ISRN KTH/OPT SYST/FR 02/1 SE
Coden: TRITA/MAT-02-OS01
ISSN 1401-2294

Dimensioning of a multicast network that uses shortest path routing distribution trees

by

Mikael Prytz and Anders Forsgren

Optimization and Systems Theory

DEPARTMENT OF MATHEMATICS
ROYAL INSTITUTE OF TECHNOLOGY
SE-100 44 STOCKHOLM, SWEDEN

ISSN 1401-2294

DIMENSIONING OF A MULTICAST NETWORK THAT USES SHORTEST PATH ROUTING DISTRIBUTION TREES*

Mikael PRYTZ† and Anders FORSGREN‡

Technical Report TRITA-MAT-2002-OS01
Department of Mathematics
Royal Institute of Technology
February 2002

Abstract

This paper studies a special case of the minimum-cost backbone communications network capacity dimensioning problem: (i) multicast source-tree and core-tree traffic demands; (ii) discrete, non-uniform capacity levels; and, (iii) a multicast routing protocol that finds distribution trees as shortest path trees with respect to a single set of pre-configured, strictly positive, link routing weights. The last feature implies that distribution tree combinations may exist that cannot simultaneously be realized in the network. Two versions of the problem are studied: free routing weight choices, and routing weights given by selected capacity levels. The latter problem incorporates the heuristic "inverse capacity" configuration rule, which is in common use. A mixed integer programming model is developed for the problem, and a branch-and-cut method is proposed including a straight-forward, effective primal heuristic. Computational results on a set of test problems indicate that the primal heuristic rapidly gives good solutions. On the smaller test problems the algorithm could also verify optimality of the solutions.

Key words. Network optimization, telecommunications, multicast traffic, shortest path routing, branch-and-cut.

1. Introduction

The network capacity dimensioning (or provisioning) problem is a well-known network design problem. It is important for communications network service providers, both in long-term planning for green-field and existing networks, and in a more short-term routing configuration setting. This paper studies the minimum-cost backbone communications network capacity dimensioning problem for the following special case: (i) multicast source-tree and core-tree traffic demands; (ii) discrete, non-uniform capacity levels; and, (iii) a multicast routing protocol that finds distribution trees as shortest path trees with respect to a single set of pre-configured,

*Research supported by the Swedish Research Council (VR).

†Network Design, Global Services, Ericsson Radio Systems AB, SE-126 25 Stockholm, Sweden; and Optimization and Systems Theory, Department of Mathematics, Royal Institute of Technology, SE-100 44 Stockholm, Sweden (mikael.prytz@math.kth.se).

‡Optimization and Systems Theory, Department of Mathematics, Royal Institute of Technology, SE-100 44 Stockholm, Sweden (anders.forsgren@math.kth.se).

strictly positive, link routing weights. The last feature means that there may exist combinations of shortest path trees for different multicast demands that can not be simultaneously realized by the routing protocol. We study two versions of the problem, one where the routing weight set can be chosen freely to match the design, and one where the routing weights are given by the selected capacity levels. The latter problem incorporates the standard "inverse capacity" heuristic configuration rule, which is in common practice for setting routing weights.

A brief technical background on multicast traffic, multicast distribution trees, and multicast routing and forwarding is given in Section 2. The network design problem is developed in Section 3, where an example is given of a cost-optimal capacity design with routing trees that cannot be realized simultanesously in the network. Section 4 describes a mixed integer programming model for the problem, and develops some valid constraints and inequalities. A branch-and-cut method, including a primal heuristic, is proposed for the two problems in Section 5. Some computational results are described in Section 6. Finally, a summary and some conclusions are given in Section 7.

2. Multicast traffic and multicast routing

This section gives a brief technical background of multicast traffic, multicast-enabled networks, multicast distribution trees, and multicast routing and forwarding. The focus is on IP (Internet Protocol) networks, although the model and solution methods are applicable to other networks with similar functionality. A detailed treatment of IP multicasting can be found in Williamson [33].

Multicast traffic is destined for a specific group of receivers in a network or across networks. It generalizes unicast traffic, which is directed to a single receiver, and broadcast traffic, which is directed to all receivers. Some examples are real-time distribution of live video and data (e.g. video conferencing, stock tickers), network gaming, and distributed database replication. The sender and receivers that are part of a multicast transmission belong to a *multicast group* G. A *multicast-enabled* network recognizes and handles multicast traffic separately so that no duplicate copies of the same information is transmitted. Besides reducing use of link and router processing capacity, a multicast-enabled network can also lower the total transmission delay by avoiding repeated unicast transmission delays.

Traffic to and from members in a multicast group is routed through the network over a *distribution tree* of links, where each link in the tree only transmits a single copy of the traffic. There are two basic types of distribution trees: *source trees* (S,G), which are specific to each sender S, and *shared trees* $(*,G)$, which are common to all members in the multicast group G, see, e.g., Williamson [33].

A source tree (S,G) is rooted at the sender S, while a shared tree has one common root placed at a pre-configured point in the network. Depending on the multicast routing protocol, this point is called *rendez-vous point* (RP) or *core*. A sender using a shared tree $(*,G)$ transmits its traffic to the RP/core, which is then responsible for multicasting the traffic out to the remaining members in the group on the RP/core's unique distribution tree. The path taken up-link to the RP/core

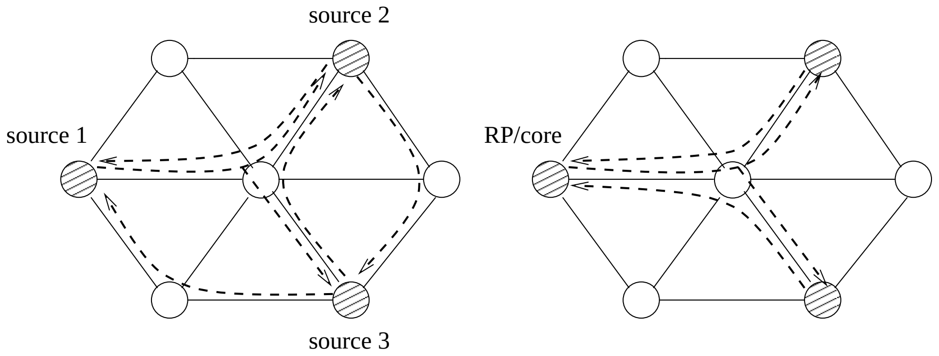

Figure 1: Source trees and a shared/core tree.

may or may not be the same as is used by the RP/core in the reverse direction. Even if the path up-link is different from the down-link path, there is a clear distinction from source trees. In a source tree, traffic to two different members use different paths, whereas in a shared tree all traffic is sent up-link to the RP. Shared trees are also called *RP-trees* or *core based trees* (CBT), while source trees are also called *shortest path trees* (SPT).

The differences between source trees and a shared tree is illustrated in Figure 1, where the three designated routers belong to some multicast group G. The network to the left uses source trees, so that, e.g., traffic from source 1 to the other members is routed over the distribution tree originating at source 1. Note that in this particular case it is only the distribution tree from source 1 that benefits fully from multicasting. The network to the right uses a shared tree where the path taken up-link to the RP/core is the same as the downlink path.

Multicast distribution trees are managed by multicast routing protocols such as PIM (Protocol Independent Multicast), DVMRP (Distance Vector Multicast Routing Protocol), CBT (Core Based Trees), and MOSPF (Multicast Open Shortest Path First), see, e.g., Stallings [31] or Williamson [33]. Multicast routing protocols handle the normal network dynamics, e.g., rerouting around failed links, but additionally also manage dynamics in group membership, i.e., group members that join and leave the group while transmission is ongoing. The distribution tree is modified by the multicast routing protocol, pruning branches with no remaining members and grafting new branches as necessary to accept new members.

The remainder of the paper focuses on the PIM protocol, which can use any underlying unicast routing protocol to find the necessary routing paths. The specific unicast protocol that we consider is OSPF (Open Shortest Path First), which is one of the most commonly used unicast intra-domain routing protocols (or Interior Gateway Protocol (IGP)) today, although any other link-state protocol that computes paths similarly, such as IS-IS (Intermediate System–Intermediate System), can be assumed. For a detailed description of OSPF see, e.g., Moy [25] or the IETF (Internet Engineering Task Force) standard RFC 2328 [26].

OSPF uses link-state information to make routing decisions. The critical piece of information for path selection is the link metric or link weight/cost, which is a strictly positive number (for OSPF actually an integer number between 1 and 2^{16}–1) configured by the network administrator. The routing paths are chosen as the *shortest paths* with respect to these weights and are calculated using Dijsktra's [10] shortest path algorithm.

The standard approach to configure the link weights w, recommended by the major backbone router vendors, is to set their values inversely proportional to the bandwidth capacities b (in bits/s) of the links, typically $w = 10^8/b$, see, e.g., Jones and Mitchell [18] and the Cisco OSPF Design Guide [9]. The motivation for this heuristic rule is to direct traffic on paths that have the greatest bandwidth capacities available. Unfortunately there are many situations when the inverse capacity rule leads to unbalanced loads and unnecessary network congestion, see, e.g., Fortz and Thorup [12]. This is one of the many reasons that traffic engineering is becoming increasingly important today. With traffic engineering, a network administrator can choose specific paths for certain demands and obtain a more balanced load pattern.

One technology that can achieve many traffic engineering objectives over any network layer protocol is the proposed IETF standard protocol MPLS (Multiprotocol Label Switching), see IETF RFC 3031 [29]. An alternative approach, which does not require MPLS and which has been studied recently, is to loosen the inverse capacity link routing weight configuration rule and instead optimize the weights to obtain better traffic loads. It has been shown that there are routing weight combinations that induce traffic patterns having overall load balances within a few percent of the optimal free routing configuration (where all routing paths can be realized), see Fortz and Thorup [12].

Finally a note on multicast addressing and packet forwarding. Each multicast group is identified by a single multicast address (a so called class D address in IP networks, see, e.g., Long [22]). Based on this group address, a multicast router determines the set of interfaces on which to forward the packet. To avoid packet cycling in the network, the router performs a Reverse Path Forwarding (RPF) check before forwarding the packet. This means that the interface leading back to the sender is determined and the packet is only forwarded if it arrived on this interface.

A potential difficulty with this simple RPF check is how to handle equal-cost multipaths, i.e., the situation when there are at least two different paths of equal length from the sender to the router. The multicast routing protocols mentioned above solve the issue by arbitrating among equal-cost multipaths. The arbitration rule can be to simply select the next-hop router with the highest numbered IP address, or to use a more involved hashing rule, see, e.g., IETF RFC 2991 [32].

3. Network design problem

We consider the minimum-cost, discrete, non-uniform capacity-level network dimensioning problem for multicast traffic demands in a backbone data communications network, e.g. tier 1 or 2 in an IP (Internet Protocol) network. The network is assumed to use a multicast routing protocol that finds distribution trees as shortest path trees based on a single set of pre-configured, strictly positive link routing

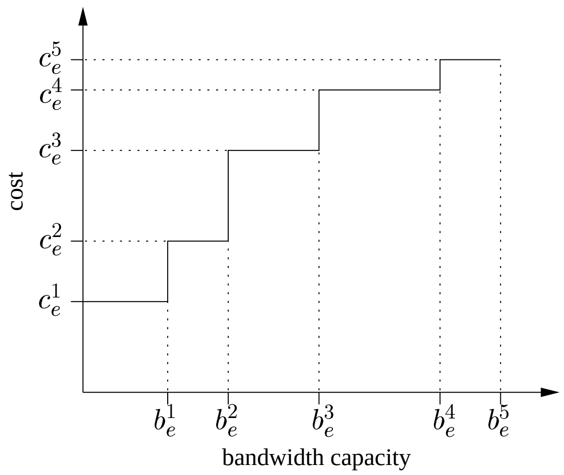

Figure 2: Capacites and costs on edge e.

weights. Note that this assumption restricts the possible combinations of distribution tree topologies that simultaneously can be realized in the network.

