
*Journal of Mathematical Modelling and Algorithms* **2:** 37–56, 2003.



# A Branch-and-Bound Algorithm to Solve a Multi-level Network Optimization Problem

F. R. B. CRUZ[1], G. R. MATEUS[2] and J. MACGREGOR SMITH[3]
[1]*Departamento de Estatística, Universidade Federal de Minas Gerais, 31270-901 – Belo Horizonte – MG, Brazil. e-mail: fcruz@est.ufmg.br*
[2]*Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, 31270-901 – Belo Horizonte – MG, Brazil. e-mail: mateus@dcc.ufmg.br*
[3]*Department of Mechanical & Industrial Engineering, University of Massachusetts, Amherst, MA 01003, U.S.A. e-mail: jmsmith@ecs.umass.edu*

(Received: 20 September 1999; accepted in final form: 13 November 2002)

**Abstract.** Multi-level network optimization problems arise in many contexts such as telecommunication, transportation, and electric power systems. A model for multi-level network design is formulated as a mixed-integer program. The approach is innovative because it integrates in the same model aspects of discrete facility location, topological network design, and dimensioning. We propose a branch-and-bound algorithm based on Lagrangian relaxation to solve the model. Computational results for randomly generated problems are presented showing the quality of our approach. We also present and discuss a real world problem of designing a two-level local access urban telecommunication network and solving it with the proposed methodology.

**Mathematics Subject Classifications (2000):** 90B10, 68M10.

**Key words:** network design, multi-level networks, topological network design, location problems, Steiner problems in graphs.

## 1. Introduction

In order to guarantee quality of service (QoS) and performance at minimum cost, network design and planning in engineering systems require policy decisions, analysis of investment strategies, and technical and development plans. Network planning must satisfy the expected demand for new services, upgrading, and improvements on the existing network. The aim is to explore the hierarchical organization of each network and to propose integrated network models as decision support systems. Within this context, we have focused solutions for basic urban mapping data capture and data analysis using a Geographic Information System (GIS) and network optimization systems [35]. The multi-level network optimization (MLNO) problem treated here is a network design model that raises optimization aspects of dimensioning, topological design, and facility location. In this sense, the model can be applied in network planning to explore design aspects at different levels in a modeling approach that integrates several hierarchical levels.

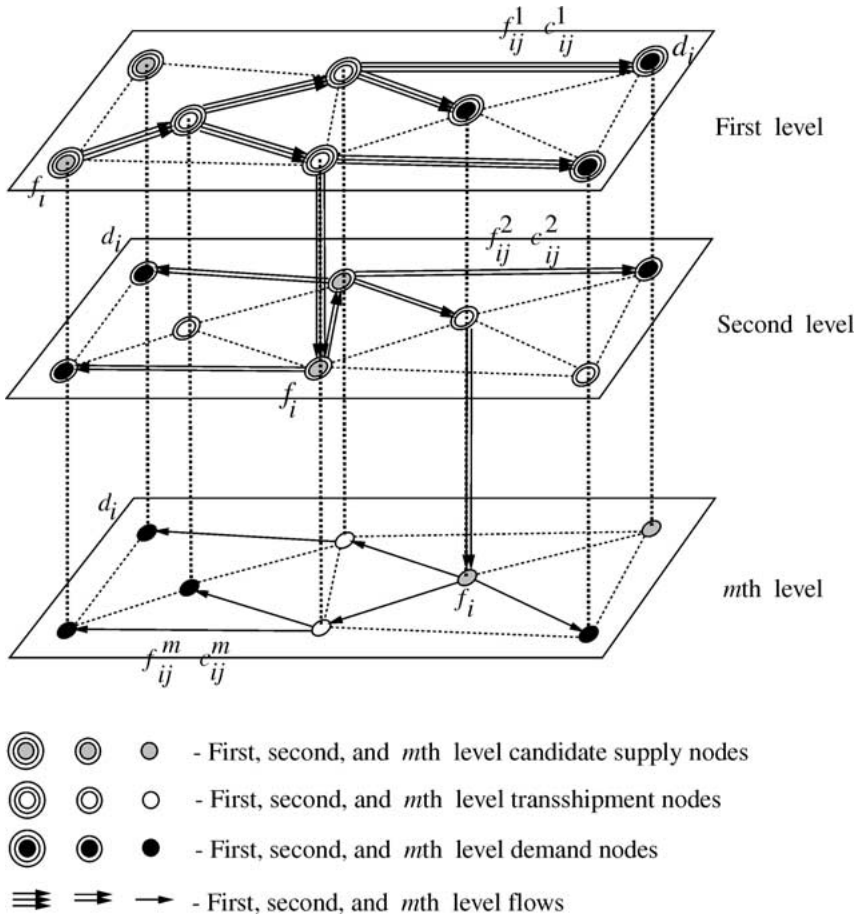



*Figure 1.* The MLNO problem.

### 1.1. MOTIVATION AND PROBLEM STATEMENT

We define the MLNO problem on a multi-weighted digraph $\mathcal{D} = (N, A)$, where $N$ is the set of nodes and $A$ is the set of arcs. Figure 1 shows a $m$-level network example containing, at each level, candidate supply nodes, demand nodes, and transshipment nodes. The objective is to determine an optimum combination of supply nodes and arcs to provide the required flow type to all demand nodes while respecting rules of flow conservation.

In economic terms, multi-level network design models are critically important. Consider, for instance, the problem of planning local access urban telecommunication networks [30]. The digraphs represent urban topologies, i.e., streets and junctions, and primary and secondary commodities represent fiber optic transmission systems and copper cables, respectively. The switching centers and certain important customers are first-level demand nodes, and households, second-level demand nodes. The network must connect the first-level demand nodes using high

bandwidth, but expensive, fiber optic systems, and the second-level demand nodes might access the network via copper cables. Road network and electric power system planning are similar applications [15]. In the transportation context, the model could be used to plan the design of all-weather highways, rough roads, and secondary roads. For electric power system design, the commodities correspond to each voltage level.

The study of multi-level networks is also important in theoretical terms because they can be viewed as generalizations of well-known topological network design (TND) problems, like TND problems oriented to telecommunications, fixed charge flow problems, Steiner problems in graphs, and uncapacitated location problems. These problems have been frequently treated by researchers in recent decades. A selective bibliography about TND problems can be found in the paper of Minoux [36]. In many works, network optimization [22, 23, 30] and dimensioning [5, 7, 32], problems oriented to telecommunication applications have been presented. In those papers, there is also a large number of references. The fixed charge flow problem was studied in [27]. The Steiner problem in graphs is also a classical model for which results have been discovered [11, 24, 25]. The uncapacitated location problem is another example of a relevant subproblem of the multi-level model. The solution of this problem has many implications in the real world and advances continue to be made for its solution [20].

