Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in DISCRETE MATHEMATICS

Electronic Notes in Discrete Mathematics 52 (2016) 189–196 www.elsevier.com/locate/endm

A Multi-Commodity Flow Based Model for Multi Layer Hierarchical Ring Network Design

Christian Schauer and Günther R. Raidl

Institute of Computer Graphics and Algorithms
TU Wien, Vienna, Austria
{cschauer|raidl}@ac.tuwien.ac.at

Abstract

We address the Multi Layer Hierarchical Ring Network Design Problem. The aim of this problem is to connect nodes that are assigned to different layers using a hierarchy of rings of bounded length. We present a multi-commodity flow based mixed integer linear programming formulation and experimentally evaluate it on various graphs. Instances up to 76 nodes and 281 edges could be solved to optimality.

Keywords: hierarchical network design, dual homing, multi-commodity flow

1 Introduction and Problem Definition

We consider the *Multi Layer Hierarchical Ring Network Design* (MLHRND) problem, which finds applications in large, hierarchically structured networks with a strong need of survivability. The problem description originates from a cooperation with an Austrian telecommunication provider.

In principle, the simplest way to achieve survivability is the use of a ring topology since the network stays connected in case of a single node or link failure. For the backbone of wide area networks a single ring would not be efficient anymore because two simultaneous failures could disconnect large

http://dx.doi.org/10.1016/j.endm.2016.03.025
1571-0653/

parts of the network. Moreover, requirements with respect to bandwidth and maximal delays physically limit the size of a ring. To avoid this problem multiple interconnected rings are frequently used as backbones.

In a *Hierarchical Ring Network* (HRN) this interconnection is often realized in a hierarchical fashion using rings on every layer of the hierarchy to allow scalability. Thomadsen and Stidsen describe in [4] such an HRN for two levels, where the node set is not partitioned a priori but to determine during the design process. They discuss different subproblems (clustering, hub selection, ring design, and routing) and present a Branch-and-Price approach. For this problem the top level ring is concatenated with any other ring over a single node (*single homing*). In this case the network can compensate a link failure but does not stay connected if the concatenation node fails.

To also cover this situation the rings must be connected over two different nodes on each ring (*dual homing*), which is addressed for instance by Karaşan et al. in [1]. These authors study a two layer network, where the first layer is connected by overlapping rings and each second layer node is connected to two different nodes on the first layer. To solve this problem two-index and three-index models are presented.

MLHRND deals with a hierarchical structure spanning nodes on multiple layers using rings of bounded length and dual homing to ensure fault tolerance in case of single link and node failures.

We introduced MLHRND in [2] for the three-layer case and described a variable neighborhood search (VNS) and a greedy randomized adaptive search procedure to solve it heuristically. We further argued that the classical *Capacitated Vehicle Routing Problem* can be reduced to MLHRND, i.e., MLHRND is NP-hard, even for the three layer case.

In [3] we extended the definition of MLHRND to an arbitrary number (≥ 3) of layers and presented a memetic algorithm (MA). In this work we model MLHRND as a mixed integer linear program (MIP) using a flow based approach and practically evaluate this formulation in experiments.

Formally we can define MLHRND as follows. Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E. A weighting function assigns costs $c_{ij} \geq 0$ to each edge $(i, j) \in E$. Moreover, V is partitioned into $K \geq 3$ disjoint subsets $V_1, \dots, V_K$, representing the layers each node belongs to. Edge set E consists of sets E_k connecting nodes within each layer $k = 1, \dots, K$ and sets $E'_k,\ k = 2, \dots, K$, connecting nodes between layer k and $k-1$.

A feasible solution to MLHRND is a subgraph $G_L = (V, E_L)$ connecting all nodes in V and satisfying the following conditions:

(i) All nodes in V_1 are connected by a single independent ring containing no other node.

(ii) The remaining layers are connected by $K - 1$ respective sets of paths containing no nodes from other layers. Each node must appear in exactly one path, i.e., the paths are node and edge disjoint to ensure reliability.