Two different versions of the problem are considered. In the first version, we assume that all combinations of distribution tree topologies, for which there exists at least one (freely choosable) strictly positive routing weight set, are feasible. In the second version we assume that a selected capacity level automatically implies a selected routing weight, as is the case, e.g., with the inverse capacity routing weight rule.

The locations of all backbone routers in the network are assumed given. The set of possible backbone point-to-point links between routers is also given. Link capacity is available in discrete, non-uniform capacity levels, where each level can correspond to a single physical transmission system, combinations of transmission systems, logical transmission systems, leased lines, etc. Each capacity level has an associated bandwidth capacity and a cost, see Figure 2. It is assumed that cost is increasing with bandwidth, and that the capacity is bidirectional. Note that at most one capacity level should be selected. We assume that the single link configuration problem, i.e., finding the cost-optimal configuration of systems to meet a certain bandwidth demand, has been solved and incorporated in the cost and capacity structure for our problem. The available capacity levels and costs may be different on the different possible links.

Traffic is given as a set of *multicast demands*, which aggregate subnets (or access networks) to the backbone and aggregate individual multicast group transmissions to some appropriate service or user group level. Each multicast demand is characterized by the set of backbone routers involved, the demanded bandwidth, and the type of distribution tree, shared or source, that the multicast sessions represented by the demand ultimately will use in the network. We assume that traffic in shared trees uses the same paths uplink to the RP/core as the RP/core uses downlink. One of the backbone routers is designated as *root*. For a shared tree the root corresponds to the RP/core, while for a source tree it is the sending router. Capacity for shared tree multicast demand traffic should be provided in both directions on all links. For source tree multicast demand traffic, capacity should only be provided in one direction since we assume separate demands for each sender, see Figure 1.

A dimensioning principle is assumed, such as busy hour, so that the demanded bandwidth is given by a fixed number (e.g. in Mbit/s) representing a peak level or some statistical time average. This number could be estimated using more sophisticated models for multicast traffic, capturing, e.g., dynamic traffic rates, hosts joining and leaving a transmission, and time lags associated with propagating updated multicast tree configurations through the network (e.g. due to protocol implementation details). The dimensioning principle and the service/user backbone aggregation also permit us to assume that the backbone routers in each demand are fixed.

3.1. Non-realizable shortest path routing example

To illustrate that the restriction to shortest path routing, using a single set of pre-configured, strictly positive link routing weights, actually may limit the realizable network topologies, consider the very small fully connected three-router example in Figure 3. There are two core (shared) tree multicast demands to provision for: the first should connect routers 1 and 2 reserving a capacity of 8 Mb/s, and the second should connect routers 1 and 3 reserving a capacity of 16 Mb/s. Note that the three possible links have different cost functions and that the distribution trees here reduce to paths since only two routers are involved per "tree".

Since there are only two paths to choose from for each demand, the design problem has only four possible solutions. All these have a total cost of four except for the solution indicated with dashed arrows, which has a total cost of three. Hence, this is the optimal solution to the dimensioning problem without the shortest path routing restriction. This solution can not, however, be realized using a common set of link routing weights w such that $w > 0$. For demand 1, the path via router 3 has been selected over the direct path from router 1 to router 2. Hence $w_{13} + w_{32} \leq w_{12}$. Similarly for demand 2 we have $w_{12} + w_{23} \leq w_{13}$. Putting these two inequalities together results in

$$w_{23} + w_{32} \leq 0,$$

which clearly cannot be fulfilled for $w > 0$. The remaining three solutions to the design problem are all realizable in this respect, hence they are all optimal solutions to the restricted design problem.

3.2. Related work

The network dimensioning problem for multicast traffic, but without the shortest path routing constraints, has previously been studied by Prytz and Forsgren [28]. A similar problem, also without the shortest path routing constraints, has independently been studied by Bienstock and Bley [4]. Related problems on multicast network design have been studied by Leung and Yung [20] and Chen, Günlük, and Yener [7]. Leung and Yung [20] consider the multicast routing problem for a *single* multicast group, but with an extra total delay requirement between the routers.

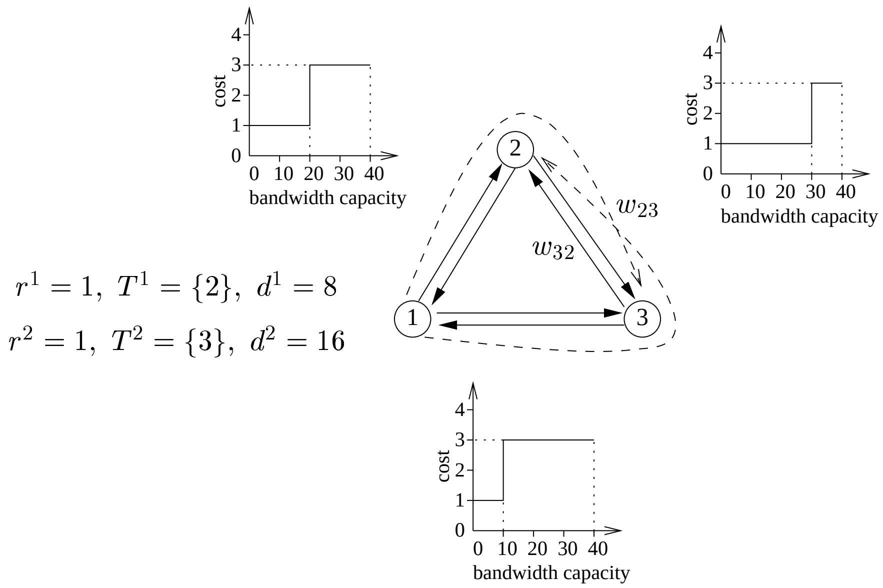

Figure 3: Example where the least-cost network design solution cannot be realized by shortest path routing on a common set of strictly positive link routing weights.

Chen, Günlük, and Yener [7] consider packing several multicast groups in an existing network such that the maximum congestion on any link is minimized.

The network design problem with shortest path routing constraints has recently been studied by Holmberg and Yuan [15]. They treat unicast, point-to-point traffic and explicitly include the possibility to use equal-cost multipaths in the model. Several heuristic methods are proposed to find good feasible solutions. The traffic engineering problem of finding shortest path routing weights in an existing network such that link congestion is minimized, has been studied by Fortz and Thorup [12]. They show that it is possible to achieve nearly optimal load balancing in a network by selecting link routing weights different from the ones given by the inverse-capacity heuristic. They also present a local search heuristic for finding a good set of link routing weights. Recently, Ericsson, Resende and Pardalos [11] have described a genetic algorithm for the same problem. Staehle, Köhler and Kohlhaas [30] have considered optimizing link routing weights to minimize the maximum utilization on any link. Lin and Wang [21] have studied a similar problem which also included multicast traffic. They suggest some primal heuristics together with a Lagrangian relaxation lower bounding method.

4. Model

The network design problem described in Section 3 is modeled here as two (mixed) integer programming problems. The first is for the case when the routing weight set is freely choosable, and the second is for the case when the selected capacity levels imply a fixed routing weight set.

Figure 4: Edges represented by two anti-parallel arcs.

Let $G = (V, E)$ be an undirected graph with $n = |V|$ vertices $v \in V$ corresponding to the routers, and $m_E = |E|$ edges $e = [i, j] \in E$ corresponding to the possible links between routers. At most one edge between any pair of vertices is used. Since source trees only need capacity in one direction, and since routing weights in OSPF are configured per outgoing interface in the routers (so that the weights may be different in the two directions of a single link), we extend the network representation to the directed graph $D = (V, A)$, where each edge $e = [i, j] \in E$ is represented by the two anti-parallel arcs $a = (i, j)$ and $a' = (j, i)$ in A, see Figure 4. The notation $(a, a') \mapsto e$, $a \mapsto e$, and $a' \mapsto e$ is used to indicate that a and a' correspond to e. The number of arcs m_A is $m_A = 2m_E$. Both G and D are used in the model as appropriate.

Let L be the set of capacity levels available for all edges e, and let

$$
\begin{aligned}
b_e^l &= \text{bandwidth capacity for level } l \text{ on edge } e,\ b_e^l > 0,\\
c_e^l &= \text{cost for capacity level } l \text{ on edge } e,\ c_e^l > 0,\\
\widetilde{w}_e^l &= \text{routing weight associated with capacity level } l \text{ on edge } e.
\end{aligned}
$$

The bandwidth capacities and costs are assumed to be in increasing order, i.e., $b_e^{l'} > b_e^l$ and $c_e^{l'} > c_e^l$ for $l' > l$. The routing weights $\widetilde{w}_e^l$ are used in the second problem version where the selected capacity levels imply the routing weights, e.g., when using the inverse capacity rule. They may be defined as $\widetilde{w}_e^l = 10^8/b_e^l$, see Jones and Mitchell [18], or to some other suitable values. Note that the same routing weights are used for both a and a' such that $(a, a') \mapsto e$.

Let K be the set of multicast demands, and let

$$
\begin{aligned}
K_c &= \textit{core} \text{ (shared) tree demands, } K_c \subseteq K,\\
K_s &= \textit{source} \text{ tree demands, } K_s = K \setminus K_c,\\
d^k &= \text{demanded bandwidth capacity for demand } k,\\
r^k &= \text{root router for demand } k,\\
T^k &= \text{remaining involved routers for demand } k,\ T^k \subseteq (V \setminus \{r^k\}),\\
R^k &= \{r^k\} \cup T^k.
\end{aligned}
$$

Note that a multicast demand k corresponds to one distribution tree. Hence, a multicast group using source trees is modeled with one demand $k \in K_s$ for each sender in the group.

Introduce the binary arc selection variables $x_a^k \in \{0, 1\}$ for $k \in K$, $a \in A$, and the binary capacity level selection variables $z_e^l \in \{0, 1\}$ for $l \in L$, $e \in E$ with the interpretation

$$x_a^k = \begin{cases} 1 & \text{if arc } a \text{ is part of selected distribution tree for demand } k, \\ 0 & \text{otherwise,} \end{cases}$$

$$z_e^l = \begin{cases} 1 & \text{if capacity level } l \text{ is used for edge } e, \\ 0 & \text{otherwise.} \end{cases}$$

For notational convenience let x^k be the vector $x^k = (x_a^k),\ a \in A$. For the first problem version where the routing weights can be chosen freely, introduce the continuous variables

$$w_a = w_{ij} = \text{routing weight assigned to arc } a = (i,j),$$

bounded by

$$w_a \geq 1, \quad a \in A. \tag{4.1}$$

The routing weights w_a are in reality integer numbers and have some upper bound $W_{\max}$ ($W = 2^{16}-1$ for OSPF), but we assume here that a fractional solution can be scaled and rounded to within $W_{\max}$ without changing the solution for the tree arc selection variables x.

The two anti-parallel arcs a and a' cannot both be on the same shortest path, with respect to w, when $w \geq 1$, as the following proposition shows.

Proposition 4.1. *The two anti-parallel arcs $a=(i,j)$ and $a'=(j,i)$ cannot both be on the same shortest path, with respect to w, from some r^k when $w > 0$.*

Proof. Let π_i^k be the final labels obtained for all vertices $i \in V$ on termination of Dijkstra's algorithm. Suppose $a = (i,j)$ and $a' = (j,i)$ are both on the shortest path from r^k. Then $\pi_j^k = \pi_i^k + w_{ij}$ and $\pi_i^k = \pi_j^k + w_{ji}$. Hence $w_{ji} + w_{ij} = 0$, which contradicts $w > 0$. ∎

Since the distribution trees induced by x^k must be such that they can be generated as shortest path trees with respect to w, a consequence of Proposition 4.1 is that at most one of the two tree design variables x_a^k and $x_{a'}^k$ can have a value equal to one, i.e.,

$$x_a^k + x_{a'}^k \leq 1, \qquad (a,a') \mapsto e. \tag{4.2}$$

Hence, there is always a one-to-one correspondence between a directed solution in D and an undirected solution in G.