### 1.2. PREVIOUS RESEARCH

The MLNO problem is $\mathcal{NP}$-hard given that it generalizes other $\mathcal{NP}$-hard optimization problems, such as the Steiner problem in graphs [21], the telephonic switching center problem [30], or the uncapacitated location problem [17]. Little research has been done on the MLNO problem. In some works, models for multi-level network design have appeared, but not as done here. Some works do not consider location aspects [15, 16], others do not consider dimensioning aspects [4, 5]. This paper integrates discrete location aspects, topological network design, and network dimensioning in the same model. Somebody might argue that the MLNO problem may not be able to capture all complexities existing in the actual network design problem. However, its solution may provide insights and a starting point for further and more accurate analyses.

### 1.3. OUTLINE OF PAPER

The paper is outlined as follows. In Section 2, we introduce the notation and present a mathematical programming formulation for the MLNO problem. In Section 3, a branch-and-bound algorithm is proposed for the problem, which is an appropriate approach to deal with $\mathcal{NP}$-hard optimization problems. We also derive a lower bound for the problem and a heuristic procedure, based on the Lagrangian relaxation, for computing feasible solutions. A preliminary version of the algorithm has

been coded in *C* and tested, and computational results are presented in Section 4. We solve a group of randomly generated problems and a sample real problem, establishing the effectiveness of our solution method. We conclude the work in Section 5, presenting and discussing some open questions.

## 2. Mixed-integer Mathematical Programming Formulation

In formulating the MLNO problem, we made some assumptions concerning the settings which are made explicit below:

(1) The arcs have cost parameters that include a non-negative fixed cost of using the arc and a non-negative cost per-unit of flow. There is a discontinuity in the zero flow values, so the total cost is a nonlinear function of the amount of flow.
(2) The total supply capacity of the first-level candidate supply nodes equals the sum of all demands in all levels.
(3) Without loss of generality, in all other levels but the first, the candidate supply nodes are 'transformation' nodes that receive flows from the respective level and 'convert' them into the immediately superior level at an 1 : 1 ratio.
(4) There may be a cost in transforming flows from one level to another. We model here possible hardware that must be present to interconnect the different networks.

### 2.1. NOTATION

As an aid to the reader, we now define the notation used:

$m$ – number of levels;
$R^l$ – set of $l$th level candidate supply nodes;
$D^l$ – set of $l$th level demand nodes.
$d_i$ – $l$th level demand node $i \in D^l$;
$T^l$ – 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 $T^m = N \setminus (R^m \cup D^m)$;
$c_{ij}^l$ – nonnegative per-unit cost for $l$th level flow on arc $(i, j) \in A$;
$x_{ij}^l$ – $l$th level flow through arc $(i, j) \in A$;
$f_{ij}^l$ – nonnegative fixed cost for using arc $(i, j) \in A$ to support $l$th level flow;
$y_{ij}^l$ – 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;
$f_i$ – nonnegative allocation cost for the $l$th level 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;
$M^l$ – capacity on all arcs in the $l$th level, but relaxed in this paper and considered a *big* enough number, i.e., $M^l = \sum_{L=l}^{m} \sum_{i \in D^L} d_i$;
$s^l$ – capacity on all $l$th level candidate supply nodes, but also relaxed in this paper, i.e., $s^l = M^l$;

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

## 2.2. FORMULATION

The mathematical programming formulation describing the MLNO problem is presented as a flow-based mixed-integer programming (MIP) model ($M$):

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

subject to

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

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

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

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

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

$$x_{ij}^l \geqslant 0, \quad \forall_{l=1,2,\ldots,m,}^{(i,j)\in A,} \tag{7}$$

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

$$z_i \in \{0, 1\}, \quad \forall_{l=1,2,\ldots,m.}^{i\in R^l,} \tag{9}$$

The objective function (1) minimizes three terms: (i) the first accounts for the total variable cost for all flow types, (ii) the second accounts for the fixed cost associated with the use of the arcs (the overhead cost), and (iii) the last considers the total cost resulting from the use of the supply nodes.

Constraints (2) ensure flow conservation between adjacent levels at each candidate supply node. Constraints (3) and (4) are the usual network flow conservation equalities at each transshipment node and demand node. From the point of view of level 1, for instance, all nodes $i \in N \setminus (R^1 \cup D^1 \cup R^2)$ are transshipment nodes (see Figure 1). Constraints (5) ensure that there is no flow transformation in a candidate supply node if that node is not selected, and constraints (6) express the fact that the flow through an arc must be zero if this arc is not included in the design.

```
algorithm Branch-and-Bound
    U_BEST ← +∞
    Γ ← {(M)^0}
    while Γ ≠ ∅ do
            /* search rule */
        select and delete a problem (M)^i from Γ
            /* bound rule */
        Compute_Lower_and_Upper_Bounds(L^i,U^i)
        update U_BEST
            /* branch rule */
        if L^i < U_BEST and (M)^i is not a leaf then
            Γ ← Γ ∪ {(M)^(2i+1)} ∪ {(M)^(2i+2)}
        end if
    end while
end algorithm
```

*Figure 2.* Branch-and-bound algorithm.

## 3. Methodology and Algorithm

An exact approach to solve any $\mathcal{NP}$-hard optimization problem normally is to implicitly enumerate all solutions. Branch-and-bound is a well-known technique largely applied to many similar problems. It can be tailored in order to get efficient solutions quickly. We propose implementations based on the branch-and-bound algorithm depicted in Figure 2. In that description, $U_{\text{BEST}}$ is the global upper bound and $\Gamma$ is a list of unexplored problems $(M)^i$, each of which is of the form $Z_M^i = \min\{\mathbf{cx}\ \mathbf{s.t.:}\ \mathbf{x} \in S^i\}$, where $S^i \subseteq S$ and $S$ is the set of feasible solutions. Associated with each problem in $\Gamma$ there are a lower bound $L^i \leqslant Z_M^i$ and an upper bound $U^i \geqslant Z_M^i$. For memory economy purposes, the search rule applied is *last-in-first-out* which yields a *depth-first* search strategy.

We shall now describe how the bounds $L^i$ and $U^i$ are computed and how the branching variable selections are made.

### 3.1. COMPUTING LOWER BOUNDS

An obvious way to compute lower bounds for MIP problems is through a linear programming relaxation (LPR) and it could be applied to the MLNO problem as well. However, we will make use of the Lagrangian relaxation (LR), which is a well-known technique to derive lower bounds, used commonly coupled with subgradient optimization procedures [18]. LR has been applied successfully to many similar combinatorial optimization problems, such as location problems [9, 20], distribution systems design [33, 34], traveling salesman problems [26], design of computer networks [22, 23], and topological network design problems [28, 29].

