A Multi-Commodity Flow Approach for the Design of the Last Mile in Real-World Fiber Optic Networks*

Daniel Wagner[1], Günther R. Raidl[2], Ulrich Pferschy[3], Petra Mutzel[4], and Peter Bachhiesl[1]

[1] Department of Telematics and Network Engineering, Carinthia University of Applied Sciences, Austria {d.wagner|p.bachhiesl}@fh-kaernten.at
[2] Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria raidl@ads.tuwien.ac.at
[3] Institute of Statistics and Operations Research, University of Graz, Austria pferschy@uni-graz.at
[4] Chair for Algorithm Engineering, University of Dortmund, Germany petra.mutzel@cs.uni-dortmund.de

Summary. We consider a generalization of the Steiner tree problem on graphs suitable for the design of the last mile in fiber optic networks and propose a multi commodity flow formulation for the exact solution of this problem. Some experimental results are discussed.

1 Introduction

We consider the problem of finding a most cost-efficient fiber optic network to connect given customer nodes to an existing network infrastructure.

Given is a connected undirected graph $G = (V, E, c, l, p, k_{\max})$ describing the topology of the surrounding area of given customer nodes. 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 $c_e \geq 0$. The existing infrastructure is given as a subgraph $I = (V_I, E_I)$ of G. Furthermore the set of nodes V consists of the customer nodes C that shall be connected and spatial nodes S (possible Steiner nodes) resulting from the underlying spatial topology. The customer set C is the disjoint union of C_1 and C_2, whereby customers C_1 require a single connection and customers C_2 need to be redundantly connected.

* 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 real-world network design we have to consider several technical and practical requirements which make the problem more difficult. We model them by the following constraints.

- *Junction constraint:* 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 $J \subseteq V_I$.
- *Biconnectivity constraint:* Customer nodes of set C_2 need to be redundantly connected to the existing infrastructure by two node-disjoint paths.
- $k_{\max}$*-redundancy constraint:* Occasionally, the biconnectivity constraint for the nodes in the set C_2 may be relaxed in the sense that such a node may be connected to any biconnected (Steiner or customer) node v via a single path of maximum Euclidean length $k_{max}(k)$.
- *Non-crossing constraint:* Cable routes are not allowed to cross each other in a geometric sense.

The *Operative Planning Task* (OPT) consists of finding a minimum-cost connected subgraph $G' = (V_{G'}, E_{G'})$ of G, $V_{G'} \subseteq V$, $E_{G'} \subseteq E$ that connects all customers so that the technical constraints are satisfied.

An extended variant is the *Strategic Simulation Task* (SST) that represents a generalization of the prize-collecting Steiner tree problem [5]. For each customer $c \in C$ a prize $p_c \geq 0$, i.e. an intended return on investment, is introduced. The problem is to find a subgraph G' of G that connects a subset of customers, minimizing

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

so that the technical constraints are satisfied as in the OPT variant.

2 An Approach Based on Multi Commodity Flows

Multi commodity flow formulations are known to yield strong descriptions for many network design tasks including various constrained spanning and Steiner tree problems [1, 3, 6]. The basic idea is to send in G one commodity from the root node 0, which results from shrinking the infrastructure I into a single node, to each customer node $k \in C$ to be connected. This already satisfies the junction constraint.

To realize the biconnectivity constraint for each customer node $k \in C_2$, two different commodities f^k and g^k are sent in G from the root node 0 to the customer node k, which may not share any edges or have nodes other than the root 0 and k in common. With respect to the $k_{\max}$-redundancy constraint we introduce an auxiliary commodity h^k in the $k_{\max}$-neighborhood of customer k that indicates the segment where commodities f^k and g^k flow along the same path according to the definition of $k_{\max}$-redundancy.

We define the following node, edge and arc sets.

