Proceedings of the XI Brazilian Congress on Automatic Control (XI CBA)
São Paulo, pp. 365–370, September, 1996

THE MULTI-LEVEL NETWORK OPTIMIZATION PROBLEM

F. R. B. Cruz[a,†,‡], G. R. Mateus[b,§], J. MacGregor Smith[a]

[a]*Department of Mechanical and Industrial Engineering,*
The University of Massachusetts,
Amherst, MA 01003, USA
E-mail: {fcruz,jmsmith}@ecs.umass.edu

[b]*Departamento de Ciência da Computação,*
Universidade Federal de Minas Gerais,
31270-010 - Belo Horizonte - MG, Brazil
E-mail: mateus@dcc.ufmg.br

Abstract: This paper addresses a new multicommodity problem of network location, topological design and dimensioning integrated in the same model. The generalized multi-level network optimization problem is defined and one possible mathematical programming formulation is presented. A branch-and-bound algorithm based on Lagrangean relaxation is proposed to solve this general model.

Key words: Network Optimization, Multi-level Networks, Topological Network Design, Location Problems, Steiner Problems in Graphs.

1. INTRODUCTION

This paper addresses a multicommodity problem of network location, topological design and dimensioning integrated in the same model. A multi-level network is depicted in Figure 1. The multi-level problem shown includes m sets of candidate supply nodes, m sets of demand nodes, and m sets of *Steiner* or transshipment nodes. The arcs have the following cost parameters: a fixed cost of using the arc and per-unit flow costs. None of the candidate supply nodes can create flows except those from the first level. However, they are able to transform flows one level down. Each time a first-level candidate supply node is chosen to provide flow or a non-first-level candidate supply node is chosen to transform flow, a fixed cost must be charged. The objective is to determine a minimum-total-cost subset of supply nodes and arcs with flows from the supply nodes to all of demand nodes.

Modern telecommunication, transportation, and electric power distribution systems, to cite just a few, are good practical examples of multi-level networks. The design problems arising in such contexts have reached a high degree of complexity and each time an improvement is sought out a large number of resources are usually required to be allocated. Because these resources are scarce the application of optimization techniques and computer aids become crucial justifying the research in new methodologies that make possible efficient and accurate algorithms.

The study of multi-level networks is also important in theoretical terms because they can be viewed as a generalization of several important network optimization problems such as topological network design problems, fixed-charge problems, or uncapacitated location problems. A selective bibliography about topological network design problems in general can be found, *e.g.* in [21]. Many network optimization problems oriented to telecommunication applications have been studied including topological network design problems [13, 3], topological design and dimensioning problems [18, 5], and routing problems [1]. The fixed-charge network flow problem [22] which is a special case that represents an important class of mixed-integer programming problems was studied in [17] and [7]. The Steiner problems in graphs [19] is probably the subproblem most studied. Although it is a classical model, recent new results are being discovered for the problem [15]. The uncapacitated location problem [9] is another relevant subproblem. The solution of this problem has many implications in the real world

[†]To whom all correspondences should be addressed. Address: Caixa Postal 702, Departamento de Ciência da Computação - UFMG, 30161-970 - Belo Horizonte - MG, Brazil.

[‡]Visiting research scholar supported by the *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (CNPq), Brazil, under grant CNPq 201046/94-6.

[§]Supported by CNPq and FAPEMIG, Brazil.