The idea of the method is to 'price out' complicating constraints by means of a Lagrangian multiplier vector, deriving a relaxed problem easier to solve than the original one. The relaxation is a lower bound for the original problem for any given *feasible* Lagrangian multipliers.

There are many ways to derive a LR for $(M)$. The relaxation presented in this paper divides the problem into the following subproblems:

(i) shortest path problem;
(ii) two subset selection integer problems.

Let us 'price out' constraints (5) using dual variables $v_i \geqslant 0$ and constraints (6) using dual variables $w_{ij}^l \geqslant 0$. Then it is possible to write the following Lagrangian function:

$$\begin{aligned}
&\mathcal{L}(\mathbf{x}, \mathbf{y}, \mathbf{z}; \mathbf{v}, \mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^m) \\
&\quad = \sum_{l=1}^{m} \Bigg[ \sum_{(i,j)\in A} (c_{ij}^l x_{ij}^l + f_{ij}^l y_{ij}^l) + \sum_{i \in R^l} f_i z_i + \\
&\qquad + \sum_{i \in R^l} v_i \Bigg( \sum_{j \in \delta^+(i)} x_{ij}^l - \sum_{j \in \delta^-(i)} x_{ji}^l - s^l z_i, \Bigg) + \\
&\qquad + \sum_{(i,j)\in A} w_{ij}^l \left( x_{ij}^l - M^l y_{ij}^l \right) \Bigg]. \qquad (10)
\end{aligned}$$

Thus, the LR is $(\mathrm{LR}_{\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m})$:

$$\begin{aligned}
&L(\mathbf{v}, \mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^m) \\
&\quad = \min_{\mathbf{v},\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^m \geqslant \mathbf{0}} \mathcal{L}(\mathbf{x}, \mathbf{y}, \mathbf{z}; \mathbf{v}, \mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^m), \qquad (11)
\end{aligned}$$

subject to

(2)–(4), (7)–(9).

Given 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 $L$ at $(\mathbf{v}, \mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^m)$ is:

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

Once feasible values for the Lagrangian multipliers $\mathbf{v}$, $\mathbf{w}^1$, $\mathbf{w}^2$, ..., and $\mathbf{w}^m$ are given, the computation of function $L(\mathbf{v}, \mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^m)$ is reduced to solve 'easy' subproblems. It is possible to write that

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

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

$$L_1(\mathbf{v}, \mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^m) = \min \sum_{l=1}^{m} \sum_{(i,j)\in A} C_{ij}^l x_{ij}^l, \tag{14}$$

subject to

(2)–(4), (7)

in which

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

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

subject to

(8),

$$L_3(\mathbf{v}, \mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^m) = \min \sum_{l=1}^{m} \sum_{i\in R^l} \left( f_i - v_i s^l \right) z_i, \tag{17}$$

subject to

(9).

The optimum of problem $(L_1)$ is obtained using a shortest path algorithm. The problem can be solved level by level in the following way. 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 presented in Figure 3.

The shortest *simple* paths algorithm for arbitrary costs is a polynomial procedure, $O(|N|\ |A|)$, if there are no negative cost circuits [10]. Thus, the algorithm

**algorithm** $L_1$
  /* $\sigma_i^l$ is the minimum per-unit cost to bring */
  /* $l$th level flow from set $R^1$ to node $i \in N$; */
  /* function SH$(i, j, l)$ returns the shortest */
  /* path length from $i$ to $j$ using costs $C_{ij}^l$; */
  $L1 \leftarrow 0$
  **for** $\forall j \in R^1$ **do**
    $\sigma_j^0 \leftarrow 0$
  **end for**
  **for** $l \leftarrow 1$ **to** $m$ **do**
    **for** $\forall j \in D^l$ **do**
      $\sigma_j^l \leftarrow \min_{i \in R^l} [\sigma_i^{l-1} + \text{SH}(i, j, l)]$
      $L1 \leftarrow L1 + \sigma_j^l * d_j$
    **end for**
    **if** $l \neq m$ **do**
      **for** $\forall j \in R^{l+1}$ **do**
        $\sigma_j^l \leftarrow \min_{i \in R^l} [\sigma_i^{l-1} + \text{SH}(i, j, l)]$
      **end for**
    **end if**
  **end for**
**end algorithm**

*Figure 3.* Algorithm for problem $(L_1)$.

presented in Figure 3, efficiently implemented, runs with complexity $O(m|N|\,|A|)$ which translates into $O(m|N|^3)$ for dense networks.

The subproblem $(L_2)$ is simply a subset selection problem. Its optimum can be found with polynomial complexity $O(m|A|)$ which is $O(m|N|^2)$ for dense networks.

Similarly, the subproblem $(L_3)$ is also a subset selection problem which is solvable with polynomial complexity $O(m|N|)$.

### 3.2. HEURISTICS FOR $(M)$

When the objective is to get quickly good quality feasible solutions, usually we are looking for heuristic procedures. This is an approach to solve approximately an $\mathcal{NP}$-hard optimization problem. Flexibility and computational simplicity are important characteristics for a heuristic method but first and foremost, it must give good quality solutions.

There are many possible ways of computing an upper bound (a feasible solution) for model $(M)$. The method proposed here uses the solution of $(L_1)$ and is

*Table I.* Linear programming relaxation versus Lagrangian relaxation

| | | | | LPR | | LR | | |
|---|---|---|---|---|---|---|---|---|
| Problem[a] | $\|N\|$ | $\|A\|$ | $\|D\|$ | Lower Bound | CPU (s)[b] | Lower Bound | Upper Bound | CPU (s) |
| B-1 | 50 | 126 | 8 | 1,154.25 | 0.20 | 1,154.13 | 1,222* | 2.00 |
| 2 | | | 12 | 2,450.25 | 0.27 | 2,450.24 | 2,520* | 2.20 |
| 3 | | | 24 | 4,860.17 | 0.27 | 4,859.98 | 5,012* | 2.20 |
| 4 | | 200 | 8 | 1,174.50 | 0.32 | 1,174.49 | 1,237 | 4.80 |
| 5 | | | 12 | 1,038.58 | 0.37 | 1,038.52 | 1,095* | 4.60 |
| 6 | | | 24 | 3,072.75 | 0.38 | 3,072.51 | 3,208 | 5.40 |
| 7 | 75 | 188 | 12 | 2,843.50 | 0.34 | 2,843.50 | 2,943* | 4.90 |
| 8 | | | 18 | 2,554.11 | 0.37 | 2,553.74 | 2,657* | 5.00 |
| 9 | | | 37 | 5,655.24 | 0.36 | 5,654.49 | 5,874* | 5.10 |
| 10 | | 300 | 12 | 1,946.08 | 0.60 | 1,946.02 | 2,053 | 11.00 |
| 11 | | | 18 | 3,881.44 | 0.55 | 3,881.27 | 3,987* | 11.00 |
| 12 | | | 37 | 6,738.16 | 0.62 | 6,737.60 | 6,948 | 12.00 |
| 13 | 100 | 250 | 16 | 4,266.50 | 0.43 | 4,265.95 | 4,432* | 8.90 |
| 14 | | | 24 | 8,876.83 | 0.53 | 8,876.22 | 9,117 | 9.30 |
| 15 | | | 49 | 11,052.51 | 0.54 | 11,051.60 | 11,383* | 9.20 |
| 16 | | 400 | 16 | 3,803.63 | 0.82 | 3,803.32 | 3,942 | 23.00 |
| 17 | | | 24 | 5,071.04 | 0.81 | 5,070.66 | 5,193 | 22.00 |
| 18 | | | 49 | 6,092.41 | 0.94 | 6,091.92 | 6,360 | 21.00 |

[a] Networks from Beasley [12], with $f_{ij}/\Omega_{ij} = 1$ and $c_{ij}/\Omega_{ij} = 10$.

