Proceedings of the XII Brazilian Automatic Control Conference - XII CBA
Vol. III, pp. 961-966 - September 14 - 18, 1998 - Uberlândia, MG, Brazil

STATISTICAL ANALYSIS OF ADDITIONAL CONSTRAINTS IN THE SOLUTION OF UNCAPACITATED FIXED-CHARGE NETWORK FLOW PROBLEMS

Frederico R. B. da Cruz[1], Geraldo R. Mateus[2], Enrico A. Colosimo[1]

[1]*Departamento de Estatística,* [2]*Departamento de Ciência da Computação,*
Universidade Federal de Minas Gerais, 31270-901 - Belo Horizonte - MG, Brazil
E-mail: {fcruz@est,mateus@dcc,enricoc@est}.ufmg.br

James MacGregor Smith

Department of Mechanical and Industrial Engineering,
The University of Massachusetts, Amherst, MA 01003, USA
E-mail: jmsmith@ecs.umass.edu

Abstract— The uncapacitated fixed-charge network flow (UFNF) problem results from the combination of the *Steiner* problem in graphs and the minimum cost network flow (MCNF) problem. It is a very important mixed-integer programming problem with many applications for the real world. In this paper, we investigate the use of an additional constraint that enforces the presence of a minimum number of arcs in the set of feasible solutions. Computational results and statistical analysis are presented showing that our approach is quite promising.

Key Words— Optimization problems; Network topologies; Operations research; Networks optimization; Mixed-integer programming.

1 Introduction

Network design and planning in engineering systems requires policy decisions, analysis of investment strategies, technical and development plans, in order to guarantee service performance and quality at minimal cost. Telecommunication, transportation and electrical system network planning must satisfy the expected demand for new services and upgrade and improve the existing network. The aim is to explore the hierarchical organization of each network and propose integrated network models as a decision support system. In this context, we have focused solutions for basic urban mapping data capture and the analysis of data, using a Geographic Information System (GIS), and systems for network optimization (Mateus et al., 1996).

The uncapacitated fixed-charge network flow (UFNF) problem is a network model that raises optimization aspects of dimensioning, topological design, routing and location of facilities. In this sense, it can be applied in network planning to explore the design aspects in the different levels of an hierarchical modeling approach.

We define the UFNF problem on a digraph $\mathcal{D} = (N, A)$, where N is the set of nodes and A is the set of arcs. One of the costs involved is the fixed cost of using an arc to send flow and the other is a variable cost dependent on the amount of flow sent through the arc. The objective is to determine a minimum cost arc combination that provides flows from certain nodes, the supply nodes, to the demand nodes, possibly using intermediate *Steiner* or transshipment nodes. Figure 1 illustrates an instance of the problem.

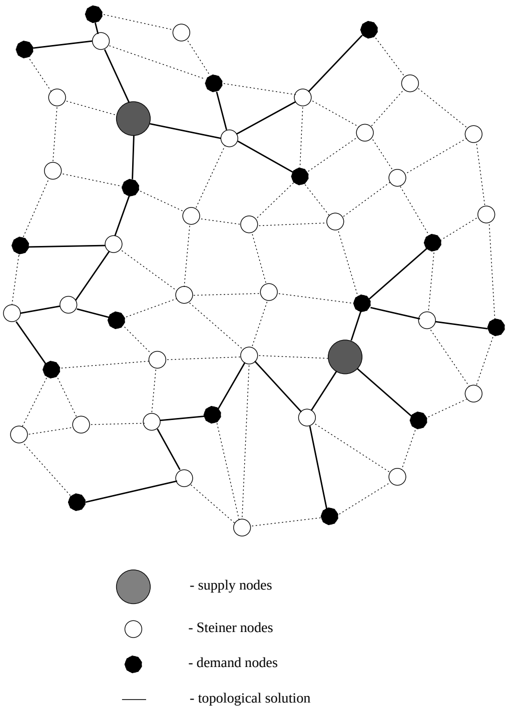


Figure 1. The UFNF Problem.