(iii) The end nodes of each path at layer $k \in \{2, \ldots, K\}$ are further connected to two different nodes (*hubs*) in layer $k - 1$, i.e., dual homing is realized. We refer to the edges in E'_k connecting paths to hubs as *uplinks*.

(iv) The two hub nodes a path is connected to must themselves be connected by a simple path at their layer, i.e., the connection to a ring may not be established via more than two layers.

(v) The lengths of layer $k \in \{2, \ldots, K\}$ paths in terms of the number of edges is bounded below and above by $b^{\mathrm{l}}_k \geq 1$ and $b^{\mathrm{u}}_k \geq b^{\mathrm{l}}_k$, respectively.

The objective is to find a feasible solution with minimum total costs.

2 A MIP Model for MLHRND

In this section we present a MIP model for MLHRND using a multi-commodity based flow to ensure all conditions mentioned in the previous section and a single-commodity flow to enforce connectivity over the whole network.

Layer $k = 1$

From condition i we can conclude that finding the layer 1 ring resembles the classical Traveling Salesman Problem (TSP) that can be solved independently. Consequently, any feasible TSP solver can be used, e.g., Concorde [1] .

Layers $k > 1$

The remaining layers influence one another due to the combination of the dual homing aspect (condition iii) and the connection of the hubs via simple paths (condition iv). Thus, these layers cannot be treated independently to obtain an optimal solution, which marks the challenging aspect of MLHRND both to model and solve it. Note that by relaxing either condition iii or condition iv, each layer could be solved independently to achieve an optimal solution.

For the given input graph $G = (V, E)$ we define a corresponding directed graph $G_D = (V, A)$, in which we have two reversely directed arcs (i, j) and

[1] `http://www.math.uwaterloo.ca/tsp/concorde/`

(j,i) for each edge $(i,j) \in E$. Let A_k be the arcs corresponding to E_k and A'_k the arcs corresponding to E'_k.

Further, we define binary variables $a_{i\alpha} \in A'_k$ with $i \in V_{k-1}$ and $\alpha \in V_k$. If set to 1 variable $a_{i\alpha}$ indicates the existence of a circle, which is realized by arc (i,α) and sending a flow of commodity α from source node α via a path in layer k, an uplink to layer $k-1$, and a path in layer $k-1$ to the sink node i.

Additionally, we use binary variables $x_{ij}, \forall (i,j) \in E$, to indicate the edges being part of the solution and flow variables $0 \leq f^{\alpha}_{ij} \leq 1$, $\forall (i,j) \in A_k \cup A_{k-1} \cup A'_k$, $\forall \alpha \in V_k$. Then we can formulate the objective function together with each layer $k \in \{2,\ldots,K\}$:

$$\min \sum_{(i,j)\in E} c_{ij} \cdot x_{ij} \tag{1}$$

$$\text{s.t.} \quad \sum_{(i,\alpha)\in A'_k} a_{i\alpha} - \sum_{(\alpha,i)\in A_k} f^{\alpha}_{\alpha i} = 0, \qquad \forall \alpha \in V_k, \tag{2}$$

$$\sum_{(i,j)\in A_k} f^{\alpha}_{ij} - \sum_{(j,i)\in A_k \cup A'_k} f^{\alpha}_{ji} = 0, \qquad \forall j \in V_k, \forall \alpha \in V_k \setminus j, \tag{3}$$

$$\sum_{(i,j)\in A_{k-1} \cup A'_k} f^{\alpha}_{ij} - \sum_{(j,i)\in A_{k-1}} f^{\alpha}_{ji} = a_{j\alpha}, \qquad \forall j \in V_{k-1}, \forall \alpha \in V_k, \tag{4}$$

$$\sum_{\alpha \in V_k} f^{\alpha}_{ij} \leq 1, \qquad \forall (i,j) \in A_k \cup A'_k, \tag{5}$$

$$f^{\alpha}_{ij} = 0, \qquad \forall (i,j) \in A'_k, i \in V_k \wedge i < \alpha, \tag{6}$$