The objective function in both versions of the problem is the cost of all selected capacity levels,

$$\sum_{e \in E} \sum_{l \in L} c_e^l z_e^l. \tag{4.3}$$

Capacity levels should be selected so that sufficient bandwidth is available on all edges to accommodate all multicast traffic demands,

$$\sum_{k \in K_s} d^k x_\gamma^k + \sum_{k \in K_c} d^k (x_a^k + x_{a'}^k) \leq \sum_{l \in L} b_e^l z_e^l, \qquad \begin{array}{c} e \in E,\ \gamma \in \{a, a'\}, \\ (a,a') \mapsto e. \end{array} \tag{4.4}$$

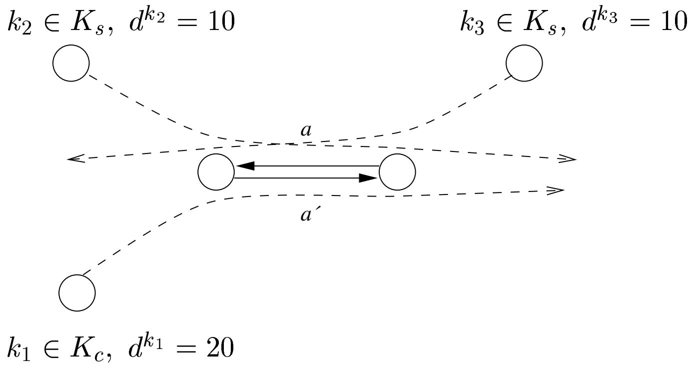

Figure 5: Two source tree demands and one core tree demand passing e, $(a, a') \mapsto e$, required capacity is 30.

Note that multicast demands $k \in K_s$ require capacity in only one of the arcs a and a', hence the need for two separate constraints (4.4) for $\gamma = a$ and $\gamma = a'$. If some other source-tree demand is routed in the reverse direction, then the two demands can share capacity. Demands $k \in K_c$ use capacity in both directions, although only one directed tree (arborescence) is used in the model to represent the distribution tree, see Figure 5. At most one capacity level should be selected per edge, i.e.,

$$\sum_{l \in L} z_e^l \leq 1, \quad e \in E, \tag{4.5}$$

The model for the shortest path constrained multicast network design problem, with freely choosable routing weights w, can now be formulated as:

$$
\begin{aligned}
(\mathrm{P}^w_{\mathrm{MCSP}}):\ \text{minimize}\quad & \sum_{e \in E} \sum_{l \in L} c_e^l z_e^l && && (4.6a)\\
\text{subject to}\quad & \sum_{k \in K_s} d^k x_\gamma^k + \sum_{k \in K_c} d^k (x_a^k + x_{a'}^k) \leq \sum_{l \in L} b_e^l z_e^l, && e \in E,\ \gamma \in \{a, a'\} \\
& && (a, a') \mapsto e, \\
& \sum_{l \in L} z_e^l \leq 1, && e \in E, \\
& x^k \in X_{\mathrm{CN}}(r^k, T^k), && k \in K, && (4.6b)\\
& (w, x^k) \in X^w_{\mathrm{SP}}(r^k, T^k), && k \in K, && (4.6c)\\
& w \geq 1, && && (4.6d)\\
& z \in \{0,1\}^{n_z},\ x \in \{0,1\}^{n_x}, && && (4.6e)
\end{aligned}
$$

where $n_x = m_A|K|$, $n_z = m_E|L|$. The second version of the problem, where the routing weights w are given by the selected capacity levels z, is similar and can be stated as:

$$
\begin{aligned}
(\mathrm{P}^z_{\text{MCSP}}):\ \text{minimize}\quad & \sum_{e\in E}\sum_{l\in L} c^l_e z^l_e && && \text{(4.7a)}\\
\text{subject to}\quad & \sum_{k\in K_s} d^k x^k_\gamma + \sum_{k\in K_c} d^k (x^k_a + x^k_{a'}) \le \sum_{l\in L} b^l_e z^l_e, && e\in E,\ \gamma\in\{a,a'\}\ (a,a')\mapsto e, &&\\
& \sum_{l\in L} z^l_e \le 1, && e\in E, &&\\
& x^k \in X_{\text{CN}}(r^k,T^k), && k\in K, && \text{(4.7b)}\\
& (z,x^k) \in X^z_{\text{SP}}(r^k,T^k), && k\in K, && \text{(4.7c)}\\
& z\in\{0,1\}^{n_z},\ x\in\{0,1\}^{n_x}. && && \text{(4.7d)}
\end{aligned}
$$

The sets $X_{\text{CN}}(r^k,T^k),\ k\in K$, in the *connectivity constraints* (4.6b) and (4.7b), and the sets $X^w_{\text{SP}}(r^k,T^k),\ k\in K$, and $X^z_{\text{SP}}(r^k,T^k),\ k\in K$, in the *shortest path routing constraints* (4.6c) and (4.7c) are discussed in the following sections.

4.1. Connectivity of distribution trees

The set $X_{\text{CN}}(r^k,T^k)$ in the connectivity constraints (4.6b) and (4.7b) contains vectors x^k such that there is a directed path from r^k to all $t\in T^k$ in the network topology $D^k(x^k)$ induced by x^k, where $D^k(x^k)\subseteq D$, $D^k(x^k)=\big(R^k\cup V(A^k(x^k)),\ A^k(x^k)\big)$ and $A^k(x^k)=\{a:\ x^k_a=1\}$. This definition allows non-tree topologies as solutions; however, since $b^l_e>0$ and $c^l_e>0$ for $e\in E$ and $l\in L$, it is always possible to find an optimal solution that induces tree network topologies for all multicast demands. Note that the connectivity constraints are the same as for the *Steiner tree problem in graphs*, see, e.g., Hwang et al. [16], and that $\mathrm{P}^w_{\text{MCSP}}$ and $\mathrm{P}^z_{\text{MCSP}}$ reduce to the Steiner tree problem, with R^1 as the terminal (required) vertices, when $|K|=1$, $|L|=1$, and $b^1_e\ge d^1,\ e\in E$. Since the Steiner tree problem is NP-hard in the general case, see, e.g., Garey and Johnson [13, pp. 208–209], this implies that $\mathrm{P}^w_{\text{MCSP}}$ and $\mathrm{P}^z_{\text{MCSP}}$ are NP-hard.

There are several ways to give a polyhedral description of $X_{\text{CN}}(r^k,T^k)$. We use a formulation based on directed graph cuts and discuss an alternative multicommodity flow formulation. Consider first the *directed cut inequalities* formulation in D. Take a subset $S\subset V$ of vertices such that $r^k\in S$ and $\bar S\cap T^k\ne\emptyset$, where $\bar S=(V\setminus S)$, and consider the cutset $\delta^+(S)=\{a=(i,j)\in A:\ i\in S,\ j\in\bar S\}$, i.e., the set of arcs leaving S (tail in S and head in $\bar S$). Then at least one of the arcs $a\in\delta^+(S)$ must be selected in order for there to be a directed path from r^k to all $t\in T^k$, i.e.,

$$\sum_{a\in\delta^+(S)} x^k_a \ge 1,\quad S\subset V,\ r^k\in S,\ \bar S\cap T^k\ne\emptyset. \tag{4.8}$$

Arc-minimal integer solutions x^k to (4.8) induce Steiner arborescences in D.

The multicommodity flow formulation of $X_{\text{CN}}(r^k,T^k)$ introduces one commodity t for each $t\in T^k$, see, e.g., Wong [34]. While keeping the arcs in the graph capacitated by the arc selection variables x_a^k, one unit of flow is sent for each commodity from r^k to the corresponding $t \in T^k$. Let $f_a^{k,t} \geq 0$ be continuous variables representing the flow on arc a for commodity t in T^k. This yields the formulation

$$N^k f^{k,t} = q^{k,t} \qquad t \in T^k, \tag{4.9a}$$
$$f_a^{k,t} \leq x_a^k, \qquad a \in A,\ t \in T^k, \tag{4.9b}$$
$$0 \leq f_a^{k,t}, \qquad a \in A,\ t \in T^k, \tag{4.9c}$$

where $q_i^{k,t} = 1$ for $i = t$, $q_i^{k,t} = 0$ for $i \in V$, $i \neq t$, $i \neq r^k$, and where N^k is the node-arc incidence matrix of D with the row corresponding to r^k deleted. Again feasible integer solutions x^k induce network topologies such that there exists a directed path from r^k to each $t \in T^k$, and arc-minimal solutions correspond to Steiner arborescences in D.

The multicommodity flow formulation (4.9) can be used as inspiration to construct a set of valid constraints for $X_{\text{CN}}(r^k, T^k)$, called *flow-balance constraints*, using only the arc selection variables x^k, see Koch and Martin [19]. Consider a vertex j and the entering arcs $\delta^-(j)$ (the arcs where j is the head vertex), and the leaving arcs $\delta^+(j)$ (the arcs where j is the tail vertex). The "in-sum" of the x_a^k-variables must be zero for r^k and equal to one for each vertex $t \in T^k$. The in-sum must also be less than or equal to one for all other vertices $j \notin R^k$, since the tree induced by x^k can enter j from at most one arc "up-stream" (the multicast split occurs "down-stream").

Next consider $j \notin R^k$: if the tree induced by x^k does not branch at j then "flow conservation" must hold at j for the x_a^k variables. Unfortunately it is not known beforehand if j is a branching vertex. However, it is always true that an "in-arc" at j can only be used if at least one "out-arc" is used. Finally the argument above can be reversed: if some "out-arc" at $j \notin R^k$ is used, then the "in-sum" must be at least one. These observations are summarized in the flow-balance constraints:

$$\sum_{a \in \delta^-(j)} x_a^k \begin{cases} = 0, & j = r^k, \\ = 1, & j \in T^k, \\ \leq 1, & j \notin R^k, \\ \leq \displaystyle\sum_{a \in \delta^+(j)} x_a^k & j \notin R^k, \\ \geq x_a^k, & a \in \delta^+(j),\ j \notin R^k. \end{cases} \tag{4.10}$$

These constraints are redundant when x^k is integer, but normally constraints like these help in tightening the LP-relaxation.

4.2. Capacity level inequalities

In this section valid inequalities for (4.6) and (4.7) are derived considering a single arc selection variable x_a^k. If this arc is used for demand k, then sufficient bandwidth capacity must be selected to accommodate for at least d^k. The inequalities are shown to be cover inequalities for the capacity constraints (4.4) and the Generalized Upper Bound (GUB) constraints (4.5). For more details on cover inequalities, see, e.g., Nemhauser and Wolsey [27, Ch. II.2.2].

Consider first the set S of feasible solutions to a 0-1 knapsack problem,

$$S = \{x \in \{0,1\}^n : \sum_{j \in N} a_j x_j \leq b\}, \tag{4.11}$$

where $N = \{1, \dots n\}$, $a_j \in Z^1_+$ for $j \in N$, and $b \in Z^1_+$. Assume that $a_j \leq b$ for all $j \in N$ (otherwise $x_j = 0$ for $a_j > b$). A set $C \subseteq N$ is called a *cover* if $\sum_{j \in C} a_j > b$. A cover C is *minimal* if $\sum_{j \in C \setminus \{i\}} a_j \leq b$ for all $i \in C$. For any cover C, the *cover inequality*,

$$\sum_{j \in C} x_j \leq |C| - 1, \tag{4.12}$$

is valid for S.