[b] CPU time using CPLEX$^{TM}$.

* Optimal solution.

quite simple. By using an optimal solution of $(L_1)$ added by the respective utilization costs of the arcs and supply nodes in this solution, we are actually ensuring feasibility of constraints (5) and (6), 'priced out' in the relaxation process, and getting an upper bound for $(M)$.

Indeed, the main advantage of using a LR to solve the MLNO problem is the possibility of deriving such a Lagragian heuristics to solve it. The quality of the upper bounds is usually very good, as seen in Table I, for one-level networks derived from the library of Beasley [12]. On the other hand, the CPU times are larger for LR and the lower bounds are not better than those computed via LPR, given that the LR obeys the integrality property [19]. The integrality property of our relaxation is an important theoretical result easily confirmed in practice, as shown in Table I.

As observed by Rardin and Wolsey [37] and Barahona [8], stronger lower bounds can be obtained by means of multicommodity formulations since these models represent more accurately the fixed costs in the arcs. Of course, such an

*Table II.* Linear programming relaxations for $f_{ij}/c_{ij} = 10$

| | | | | SC Formulation | | MC Formulation | |
|---|---|---|---|---|---|---|---|
| Problem[a] | $\|N\|$ | $\|A\|$ | $\|D\|$ | Lower Bound (%) | CPU (s)[b] | Lower Bound (%) | CPU (s)[b] |
| B-1 | 50 | 126 | 8 | 27.46 | 0.17 | 100.00 | 7.60 |
| 2 | | | 12 | 41.06 | 0.18 | 100.00 | 59.00 |
| 3 | | | 24 | 33.79 | 0.20 | 96.11 | 370.00 |
| 4 | | 200 | 8 | 35.37 | 0.25 | 100.00 | 180.00 |
| 5 | | | 12 | 26.01 | 0.32 | 100.00 | 370.00 |
| 6 | | | 24 | 25.17 | 0.31 | 93.61 | 3,700.00 |
| 7 | 75 | 188 | 12 | 36.85 | 0.26 | 100.00 | 210.00 |
| 8 | | | 18 | 29.35 | 0.27 | 99.48 | 320.00 |
| 9 | | | 37 | 25.34 | 0.29 | * | ? |
| 10 | | 300 | 12 | 30.96 | 0.49 | 95.63 | 850.00 |
| 11 | | | 18 | 40.49 | 0.41 | 93.73 | 3,800.00 |
| 12 | | | 37 | 30.87 | 0.51 | * | ? |
| 13 | 100 | 250 | 16 | 32.47 | 0.37 | 100.00 | 730.00 |
| 14 | | | 24 | 37.26 | 0.39 | 98.19 | 1,700.00 |
| 15 | | | 49 | 28.90 | 0.43 | * | ? |
| 16 | | 400 | 16 | 35.28 | 0.73 | 95.92 | 6,600.00 |
| 17 | | | 24 | 37.52 | 0.64 | * | ? |
| 18 | | | 49 | 22.09 | 0.72 | * | ? |

[a] Networks from Beasley [12], with $f_{ij}/\Omega_{ij} = 10$ and $c_{ij}/\Omega_{ij} = 1$.

[b] CPU time using CPLEX$^{TM}$.

* Number of constraints $> 16.000$.

improvement is obtained at the cost of a substantial increase in the number of constraints. Table II shows comparisons between single commondity (SC) versus multicommotity (MC) formulations, in terms of the quality of the LPR lower bounds. The improvement on the lower bounds is noticeable as much as the CPU time explosion to compute it.

However, for multicommodity formulations coupled with LR, the performance is expected to be superior than LPR, given that the quality of the LR lower bounds is comparable to the LPR, as seen in Table I, and that the high number of constraints could be implicity and more efficiently handled by LR algorithms.

### 3.3. CHOOSING BRANCHING VARIABLES

It is common that a branching strategy works very well in some cases and poorly for others. The objective which we aim at is to reduce the number of nodes visited in the branch-and-bound search tree. In this paper, for reason of simplicity we propose

greedy-type heuristics. The branching variable will be that one for which one gets the maximum expected increment in the lower bound. So, from Equation (10), the branching variable should be:

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

where

$$\Delta_{z_k} = \begin{cases} \left| v_k \left( \sum_{j \in \delta^+(k)} {x_{kj}^l}^* - \sum_{j \in \delta^-(k)} {x_{jk}^l}^* - s^l {z_k}^* \right) \right|, & \text{if decision variable } z_k \text{ is currently free,} \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\Delta_{y_{ij}^l} = \begin{cases} \left| w_{ij}^l \left( {x_{ij}^l}^* - M^l {y_{ij}^l}^* \right) \right|, & \text{if decision variable } y_{ij}^l \text{ is currently free,} \\ 0, & \text{otherwise,} \end{cases}$$

for the last $\mathbf{x}^*$, $\mathbf{y}^*$, $\mathbf{z}^*$, $\mathbf{v}$, $\mathbf{w}^1$, $\mathbf{w}^2$, ..., and $\mathbf{w}^m$, obtained in the current node in the branch-and-bound search tree. This choice seems to be more effective than merely to choose the first free variable encountered. The complexity of this method is polynomial, $O(m|A|)$, which is $O(m|N|^2)$ for dense networks.

## 4. Experimental Results

A version of the algorithms described here was coded in *C* and is available from the authors upon request. All tests presented were performed using a workstation *Sun* Ultra 1 Model 140, RAM 128 MB, running the SunOS (*Sun* operating system), Release 5.5.1.