$$\sum_{(g,h)\in A_k} f^{\alpha}_{gh} \geq a_{i\alpha} \cdot b^{\mathrm{l}}_k, \qquad \forall \alpha \in V_k, \forall (i,\alpha) \in A'_k, \tag{7}$$

$$\sum_{(i,j)\in A_k} f^{\alpha}_{ij} \leq b^{\mathrm{u}}_k, \qquad \forall \alpha \in V_k, \tag{8}$$

$$\sum_{(g,h)\in A_{k-1}} f^{\alpha}_{gh} \geq a_{i\alpha}, \qquad \forall \alpha \in V_k, \forall (i,\alpha) \in A'_k, \tag{9}$$

$$\sum_{(i,j)\in A_1} f^{\alpha}_{ij} \leq |V_1| - 1, \qquad \forall \alpha \in V_2, \tag{10}$$

$$\sum_{(i,j)\in E_k \cup E'_k} x_{ij} = 2, \qquad \forall i \in V_k, \tag{11}$$

$$\sum_{\alpha \in V_k} f^{\alpha}_{ij} = x_{ij}, \qquad \forall (i,j) \in A_k, \tag{12}$$

$$\sum_{\alpha \in V_k} f_{ij}^{\alpha} = x_{ij}, \qquad \forall (i,j) \in A'_k, i \in V_k, j \in V_{k-1}, \tag{13}$$

$$f_{ij}^{\alpha} \leq x_{ij}, \qquad \forall (i,j) \in A_{k-1}, \forall \alpha \in V_k, \tag{14}$$

$$a_{i\alpha} = x_{i\alpha}, \qquad \forall (i,\alpha) \in A'_k, i \in V_{k-1}, \alpha \in V_k, \tag{15}$$

$$f_{ij}^{\alpha} \in [0,1], \qquad \forall (i,j) \in A_k \cup A_{k-1} \cup A'_k, \forall \alpha \in V_k, \tag{16}$$

$$x_{ij} \in \{0,1\}, \qquad \forall (i,j) \in E, \tag{17}$$

$$a_{i\alpha} \in \{0,1\}, \qquad \forall (i,\alpha) \in A'_k, i \in V_{k-1}, \alpha \in V_k. \tag{18}$$

The objective function (1) minimizes the costs over all selected edges. Equations (2) to (4) model the flows. A unit flow of commodity α is initiated at node α, if there exists an $a_{i\alpha}$ with value 1, i.e., (i,α) is an uplink (2). Flow conservation within layer k is formulated by equations (3) including the flow on the second uplink to the first hub, while equations (4) model the flow conservation on the preceding layer $k-1$ with the flow ending at the second hub node j for which $a_{j\alpha} = 1$. Inequalities (5) ensure that on every edge of the current layer k and its potential uplinks only one commodity can flow, i.e., rings must be node and edge disjoint (condition ii).

By defining a natural order for all nodes V, we can break symmetries by demanding that the flow's initial node is always smaller than the end node of the path in A_k. We achieve this by forbidding a flow for commodity α on uplinks going from $i \in V_k$ to $j \in V_{k-1}$ for all nodes $i < \alpha$ (6).

The lower and upper bounds for the path lengths are enforced by summing up the edges with flow along the path in A_k in (7) and (8), respectively. As the lower bound must only hold, if there is a flow of the respective commodity α, we again use variables $a_{i\alpha}$ for enabling the corresponding constraint in (7).

Dual homing is enforced by equations (9), where we use the same trick as before. If there is a flow for commodity α, i.e., an uplink is connected to α, then this flow must span at least one arc in layer $k-1$. For the dual homing of layer 2 commodities α we have to limit the flow within layer 1 to a maximum length of $|V_1| - 1$ (10). To ensure that each node in V_k appears exactly in one path, equations (11) enforce node degree two with respect to $E_k \cup E'_k$. Since every node, except nodes in the last layer K, can serve as a hub, the overall degree in G_L of each node is possibly greater than two.