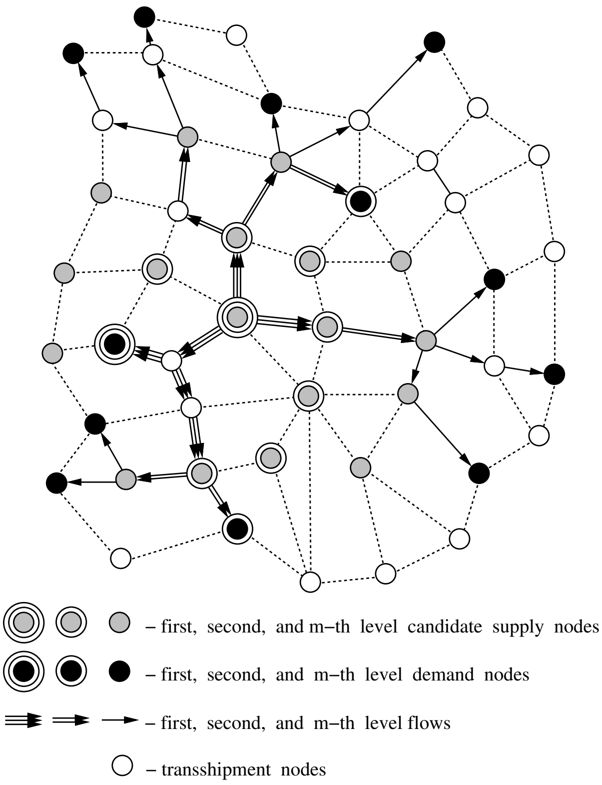

Figure 1: A Multi-level Network

and recent advances continue to be made [11].

Little research has been done on the multi-level network optimization problem. Multi-level networks have appeared in some recent works but they do not consider the integration of location, design, and dimensioning aspects in the same model [8, 3, 2] nor do they provide the mathematical formulation and bounds for the general problem [20], as done here. However, a similar model was studied in [6].

This paper is organized as follows. In section 2, a mathematical programming formulation of the multi-level network optimization problem is proposed. In section 3, a branch-and-bound algorithm based on Lagrangean relaxation is proposed. Section 4 concludes this paper with final remarks.

2. MATHEMATICAL FORMULATION

Let $\mathcal{D} = (N, A)$ be a multi-weighted digraph, where N is the set of nodes and A the set of arcs. The multi-level network optimization problem with m levels is defined on $\mathcal{D}$.

The set of nodes are partitioned into the following subsets: (i) demand nodes D^l, (ii) *Steiner* or transshipment nodes T^l, and (iii) uncapacitated candidate supply nodes R^l, where $l = 1, 2, \ldots, m$. The demand nodes $i \in D^l$ require a non-negative amount of l-th level flow, $d_i \geq 0$. The supply nodes $i \in R^l$ are only able to provide l-th level flow, x^l_{ij}. Since there is also demand for l'-th level flow, other supply nodes $i \in R^{l'}$ must be opened at fixed cost f_i whose function is to convert the supposed available $(l'-1)$-th level flow to the required l'-th level flow. The location of the supply nodes is not known in advance and the solution of this uncapacitated location problem is also provided by the model.

The arcs $(i, j) \in A$ have a non-negative fixed cost associated with their use, $f^l_{ij} \geq 0$, and a non-negative per unit cost, $c^l_{ij} \geq 0$, for each flow level. Thus, considering that the arbitrary arc (i, j) supports positive l-th level flow, then its total cost is a nonlinear function, $c^l_{ij}x^l_{ij} + f^l_{ij}$. Although this is an additional complicating factor in the problem resolution, the resulting model is able to represent the important economy of scale effect of the arcs.

The objective is to minimize the total fixed and variable costs of the chosen facilities ensuring the imposed demand requirements.

2.1. NOTATION

The following notation is used in the mathematical programming formulation of multi-level network optimization problem:

R^l - set of l-th level candidate supply nodes;

D^l - set of l-th level demand nodes;

d_i - demand on node $i \in D^l$ for l-th level flow;