### 4.1. DATA STRUCTURES

Data structures must be carefully designed. As it is well-known, they play a key role in algorithm performance and storage requirements. The main abstract data type we have on hand is the multi-weighted digraph $\mathcal{D} = (N, A)$. Alternative data structures could be used to represent the digraph and the appropriate choice depends on the operations required and the amount of information stored [1]. Looking closely upon problems $(L_1)$, $(L_2)$, and $(L_3)$, we propose adjacency lists. This is a very convenient way of representing digraphs because adjacency lists can support with efficiency commonly used operations. Additionally, storage space is saved for sparse digraphs.

### 4.2. SOLVING RANDOMLY GENERATED PROBLEMS

All testing problems were generated by a random procedure similar to the method proposed by Aneja [2]. Thus, node positions, arc extremities, basic arc weights, $\Omega_{ij}$, and candidate supply and demand nodes were chosen by means of an uniform probability distribution (see [2], for futher details). The problems solved were the directed version of the graph generated, each edge being substituted by two opposite arcs with same weight. All demands were considered unitary. The costs $f_{ij}^l$ and $c_{ij}^l$ were derived from the weights $\Omega_{ij}$ by using the constant factors shown in Table III.

For the first node in the search tree, the results presented are the best upper bound, the GAP $= 100\% \times (U_{\text{BEST}} - L^0)/U_{\text{BEST}}$, and the CPU time in seconds. For the whole search tree, the results presented are the optimal solution, the number of explored nodes, and the CPU time in seconds. Some optimal solutions are not available since the maximum time allowed was 8,000 seconds. All CPU times reported are the elapsed time in order to solve the hardest instance out of 5 different ones which were tested, excluding all I/O operations and considering that only a single process was running on the machine.

Table III presents results for one-level networks with 16 and 32 nodes. All networks have only one supply node. For all groups, three different $f_{ij}^1/c_{ij}^1$ ratios were used. The ratio 1 : 10 creates problems 'closer' to the pure shortest path problem (polynomially solvable) which are expected to be easier than those created with the ratio 10 : 1, that are 'almost' pure Steiner problems ($\mathcal{NP}$-hard). The ratio 1 : 1 is merely an equally weighted mix of both problems. We remind the reader that all these combinations are still $\mathcal{NP}$-hard problems.

It is possible to see that problems with large $f_{ij}^1$ present weak bounds spending considerably more CPU time than those with large $c_{ij}^1$. The density also plays a key role in the problem hardness. In a reasonable amount of time, it was possible to solve sparse problems up to 32 nodes. In all cases solved to optimality, the solutions found in the first node proved to be the optimum, indicating that the proposed heuristic achieved good primal solutions for the problems tested. Observing the number of visited branch-and-bound nodes, it is remarkable how difficult it really is to prove optimality.

### 4.3. SOLVING A REAL CASE

We shall now solve a real case. The problem presented arose at TELEMIG, the former telephone company of *Minas Gerais* State, Brazil. This is the problem of planning a local access urban telecommunication network [30], that uses both fiber optics and copper cable links. In this application, first and second levels are meant to represent both different transmission media, fiber optical and copper cables. The fixed costs associated with the transformation nodes (second-level candidate supply nodes) mean the cost of the hardware necessary to convert the optical signal into

*Table III.* Results for one-level random network problems

| | | | | | | | | First node | | | Branch-and-bound | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Set | $\|N\|$ | $\|A\|$ | $\|R^1\|$ | $\|D^1\|$ | $\frac{f^1_{ij}}{\Omega_{ij}}$ | $\frac{c^1_{ij}}{\Omega_{ij}}$ | $f_i$ | $U_{\text{best}}$ | GAP (%) | CPU (s) | $U_{\text{opt}}$ | Nodes | CPU (s) |
| 1 | 16 | 30 | 1 | 2 | 1 | 10 | 0 | 2,658 | 1.20 | 0.06 | 2,658 | 9 | 0.18 |
| 2 | | | | | 1 | 1 | 0 | 408 | 8.10 | 0.06 | 408 | 7 | 0.12 |
| 3 | | | | | 10 | 1 | 0 | 1,830 | 18.00 | 0.07 | 1,830 | 63 | 1.60 |
| 4 | | | | 4 | 1 | 10 | 0 | 5,972 | 1.48 | 0.06 | 5,972 | 409 | 8.80 |
| 5 | | | | | 1 | 1 | 0 | 806 | 11.00 | 0.06 | 806 | 2,315 | 48.00 |
| 6 | | | | | 10 | 1 | 0 | 2,894 | 31.00 | 0.07 | 2,894 | 493 | 11.00 |
| 7 | | | | 8 | 1 | 10 | 0 | 12,250 | 2.10 | 0.07 | 12,250 | 1,879 | 43.00 |
| 8 | | | | | 1 | 1 | 0 | 1,585 | 16.00 | 0.06 | 1,585 | 5,183 | 120.00 |
| 9 | | | | | 10 | 1 | 0 | 5,185 | 49.00 | 0.07 | 5,183 | 16,365 | 410.00 |
| 10 | | 60 | 1 | 2 | 1 | 10 | 0 | 2,332 | 2.30 | 0.15 | 2,332 | 209 | 11.00 |
| 11 | | | | | 1 | 1 | 0 | 379 | 14.00 | 0.15 | 379 | 1,783 | 87.00 |
| 12 | | | | | 10 | 1 | 0 | 1,757 | 26.00 | 0.18 | 1,757 | 137 | 9.00 |
| 13 | | | | 4 | 1 | 10 | 0 | 4,066 | 4.20 | 0.15 | 4,066 | 7,265 | 350.00 |
| 14 | | | | | 1 | 1 | 0 | 646 | 26.00 | 0.15 | 646 | 14,871 | 760.00 |
| 15 | | | | | 10 | 1 | 0 | 2,400 | 45.00 | 0.17 | 2,400 | 5,537 | 310.00 |
| 16 | | 120 | 1 | 2 | 1 | 10 | 0 | 1,749 | 3.30 | 0.42 | 1,749 | 243 | 44.00 |
| 17 | | | | | 1 | 1 | 0 | 290 | 17.00 | 0.43 | 290 | 255 | 44.00 |
| 18 | | | | | 10 | 1 | 0 | 1,364 | 29.00 | 0.49 | 1,364 | 311 | 62.00 |
| 19 | | | | 4 | 1 | 10 | 0 | 3,120 | 4.70 | 0.42 | 3,120 | 7,321 | 0 |
| 20 | | | | | 1 | 1 | 0 | 500 | 28.00 | 0.43 | ** | $>$ 54,500 | $>$ 8,000.00 |
| 21 | | | | | 10 | 1 | 0 | 1,915 | 47.00 | 0.47 | 1,915 | 40,153 | 6,600.00 |
| 22 | 32 | 62 | 1 | 2 | 1 | 10 | 0 | 2,634 | 3.90 | 0.20 | 2,634 | 567 | 37.00 |
| 23 | | | | | 1 | 1 | 0 | 465 | 22.00 | 0.20 | 465 | 21 | 1.60 |
| 24 | | | | | 10 | 1 | 0 | 2,481 | 42.00 | 0.21 | 2,481 | 11 | 0.80 |
| 25 | | | | 4 | 1 | 10 | 0 | 7,645 | 4.00 | 0.20 | 7,645 | 4,799 | 310.00 |
| 26 | | | | | 1 | 1 | 0 | 1,201 | 25.00 | 0.20 | 1,201 | 33,799 | 2,200.00 |
| 27 | | | | | 10 | 1 | 0 | 5,566 | 55.00 | 0.20 | 5,566 | 2,383 | 160.00 |
| 28 | | | | 8 | 1 | 10 | 0 | 12,349 | 3.60 | 0.20 | 12,349 | 1,073 | 73.00 |
| 29 | | | | | 1 | 1 | 0 | 1,765 | 25.00 | 0.20 | 1,765 | 3,317 | 220.00 |
| 30 | | | | | 10 | 1 | 0 | 7,066 | 63.00 | 0.22 | 7,066 | 5,489 | 380.00 |
| 31 | | 124 | 1 | 2 | 1 | 10 | 0 | 2,981 | 4.50 | 0.54 | 2,981 | 10,711 | 1,700.00 |
| 32 | | | | | 1 | 1 | 0 | 536 | 24.00 | 0.56 | 536 | 747 | 150.00 |
| 33 | | | | | 10 | 1 | 0 | 2,516 | 36.00 | 0.64 | 2,516 | 1,697 | 310.00 |