- Arcs connecting the root with spatial nodes: $A_0 = \{(0, j) \in E \mid j \in S\}$
- Edges connecting two spatial nodes with respect to customer $k \in C$: $E_S(k) = \{(i, j) \mid i, j \in V \setminus \{0, k\}\}$
- Pairs of reversely directed arcs corresponding to the edges in $E_S(k)$: $A_S(k) = \{(i, j),\ (j, i) \mid (i, j) \in V \setminus \{0, k\}\}$.
- Arcs leading to customer node k: $A(k) = \{(i, k) \in E\}$
- Let $A'(k) = A_0 \cup A_S(k) \cup A(k)$ be the set of all arcs relevant for one customer node $k \in C$.
- With respect to the non-crossing constraint let K be the set of all pairs of edges that are crossing.
- Let $N(k)$ be the set of all nodes with a path to $k \in C_2$, of length not longer than $k_{\max}(k)$ and $B(k)$ be the set of all arcs $(i, j) \in A'(k)$, $i, j \in N(k)$.

The sets K, $N(k)$ and $B(k)$ can be efficently determined in advance.

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

- Variables $x_{ij} \in \{0, 1\}$, $\forall (i, j) \in E$, indicate whether edge (i, j) is used $(x_{ij} = 1)$ or not $(x_{ij} = 0)$.
- For each $k \in C$ and each $(i, j) \in A'(k)$, we define a variable $0 \leq f^k_{ij} \leq 1$. They represent the flow of commodity f^k associated to node $k \in C$ from node i to node j via edge (i, j).
- For each $k \in C_2$ and each $(i, j) \in A'(k)$, we further define the variable $0 \leq g^k_{ij} \leq 1$. They represent the flow of a second commodity g^k for each node $k \in C_2$ in order to achieve the required redundancy.
- Variables $y_i \in \{0, 1\}$, $\forall i \in C$, indicate in the SST problem whether customer node i is connected or not. In the OPT problem, these variables are fixed to 1.
- Variables $0 \leq h^k_{ij} \leq 1$, $\forall (i, j) \in B(k)$, $\forall k \in C_2$, indicate the used edges (i, j) forming the branch (i.e., non-redundant path) leading to node k.

The formulation as an integer linear program is as follows:

Minimize

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

subject to

$$\sum_{(i,j)\in A'(k)} f^k_{ij} - \sum_{(j,i)\in A'(k)} f^k_{ji} = \begin{cases} -y_k & \text{if } j = 0 \\ y_k & \text{if } j = k \\ 0 & \text{otherwise} \end{cases} \qquad \forall k \in C,\ \forall j \in V \tag{2}$$

$$\sum_{(i,j)\in A'(k)} g^k_{ij} - \sum_{(j,i)\in A'(k)} g^k_{ji} = \begin{cases} -y_k & \text{if } j = 0 \\ y_k & \text{if } j = k \\ 0 & \text{otherwise} \end{cases} \qquad \forall k \in C_2,\ \forall j \in V \tag{3}$$

$$f_{ij}^k \le x_{ij} \qquad \forall (i,j) \in A_0 \cup A(k),\ \forall k \in C \tag{4}$$

$$f_{ij}^k + f_{ji}^k \le x_{ij} \qquad \forall (i,j) \in E_S(k),\ \forall k \in C \tag{5}$$

$$g_{ij}^k \le x_{ij} \qquad \forall (i,j) \in A_0 \cup A(k),\ \forall k \in C_2 \tag{6}$$

$$g_{ij}^k + g_{ji}^k \le x_{ij} \qquad \forall (i,j) \in E_S(k),\ \forall k \in C_2 \tag{7}$$

$$f_{ij}^k + g_{ji}^k \le x_{ij} \qquad \forall (i,j) \in A_S(k),\ \forall k \in C_2 \tag{8}$$

$$f_{ij}^k + g_{ij}^k \le x_{ij} \qquad \forall (i,j) \in A'(k) \setminus B(k),\ \forall k \in C_2 \tag{9}$$

$$f_{ij}^k + g_{ij}^k - h_{ij}^k \le x_{ij} \qquad \forall (i,j) \in B(k),\ \forall k \in C_2 \tag{10}$$

$$h_{ij}^k \le f_{ij}^k \qquad \forall (i,j) \in B(k),\ \forall k \in C_2 \tag{11}$$

$$h_{ij}^k \le g_{ij}^k \qquad \forall (i,j) \in B(k),\ \forall k \in C_2 \tag{12}$$