The linking constraints for layer k are formulated by (12) and (13) for the path edges and uplinks, respectively. If there is a flow of a commodity on edge $(i,j) \in E_k \cup E'_k$ then the corresponding x_{ij} must be set. If on the other hand x_{ij} is 1, then there must be a flow on the respective edge (i,j). For the flow on layer $k-1$ the situation is different (14). If there is a flow on the edge $(i, j) \in E_{k-1}$ then x_{ij} must be set. Now flows of different commodities are allowed along the same edge and not every selected edge must carry a flow from layer k, since this edge can also be set through a commodity for layer $k-1$. Equations (15) link the variables $a_{i\alpha}$ inducing flows from α to i with the corresponding circle-closing edge variables $x_{i\alpha}$.

Ensuring Connectivity with Single-Commodity Flows

With the model considered so far two issues still remain to obtain a feasible solution. First, there are no equations that enforce arc variables $a_{i\alpha}$ to be set to 1. Second, we cannot exclude subtours for commodities α. To solve both we decided to adapt the classical single-commodity flow (SCF) formulation for the spanning tree polytope to ensure connectivity over the whole network. Using flow variables g_{ij}, $\forall (i, j) \in A$, we send one unit of flow from some dedicated source node 0 in layer 1 to each other node in the graph.

$$\sum_{(0,i) \in A_1 \cup A'_2} g_{0i} = |V| - 1, \tag{19}$$

$$\sum_{(i,j) \in A_1} g_{ij} - \sum_{(j,i) \in A_1} g_{ji} = 1, \qquad \forall j \in V_1 \setminus \{0\}, \tag{20}$$

$$\sum_{(i,j) \in A_k \cup A'_k} g_{ij} - \sum_{(j,i) \in A_k} g_{ji} = 1, \qquad \forall j \in V \setminus V_1, \tag{21}$$

$$g_{ij} \le (|V| - 1) \cdot x_{ij}, \qquad \forall (i, j) \in A, \tag{22}$$

$$g_{ij} \in [0, |V| - 1] \qquad \forall (i, j) \in A. \tag{23}$$

Equation (19) initiates the flow at node 0 with $|V| - 1$ units. The flow conservation for all other layer 1 nodes considers only layer 1 arcs A_1 (20). For the flow conservation of the remaining layers the incoming flow over the uplinks of the current layer must be taken into account, too (21). The flow variables are finally linked with the edge variables in (22).

3 Experimental Results

We tested our model on the $K = 3$ layer scenario and used the same benchmark instances as in [3], which are based on TSPLIB[2] and randomly generated graphs. The original “dense” instances were graphs that contained no

[2] `http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/`

Table 1
Excerpt of our experimental results.

Instance	b	lp	lb	g[%]	t[s]	#BnB	best ub	algo
rand30-3-10-384-d	4-6	78659.25	84801.63	**0.0**	565	11311	**84801.63**	mip,ma,vns
rand35-3-10-529-d	4-4	81641.30	86242.90	2.0	20000	163073	87942.81	ma,vns
rand35-3-10-529-d	4-6	81641.30	85056.55	**0.0**	824	7214	**85056.55**	mip,ma,vns
rand45-5-15-865-d	4-6	91655.12	94477.87	2.4	20000	47186	96835.30	mip,vns
rand45-5-15-197-s	4-6	91655.12	96835.30	**0.0**	1855	33294	**96835.30**	mip
att48-4-10-992-d	4-6	54840.60	57072.71	6.4	20000	13004	58476.25	ma,vns
att48-4-10-167-s	4-6	54842.42	58591.88	**0.0**	8661	70817	**58591.88**	mip
berlin52-5-15-1166-d	4-6	12628.36	12727.43	13.2	20000	14021	13295.40	ma
berlin52-5-15-201-s	4-6	12628.36	13294.29	**0.0**	6480	52401	**13294.29**	mip
eil76-5-20-2595-d	7-11	834.03	842.30	—	20000	1017	853.50	ma,vns
eil76-5-20-281-s	7-11	839.34	856.15	**0.0**	12540	21734	**856.15**	mip