*Table III.* (Continued.)

| | | | | | | | | First node | | | Branch-and-bound | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Set | $\|N\|$ | $\|A\|$ | $\|R^1\|$ | $\|D^1\|$ | $\frac{f_{ij}^1}{\Omega_{ij}}$ | $\frac{c_{ij}^1}{\Omega_{ij}}$ | $f_i$ | $U_{\text{best}}$ | GAP (%) | CPU (s) | $U_{\text{opt}}$ | Nodes | CPU (s) |
| 34 | | | | 4 | 1 | 10 | 0 | 6,891 | 3.20 | 0.54 | ** | > 50,700 | > 8,000.00 |
| 35 | | | | | 1 | 1 | 0 | 1,026 | 21.00 | 0.56 | ** | > 48,400 | > 8,000.00 |
| 36 | | | | | 10 | 1 | 0 | 3,878 | 41.00 | 0.59 | ** | > 48,400 | > 8,000.00 |

** Not available, time overflow (> 8,000.00 sec.).

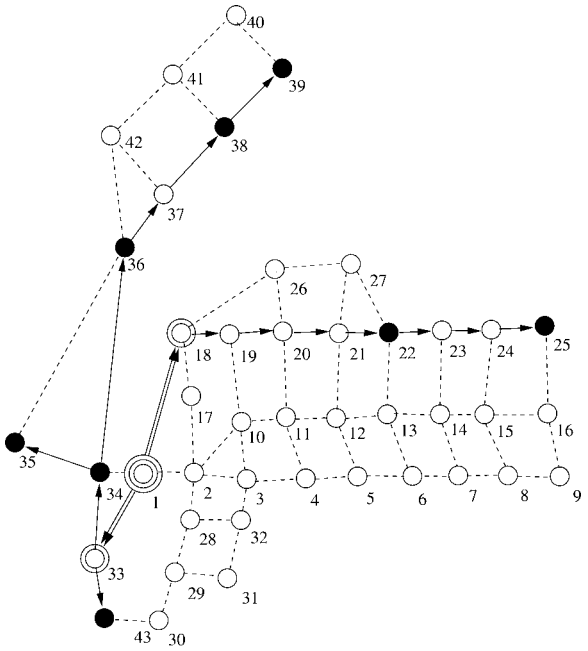
*Table IV.* Real case settings

| | |
|---|---|
| $m$ | $= 2$ |
| $N$ | $= \{1 - 43\}$ (43 nodes) |
| $D^1$ | $= \emptyset$ (no demand nodes at first level) |
| $R^1$ | $= \{1\}$ (only one supply node at first level) |
| $D^2$ | $= \{22, 25, 34 - 36, 38 - 39, 43\}$ |
| $R^2$ | $= N \setminus (D^1 \cup D^2 \cup R^1) = \{2 - 21, 23 - 24, 26 - 33, 37, 40 - 42\}$ |
| $d^i$ | $= 1, \ \forall i \in D^2$ |
| $T^1$ | $= N \setminus (R^1 \cup D^1 \cup R^2) = D^2$ |
| $T^2$ | $= N \setminus (R^2 \cup D^2) = \emptyset$ |

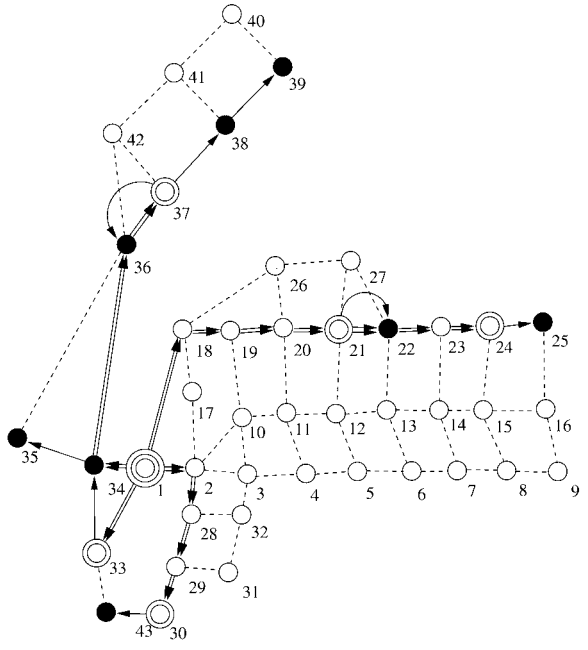
the electrical signal and vice-versa. The fixed costs in arcs usually represent the infrastructure costs, which in some sense are independent of the amount of flows the arc carries. In this context, the flow costs are obvious. We assume that the supply node in the first-level is able to provide as much flow as it is required. In other words, we assume that its capacity equals the sum of all demands.

The graph in Figure 4 represents the urban topology of *Monlevade*, a medium sized town in *Minas Gerais* State. As said previously, the arcs represent main streets and nodes represent concentrators of telephonic pairs or crosses. There is a total of 43 nodes, 68 edges (136 arcs) and 8 second-level demand nodes, considered *unitary* in this application example, although in the real world, the demand nodes usually concentrate 300 household subscribers or more. The first-level supplier is node 1 and the second-level candidate supply nodes are all remaining nodes. The MLNO problem settings are presented in Table IV.