$$\sum_{(i,j) \in B(k)} (f_{ij}^k + g_{ij}^k - h_{ij}^k) + \sum_{(i,j) \in A'(k) \setminus B(k)} (f_{ij}^k + g_{ij}^k) \le 1 \qquad \forall i \in V \setminus \{0,k\}, \forall k \in C_2 \tag{13}$$

$$\sum_{(i,j) \in B(k)} l_{ij} h_{ij}^k \le k_{\max}(k) \qquad \forall k \in C_2 \tag{14}$$

$$x_e + x_{e'} \le 1 \qquad \forall (e,e') \in K \tag{15}$$

The objective function (1) corresponds to the usual one in the prize-collecting Steiner tree problem [4]. Equalities (2) and (3) are the flow conservation constraints, in which we distinguish root node, connection objects and possible Steiner nodes. Constraints (4) to (8) relate the flow variables to x_{ij} and prevent reversely directed flows over the same edge. Constraints (9) ensure for each arc (i,j) outside of the $k_{\max}$-neighborhood of node $k \in C_2$ that the flows of commodities f^k and g^k do not both use it.

Inequalities (10) force $h_{ij}^k = 1$ if $f_{ij}^k = 1 \wedge g_{ij}^k = 1$, i.e., both commodities associated to a $k \in C_2$ are routed over the same arc (i,j). If $f_{ij}^k = 0 \vee g_{ij}^k = 0$, inequalities (11) and (12) force $h_{ij}^k = 0$. Inequalities (13) limit the outgoing flow of each node $i \in S$ with respect to commodities f^k and g^k to one except for the case when both commodities leave the node over the same edge being part of the branch ($h_{ij}^k = 1$). Constraints (14) limit for each node $k \in C_2$ the total length of all edges for which $h_{ij}^k = 1$ to $k_{\max}(k)$, i.e., the maximum branch lengths are enforced. Finally, inequalities (15) guarantee the non-crossing constraint.

The number of variables (16) and constraints (17) used by the formulation is as follows:

$$|E| + |C| + \sum_{k \in C} |A'(k)| + \sum_{k \in C_2} (|A'(k)| + |B(k)|) \tag{16}$$

$$(|V| + |E|)\ (|C| + |C_2|) + \sum_{k \in C_2} (|A_S(k)| + |A'(k)| + 2|B(k)|) + |C_2|\ (|V| - 1) + |K| \tag{17}$$

3 Experimental Results

We tested our approach on Intel Xeon 3.6GHz machines with 4GB memory using the general purpose ILP solver CPLEX 10.0.1 from ILOG. For each individual run a time limit of 7200 seconds was used. The tests were performed with artificial grid-graph instances and instances constructed from real-world data from a German city [2].

The tests were run for both the OPT and SST problem variants with and without the non-crossing constraint (NCR) enabled. These different problem configurations were tested with (Table 2) and without (Table 1) $k_{\max}$-redundancy. The tests with $k_{\max}$-redundancy enabled used a $k_{\max}$ of 20 for the G0100-I{1,2,3} instances, 10 for G0400-I{1,2}, 30 for the ClgS-E instances, 100 for the ClgM-E instances and 150 for the ClgN1B-I{1,2} sets.

The column *%-Gap* shows the average gap between the best integer feasible solution and the best lower bound from the branch and bound tree, the column *Opt* shows the number of instances from a set that were solved to optimality.