This problem is clearly $\mathcal{NP}$-hard since it generalizes the *Steiner* problem in graphs (Beasley, 1984), which is known to be $\mathcal{NP}$-hard (Garey and Johnson, 1979). The model has applications for distribution, transportation, and communication problems and it is also useful for certain routing problems.

In the field of distribution systems, a problem of designing offshore natural-gas pipeline systems was treated by Rothfarb et al. (1970), using this model. In the problem, the central separation plant is located on land and typically serves multiple offshore under-water gas wells. The problem is how to transport gas from the offshore wells to the separation plant at minimum cost. An important characteristic of gas pipelines systems is that the construction cost consists of fixed components that are independent of the amount of flow and variable components which are proportional to the amount of flow.

In the work of Luna et al. (1987), the switching center network problem was solved using a similar model. The switching center network design problem consists of looking for a topology on the urban street network that minimizes the total cost of cables and subterranean piping infrastructure necessary to link a telephone center and its subscribers. The fixed cost models the subterranean infrastructure and the flow-dependent cost models the cables.

The UFNF model is also useful in those cases where the network topology exists. Important routing problems emerge in existing networks. Suppose for example that one specific node in the network has to send messages at minimum cost addressed to a specific collection of other nodes in the network. The UFNF model is applicable since it is a reasonable assumption that an initial set up cost is associated with each link selected independent on the flow as well as a variable cost dependent on how much information has to be sent.

The main purpose of this paper is to investigate possible performance improvements, in the context of exact branch-and-bound algorithms, caused by enforcing a minimum number of arcs to be present in all feasible solutions of the UFNF problem.

The paper is outlined as follows. In Section 2, we present a mathematical programming formulation for the UFNF problem, discuss some algorithms studied previously, and propose a new one. In Section 3 and 4, computational results and statistical analysis are presented and discussed. Section 5 closes the paper with the presentation of open questions and final remarks.

2 Formulation and Solution

One possible mixed-integer mathematical programming formulation for the UFNF problem is:

(M):

$$\min \sum_{(i,j)\in A} \left(c_{ij}x_{ij} + f_{ij}y_{ij} \right), \tag{1}$$

s.t.:

$$\sum_{j\in\delta^-(i)} x_{ji} - \sum_{j\in\delta^+(i)} x_{ij} = \begin{cases} -\sum_{k\in D} d_k, & \forall i \in S, \\ 0, & \forall i \in T, \\ d_i, & \forall i \in D, \end{cases} \tag{2}$$

$$x_{ij} \le \left(\sum_{k\in D} d_k \right) y_{ij}, \forall (i,j) \in A, \tag{3}$$

$$x_{ij} \ge 0, \forall (i,j) \in A, \tag{4}$$

$$y_{ij} \in \{0,1\}, \forall (i,j) \in A, \tag{5}$$

where c_{ij} is the variable cost per unit of flow on arc (i,j), f_{ij} is the fixed cost of having flow on arc (i,j), $\delta^+(i) = \{j|\ (i,j) \in A\}$, $\delta^-(i) = \{j|\ (j,i) \in A\}$, d_i is the demand at node i, S is the set of supply nodes, T is the set of *Steiner* nodes, and D is the set of demand nodes.

Of course, multiple supply nodes are possible with the model but, for the sake of the argument, only one supply node is treated in this paper. We also assume the fixed costs f_{ij} are non-negative and the variable costs c_{ij} are unconstrained. However, to make sure that the objective function is bounded from below, we assume that there are no negative-cost direct cycles with respect to c_{ij}.

2.1 Previous Algorithms

Some experimental work concerning approximate and exact solutions for the UFNF problems and their special cases has been done previously. An analysis of offshore natural-gas systems was done in Rothfarb et al. (1970), with the cost model being simplified including only fixed costs. In Luna et al. (1987), the model studied was even more complex than the UFNF problem, presenting some additional features. However, only heuristic procedures and local optimization techniques were considered.