T^l - set of l-th level transshipment nodes, defined as follows: $T^l = N \setminus (R^l \cup D^l \cup R^{l+1})$ for $l = 1, 2, \ldots, (m-1)$, and $T^m = N \setminus (R^m \cup D^m)$;

c^l_{ij} - non-negative per unit cost on arc $(i, j) \in A$ for l-th level flow;

x^l_{ij} - l-th level flow through arc $(i, j) \in A$;

f^l_{ij} - non-negative fixed cost for using the arc $(i, j) \in A$ to support l-th level flow;

y^l_{ij} - boolean variable which assumes the value 1 or 0 depending on whether or not the arc (i, j) is being used to support l-th level flow;

f_i - non-negative allocation cost for using the candidate supply node $i \in R^l$;

z_i - boolean variable which is set to 1 or 0 depending on whether or not the node $i \in R^l$ is being selected to provide l-th level flow;

M^l - l-th level flow capacity on arcs but relaxed in this paper, *i.e.* $M^l = \sum_{L=l}^{m} \sum_{i \in D^L} d_i$;

s^l - l-th level flow supplying capacity on candidate supply nodes also relaxed, *i.e.* $s^l = M^l$;

$\delta^+(i)$ - set $\{j | (i, j) \in A\}$;

$\delta^-(i)$ - set $\{j | (j, i) \in A\}$.

2.2. MODEL (M)

The mathematical programming formulation to describe the multi-level network optimization problem is presented below, as a flow-based mixed integer programming model:

(M):

$$\min \sum_{l=1}^{m} \left[\sum_{(i,j)\in A} \left(c_{ij}^l x_{ij}^l + f_{ij}^l y_{ij}^l \right) + \sum_{i\in R^l} f_i z_i \right], \quad (1)$$

s.t.:

$$\sum_{j\in\delta^+(i)} x_{ij}^l - \sum_{j\in\delta^-(i)} x_{ji}^l = -\left(\sum_{j\in\delta^+(i)} x_{ij}^{l-1} - \sum_{j\in\delta^-(i)} x_{ji}^{l-1} \right), \quad \forall \; i\in R^l, \; l=2,3,\ldots,m, \quad (2)$$

$$\sum_{j\in\delta^+(i)} x_{ij}^l - \sum_{j\in\delta^-(i)} x_{ji}^l = 0, \quad \forall \; i\in T^l, \; l=1,2,\ldots,m, \quad (3)$$

$$\sum_{j\in\delta^+(i)} x_{ij}^l - \sum_{j\in\delta^-(i)} x_{ji}^l = -d_i, \quad \forall \; i\in D^l, \; l=1,2,\ldots,m, \quad (4)$$

$$\sum_{j\in\delta^+(i)} x_{ij}^l - \sum_{j\in\delta^-(i)} x_{ji}^l \le s^l z_i, \quad \forall \; i\in R^l, \; l=1,2,\ldots,m, \quad (5)$$

$$x_{ij}^l \le M^l y_{ij}^l, \quad \forall \; (i,j)\in A, \; l=1,2,\ldots,m, \quad (6)$$

$$x_{ij}^l \ge 0, \quad \forall \; (i,j)\in A, \; l=1,2,\ldots,m, \quad (7)$$

$$y_{ij}^l \in \{0,1\}, \quad \forall \; (i,j)\in A, \; l=1,2,\ldots,m, \quad (8)$$

$$z_i \in \{0,1\}, \quad \forall \; i\in R^l, \; l=1,2,\ldots,m. \quad (9)$$

The objective function (1) minimizes the following three terms: (i) the total flows' variable cost, (ii) the total fixed cost associated with using the arcs (the overhead cost), and (iii) the total cost resulting from the use of the supplying nodes.

Constraints (2) ensure the network flow conservation between adjacent levels at each candidate supply node, constraints (3), and (4) are the usual network flow conservation equalities at each *Steiner* or transshipment node and at each demand node. Constraints (5) ensure there is no flow transformation in a candidate supply node if it is not selected, and constraints (6) express the fact the flow through an arc must be zero if this arc is not included in the design.