Next restrict the set S to be the feasible solutions to a 0-1 knapsack problem with GUB constraints,

$$S = \{x \in \{0,1\}^n : \sum_{j \in N} a_j x_j \leq b, \ \sum_{j \in V^k} x_j = 1, \ k \in K\}, \tag{4.13}$$

where $V^k \cap V^l = \emptyset$, $k \neq l$, $k, l \in K$, and $\bigcup_{k \in K} V^k = N$. A set $C \subseteq N$ is called a *GUB cover* if C is a cover for the knapsack constraint $\sum_{j \in N} a_j x_j \leq b$ and $|C \cap V^k| \leq 1$ for all $k \in K$. A GUB cover C is *minimal* if $\sum_{j \in C \setminus \{i\}} a_j \leq b$ for all $i \in C$. For any GUB cover C, the *GUB cover inequality* $\sum_{j \in C} x_k \leq |C| - 1$ is valid for S.

Proposition 4.2. *The capacity inequalities*

$$\sum_{l:\ b^l_e \geq d^k} z^l_e \geq x^k_a, \qquad a \mapsto e, \tag{4.14}$$

for $k \in K$ and $a \in A$ are valid for (4.6) *and* (4.7).

Proof. Consider a solution $(\bar{z}, \bar{x})$ to either (4.6) or (4.7). If $\bar{x}^k_a = 0$ then (4.14) is satisfied. Otherwise $\bar{x}^k_a = 1$ and, by (4.4), some capacity level $\bar{l}$ has been selected. The capacity $b^{\bar{l}}_e$ is sufficient for all traffic over e, in particular for the traffic d^k from demand k, i.e., $b^{\bar{l}}_e \geq d^k$, and hence $\sum_{l:\ b^l_e \geq d^k} \bar{z}^l_e \geq 1$, so that (4.14) is satisfied. ∎

Proposition 4.3. *The capacity inequalities* (4.14) *are minimal GUB cover inequalities for the the capacity constraints* (4.4) *and* (4.5).

Proof. Let $\bar{z}^l_e = 1 - z^l_e$ in (4.4) and (4.5), which gives

$$\sum_{k \in K_s} d^k x^k_\gamma + \sum_{k \in K_c} d^k (x^k_a + x^k_{a'}) + \sum_{l \in L} b^l_e \bar{z}^l_e \leq \sum_{l \in L} b^l_e, \qquad \begin{array}{c} e \in E, \ \gamma \in \{a, a'\} \\ (a, a') \mapsto e, \end{array} \tag{4.15a}$$

$$\sum_{l \in L} \bar{z}^l_e \geq |L| - 1, \quad e \in E. \tag{4.15b}$$

Constraints (4.15a) are now standard knapsack constraints in the binary variables x and $\bar{z}$ with positive coefficients and a positive capacity $\sum_{l \in L} b^l_e$. Consider a multicast demand k and an arc a. For $k \in K_s$ consider the constraint (4.15a) with $\gamma = a$ where x_a^k occurs, and for $k \in K_c$ pick either of $\gamma = a$ or $\gamma = a'$. Let $L_e^k = \{l : l \in L, b_e^l \geq d^k\}$. Suppose first that $|L_e^k| \geq |L| - 1$. Then $d^k + \sum_{l \in L_e^k} b_e^l > \sum_{l \in L} b_e^l$, and

$$x_a^k + \sum_{l \in L_e^k} \bar{z}_e^l \leq (1 + |L_e^k|) - 1 \tag{4.16}$$

is a minimal cover inequality for the knapsack constraint. Resubstituting $\bar{z}_e^l = 1 - z_e^l$ yields the capacity inequality (4.14). Next suppose $|L_e^k| \leq |L| - 2$, i.e., there are at least two levels with insufficient capacity for d^k. However, by (4.15b) all of these levels, except at most one, must be selected. Let $\bar{l}$ be the non-selected level, i.e., $\bar{l} \notin L_e^k$ and $\bar{z}_e^{\bar{l}} = 0$. Since $d^k > b_e^{\bar{l}}$ we have $d^k + \sum_{l \in L_e^k} b_e^l > \sum_{l \in L} b_e^l - \sum_{l \notin L_e^k,\ l \neq \bar{l}} b_e^l$ and (4.16) is a minimal GUB cover inequality for the knapsack constraint under (4.15b). ∎

4.3. Shortest path routing in $\mathbf{P}^w_{\mathbf{MCSP}}$, free w

The set $X_{\mathrm{SP}}^w(r^k, T^k)$ in the shortest path routing constraints (4.6c) contains pairs of vectors w and x^k such that w induces one shortest path tree (either unique or arbitrated) from r^k to all $t \in T^k$, given by applying Dijkstra's algorithm [10] in turn from r^k to each $t \in T^k$ using w as arc lengths, and such that the shortest path tree is contained in the topology given by x^k, i.e.,

$$X_{\mathrm{SP}}^w(r^k, T^k) = \{(w, x^k) : A_{\mathrm{SP}}^k(w) \text{ unique/arbitrated and } A_{\mathrm{SP}}^k(w) \subseteq A^k(x^k)\},$$

where $A_{\mathrm{SP}}^k(w) = \{a \in A : a \text{ on some } (r^k, t) - \text{shortest path w.r.t. } w,\ t \in T^k\}$, and $A^k(x^k) = \{a : x_a^k = 1\}$. Note that the same routing weights w are used by all multicast demands $k \in K$. Optimal solutions $\bar{w}$ and $\bar{x}^k$ to $\mathrm{P}_{\mathrm{MCSP}}^w$ will have $A_{\mathrm{SP}}^k(\bar{w}) = A^k(\bar{x}^k)$ by Proposition 4.1 and noting that $b_e^l > 0$ and $c_e^l > 0$ for $e \in E$ and $l \in L$.

An explicit polyhedral description of $X_{\mathrm{SP}}^w(r^k, T^k)$ for the case when $A_{\mathrm{SP}}^k(w)$ is required to be unique, can be developed from the dual of the minimum cost flow formulation for the shortest path problem from r^k to each $t \in T^k$ (with respect to w):

$$\begin{aligned} \text{minimize} \quad & w^T f^k, \\ \text{subject to} \quad & N^k f^k = q^k, \\ & f_a^k \geq 0, \qquad a \in A, \end{aligned} \tag{4.17}$$

where $q_t^k = 1$ for $t \in T^k$, and $q_i^k = 0$ for $i \in V,\ i \notin T^k,\ i \neq r^k$, and where the row corresponding to the root node r^k has been deleted in the node-arc incidence matrix N^k. The idea is to consider the constraints of the dual to (4.17) and modify them so that strict complementarity (strict because of the unique paths requirement) is coupled to the values of the design variables x_a^k, see Holmberg and Yuan [15]. Introduce the continuous variables

$$\pi_i^k \;=\; \text{shortest path dual value for vertex } i \text{ in tree } k,$$

and fix $\pi^k_{r^k} = 0$ (since one flow conservation constraint is redundant). Consider the big-M formulation

$$w_a + \pi^k_i - \pi^k_j \geq \frac{1}{M_1}(1 - x^k_a), \quad a = (i,j) \in A, \tag{4.18a}$$

$$w_a + \pi^k_i - \pi^k_j \leq M_2(1 - x^k_a), \quad a = (i,j) \in A. \tag{4.18b}$$

Proposition 4.4. *Let $\bar{w}$ and $\bar{x}$ be such that $\bar{x} \in \{0,1\}^{n_x}$, $\bar{x}^k \in X_{CN}(r^k, T^k)$, $k \in K$, and $(\bar{w}, \bar{x}^k)$ satisfies (4.18) for $k \in K$, some $\bar{\pi}$, and some choice of M_1 and M_2. Then there exists a shortest path tree, with respect to $\bar{w}$, from r^k to each $t \in T^k$ using only arcs a for which $\bar{x}^k_a = 1$, $k \in K$. Furthermore, any arcs a, for which $\bar{x}^k_a = 0$, can not be on any shortest path from r^k, $k \in K$.*

Proof. The solution $\bar{\pi}^k$ is clearly feasible for the dual of the shortest path tree problem (4.17), since, by construction, $\bar{w}_a + \bar{\pi}^k_i - \bar{\pi}^k_j \geq 0$ for all $a = (i,j) \in A$. Since $\bar{x}^k \in X_{\text{CN}}(r^k, T^k)$ there is at least one directed path $P^{k,t}$ from r^k to $t \in T^k$ such that $\bar{x}^k_a = 1$ for all $a \in P^{k,t}$. By (4.18), $\bar{\pi}^k_i - \bar{\pi}^k_j = \bar{w}_a$, $a \in P^{k,t}$ and $\bar{\pi}^k_t = \sum_{(i,j) \in P^{k,t}} (\bar{\pi}^k_i - \bar{\pi}^k_j) = \sum_{a \in P^{k,t}} \bar{w}_a$. Let $\bar{f}^k_a = |\{t : \ a \in P^{k,t}, \ t \in T^k\}|$ for $a \in P^{k,t}$, $t \in T^k$ and $\bar{f}^k_a = 0$ otherwise. Now, $\bar{f}^k$ is feasible for (4.17), $\bar{f}^k$ and $\bar{\pi}^k$ satisfies complementarity, and $\bar{w}^T \bar{f}^k = \sum_{a \in A} \bar{w}_a \bar{f}^k_a = \sum_{t \in T^k} \sum_{a \in P^{k,t}} \bar{w}_a = \sum_{t \in T^k} \bar{\pi}^k_t$, i.e., the primal and dual objective values are equal. Hence, $\bar{f}^k$ is an optimal shortest path tree solution to (4.17) with respect to $\bar{w}$, which shows the first part of the proposition. For the second part, if $\bar{x}^k_a = 0$, then (4.18) implies $\bar{w}_a + \bar{\pi}^k_i - \bar{\pi}^k_j > 0$. Hence, by complementarity, we must have $\bar{f}^k_a = 0$. ∎

Corollary 4.1. *If $\bar{w}$ and $\bar{x}$ are as in Proposition 4.4, for some $\bar{\pi}$ and some choice of M_1 and M_2, and, in addition, all sets $\{a : \ \bar{x}^k_a = 1\}$ have minimal cardinality for each $k \in K$, then the tree induced by $\bar{x}^k$ must be equal to the unique shortest path tree from r^k to all $t \in T^k$ with respect to $\bar{w}$ for all $k \in K$.* ∎

A straightforward upper bound on M_2 is $|A| W_{\max}$ (since a path can use at most all arcs), while an upper bound for M_1 depends on the floating point representation accuracy. Unfortunately big-M formulations in integer programs often have poor LP relaxations. Since the scaling factor in (4.18a) together with (4.18b) is $M_1 M_2$, it is likely that the condition number of the constraint matrix is large, and numerical difficulties can be expected when solving the LP relaxation. Numerical computations on some test problems confirm these expectations, see Section 6.

If the unique shortest paths requirement is relaxed, i.e., if the right hand side of (4.18a) is replaced by zero, then the remaining big-M formulation,

$$w_a + \pi^k_i - \pi^k_j \geq 0, \qquad a = (i,j) \in A, \tag{4.19a}$$

$$w_a + \pi^k_i - \pi^k_j \leq M_2(1 - x^k_a), \quad a = (i,j) \in A, \tag{4.19b}$$

can be made more tight. Since w is continuous, $W_{\max}$ can be scaled down to a small value, say $W_{\max} = 2$, which leads to a much smaller value for M_2. The formulation (4.19) is valid for $\mathrm{P}^w_{\text{MCSP}}$ both under unique paths and some path arbitration rule.