The special case without *Steiner* nodes was treated by Magnanti et al. (1986), and by Hochbaum and Segev (1989). In the former, an exact branch-and-bound algorithm combined with Benders cuts was studied and, in the latter, a set of heuristic procedures based on Lagrangean relaxation techniques (Geoffrion, 1974) was developed. Some other special cases were solved exactly by Barr et al. (1981), Cabot and Erengue (1984), and Suhl (1985), by means of fractional cutting-plane algorithms.

Concerning the general UFNF problem, we have noticed contributions on approximate and exact methods. In Mateus et al. (1994), *ADD* and *DROP* heuristic approaches were studied. In Cruz et al. (1994), a Lagrangean relaxation based heuristics was proposed. In another work of Cruz et al. (1998), a successful exact branch-and-bound algorithm was developed. The exact algorithm proposed by them employs the Lagrangean relaxation of constraints (3) to compute the lower and upper bounds of model (M).

Indeed, defining $K_0 \subseteq A$ as the set of arcs that have been positively identified, by some reduction method, as not-present in an optimal solution, $K_1 \subseteq A$ as the set of arcs that have been positively identified as present in an optimal solution, and $K = A \setminus K_0 \setminus K_1$ as the set of free or undefined arcs, and dropping the capacity constraints (3) by means of dual variables $w_{ij} \geq 0, \forall (i,j) \in K$, the following Lagrangean function results:

$$\mathcal{L}(\mathbf{x},\mathbf{y};\mathbf{w}) = \sum_{(i,j)\in A} (c_{ij}x_{ij} + f_{ij}y_{ij}) + \sum_{(i,j)\in K} w_{ij}\left[x_{ij} - \left(\sum_{k\in D} d_k\right) y_{ij}\right]. \quad (6)$$

Consequently, the Lagrangean relaxation of the model (M) may be written:

$(LR_{\mathbf{w}})$**:**

$$L(\mathbf{w}) = \min \left\{ \mathcal{L}(\mathbf{x},\mathbf{y};\mathbf{w}) \text{ s.t.: } (2), (4), (5), \mathbf{w} \geq 0 \right\}. \quad (7)$$

For any *feasible* Lagrangean multiplier vector, $\mathbf{w} \geq 0$, the solution of (7) is a lower bound for the original problem since the quantity $\sum_{(i,j)\in K} w_{ij}[x_{ij}{}^* - (\sum_{k\in D} d_k)y_{ij}{}^*]$ is always non-positive, considering $L(\mathbf{w}) = \mathcal{L}(\mathbf{x}^*,\mathbf{y}^*;\mathbf{w})$ (Fisher, 1985). Thus, the lower bound computations reduce to solving two easy (polynomial) subproblems: (i) a minimum cost network flow (MCNF) problem in $\mathbf{x}$ and (ii) a set-selection problem in $\mathbf{y}$.

The problem in $\mathbf{x}$, model (LR_1) below, is solvable in polynomial time applying the $O(|N|\ |A|)$ shortest *simple* paths algorithm for arbitrary costs developed by Glover et al. (1985):

(LR_1)**:**

$$\min \sum_{(i,j)\in A} C_{ij}x_{ij}, \quad (8)$$

s.t.:

$$\sum_{j\in\delta^-(i)} x_{ji} - \sum_{j\in\delta^+(i)} x_{ij} = \begin{cases} -\sum_{k\in D} d_k, & \forall i \in S, \\ 0, & \forall i \in T, \\ d_i, & \forall i \in D, \end{cases} \quad (9)$$

$$x_{ij} \geq 0, \forall (i,j) \in K, \quad (10)$$

$$x_{ij} \leq \sum_{k\in D} d_k, \forall (i,j) \in K_1, \quad (11)$$

$$x_{ij} \geq 0, \forall (i,j) \in K_1, \quad (12)$$

$$x_{ij} = 0, \forall (i,j) \in K_0, \quad (13)$$

where

$$C_{ij} = \begin{cases} c_{ij} + w_{ij}, & \forall (i,j) \in K, \\ c_{ij}, & \forall (i,j) \in A \setminus K. \end{cases} \quad (14)$$