3. SOLUTION METHODOLOGY

The multi-level network optimization problem is $\mathcal{NP}$-hard since it generalizes other $\mathcal{NP}$-hard optimization problems such as the Steiner problem in graphs [12] or the uncapacitated location problem [9]. The only known exact approach to solve an $\mathcal{NP}$-hard problem is through the complete enumeration of all solutions. Branch-and-bound is a well-known technique largely applied. Although the algorithm is exponential, it is acceptable for small sized problem instances. Figure 2 presents a template of a recursive version of the branch-and-bound algorithm using depth-first search[1].

```
algorithm Solve(M)

    /* bounding */
  compute lower bound L
  compute upper bound U and update U_BEST
  GAP ← (U−L)/L

    /* branching */
  if L > U_BEST then
      write 'Infeasible node reached.'
  else if GAP ≤ ε then
      write 'Optimum reached.'
  else if P has a free decision variable then
          choose free decision variable
          create new problem fixing it to 1
          Solve(New_M)
          create new problem fixing it to 0
          Solve(New_M)
  end if
  end if
  end if
end algorithm
```

Figure 2: Recursive Branch-and-bound Algorithm

Three statements shall be clarified in the branch-and-bound algorithm presented in Figure 2: the computation of (i) lower bounds, (ii) upper bounds, and (iii) the strategy of choosing the branching variables.

3.1. LOWER BOUND COMPUTATION

A well-known technique to derive lower bounds is Lagrangean relaxation [10] which is usually coupled with a subgradient optimization procedure [16]. There are many ways to derive a Lagrangean relaxation for model (M). The relaxation we propose divides the problem into a shortest paths problem and subset selection problems.

Let us drop constraints (5) using the dual variables $v_i \ge 0$ and constraints (6) using the dual variables $w_{ij}^l \ge 0$. Then, the Lagrangean function below follows:

$$\mathcal{L}(\mathbf{x}, \mathbf{y}, \mathbf{z}; \mathbf{v}, \mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^m) =$$

[1] In opposition to breadth-first search.

$$\sum_{l=1}^{m}\left[\sum_{(i,j)\in A}(c_{ij}^l x_{ij}^l + f_{ij}^l y_{ij}^l) + \sum_{i\in R^l} f_i z_i\right] +$$
$$\sum_{l=1}^{m}\sum_{i\in R^l} v_i \left(\sum_{j\in\delta^+(i)} x_{ij}^l - \sum_{j\in\delta^-(i)} x_{ji}^l - s^l z_i,\right) +$$
$$\sum_{l=1}^{m}\sum_{(i,j)\in A} w_{ij}^l \left(x_{ij}^l - M^l y_{ij}\right), \quad (10)$$

which results in the following Lagrangean relaxation:

$(LR_{\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m})$**:**

$$L(\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m) =$$
$$\min_{\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m\geq\mathbf{0}} \mathcal{L}(\mathbf{x},\mathbf{y},\mathbf{z};\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m), \quad (11)$$

s.t.:

(2), (3), (4), (7), (8), (9).

Supposing that $L(\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m) = \mathcal{L}(\mathbf{x}^*, \mathbf{y}^*,\mathbf{z}^*;\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m)$, the subgradient vector of the function L at point $(\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m)$ is:

$$\left[\left(\sum_{j\in\delta^+(i)} {x_{ij}^l}^* - \sum_{j\in\delta^-(i)} {x_{ji}^l}^* - s^l {z_i}^*,\right)_{\substack{i\in R^l,\\ l=1,2,\ldots,m}}, \left({x_{ij}^l}^* - M^l {y_{ij}^l}^*\right)_{\substack{(i,j)\in A,\\ l=1,2,\ldots,m}}\right]. \quad (12)$$

Once feasible values for the Lagrangean multipliers $\mathbf{v}$, $\mathbf{w}^1$, $\mathbf{w}^2$,..., and $\mathbf{w}^m$ are given, the computation of the function $L(\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m)$ is reduced to solve easy[2] subproblems:

$$L(\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m) =$$
$$L_1(\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m) + L_2(\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m) + L_3(\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m), \quad (13)$$

where $L_1(\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m)$, $L_2(\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m)$, and $L_3(\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m)$ are optimal solutions of the subproblems shown below.

