_{\max}$-redundancy enabled.

The tests were run for both the (STR) and (PCS) problem variants with and without the non-crossing constraint (NCR) enabled. These different problem configurations were tested without and with $b_{\max}$-redundancy, see Tables 2 and 3, respectively. Columns *%-Gap* show the average relative gaps between the best integer feasible solutions and the best lower bounds from the branch and bound tree, columns *Opt* list numbers of instances from each set that were solved to provable optimality.

These preliminary results show that smaller or rather sparse problem instances can be solved with our approach within reasonable time. Furthermore the results indicate that already for small instances the $b_{\max}$-redundancy constraint seems to have great impact on the solvability, although we assumed moderate numbers for $b_{\max}$ to limit the number of additional variables and constraints that have to be introduced. We also observed that for the $b_{\max}$-redundancy problem instances, the quality of the obtained feasible solutions is quite good (already in early stages of the overall process), but the achieved lower bound is poor. Additional experiments with larger problem instances are currently on the way.

Table 1: Characteristics of used instance groups.

	#	Dimension				Euclid. dim.		$b_{\max}$-neighborhood (avg.)		
Set	Inst.	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert C_1\rvert$	$\lvert C_2\rvert$	width	height	$b_{\max}$	$\lvert N(k)\rvert$	$\lvert B(k)\rvert$
G0100-I1	15	100	342	10	7	100.00	100.00	20	7.38	6.38
G0100-I2	15	100	342	8	4	100.00	100.00	20	7.69	6.69
G0100-I3	20	100	342	6	4	100.00	100.00	20	5.70	4.70
G0400-I1	15	400	1482	7	4	100.00	100.00	10	8.58	7.58
G0400-I2	15	400	1482	9	7	100.00	100.00	10	8.10	7.10
ClgSE-I1	25	190	377	4	2	189.53	164.54	30	12.93	17.05
ClgSE-I2	15	190	377	9	4	189.53	164.54	30	10.23	12.54
ClgSE-I3	15	190	377	5	3	189.53	164.54	30	10.61	13.27

Table 2: Results without $b_{\max}$-redundancy.

	#	STR		STR NCR		PCS		PCS NCR	
Set	Inst.	Opt	%-Gap	Opt	%-Gap	Opt	%-Gap	Opt	%-Gap
G0100-I1	15	15	0.00	15	0.00	15	0.00	15	0.00
G0100-I2	15	15	0.00	15	0.00	15	0.00	15	0.00
G0100-I3	20	20	0.00	20	0.00	20	0.00	20	0.00
G0400-I1	15	5	6.71	5	6.91	6	7.49	6	8.13
G0400-I2	15	4	3.68	3	9.12	4	4.04	3	6.40
ClgSE-I1	25	25	0.00	25	0.00	25	0.00	25	0.00
ClgSE-I2	15	15	0.00	15	0.00	15	0.00	15	0.00
ClgSE-I3	15	15	0.00	15	0.00	15	0.00	15	0.00

4. Conclusions and Future Outlook

We presented variants of the Steiner tree problem in graphs which considers biconnectivity and allows to make a tradeoff between survivability of the designed cable-laying and the construction costs for the connection of customers in access networks.

For this problem we proposed a directed cut formulation. Our experiments showed that the $b_{\max}$-redundancy variant is much harder to solve than the strict biconnectivity variant.

A detailed comparison of the specific advantages and limits of the directed cut and the multi-commodity flow model, especially the influence of the number of customers related to the size of the graph is ongoing work. We are also investigating if we can improve our existing approaches, e.g. by the possibilities offered by a column generation for the edge variables in our directed cut and multi-commodity flow model. Experimental work with an alternative cut formulation is currently under investigation, too. Furthermore we are considering a polyhedral analysis on our models in order to come up with additional cutting planes. In further experiments we also plan to use additional benchmark instances used in the literature for similar problems (e.g., in [**?**, **?**]).

Table 3: Results with $b_{\max}$-redundancy.

	#	STR		STR NCR		PCS		PCS NCR	
Set	Inst.	Opt	%-Gap	Opt	%-Gap	Opt	%-Gap	Opt	%-Gap
G0100-I1	15	6	7.65	5	8.61	3	10.76	2	12.32
G0100-I2	15	9	5.08	8	6.61	7	6.90	7	7.54
G0100-I3	20	20	0.00	20	0.00	20	0.00	20	0.00
ClgSE-I1	25	24	0.20	24	0.11	22	0.37	25	0.00
ClgSE-I2	15	2	18.84	1	21.08	2	18.38	1	17.31
ClgSE-I3	15	10	5.72	2	17.78	9	5.38	2	18.95