4.4. Sub-path constraints

A set of valid inequalities for $X^w_{\text{SP}}(r^k, T^k)$ are the so called *sub-path constraints*, which are derived from the observation that if one multicast demand is routed so that it reaches the source of another demand, then the subsequent paths taken by the two demands must be equal. Similarly, if the last hop in the path for some demand passes via the destination router of some other demand, then the paths preceding the router for the two demands must be equal. These observations lead to the constraints

$$x^{k_1}_a + x^{k_2}_{\widehat{a}} - x^{k_1}_{\widehat{a}} \leq 1, \quad \begin{array}{l} r^{k_1} \neq r^{k_2},\ t_1 = t_2, \\ a = (r^{k_1}, r^{k_2}),\ \widehat{a} \neq a,\ \widehat{a} \in A, \end{array} \tag{4.20a}$$

$$x^{k_1}_a + x^{k_2}_{\widehat{a}} - x^{k_1}_{\widehat{a}} \leq 1, \quad \begin{array}{l} r^{k_1} = r^{k_2},\ t_1 \neq t_2, \\ a = (\widehat{t}, t),\ \widehat{a} \neq a,\ \widehat{a} \in A, \end{array} \tag{4.20b}$$

for all demands $k_1 \in K$ and $k_2 \in K$, and all destination vertices $t_1 \in T^{k_1}$ and $t_2 \in T^{k_2}$. The constraints are similar to the sub-path constraints given by Staehle, Köhler and Kohlhaas in [30] for a related unicast traffic network design problem, which has demands between all possible combinations of sending and receiving routers. The constraints (4.20) do not completely describe $X^w_{\text{SP}}(r^k, T^k)$ as is shown in the following small example, adapted from an example by Gourdin [14], see Figure 6. There are five point-to-point demands in a network with six routers, which yields a total of 136 possible sub-path constraints (4.20). The solution in the figure satisfies all sub-path constraints, but it can not be generated with unique shortest paths. To be unique, the paths in the figure must satisfy

$$\begin{aligned} w_{12} + w_{23} &< w_{15} + w_{53}, && \text{(demand 1)} \\ w_{53} + w_{34} &< w_{56} + w_{64}, && \text{(demand 5)} \\ w_{15} + w_{56} &< w_{12} + w_{26}, && \text{(demand 3)} \\ w_{26} + w_{64} &< w_{23} + w_{34}, && \text{(demand 4)} \end{aligned}$$

which, when taken together, leads to the contradiction

$$\begin{aligned} w_{12} + w_{23} + w_{34} &< w_{15} + w_{53} + w_{34} \\ &< w_{15} + w_{56} + w_{64} \\ &< w_{12} + w_{26} + w_{64} < w_{12} + w_{23} + w_{34}. \end{aligned}$$

As seen above, neither the relaxed shortest path dual constraints (4.19) nor the sub-path constraints (4.20) are sufficient to describe unique shortest path routing. A natural question is whether either set of constraints includes the other. The following two counter examples demonstrate that this is not the case.

Both examples have two point-to-point demands, and the proposed integer solutions satisfy the connectivity constraints $x^k \in X_{\text{CN}}(r^k, T^k),\ k \in K$. The solution in the first example, Figure 7, satisfies the relaxed shortest path dual constraints (4.19) but violates the sub-path constraints (4.20). In the second example, Figure 8, there are no sub-path constraints, but the solution violates the shortest path dual constraints (for the same reason as in the example of Figure 3).

4.5. Unique shortest path routing separation LP for $\mathbf{P}^w_{\mathbf{MCSP}}$

To avoid the numerical drawbacks of the full big-M formulation in (4.18), we propose using the relaxed big-M constraints (4.19) with a small value for M_2, and the sub-path constraints (4.20). A feasible solution $(\bar{w}, \bar{x})$, such that $\bar{x}$ is integer and satisfies the connectivity constraints $x^k \in X_{\text{CN}}(r^k, T^k),\ k \in K$, can then be separated from $X^w_{\text{SP}}(r^k, T^k)$ by solving a feasibility problem for a set of linear inequalities generated from $\bar{x}$. This approach is an extension to the multicast case from similar work presented by Bley [6] for a related unicast network design problem.

Let $w_a,\ a \in A$, and $\pi^k_i,\ i \in V,\ k \in K$, again be variables for the routing weights and the shortest path dual variables. If the system

$$
\begin{aligned}
\pi^k_i + w_a - \pi^k_j &= 0, &\quad& \text{for } k \in K,\ a \in A, \text{ such that } \bar{x}^k_a = 1, && \text{(4.21a)}\\
\pi^k_i + w_a - \pi^k_j &\geq 1, && \text{for } k \in K,\ a \in A, \text{ such that } \bar{x}^k_a = 0, && \text{(4.21b)}\\
w_a &\geq 1, && a \in A, && \text{(4.21c)}\\
\pi^k_i &\geq 0, && k \in K,\ i \in V, && \text{(4.21d)}
\end{aligned}
$$

where $a = (i, j)$, is feasible, then the routing weights w will realize the distribution

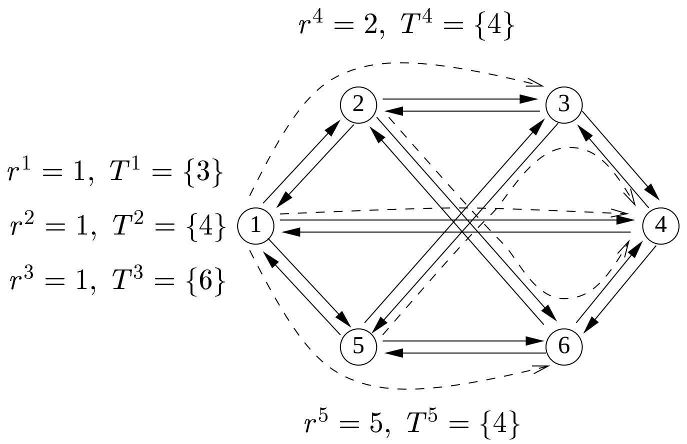

Figure 6: Example with a solution that satisfies the sub-path constraints, but which cannot be generated with unique shortest paths.

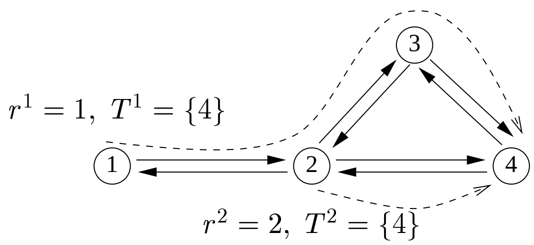

Figure 7: Example with a solution satisfying shortest path dual constraints, but violating sub-path constraints.

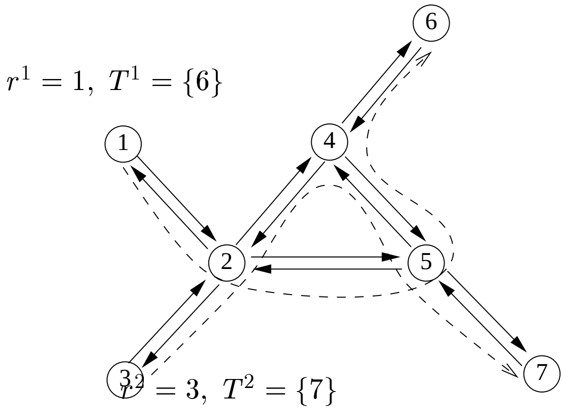

Figure 8: Example without sub-path constraints and with a solution that violates shortest path dual constraints.

trees induced by $\bar{x}$, and these trees will be unique.

If, on the contrary, the system (4.21) is infeasible, then there is no routing weight set w that uniquely can realize $\bar{x}$. A valid inequality for $X^w_{\text{SP}}(r^k, T^k)$ can be generated that separates $\bar{x}$. Find an infeasibility system of rows I in (4.21a) and (4.21b), where $I = I_1 \cup I_0$, $I_1 = \{(k,a) :\ \bar{x}^k_a = 1\}$, and $I_0 = \{(k,a) :\ \bar{x}^k_a = 0\}$, and then form the inequality

$$\sum_{(k,a)\in I_1} x^k_a - \sum_{(k,a)\in I_0} x^k_a \le |I_1| - 1, \tag{4.22}$$

which is violated by $\bar{x}$, but is satisfied for any feasible solution to (4.6).

The quality of (4.22) can be improved by generating the "smallest" possible irreducible infeasibility system I. Furthermore, it follows from the "flow balance" constraints (4.10) that all $(k,a) \in I_0$, such that a is incident to r^k, can be removed from I_0 since the corresponding x^k_a will always be zero. Also, if $(k,a) \in I_1,\ a = (i,j)$, then all $(k,\widehat{a}) \in I_0,\ \widehat{a} = (i',j)$ can be removed from I_0 since, by (4.10), they will automatically have value zero when $x^k_a = 1$.

If the system (4.21) is feasible we would like to find w as small as possible to ensure $w \le W_{max}$. The two objectives, determining feasibility and minimizing w, can be achieved by solving the linear program

$$\begin{array}{lll}
\text{minimize} & \sum\limits_{a\in A} w_a & \\
\text{subject to} & \pi^k_i + w_a - \pi^k_j = 0, & \text{for } k \in K,\ a \in A \text{ such that } \bar{x}^k_a = 1, \\
& \pi^k_i + w_a - \pi^k_j \ge 1, & \text{for } k \in K,\ a \in A \text{ such that } \bar{x}^k_a = 0, \\
& w_a \ge 1, & a \in A, \\
& \pi^k_i \ge 0, & k \in K,\ i \in V.
\end{array} \tag{4.23}$$

Since $w \ge 1$, (4.23) has an optimal solution if (4.21) is feasible.

4.6. Shortest path routing in $\mathrm{P}^z_{\mathrm{MCSP}}$, w given by z

In $\mathrm{P}^z_{\mathrm{MCSP}}$, where the routing weights are given by the choice of capacity levels z, the set $X^z_{\mathrm{SP}}(r^k, T^k)$ in (4.7c) is similar to the set $X^w_{\mathrm{SP}}(r^k, T^k)$ described above, except that $X^z_{\mathrm{SP}}(r^k, T^k)$ contains pairs of vectors z and x^k, and that the routing weights $w(z)$, for integer z, are given by

$$w_a(z) = w_{a'}(z) = \begin{cases} \sum_{l\in L} \widetilde{w}^l_e z^l_e, & \text{if } \sum_{l\in L} z^l_e = 1, \\ \infty & \text{if } \sum_{l\in L} z^l_e = 0, \end{cases} \qquad (a, a') \mapsto e,\ e \in E, \tag{4.24}$$

where $\widetilde{w}^l_e$ was defined in Section 4 as the routing weight associated with capacity level l on edge e.

Again, $w(z)$ should induce one shortest path tree (either unique or arbitrated) from r^k to all $t \in T^k$, given by applying Dijkstra's algorithm [10] in turn from r^k to each $t \in T^k$ using $w(z)$ as arc lengths, such that the shortest path tree is contained in the topology given by x^k, i.e., $A^k_{\mathrm{SP}}(w(z))$ unique or arbitrated and $A^k_{\mathrm{SP}}(w(z)) \subseteq A^k(x^k)$, where $A^k_{\mathrm{SP}}(w(z)) = \{a \in A : \ a \text{ on some } (r^k, t) -$ shortest path w.r.t. $w(z),\ t \in T^k\}$, and $A^k(x^k) = \{a :\ x^k_a = 1\}$. Note that one path is selected for each (r^k, t)-pair, which corresponds to vertex, i.e., integer, optimal solutions to (4.17). The same routing weights $w(z)$ are used by all multicast demands $k \in K$. Optimal solutions $\bar{x}^k$ and $\bar{z}$ to $\mathrm{P}^z_{\mathrm{MCSP}}$ will be arc-minimal and have $A^k_{\mathrm{SP}}(w(\bar{z})) = A^k(\bar{x}^k)$ by Proposition 4.1 and using $b^l_e > 0$ and $c^l_e > 0$ for $e \in E$ and $l \in L$.