edges between layer 1 and 3, since they cannot occur in a feasible solution. We additionally created "sparse" graphs, where we defined layer 1 to be fully connected, added a simple random solution, ensured for each layer to be two-connected, and then randomly added edges to the closest nodes and hubs. We implemented our MIP approach in Java 1.6 using IBM CPLEX 12.5. All tests were performed on a single core of an Intel Xeon (Nehalem) Quadcore CPU with 2.53 GHz and 3GB of RAM. As stopping criterion we defined a time limit of 20 000 seconds for each test case. For the lower bound b_k^l we always assumed one edge as the minimum length for all paths.

Due to space limitations we can only present an excerpt from our results in Table 1, while all test results together with the instances can be downloaded from `www.ads.tuwien.ac.at/w/Research/Problem_Instances`. Column *Instance* refers to the underlying graphs, either the names from the TSPLIB or *rand* for the random instances. The following values represent the numbers of nodes in the graphs, layers 1, layers 2, and edges, while *d* indicates a dense and *s* a sparse graph. Column *b* lists the upper bound path lengths for layers 2 and 3. Columns *lp*, *lb*, and *g* list the linear programming (LP) relaxations, lower bounds and the percentage gaps ((upper bound − lower bound) / upper bound) obtained by CPLEX, a dash indicates that no integral solution was found. Moreover, the total run times (t) in seconds and the numbers of branch-and-bound nodes (*#BnB*) are provided. Column *best ub* presents the best upper bounds known together with the corresponding algorithms (*algo*) from [2,3] that generated these solutions. Note that only the MIP approach can solve sparse instances.

The run times to obtain the LP values for instances with up to 30 nodes are always less than one second. With the increasing number of nodes the run time also increases up to 650 seconds for the dense instances, while for the sparse it never exceeds three seconds. When solving the problem to integrality almost all test cases with up to 35 nodes are solved to optimality. For all test cases the lower bound is improved in comparison to the LP-value. The scalability for the larger dense instances show the limits of our approach. For rand45 we find a gap of 2.4%, for att48 a gap of 6.4%, and for berlin52 a gap of 13.2%, while even finding an integral solution for eil76 is not possible. On the sparse instances our approach scales much better and finds optimal solutions even for test cases berlin52 and eil76.

4 Conclusions and Future Work

We presented a multi-commodity flow based formulation to solve the Multi Layer Hierarchical Ring Network Design problem and experimentally evaluated it. To ensure connectivity for the whole network we used a single-commodity flow approach. We were able to solve dense instances with up to 35 nodes and sparse with up to 76 nodes and 281 edges. In the future we will focus on decomposition techniques, e.g., Benders decomposition, to solve larger instances. In a hybrid metaheuristic context, we expect good results by utilizing the proposed MIP model within a large neighborhood search approach.

References

[1] Karaşan, O. E., A. R. Mahjoub, O. Özkök and H. Yaman, *Survivability in Hierarchical Telecommunications Networks Under Dual Homing*, INFORMS Journal on Computing, Articles in Advance (2013), pp. 1–15.

[2] Schauer, C. and G. R. Raidl, *Variable Neighborhood Search and GRASP for Three-Layer Hierarchical Ring Network Design*, in: C. A. Coello Coello et al., editors, *Parallel Problem Solving from Nature–PPSN XII*, LNCS **7491** (2012), pp. 458–467.

[3] Schauer, C. and G. R. Raidl, *A Memetic Algorithm For Multi Layer Hierarchical Ring Network Design*, in: T. Bartz-Beielstein et al., editors, *Parallel Problem Solving from Nature–PPSN XIII*, LNCS **8672** (2014), pp. 832–841.

[4] Thomadsen, T. and T. Stidsen, *Hierarchical Ring Network Design Using Branch-and-Price*, Telecommunication Systems **29** (2005), pp. 61–76.