		Dimension				OPT		OPT NCR		SST		SST NCR	
Set	#	$\|V\|$	$\|E\|$	$\|C_1\|$	$\|C_2\|$	Opt	%-Gap	Opt	%-Gap	Opt	%-Gap	Opt	%-Gap
G0100-I1	15	100	342	10	7	15	0.00	15	0.00	15	0.00	15	0.00
G0100-I2	15	100	342	8	4	15	0.00	15	0.00	15	0.00	15	0.00
G0100-I3	20	100	342	6	4	20	0.00	20	0.00	15	0.00	15	0.00
G0400-I1	15	400	1482	7	4	10	1.74	10	0.96	9	1.83	8	2.74
G0400-I2	15	400	1482	9	7	9	0.90	10	1.16	8	1.20	8	1.68
ClgS-E	25	190	377	4	2	25	0.00	25	0.00	25	0.00	25	0.00
ClgM-E	25	1757	3877	5	2	14	0.90	16	0.63	14	0.86	17	0.95
ClgN1B-I1	20	2804	3082	9	3	17	0.13	17	0.11	18	0.12	19	0.05
ClgN1B-I2	20	2804	3082	4	5	19	0.00	19	0.00	20	0.00	20	0.00
ClgN1E-I1	20	3867	8477	4	7	3	3.48	4	4.72	2	6.82	4	4.47
ClgN1E-I2	20	3867	8477	6	4	1	5.11	0	4.84	1	5.00	1	5.66
ClgN1E-I3	20	3867	8477	7	6	2	8.81	1	8.81	1	9.18	1	6.67

Table 1. Results without $k_{\max}$-redundancy.

This results show that at least smaller or rather sparse problem instances can be solved with our model within reasonable time. Some larger problem instances could be solved to provable optimality, too. Furthermore the results indicate that the $k_{\max}$-redundancy constraint has a great impact on the solvability of the problem instances, although we used moderate numbers for $k_{\max}$ to limit the number of additional variables and constraints that have to be introduced.

4 Conclusions and Future Outlook

We presented a generalization of the Steiner tree problem in graphs which allows to make a tradeoff between survivability and costs for the connection of customers in access networks.

Set	#	Dimension				OPT		OPT NCR		SST		SST NCR	
		$\|V\|$	$\|E\|$	$\|C_1\|$	$\|C_2\|$	Opt	%-Gap	Opt	%-Gap	Opt	%-Gap	Opt	%-Gap
G0100-I1	15	100	342	10	7	13	0.56	13	0.78	13	0.75	13	0.75
G0100-I2	15	100	342	8	4	15	0.00	15	0.00	15	0.00	15	0.00
G0100-I3	20	100	342	6	4	20	0.00	20	0.00	20	0.00	20	0.00
G0400-I1	15	400	1482	7	4	1	6.76	1	7.18	1	7.25	2	7.63
G0400-I2	15	400	1482	9	7	0	8.75	1	8.24	0	8.35	0	8.75
ClgS-E	25	190	377	4	2	25	0.00	25	0.00	25	0.00	25	0.00
ClgM-E	25	1757	3877	5	2	0	4.26	1	7.93	0	3.86	1	4.35
ClgN1B-I1	20	2804	3082	9	3	4	2.97	4	3.44	3	3.26	3	3.74
ClgN1B-I2	20	2804	3082	4	5	3	7.34	2	6.55	2	8.47	3	6.41

Table 2. Results with $k_{\max}$-redundancy.

Furthermore we proposed a multi commodity flow formulation for this problem. Our experiments showed that although the $k_{\max}$-redundancy constraint is a relaxation of the more strict biconnectivity constraint it is much harder to solve.

Currently, we are investigating a formulation based on directed cuts. Cut formulations have shown to be superior to flow formulations in the context of Steiner tree problems and prize-collecting Steiner tree problems. So we expect to gain improved results with this aproach.

References

1. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications.* Prentice-Hall, 1993.
2. P. Bachhiesl. The OPT- and the SST-problems for real world access network design – basic definitions and test instances. Working Report Nr. 01/2005, Department of Telematics and Network Engineering, Carinthia Tech Institute, Austria, 2005.
3. H. A. Eiselt and C.-L. Sandblom. *Integer Programming and Network Models.* Springer, 2000.
4. M. X. Goemans and D. P. Williamson. *The Primal-Dual Method for Approximation Algorithms and its Application to Network Design Problems*, chapter 4, pages 144–191. PWS Publishing, 1997.
5. I. Ljubić, R. Weiskircher, U. Pferschy, G. W. Klau, P. Mutzel, and M. Fischetti. An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem. *Mathematical Programming*, 105(2–3):427–449, February 2006.
6. M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks.* Morgan Kaufmann Publishers, 2004.