The unique shortest path approach can be modeled by changing constraints (4.18) to

$$\sum_{l\in L} \widetilde{w}^l_e z^l_e + M_2(1 - \sum_{l\in L} z^l_e) + \pi^k_i - \pi^k_j \geq \frac{1}{M_1}(1 - x^k_a), \quad a \in A,\ a \mapsto e, \tag{4.25a}$$

$$\sum_{l\in L} \widetilde{w}^l_e z^l_e + M_2(1 - \sum_{l\in L} z^l_e) + \pi^k_i - \pi^k_j \leq M_2(1 - x^k_a), \quad a \in A,\ a \mapsto e, \tag{4.25b}$$

where $a = (i, j)$, and where the term $M_2(1 - \sum_{l\in L} z^l_e)$ corresponds to the "infinite" routing weight case when no bandwidth capacity is installed. These constraints can be relaxed, as in (4.19), by replacing the right hand side of (4.18a) with zero. The sub-path constraints (4.20) are also valid for $X^z_{\mathrm{SP}}(r^k, T^k)$ and can be used.

4.7. Valid inequalities for $\mathrm{P}^z_{\mathrm{MCSP}}$

Since an integer solution $\bar{z}$ to $\mathrm{P}^z_{\mathrm{MCSP}}$ uniquely determines $\bar{w}$ through (4.24) and, hence, determines the shortest path routing distribution trees given by $\bar{x}$, it is natural to consider valid inequalities in z only. The first set of inequalities are a reformulation of the capacity cut inequalities in Barahona [3] to the non-uniform capacity level case. See also Bienstock et al. [5] where similar inequalities are used.

Let U be any subset of vertices and let $\bar{U} = (V \setminus U)$. The cut inequality

$$\sum_{e\in\delta(U)} \sum_{l\in L} b^l_e z^l_e \geq \sum_{k\in\gamma(U)} d^k, \tag{4.26}$$

where $\delta(U)$ is the cutset of edges between U and $\bar{U}$, and $\gamma(U) = \{k : r^k \in U, T^k \cap \bar{U} \neq \emptyset\}$, is valid for P^z_{MCSP}. The interpretation is that the total capacity over the cutset $\delta(U)$ must be sufficient for the demanded capacity of all multicast demands separated by the cut. If an integer solution $\bar{z}$ is such that there does not exist a directed path from some r^k to some $t \in T^k$ with respect to $w(\bar{z})$, then a separating inequality (4.26) can be generated with $\bar{U}$ as the unlabeled nodes on termination of Dijkstra's algorithm.

Next suppose that $w(\bar{z})$ are such that all $t \in T^k$ are reachable as shortest paths from all r^k, $k \in K$ with respect to $w(\bar{z})$, i.e., problem (4.17) has optimal shortest path tree solutions $\bar{f}^k$, $k \in K$, which are either unique or arbitrated. The corresponding solution $\bar{x}$ is given by

$$\bar{x}^k_a = \begin{cases} 1 & \text{if } \bar{f}^k_a \geq 1, \\ 0 & \text{if } \bar{f}^k_a = 0. \end{cases}$$

This solution must be such that $(\bar{z}, \bar{x})$ satisfies the sufficient bandwidth capacity constraints (4.4).

Proposition 4.5. *Suppose $\bar{z}$ satisfies $\bar{z} \in \{0,1\}^{n_z}$ and $\sum_{l \in L} \bar{z}^l_e \leq 1$ for $e \in E$. Let $w(\bar{z})$ be given by* (4.24) *and suppose that* (4.17) *has one optimal shortest path tree solution $\bar{f}^k$ (unique or arbitrated), with respect to $w(\bar{z})$, for each $k \in K$. Let $\bar{x}$ be the solution induced by $\bar{f}$ and suppose that $(\bar{z}, \bar{x})$ violates the sufficient bandwidth capacity constraints* (4.4). *Then,*

$$\sum_{(e,l) \in I_1} z^l_e - \sum_{e \in I_0} \sum_{l \in L} z^l_e \leq |I_1| - 1, \tag{4.27}$$

where $I_1 = \{(e,l) : \bar{z}^l_e = 1\}$ and $I_0 = \{e : \sum_{l \in L} \bar{z}^l_e = 0\}$, is violated by $\bar{z}$ but satisfied by any feasible solution to (4.7).

Proof. In any integer solution z to (4.7) we have, for any e, either all $z^l_e = 0$, $l \in L$, or exactly one $z^{\widehat{l}}_e = 1$, and $z^l_e = 0$, $l \neq \widehat{l}$. The inequality (4.27) is only violated for the solution $z^l_e = 1$, $(e,l) \in I_1$ and $z^l_e = 0$ for $e \in I_0$, $l \in L$, i.e., for $z = \bar{z}$, which, by assumption, is not a solution to (4.7). ∎

If either $\bar{x}^k_a = 1$ or $\bar{x}^k_{a'} = 1$ for all multicast demands $k \in K$ on some edge e, $(a, a') \mapsto e$, then changing the capacity level l on e so that $w_a(\bar{z})$ and $w_{a'}(\bar{z})$ decreases will not change the shortest path tree solutions, and if the new capacity level is still not sufficient for $\bar{x}$, then another separating inequality similar to (4.27) can be generated.

The following valid inequalities can be used to connect the "correct" x with a given z.

Proposition 4.6. *Suppose $\bar{z}$ satisfies $\bar{z} \in \{0,1\}^{n_z}$ and* (4.5)*, let $w(\bar{z})$ be given by* (4.24)*, and suppose that (4.17) has one optimal shortest path tree solution $\bar{f}^k$ (unique or arbitrated), with respect to $w(\bar{z})$, for all $k \in K$. Let $\bar{x}$ be the solution* *induced by $\bar{f}$ and assume that $(\bar{z}, \bar{x})$ satisfies the sufficient bandwidth capacity constraints* (4.4)*. Then,*

$$\begin{aligned} &\sum_{(e,l)\in I_1} z_e^l - \sum_{e\in I_0}\sum_{l\in L} z_e^l + x_a^k \le |I_1|, && \text{if } \bar{x}_a^k = 0, \\ &\sum_{(e,l)\in I_1} z_e^l - \sum_{e\in I_0}\sum_{l\in L} z_e^l + (1 - x_a^k) \le |I_1|, && \text{if } \bar{x}_a^k = 1, \end{aligned} \tag{4.28}$$

where $I_1 = \{(e,l) : \bar{z}_e^l = 1\}$ and $I_0 = \{e : \sum_{l\in L} \bar{z}_e^l = 0\}$, are $m_E|K|$ valid inequalities for any solution to (4.7)*.*

Proof. Let $z = \bar{z}$, which gives $\sum_{(e,l)\in I_1} z_e^l - \sum_{e\in I_0}\sum_{l\in L} z_e^l = |I_1|$. The only solution x that satisfies (4.28) is $x = \bar{x}$, and, by assumption, $(\bar{z}, \bar{x})$ is a feasible solution to (4.7). Next take any feasible solution (z, x) to (4.7) such that $z \neq \bar{z}$. Then $\sum_{(e,l)\in I_1} z_e^l - \sum_{e\in I_0}\sum_{l\in L} z_e^l \le |I_1| - 1$, so (4.28) is satisfied since $x \in \{0,1\}^{n_x}$.
∎

5. Solving $\mathbf{P}^w_{\mathbf{MCSP}}$ and $\mathbf{P}^z_{\mathbf{MCSP}}$

The strategies proposed to solve the two versions of the problem, $\mathrm{P}^w_{\mathrm{MCSP}}$ and $\mathrm{P}^z_{\mathrm{MCSP}}$, are slightly different, although the basic approaches are the same. For both problems, the multicommodity flow formulation (4.9) for $X_{\mathrm{CN}}(r^k, T^k)$ and the full big-M formulations (4.18) and (4.25) for $X^w_{\mathrm{SP}}(r^k, T^k)$ and $X^z_{\mathrm{SP}}(r^k, T^k)$ respectively, yield a compact mixed integer program (MIP) that can be solved by any general purpose MIP solver. As already mentioned, the two big-Ms in (4.18) and (4.25) unfortunately introduce ill-conditioning, which leads to numerical difficulties when solving the LP relaxations. To overcome these difficulties, we propose using a branch-and-cut approach.

5.1. Branch-and-cut

Although the Steiner tree problem in graphs is NP-hard, several effective algorithms have been described in the literature that can solve fairly large instances of the problem. Many of these are based on the directed cut formulation (4.8) for $X_{\mathrm{CN}}(r^k, T^k)$. Koch and Martin [19] have described a branch-and-cut algorithm that starts with a subset of the constraints (4.8), generates violated constraints as needed, and finally uses branch-and-bound (B&B) to obtain integer solutions. Another branch-and-cut algorithm for the Steiner tree problem in graphs has been suggested by Lucena and Beasley in [23]. Based on the success of these algorithms we have chosen to use the cut inequalities formulation for $X_{\mathrm{CN}}(r^k, T^k)$.

In our branch-and-cut strategy we solve an LP relaxation of $\mathrm{P}^w_{\mathrm{MCSP}}$ and $\mathrm{P}^z_{\mathrm{MCSP}}$ in the variables z, x, π, and, for $\mathrm{P}^w_{\mathrm{MCSP}}$, in w, initially dropping the cut inequalities (4.8), but including the flow-balance constraints (4.10), the relaxed shortest path routing big-M constraints (4.19) (for $\mathrm{P}^z_{\mathrm{MCSP}}$, the equivalent relaxation of (4.25)), the sub-path constraints (4.20), and the capacity level inequalities (4.14). All LP relaxations are solved using dual simplex.

Each solution $\bar{x}$ to the LP relaxation is used to separate violated cut inequalities (4.8) for each $k \in K$ by solving maxflow/mincut problems in D from r^k to each $t \in T^k$ using $\bar{x}^k$ as *capacities* on the arcs. As suggested by Koch and Martin [19], a tiny ϵ-capacity is also added to all arcs in order to find the arc-minimal cut among all the capacity-minimal cuts. If any of the mincut solutions have capacities less than one, then violated cut inequalities, with $\delta^+(S)$ equal to the arcs in the mincuts, have been found. These cut inequalities are added to the LP, which is then resolved, and the new solution is again checked for violation of cut inequalities. This process continues until $\bar{x}$ does not violate any cut inequalities.

If a solution $(\bar{z}, \bar{x})$ to the LP relaxation is not integer, then B&B is used, giving priority in $\mathrm{P}^z_{\mathrm{MCSP}}$ to the capacity level selection variables z and using the fact that the "select at most one capacity level" constraints (4.5) are generalized upper bound (GUB) constraints. The B&B node selection strategy is best-bound first, slightly modified by a lowest-depth rule among active nodes with bounds close to the best-bound (tolerance 10^{-4}).

When an integer solution $\bar{x}$ to $\mathrm{P}^w_{\mathrm{MCSP}}$ is obtained, the unique shortest path routing separation LP problem (4.23) is solved. If this problem is infeasible, it means that $\bar{x}$ can not be realized by unique shortest path trees, and a violated inequality (4.22) is found as described in Section 4.5. If the separation LP is feasible and $\bar{z}$ is also integer, then $(\bar{z}, \bar{x})$ is an optimal solution to $\mathrm{P}^w_{\mathrm{MCSP}}$. The optimal routing weights are given by the solution to the separation LP.

Similarly, when an integer solution $\bar{z}$ is obtained for $\mathrm{P}^z_{\mathrm{MCSP}}$, the implied routing weights $w(z)$ in (4.24) are computed and Dijkstra's algorithm is used to find $\widehat{x}^k$, $k \in K$, corresponding to the shortest path trees from r^k to each $t \in T^k$, $k \in K$. If all shortest path trees are unique, and if $\bar{x}$ is integer and $\bar{x} = \widehat{x}$, then $(\bar{z}, \bar{x})$ is an optimal solution to $\mathrm{P}^z_{\mathrm{MCSP}}$. If $(\bar{z}, \bar{x})$ does not satisfy the sufficient bandwidth capacity constraints (4.4), then inequalities (4.27) are added to the problem. Note that (4.4) is satisfied by $(\bar{z}, \bar{x})$ but not necessarily by $(\bar{z}, \widehat{x})$. The same type of inequality can also be added if some shortest path tree is not unique. Finally, if all shortest path trees are unique and $(\bar{z}, \widehat{x})$ satisfies (4.4) but $\widehat{x} \neq \bar{x}$, then the violated inequalities (4.28) can be added to couple the correct x to $\bar{z}$. The problem is again resolved and checked for any constraint violation.