Table V shows the arc distances $\Omega_{ij}$ in meters. We solved two cases. The arc costs used (fictitious) are proportional to the arc distances, as they are in practice, and are shown in Table VI. Table VI and Figure 4 represent the optimal solutions found. Case I shows what would happen in a scenario in which the second-level media costs were lower than first-level media costs. We see a small number of concentrators and first-level arcs. As a matter of fact, the optimum solution would be to contract the first-level network as much as possible, avoiding the use of first-level flows. On the other hand, case II represents a more practical scenario. If we




(a) Case I




(b) Case II

*Figure 4.* Results for the sample example.

*Table V.* Basic arc costs $\Omega_{ij}$ for the sample problem

| $(i, j)$ | $\Omega_{ij}$ | $(i, j)$ | $\Omega_{ij}$ | $(i, j)$ | $\Omega_{ij}$ | $(i, j)$ | $\Omega_{ij}$ |
|---|---|---|---|---|---|---|---|
| (1, 2) | 130 | (12, 5) | 85 | (18, 19) | 40 | (36, 35) | 315 |
| (3, 2) | 50 | (13, 6) | 100 | (18, 17) | 105 | (36, 37) | 120 |
| (3, 4) | 65 | (14, 7) | 110 | (17, 2) | 50 | (37, 38) | 290 |
| (4, 5) | 55 | (15, 8) | 130 | (2, 10) | 50 | (38, 39) | 160 |
| (5, 6) | 65 | (19, 10) | 100 | (18, 1) | 130 | (39, 40) | 140 |
| (6, 7) | 55 | (19, 20) | 70 | (26, 20) | 90 | (40, 41) | 200 |
| (7, 8) | 60 | (20, 21) | 60 | (26, 27) | 60 | (41, 42) | 180 |
| (8, 9) | 75 | (21, 22) | 65 | (27, 21) | 70 | (42, 37) | 235 |
| (9, 16) | 125 | (22, 23) | 60 | (27, 22) | 100 | (41, 38) | 141 |
| (16, 15) | 70 | (23, 24) | 60 | (1, 34) | 60 | (42, 36) | 340 |
| (15, 14) | 60 | (24, 25) | 80 | (1, 33) | 150 | (33, 34) | 150 |
| (14, 13) | 60 | (16, 25) | 100 | (32, 3) | 55 | (18, 26) | 130 |
| (13, 12) | 60 | (20, 11) | 100 | (28, 32) | 95 | (33, 43) | 150 |
| (12, 11) | 55 | (21, 12) | 95 | (32, 31) | 50 | (43, 30) | 50 |
| (11, 10) | 70 | (22, 13) | 100 | (29, 31) | 125 | (30, 29) | 60 |
| (10, 3) | 55 | (23, 14) | 105 | (34, 36) | 215 | (29, 28) | 60 |
| (11, 4) | 70 | (24, 15) | 100 | (35, 34) | 230 | (28, 2) | 80 |

had first-level facility costs lower than second-level ones, we would tend to increase the size of the first-level network. That is exactly what happens nowadays. Because of a continuous reduction of their costs, we are watching an increasing spread of high speed networks.

## 5. Summary and Conclusions

A multi-level network optimization (MLNO) problem that integrates location, topological network design, and dimensioning aspects was discussed. One possible mathematical programming formulation for the MLNO problem was presented and a branch-and-bound algorithm based on this formulation was used to solve it.

The computational results presented here were for one and two-level networks but multi-level networks could easily be handled by the proposed methodology because all procedures were developed for the general case. In these examples, the difficulty of the problem was analyzed in terms of network size, density, number of candidate supply nodes, and in terms of allocation costs.

In order to increase the size of the manageable instances and to make larger practical networks tractable, future research might further explore reduction tests. In fact, effective reduction tests were used to solve subproblems of the MLNO problem such as large uncapacitated location problems [13], uncapacitated fixed-charge network flow problems [14], and Steiner problems in graphs [31]. With

*Table VI.* Results for the sample problems

| Case | $\frac{f_{ij}^1}{\Omega_{ij}}$ | $\frac{c_{ij}^1}{\Omega_{ij}}$ | $\frac{f_{ij}^2}{\Omega_{ij}}$ | $\frac{c_{ij}^2}{\Omega_{ij}}$ | $f_i$ | $U_{\text{opt}}$ | $\{i \mid z_i = 1\}$ | Paths* |
|---|---|---|---|---|---|---|---|---|
| I | 2 | 20 | 1 | 10 | 1 | 59,763 | {1, 18, 33} | $1 \Rightarrow 18 \rightarrow 19 \rightarrow 20 \rightarrow 21 \rightarrow 22 \rightarrow 23 \rightarrow 24 \rightarrow 25$<br>$1 \Rightarrow 18 \rightarrow 19 \rightarrow 20 \rightarrow 21 \rightarrow 22$<br>$1 \Rightarrow 33 \rightarrow 34 \rightarrow 36 \rightarrow 37 \rightarrow 38 \rightarrow 39$<br>$1 \Rightarrow 33 \rightarrow 34 \rightarrow 36 \rightarrow 37 \rightarrow 38$<br>$1 \Rightarrow 33 \rightarrow 34 \rightarrow 36$<br>$1 \Rightarrow 33 \rightarrow 34 \rightarrow 35$<br>$1 \Rightarrow 33 \rightarrow 34$<br>$1 \Rightarrow 33 \rightarrow 43$ |
| II | 1 | 10 | 2 | 20 | 1 | 61,356 | {1, 24, 21, 30, 33, 37} | $1 \Rightarrow 18 \Rightarrow 19 \Rightarrow 20 \Rightarrow 21 \Rightarrow 22 \Rightarrow 23 \Rightarrow 24 \rightarrow 25$<br>$1 \Rightarrow 18 \Rightarrow 19 \Rightarrow 20 \Rightarrow 21 \rightarrow 22$<br>$1 \Rightarrow 2 \Rightarrow 28 \Rightarrow 29 \Rightarrow 30 \rightarrow 43$<br>$1 \Rightarrow 33 \rightarrow 34 \rightarrow 35$<br>$1 \Rightarrow 33 \rightarrow 34$<br>$1 \Rightarrow 34 \Rightarrow 36 \Rightarrow 37 \rightarrow 38 \rightarrow 39$<br>$1 \Rightarrow 34 \Rightarrow 36 \Rightarrow 37 \rightarrow 38$<br>$1 \Rightarrow 34 \Rightarrow 36 \Rightarrow 37 \rightarrow 36$ |

* $\Rightarrow$ first-level flow, $\rightarrow$ second-level flow.

the emerging emphasis on topological robustness and reliability, the study of enhanced models that incorporate connectivity constraints is a very promising area for investigations [3, 6, 38].