3.1.1. SUBPROBLEM (L_1)

The subproblem (L_1) is:

(L_1)**:**

$$L_1(\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m) =$$

$$\min \sum_{l=1}^{m}\sum_{(i,j)\in A} C_{ij}^l x_{ij}^l, \quad (14)$$

s.t.:

(2), (3), (4), (7),

where

$$C_{ij}^l = \begin{cases} c_{ij}^l + w_{ij}^l & i\notin R^l, \quad j\notin R^l, \\ c_{ij}^l + w_{ij}^l + v_i & i\in R^l, \quad j\notin R^l, \\ c_{ij}^l + w_{ij}^l - v_j, & i\notin R^l, \quad j\in R^l, \\ c_{ij}^l + w_{ij}^l + v_i - v_j, & i\in R^l, \quad j\in R^l. \end{cases} \quad (15)$$

The optimum of (L_1) is easily reached using a shortest paths algorithm. The problem can be solved level by level. The optimum for the first level is to connect nodes in D^1 to nodes in R^1 using the shortest paths. For the second level, the optimum is to connect nodes in D^2 also to nodes in R^1 via shortest paths but using one node of R^2, and so on, for the other levels, as shown in the algorithm seen in Figure 3.

```
algorithm
/* σ_i^l is the minimum per unit cost to bring */
/* l-th level flow from set R^1 to node i ∈ N; */
/*  function SH(i, j, l) returns the shortest  */
/*  path length from i to j using costs C_ij^l; */
    L1 ← 0
    for all j ∈ R^1 do
        if j ∈ J then
            σ_j^0 ← 0
        else
            σ_j^0 ← +∞
        end if
    end for
    for l ← 1 to m do
        for all j ∈ D^l do
            σ_j^l ← min_{i∈R^l} [σ_i^{l-1} + SH(i, j, l)]
            L1 ← L1 + σ_j^l * d_j
        end for
        if l ≠ m do
            for all j ∈ R^{l+1} do
                if j ∈ J then
                    σ_j^l ← min_{i∈R^l} [σ_i^{l-1} + SH(i, j, l)]
                else
                    σ_j^l ← +∞
                end if
            end for
        end if
    end for
end algorithm
```

Figure 3: Algorithm for Solving Problem (L_1)

[2]In such context, *easy* is used in reference to polynomially solvable problems.

The shortest *simple* paths algorithm for arbitrary costs is $O(|N||A|)$ if there are no negative cost circuits [4]. Thus, from the Theorem 1 below, the algorithm for solving problem (L_1) presented in Figure 3 efficiently implemented has worst case time complexity $O(m|N||A|)$ which translates to $O(m|N|^3)$ in dense networks.

Theorem 1 *The problem (L_1) on the digraph $\mathcal{D} = (N, A)$ with weights as defined in Equation (15) does not have negative cost circuits.*

Proof (by construction): *Let C^l be an arbitrary circuit in the l-th level, $A(C^l) \subseteq A$ be the set of arcs in that circuit and $N(C^l) \subseteq N$ be the set of nodes in the circuit. From Equation (15), the per unit cost associated with circuit C^l must be non-negative:*

$$\begin{aligned}\sum_{(i,j)\in A(C^l)} C_{ij}^l &= \sum_{(i,j)\in A(C^l)} c_{ij}^l + \sum_{(i,j)\in A(C^l)} w_{ij}^l + \\ &\quad \sum_{i\in N(C^l)\cap R^l} v_i - \sum_{j\in N(C^l)\cap R^l} v_j \\ &= \sum_{(i,j)\in A(C^l)} c_{ij}^l + \sum_{(i,j)\in A(C^l)} w_{ij}^l \\ &\geq 0.\end{aligned}$$

■

3.1.2. SUBPROBLEM (L_2)

The subproblem (L_2) is a subset selection problem:

(L_2)**:**

$$L_2(\mathbf{v}, \mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^m) = \min \sum_{l=1}^{m} \sum_{(i,j)\in A} \left(f_{ij}^l - w_{ij}^l M^l\right) y_{ij}^l, \quad (16)$$

s.t.:

$$y_{ij}^l \in \{0,1\}, \quad \forall \quad \begin{matrix}(i,j)\in A, \\ l=1,2,\ldots,m,\end{matrix} \quad (17)$$

which can be solved by an algorithm with time complexity $O(m|A|)$, or $O(m|N|^2)$ for dense networks.