5.2. Primal heuristic

Feasible solutions to $\mathrm{P}^w_{\mathrm{MCSP}}$ and $\mathrm{P}^z_{\mathrm{MCSP}}$ are generated frequently in the branch-and-cut method. For any set of routing weights $w \geq 1$, Dijkstra's algorithm is used to find shortest path trees $\widehat{x}^k$, $k \in K$. If the trees are unique a primal feasible solution can be generated. Set $\widetilde{x} = \widehat{x}$, compute the total demanded bandwidths $\widetilde{d}_e$ on each edge e as the maximum of $\sum_{k \in K_s} d^k \widehat{x}^k_\gamma + \sum_{k \in K_c} d^k (x^k_a + x^k_{a'})$ for $\gamma = a$ and $\gamma = a'$ (where $(a, a') \mapsto e$), and set $\widetilde{z}_e$ to $\widetilde{z}^{\tilde{l}}_e \leftarrow 1$, $\tilde{l} = \min_{l \in L}\{l : b^l_e \geq \widetilde{d}_e\}$ and $\widetilde{z}^l_e \leftarrow 0$, $l \neq \tilde{l}$. In $\mathrm{P}^w_{\mathrm{MCSP}}$ the w is included in the model and can be retrieved for the heuristic. A feasible solution to the shortest path separation LP (4.23) also yields a w that can be used in the heuristic.

The routing weights w are not available directly in $\mathrm{P}^z_{\mathrm{MCSP}}$. For any integer solution z the routing weights $w(z)$ are given by (4.24). If $w(z)$ are used as routing weights a solution $\widetilde{x}$ is obtained, and if this solution fits in the capacities given by z then $\widetilde{x}$ is a feasible solution to P^z_{MCSP}. If not, the heuristic gives a solution $\widetilde{z}$, which can be used to find new routing weights $w(\widetilde{z})$ that in turn can be plugged into the heuristic. For non-integer solutions z we apply two different rounding strategies. If the fractional solution is sufficiently small, then no capacity level is selected. Otherwise the level is selected corresponding to the largest fractional value or the largest cost- weighted fractional value. For both strategies we iterate twice over the heuristic as described above.

5.3. Implementation

The overall program flow for the branch-and-cut method has been implemented in MATLAB 6 [24]. The LP relaxations and the shortest path routing separation LP are solved using the fast and robust simplex solvers in CPLEX 6.5 [17]. A MEX interface has been written to use CPLEX interactively from within MATLAB. The interface permits several CPLEX problem objects to reside in memory simultaneously and allows CPLEX to warm start from previous solutions, which decreases the total solution time in the branch-and-cut approach dramatically. The CPLEX Irreducible Infeasible Set (IIS) finder is used when the unique shortest path routing problem (4.23) is infeasible. The IIS finder computes the smallest possible subset of constraints that induce the infeasibility. This infeasibility subset is used to construct violated shortest path routing inequalities (4.22).

The maxflow/mincut problem that separates violated cut inequalities (4.8) is solved by an implementation in C, accessed via a MEX interface, of a highest-label preflow-push algorithm by Cherkassky and Goldberg [8]. The shortest path computations necessary for the primal heuristic and for separating unique shortest path trees in P^z_{MCSP}, are found using a MATLAB implementation of a modified label correcting algorithm with a first-in first-out queue, see e.g., Ahuja, Magnanti and Orlin [1].

6. Computations

The branch-and-cut algorithm has been tested on a set of problems that have been generated from a small number of real network problems problems provided by Ericsson Radio Systems [2]. The problems are a subset of the test problems used in the previous study of the non-routing constrained problem, see Prytz and Forsgren [28]. The network structures, the number and locations of backbone routers for each multicast demand, and the demanded bandwidths have been randomly generated. One half of the test problems have a real, non-uniform cost and bandwidth capacity structure, while the other half have a uniform structure, i.e., the capacity levels are multiples of a base level.

All computations have been performed on a shared SUN Ultra-Enterprise 4000 server using one 250 MHz SPARC CPU processor (83 MHz system clock and 1 Gb of shared, internal memory). A CPU time limit of 10 000s. has been used for all problems. The overall optimality tolerance has been 1%, while the LP subproblems have been solved to within an optimality tolerance of 10^{-6}. The absolute integer feasibility tolerance has been 10^{-5}. The feasibility tolerance for violated Steiner cut inequalities has been 10^{-4}. The ϵ-capacity added to the maxflow/mincut capacities (to find an arc-minimal cut among the capacity-minimal cuts) has been 10^{-5}.

Test results for $\mathrm{P}^w_{\mathrm{MCSP}}$ are reported in Table 1 and Table 2 for the non-uniform and uniform cost and capacity level test problems respectively. Table 3 and Table 4 contains the corresponding results for $\mathrm{P}^z_{\mathrm{MCSP}}$. The number of routers is given by $|V|$, the number of possible edges by $|E|$, and the number of multicast demands by $|K|$. The total number of added connectivity cut inequalities (4.8) is listed under *cuts/CN*, and the total number of added shortest path routing inequalities is given by *cuts/SP*. Solution time and the number of examined B&B-nodes is also given. The objective values are reported for the root node relaxation, and for the best found lower and upper bound solutions obtained within the time limit.

Both $\mathrm{P}^w_{\mathrm{MCSP}}$ and $\mathrm{P}^z_{\mathrm{MCSP}}$ are difficult problems (having the NP-hard Steiner tree problem as a special case), which is reflected in the test results. On the smaller test problems the root node lower bound is quite good, and only a small number of B&B-iterations are required to close the optimality gap (to within the 1% tolerance). The optimal solution, or a solution very close to the optimal, has been found during the first few B&B-iterations on all of these smaller problems, which seems to indicate that the primal heuristic described in Section 5.2 is effective. The remaining iterations have been spent improving the lower bound. On the larger test problems the quality of the lower bound deteriorates, and the algorithm has been unable to verify optimality within the given time limit. Note that these problems are quite large: test problem #20 has, e.g, in the root node, 4 618 variables, of which 4 104 are binary, and 34 676 constraints. The effect on the lower bound can be seen, e.g., in the medium sized test problem #6, Table 1, where the optimal solution has been verified, and where the root node lower bound gap relative to the optimal objective value is 30%.

The total number of added connectivity cut inequalities (4.8) is very small compared to the potential number of these inequalities. Most of the cuts have been added in the root node and early iterations. The total number of added shortest path routing inequalities is small for $\mathrm{P}^w_{\mathrm{MCSP}}$, and zero on most of the smaller problems. On some problems the shortest path routing constraints are sufficiently captured by the subpath constraints (4.20) and the relaxed big-M constraints (4.19), and on some problems the shortest path routing constraints are not binding. Test problem #12, which is feasible without shortest path routing constraints, turns out to be infeasible with the shortest path routing constraints. For $\mathrm{P}^z_{\mathrm{MCSP}}$ the number of added shortest path routing related constraints is much larger. The majority of these are constraints (4.27) for solutions $\bar{z}$ with insufficient bandwidth capacity.

Although $\mathrm{P}^z_{\mathrm{MCSP}}$ contains fewer variables than $\mathrm{P}^w_{\mathrm{MCSP}}$, the LP relaxations for $\mathrm{P}^z_{\mathrm{MCSP}}$ has taken longer to solve than the corresponding LP relaxations for $\mathrm{P}^w_{\mathrm{MCSP}}$. This is due to the the larger number of added inequalities and the less favorable numerical properties of the relaxed big-M constraints (4.19). Since $\mathrm{P}^z_{\mathrm{MCSP}}$ is more restrictive than $\mathrm{P}^w_{\mathrm{MCSP}}$ the optimal value of $\mathrm{P}^z_{\mathrm{MCSP}}$ is greater than or equal to the optimal value of $\mathrm{P}^w_{\mathrm{MCSP}}$. However, on some of the larger test problems (where opti-

	problem			cuts			B&B	objective values		
#	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert K\rvert$	CN	SP	time/s	iter	root	lb	ub
1	10	15	2	30	0	2.6	35	7.27	7.91	8.00
2	10	15	10	139	56	1.0E4	4502	32.2	40.4	50.9
3	10	19	2	25	0	2.2	18	11.2	11.7	11.8
4	10	19	10	342	3	1.0E4	3070	24.6	28.2	37.0
5	15	27	3	103	0	18	55	16.6	17.6	17.8
6	15	27	15	426	0	1.4E3	93	48.3	68.3	69.0
7	15	35	3	399	0	6.1E3	8897	15.3	18.0	18.2
8	15	35	15	759	33	1.0E4	462	42.0	44.1	70.1
9	20	56	4	724	0	7.2E2	243	20.6	23.8	24.0
10	20	56	20	1229	0	1.0E4	29	70.2	70.8	134

Table 1: $\mathrm{P}^w_{\mathrm{MCSP}}$, non-uniform costs and capacities.

mality have not been verified) the best found solution is actually better for $\mathrm{P}^z_{\mathrm{MCSP}}$ than for $\mathrm{P}^w_{\mathrm{MCSP}}$, which seems to indicate that the primal heuristic for $\mathrm{P}^z_{\mathrm{MCSP}}$ is even more effective than for $\mathrm{P}^w_{\mathrm{MCSP}}$. The best found solution for $\mathrm{P}^z_{\mathrm{MCSP}}$ is of course feasible for $\mathrm{P}^w_{\mathrm{MCSP}}$.

The impact of the different valid constraints and inequalities derived for the problem has been studied for test problem #14. Figure 9 shows how the global lower bound evolves over the B&B-iterations for different combinations of the relaxed big-M constraints (4.19), (m), the subpath constraints (4.20), (s), and the capacity level inequalities (4.14), (c). Note that without the subpath constraints the lower bound improves by very little, almost regardless of the other two groups of constraints. With the subpath constraints the performance improves significantly. Note that the capacity level inequalities improve the initial lower bounds, but that they in the long run slightly impede the B&B lower bounding progress. The global upper bounds in the figure have been obtained for the (mcs) case. Note that the optimal solution (within the given tolerance) is found after relatively few iterations, and that the remaining time is spent in improving the lower bound. This behavior has been consistent on all of the test problems for which optimality has been verified. The remaining gap in the figure is due to the optimality tolerance.