## Acknowledgments

The authors wish to thank Prof. Nelson Maculan for his careful reading and valuable comments on an earlier version of this paper. The work of Frederico R. B. Cruz has been partially funded by the CNPq (*Conselho Nacional de Desenvolvimento Científico e Tecnológico*) of the Ministry for Science and Technology of Brazil, grants 301809/96-8 and 201046/94-6, the FAPEMIG, grants CEX-289/98 and CEX-855/98, and PRPq-UFMG, grant 4081-UFMG/ RTR/ FUNDO/ PRPq/ 99. The work of Geraldo Robson Mateus is supported in part by grants from the CNPq and FAPEMIG.

## References

1. Aho, A. V., Hopcroft, J. E. and Ullman, J. D.: *Data Structures and Algorithms*, Addison-Wesley, Reading, Mass., 1983.
2. Aneja, Y. P.: An integer linear programming approach to the Steiner problem in graphs, *Networks* **10** (1980), 167–178.
3. Balakrishnan, A. and Altinkemer, K.: Using a hop-constrained model to generate alternative communication network design, *ORSA J. Comput.* **4** (1992), 192–205.
4. Balakrishnan, A., Magnanti, T. L. and Mirchandani, P.: A dual-based algorithm for multi-level network design, *Manag. Sci.* **40**(7) (1994), 567–581.
5. Balakrishnan, A., Magnanti, T. L. and Mirchandani, P.: Modeling and heuristic worst-case performance analysis of two-level network design problem, *Manag. Sci.* **40**(7) (1994), 846–867.
6. Balakrishnan, A., Magnanti, T. L. and Mirchandani, P.: Designing Hierarchical Survivable Networks, *Oper. Res.* **46**(1) (1998), 116–136.

7. Balakrishnan, A., Magnanti, T. L., Shulman, A. and Wong, R. T.: Models for planning capacity expansion in local access telecommunication networks, *Ann. Oper. Res.* **33** (1991), 239–284.
8. Barahona, F.: Network design using cut inequalities, *SIAM J. Optim.* **6**(3) (1996), 823–837.
9. Barcelo, J. and Casanovas, J.: A heuristic Lagrangian algorithm for the capacitated plant location problem, *Europ. J. Oper. Res.* **15** (1984), 212–226.
10. Bazaraa, M. S., Jarvis, J. J. and Sherali, H. D.: *Linear Programming and Networks Flows*, 2nd edn, Wiley, New York, 1990.
11. Beasley, J. E.: An SST-based algorithm for the Steiner problem in graphs, *Networks* **19** (1989), 1–16.
12. Beasley, J. E.: OR-library: distributing test problems by electronic Mail, *J. Oper. Res. Soc.* **41**(11) (1990), 1069–1072.
13. Christofides, N. and Beasley, J. E.: A tree search algorithm for the P-median problem, *Europ. J. Oper. Res.* **10** (1982), 196–204.
14. Cruz, F. R. B., MacGregor Smith, J. and Mateus, G. R.: Solving to optimality the uncapacitated fixed-charge network flow problem, *Comp. Oper. Res.* **25**(1) (1998), 67–81.
15. Current, J. R., ReVelle, C. S. and Cohon, J. L.: The hierarchical network design problem, *Europ. J. Oper. Res.* **27** (1986), 57–66.
16. Duin, C. W. and Volgenant, A.: Reducing the hierarchical network design problem, *Europ. J. Oper. Res.* **39** (1989), 332–344.
17. Erlenkotter, D.: A dual-based procedure for uncapacitated facility location, *Oper. Res.* **26**(6) (1978), 992–1009.
18. Fisher, M. L.: The Lagrangian relaxation method for solving integer programming problems, *Manag. Sci.* **27** (1980), 1–18.
19. Fisher, M. L.: An application oriented guide to Lagrangian relaxation, *Interfaces* **15** (1985), 10–21.
20. Galvão, R. D. and Raggi, L. A.: A method for solving to optimality uncapacitated location problems, *Ann. Oper. Res.* **18** (1989), 225–244.
21. Garey, M. R. and Johnson, D. S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York, 1979.
22. Gavish, B.: Topological design of telecommunication networks – local access design methods, *Ann. Oper. Res.* **33** (1991), 17–71.
23. Gavish, B.: Topological design of computer communication networks – The overall design problem, *Europ. J. Oper. Res.* **58** (1992), 149–172.
24. Goemans, M. X.: The Steiner tree polytope and related polyhedra, *Math. Progr.* **63** (1994), 157–182.
25. Goemans, M. X. and Myung, Y.: A catalog of Steiner tree formulations, *Networks* **23** (1993), 19–28.
26. Held, M. and Karp, R. M.: The traveling salesman problem and minimum spanning trees, *Oper. Res.* **18** (1970), 1138–1162.
27. Hochbaum, D. S. and Segev, A.: Analysis of a flow problem with fixed charges, *Networks* **19** (1989), 291–312.
28. Holmberg, K. and Hellstrand, J.: Solving the uncapacitated network design problem by a Lagrangian heuristic and branch-and-bound, *Oper. Res.* **46**(2) (1998), 247–259.
29. Holmberg, K. and Yuan, D.: A Lagrangian approach to network design problems, *Internat. Trans. Oper. Res.* **5**(6) (1998), 529–539.
30. Luna, H. P. L., Ziviani, N. and Cabral, R. M. B.: The telephonic switching centre network problem: Formalization and computational experience, *Discrete Appl. Math.* **18** (1987), 199–210.
31. Maculan, N., Souza, P. and Vejar, A. C.: An approach for the Steiner problem in directed graphs, *Ann. Oper. Res.* **33** (1991), 471–480.

32. Mateus, G. R., Cruz, F. R. B. and Luna, H. P. L.: An algorithm for hierarchical network design, *Location Science* **2**(3) (1994), 149–164.
33. Mateus, G. R. and Franqueira, R. V. L.: Model and heuristic for a generalized access network design problem, *Telecom. Systems* **15**(3–4) (2000), 257–271.
34. Mateus, G. R., Luna, H. P. L. and Sirihal, A. B.: Heuristic for distribution network design in telecommunication, *J. Heuristics* **6** (2000), 131–148.
35. Mateus, G. R., Pádua, C. I. P. S. and Luna, H. P. L.: Integrated network models for local access network design, In: *Proc. Internat. Telecom. Sympos. 1996*, Acapulco, Mexico, 1996, pp. 6–10.
36. Minoux, M.: Network synthesis and optimum network design problems: Models, solution methods and applications, *Networks* **19** (1989), 313–360.
37. Rardin, R. L. and Wolsey, L. A.: Valid inequalities and projecting the multicommodity extended formulation for uncapacitated fixed charge network flow problems, *Europ. J. Oper. Res.* **71** (1993), 95–109.
38. Voß, S.: The Steiner tree problem with Hop constraints, *Ann. Oper. Res.* **86** (1999), 321–345.