3.1.3. SUBPROBLEM (L_3)

Similarly, the problem (L_3) is also a subset selection problem:

(L_3)**:**

$$L_3(\mathbf{v}, \mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^m) = \min \sum_{l=1}^{m} \sum_{i\in R^l} \left(f_i - v_i s^l\right) z_i, \quad (18)$$

s.t.:

$$z_i \in \{0,1\}, \quad \forall \quad \begin{matrix} i\in R^l, \\ l=1,2,\ldots,m,\end{matrix} \quad (19)$$

solvable by an $O(|N|)$ algorithm.

3.2. UPPER BOUND COMPUTATION

The objective here is to propose a heuristic procedure to get feasible solutions quickly. The principal characteristics of heuristic procedures are flexibility and computational simplicity.

There are many possibilities of computing an upper bound for model (M). The method proposed here takes advantage of the problem (L_1) optimal solution. Actually, this solution may be turned into a feasible solution for model (M) if it is ensured feasibility of the previously dropped constraints (5) and (6). Using the same arcs used in the optimal solution of problem (L_1) and computing the overhead costs[3], an upper bound is reached.

3.3. BRANCHING VARIABLE CHOICE

It is an inglorious task to choose branching variables because it is common that one strategy works very well in some instances and poorly for others. The objective is to minimize the number of nodes visited in the branch-and-bound tree. We decided in favor of simplicity. The branching variable will be that one for which one get the maximum expected increment in the lower bound. So, from Eq. (10), the branching variable will be:

$$\begin{cases} z_k, & \text{if } \max \Delta_{z_k} > \max \Delta_{y_{ij}^l}, \\ y_{ij}^l, & \text{otherwise,}\end{cases}$$

where

$$\Delta_{z_k} = \left| v_k \left(\sum_{j\in\delta^+(k)} x_{kj}^{l\;*} - \sum_{j\in\delta^-(k)} x_{jk}^{l\;*} - s^l z_k^{\,*} \right) \right|,$$

if the decision variable z_k is free in the current branch-and-bound tree node and $\Delta_{z_k} = 0$, otherwise. Additionally,

$$\Delta_{y_{ij}^l} = \left| w_{ij}^l \left(x_{ij}^{l\;*} - M^l y_{ij}^{l\;*} \right) \right|,$$

if the decision variable y_{ij}^l is free in the current branch-and-bound tree node and $\Delta_{y_{ij}^l} = 0$, otherwise. Values Δ_{z_k} and $\Delta_{y_{ij}^l}$ are computed using the latest vectors $\mathbf{x}^*$, $\mathbf{y}^*$, $\mathbf{z}^*$, $\mathbf{v}$, $\mathbf{w}^1$, $\mathbf{w}^2$,... , and $\mathbf{w}^m$ obtained.

The method above seems to be a more effective branching strategy than simply to choose the first free variable encountered and is an $O(m|A|)$ procedure (or $O(m|N|^2)$ in dense networks).

4. SUMMARY AND CONCLUSIONS

The multi-level network optimization problem integrating location, topological design, and dimensioning in the same model was defined and its im-

[3] Setup cost of arcs, f_{ij}^l, and supply nodes, f_i.

portance was discussed. A useful mathematical programming formulation was proposed and then a branch-and-bound algorithm was developed.