7. Summary and conclusions

The backbone communications network capacity dimensioning problem has been studied for the following special case: (i) multicast source-tree and core-tree traffic demands; (ii) discrete, non-uniform capacity levels; and, (iii) a multicast routing protocol that finds distribution trees as shortest path trees with respect to a single set of pre-configured, strictly positive, link routing weights. Two versions of the problem have been described, one where the unicast shortest path routing link weights can be chosen freely to match the design, $\mathrm{P}^w_{\mathrm{MCSP}}$, and one where the link weights are given by the selected capacity levels, $\mathrm{P}^z_{\mathrm{MCSP}}$. The latter problem incorporates the

problem				cuts			B&B	objective values		
#	$\|V\|$	$\|E\|$	$\|K\|$	CN	SP	time/s	iter	root	lb	ub
11	10	13	2	36	0	49	931	18.5	21.8	22.0
12	10	14	10	-	-	3.0	-	∞	∞	∞
13	10	18	2	3	0	0.29	3	11.4	11.5	11.6
14	10	19	10	129	0	2.0E2	193	73.4	78.7	79.5
15	15	28	3	301	2	2.5E3	11573	23.9	28.1	28.4
16	15	30	15	508	66	1.0E4	1591	130	133	141
17	15	37	3	230	0	1.6E3	3473	17.0	20.1	20.3
18	15	38	15	475	75	1.0E4	1169	90.1	91.4	102
19	20	57	4	1028	20	1.0E4	1440	36.1	37.2	45.0
20	20	57	20	572	45	1.0E4	103	156	158	213

Table 2: $\mathrm{P}^{w}_{\mathrm{MCSP}}$, uniform costs and capacities.

problem				cuts			B&B	objective values		
#	$\|V\|$	$\|E\|$	$\|K\|$	CN	SP	time/s	iter	root	lb	ub
1	10	15	2	22	21	4.4	23	7.27	7.91	8.00
2	10	15	10	194	4335	1.0E4	1116	32.2	38.4	45.9
3	10	19	2	77	482	1.2E2	257	11.2	13.7	13.8
4	10	19	10	274	2902	1.0E4	673	24.6	27.4	36.0
5	15	27	3	91	44	26	23	16.6	17.6	17.8
6	15	27	15	602	509	1.0E4	152	48.3	56.0	69.0
7	15	35	3	487	3736	1.0E4	913	15.3	17.2	18.8
8	15	35	15	515	632	1.0E4	107	42.0	43.5	62.8
9	20	56	4	1614	1360	1.0E4	243	17.4	22.0	25.0
10	20	56	20	685	5	1.0E4	1	70.2	70.2	134

Table 3: $\mathrm{P}^{z}_{\mathrm{MCSP}}$, non-uniform costs and capacities.

problem				cuts			B&B	objective values		
#	$\|V\|$	$\|E\|$	$\|K\|$	CN	SP	time/s	iter	root	lb	ub
11	10	13	2	39	814	1.8E2	765	18.5	21.8	22.0
12	10	14	10	-	-	5.8	-	∞	∞	∞
13	10	18	2	5	2	0.85	3	11.4	11.5	11.6
14	10	19	10	155	620	1.2E3	247	73.4	78.7	79.5
15	15	28	3	297	5235	1.0E4	1948	23.9	26.2	28.4
16	15	30	15	356	1668	1.0E4	326	130	132	135
17	15	37	3	635	2372	1.0E4	1234	17.0	19.7	20.8
18	15	38	15	385	1277	1.0E4	203	90.1	91.1	106
19	20	57	4	739	1766	1.0E4	349	36.1	36.9	45.6
20	20	57	20	433	214	1.0E4	33	156	158	210

Table 4: $\mathrm{P}^{z}_{\mathrm{MCSP}}$, uniform costs and capacities.

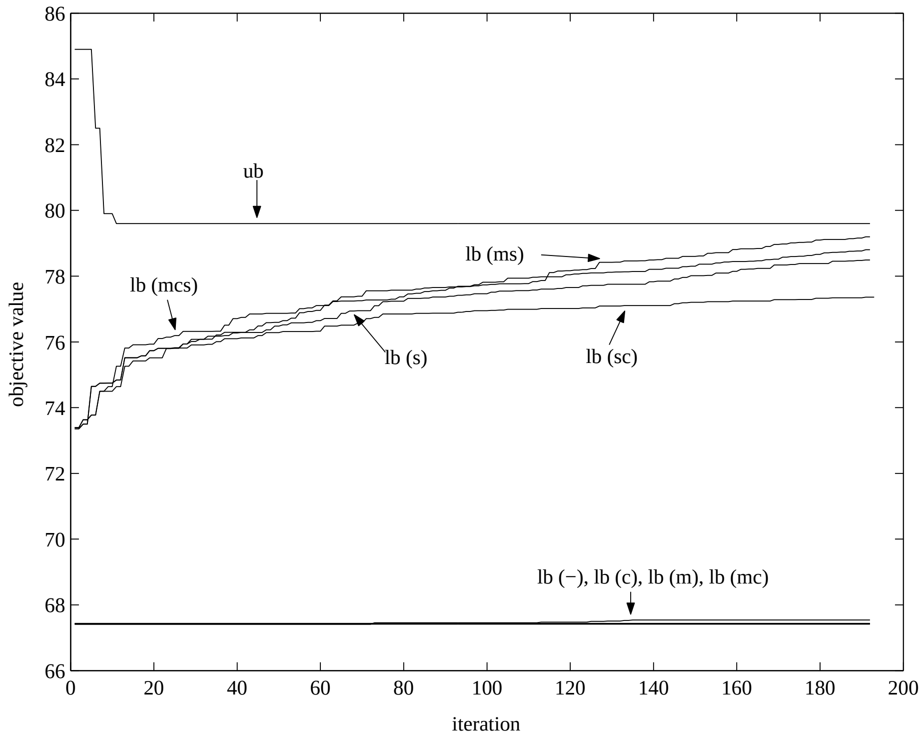

Figure 9: Lower bounds obtained with different combinations of valid inequalities (m = relaxed big-M constraints (4.19), s = subpath constraints (4.20), c = capacity cover inequalities (4.14)).

standard "inverse capacity" heuristic configuration rule for routing weights. The shortest path routing constraints have been shown, by example, to restrict the set of simultaneously realizable distributions trees.

Both problems have been modeled as mixed integer programming problems. The distribution tree connectivity constraints have been modeled using graph cut inequalities. A set of valid capacity level inequalities has been given and shown to be GUB cover inequalities for the capacity constraints. A slightly restricted version of the shortest path routing constraints has been described in terms of a big-M formulation constructed from the dual constraints for the shortest path problem. Since big-M formulations are generally poor, a relaxed version has been considered. A second set of valid constraints, the so called subpath constraints, has also been described.

A separation LP for verifying unique shortest path routing, and a strategy to generate valid, violated inequalities, have been described for $\mathrm{P}^{w}_{\mathrm{MCSP}}$. Finally some straightforward separation inequalities for $\mathrm{P}^{z}_{\mathrm{MCSP}}$ have been given.

Both problems contain the difficult, NP-hard, Steiner tree problem as a subproblem, and, hence, are difficult to solve exactly for larger problem instances. A branch-and-cut algorithm has been described that solve LP relaxations of the problems, separate violated connectivity cut inequalities by solving maxflow/mincut problems, branch on fractional variable values, and separate violated shortest path routing constraints for integer solutions (arc selection variables integer for $\mathrm{P}^{w}_{\mathrm{MCSP}}$, and capacity level selection variables integer for $\mathrm{P}^{z}_{\mathrm{MCSP}}$). A primal heuristic has been described that computes shortest path trees and selects appropriate capacity levels. The routing weights for the heuristic are taken from LP relaxation (or the separation LP) in $\mathrm{P}^{w}_{\mathrm{MCSP}}$. For $\mathrm{P}^{z}_{\mathrm{MCSP}}$ the routing weights are computed from integer capacity level solutions or by from two rounding heuristics on the capacity level variables.

The proposed branch-and-cut algorithm has been implemented in MATLAB and CPLEX. Computations have been made on twenty different test problems for the two problem versions. The algorithm has been able to find optimal solutions, within a 1% tolerance, on the smaller test problems. The results indicate that the primal heuristic produces good solutions (optimal on almost all of the problems where optimality have been verified) during the first few B&B-iterations. On the larger test problems the quality of the lower bound is not as good as for the smaller problems, and the algorithm has been unable to verify optimality. The effect of the different valid constraints and inequalities for the problem has been studied for one test problem. The results of this study suggests that the subpath constraints are critical for the lower bound quality.

Although the branch-and-cut algorithm has not been able to verify optimality within the time limit for the larger test problems, we believe that the quality of the primal feasible solutions obtained are comparable to what can be obtained by pure heuristic methods. The branch-and-cut method also provides an optimizing framework to verify optimal solutions or provide a lower bound within a given time limit.

References

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. Network Flows. Theory, Algorithms and Applications. Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[2] I. Andersson. Private communication. Ericsson Radio Systems AB, Stockholm, Sweden, 2000.

[3] F. Barahona, Network design using cut inequalities, SIAM J. Optimization 6 (1996), 823–837.

[4] D. Bienstock and A. Bley. Capacitated network design with multicast commodities. ZIB-report 00-14, Konrad-Zuse-Zentrum für informationstechnik Berlin, 2000.

[5] D. Bienstock, S. Chopra, O. Günlük, and C.-Y. Tsai, Minimum cost capacity installation for multicommodity network flows, Math. Program. 81 (1998), 177–200.

[6] A. Bley. Design of IP-networks, August 2000. Presentation at ISMP2000 (International Symposium on Mathematical Programming).

[7] S. Chen, O. Günlük, and B. Yener, The multicast packing problem, IEEE/ACM Transactions on Networking 8 (2000), 311–318.

[8] B.V. Cherkassky and A.V. Goldberg, On implementing the push-relabel method for the maximum flow problem, Algorithmica 19 (1997), 390–410.

[9] Cisco Systems Inc., San Jose, CA, USA. OSPF design guide, 2001.

[10] E.W. Dijkstra, A note on two problems in connexion with graphs, Numerische Mathematik 1 (1959), 269–271.

[11] M. Ericsson, M.G.C. Resende, and P.M. Pardalos. A genetic algorithm for the weight setting problem in OSPF routing. Technical report, AT&T Labs Research, October 2001.

[12] B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. In Proceedings of IEEE INFOCOM'2000, 519–528. IEEE, 2000.

[13] M.S. Garey and D.S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, New York, 1979.

[14] E. Gourdin, Optimizing internet networks, OR/MS Today (2001).

[15] K. Holmberg and D. Yuan. Optimization of internet protocol network design and routing. Technical Manuscript, Linköping University, Submitted to Networks, 2001.

[16] F.K. Hwang, D.S. Richards, and P. Winter. The Steiner Tree Problem. North-Holland, Amsterdam, The Netherlands, 1992.

[17] ILOG Inc, CPLEX Division, Incline Village, NV, USA. ILOG CPLEX 6.5 User's Manual, 1999.

[18] M. Jones and D. Mitchell. JUNOS Internet Software Configuration Guide: Routing and Routing Protocols, Release 4.3. Juniper Networks Inc., Sunnyvale, CA, USA, 2001.

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

[20] Y.-W. Leung and B.-T. Yang, Lower bound for multimedia multicast routing, IEE Proc.-Commun. 145 (1998), 87–90.

[21] F.Y.S. Lin and J.L. Wang. Minimax open shortest path first routing algorithms in networks supporting the SMDS service. In Technical Program, Conference Record, IEEE International Conference on Communications, volume 2, 666–670, 1993.

[22] C. Long. IP Network Design. Osborne/McGraw-Hill, 2001.

[23] A. Lucena and J.E. Beasley, A branch and cut algorithm for the Steiner problem in graphs, Networks 31 (1998), 39–59.

[24] The MathWorks, Inc., Natick, MA, USA. MATLAB User's Guide, 1998.

[25] J. Moy. OSPF: Anatomy of an Internet routing protocol. Addison Wesley, 1998.

[26] J. Moy. OSPF version 2. Request for Comments 2328, Internet Engineering Task Force, April 1998.

[27] G.L. Nemhauser and L.A. Wolsey. Integer and Combinatorial Optimization. Wiley, 1988.

[28] M. Prytz and A. Forsgren. Dimensioning multicast-enabled communications networks. Report TRITA-MAT-2001-OS2, Department of Mathematics, Royal Institute of Technology, 2001.

[29] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture. Request for Comments 3031, Internet Engineering Task Force, January 2001.

[30] D. Staehle, S. Köhler, and U. Kohlhaas. Towards an optimization of the routing parameters for IP networks. Research report TR 258, Department of Computer Science, University of Würzburg, 2000.

[31] W. Stallings. High-Speed Networks, TCP/IP and ATM Design Principles. Prentice Hall, 1998.

[32] D. Thaler and C. Hopps. Multipath issues in unicast and multicast next-hop selection. Request for Comments 2991, Internet Engineering Task Force, November 2000.

[33] B. Williamson. Developing IP Multicast Networks: The Definitive Guide to Designing and Deploying CISCO IP Multicast Networks. Cisco Press, 2000.

[34] R.T. Wong, A dual ascent approach for Steiner tree problems on a directed graph, Math. Program. 28 (1984), 271–287.