Some questions remains open such as how effective the bounds actually are and how large the instances solvable in a reasonable amount of time are. Future work might include the investigation of these questions as well as the study of enhanced models that incorporate connectivity constraints which are very important issues in the emerging emphasis on topological robustness and reliability [14, 2].

5. REFERENCES

[1] A. Balakrishnan and K. Altinkemer. Using a hop-constrained model to generate alternative communication network design. *ORSA Journal on Computing*, 4:192–205, 1992.

[2] A. Balakrishnan, T.L. Magnanti, and P. Mirchandani. Designing hierarchical survivable networks. Working Paper OR 291-94, Sloan School of Management, MIT, 1994.

[3] A. Balakrishnan, T.L. Magnanti, and P. Mirchandani. A dual-based algorithm for multi-level network design. *Management Science*, 40(7):567–581, 1994.

[4] M.S. Bazaraa, J.J. Jarvis, and H.D. Sherali. *Linear Programming and Networks Flows*. John Wiley & Sons, New York, 2nd edition, 1990.

[5] F.R.B. Cruz, J.A. Almeida, and G.R. Mateus. Análise do problema de planejamento de redes telefônicas de alimentação. In M.A. Silveira and P.M.G. Ferreira, editors, *Anais do 10º Congresso Brasileiro de Automática*, volume 1, pages 572–577, Rio de Janeiro, Brazil, 1994. SBA.

[6] F.R.B. Cruz, J. MacGregor Smith, and G.R. Mateus. A branch-and-bound algorithm to solve the multi-level network optimization problem. Technical Report RT030/95, Departamento de Ciência da Computação, UFMG, Belo Horizonte, Brazil, 1995. (under review with *Networks*).

[7] F.R.B. Cruz, J. MacGregor Smith, and G.R. Mateus. Solving to optimality the uncapacitated fixed-charge network flow problem. Technical Report RT031/95, Departamento de Ciência da Computação, UFMG, Belo Horizonte, Brazil, 1995. (under review with *Computers & Operations Research*).

[8] J.R. Current, C.S. ReVelle, and J.L. Cohon. The hierarchical network design problem. *European Journal of Operational Research*, 27:57–66, 1986.

[9] D. Erlenkotter. A dual-based procedure for uncapacitated facility location. *Operations Research*, 26(6):992–1009, 1978.

[10] M.L. Fisher. The Lagrangean relaxation method for solving integer programming problems. *Management Science*, 27:1–18, 1981.

[11] R.D. Galvão. The use of Lagrangean relaxation in the solution of uncapacitated facility location problems. *Location Science*, 1(1):57–79, 1993.

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

[13] B. Gavish. Topological design of computer communication networks - The overall design problem. *European Journal of Operational Research*, 58:149–172, 1992.

[14] M. Gendreau, M. Labbé, and G. Laporte. Efficient heuristics for the design of ring networks. *Telecommunication Systems*, 4(3-4):177–188, 1995.

[15] M.X. Goemans. The Steiner tree polytope and related polyhedra. *Mathematical Programming*, 63:157–182, 1994.

[16] M. Held, P. Wolfe, and H.D. Crowder. Validation of subgradient optimization. *Mathematical Programming*, 6:62–88, 1974.

[17] D.S. Hochbaum and A. Segev. Analysis of a flow problem with fixed charges. *Networks*, 19:291–312, 1989.

[18] H.P.L. Luna, N. Ziviani, and R.M.B. Cabral. The telephonic switching centre network problem: Formalization and computational experience. *Discrete Applied Mathematics*, 18:199–210, 1987.

[19] N. Maculan. The Steiner problem in graphs. *Annals of Discrete Mathematics*, 31:185–212, 1987.

[20] G.R. Mateus, F.R.B. Cruz, and H.P.L. Luna. An algorithm for hierarchical network design. *Location Science*, 2(3):149–164, 1994.

[21] M. Minoux. Network synthesis and optimum network design problems: Models, solution methods and applications. *Networks*, 19:313–360, 1989.

[22] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, 1988.