The problem in $\mathbf{y}$, model (LR_2) below, is also polynomially solvable employing an $O(|A|)$ algorithm, since all we need to do is to set to 1 all y_{ij}'s for which $F_{ij} < 0$ holds:

(LR_2)**:**

$$\min \sum_{(i,j)\in A} F_{ij}y_{ij}, \quad (15)$$

s.t.:

$$y_{ij} \in \{0,1\}, \quad \forall (i,j) \in K, \quad (16)$$

$$y_{ij} = 1, \quad \forall (i,j) \in K_1, \quad (17)$$

$$y_{ij} = 0, \quad \forall (i,j) \in K_0, \quad (18)$$

where

$$F_{ij} = \begin{cases} f_{ij} - w_{ij}(\sum_{k\in D} d_k) & \forall (i,j) \in K, \\ f_{ij} & \forall (i,j) \in A \setminus K. \end{cases} \quad (19)$$

2.2 Proposed Algorithm

Our proposal is based on the algorithm developed by Cruz et al. (1998). As noted by them, the lower bounds obtained by the Lagrangean relaxation $(LR_{\mathbf{w}})$ are usually poor and the objective of this work is to present computational experiments in which the modified set-selection problem is applied instead:

(LR_2')**:**

$$\min \sum_{(i,j)\in A} F_{ij}y_{ij}, \quad (20)$$

s.t.:

$$y_{ij} \in \{0,1\}, \quad \forall (i,j) \in K, \quad (21)$$

$$y_{ij} = 1, \quad \forall (i,j) \in K_1, \quad (22)$$

$$y_{ij} = 0, \quad \forall (i,j) \in K_0, \quad (23)$$

$$|Y| \geq \gamma, \quad (24)$$

where $Y = \{(i,j)|\ y_{ij} = 1\}$ and γ is the minimum number of arcs that must be present in the solution, *i.e.* the minimum cardinality of set Y.

For *small* values of γ, the added constraint (24) is clearly redundant. Its inclusion does not change the optimum of the original model (M). This new problem in $\mathbf{y}$, (LR_2'), although computationally harder, can also be solved polynomially, $O(|A|\ \log|A|)$. That increment in the complexity is caused by a sorting step that must be included in the solution algorithm.

An important question is whether or not constraint (24) can tighten up the lower bounds consequently improving the performance of branch-and-bound algorithms.

3 Experimental Results

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

[a]Sun Microsystems, Inc.

Table 1. Computational Results ($|N| = 16$, $|A| = 60$, and $|D| = 6$).

					Algorithms Based on Model (LR_2')									
			Based on (LR_2)		$\gamma = 0$			$\gamma = \lvert D\rvert$			$\gamma = \lvert Y\rvert^*$			
f_{ij}/c_{ij}	Block	$\lvert Y\rvert^*$	Nodes	CPU(s)	Gap‡	Nodes	CPU(s)	Gap‡	Nodes	CPU(s)	Gap‡	Nodes	CPU(s)
100	B_1	10	3,875	110.00	71.00	3,875	170.00	60.00	6,053	270.00	42.00	2,231	110.00
	B_2	9	201	8.10	69.00	201	12.00	56.00	215	14.00	38.00	65	5.00
	B_3	7	157	6.70	75.00	157	11.00	50.00	283	17.00	39.00	135	7.70
	B_4	7	359	12.00	62.00	359	17.00	38.00	255	12.00	31.00	55	3.40
	B_5	8	683	21.00	73.00	683	33.00	60.00	831	41.00	50.00	519	25.00
10	B_1	10	2,533	71.00	62.00	2,533	110.00	53.00	4,201	180.00	36.00	1,431	69.00
	B_2	9	183	7.00	61.00	183	10.00	48.00	183	11.00	31.00	49	3.40
	B_3	7	97	4.40	66.00	97	6.90	44.00	263	15.00	36.00	131	7.30
	B_4	7	119	4.20	52.00	119	6.40	32.00	63	3.40	26.00	15	1.10
	B_5	8	497	15.00	65.00	497	23.00	53.00	565	26.00	44.00	347	17.00
1	B_1	10	245	7.70	28.00	245	12.00	24.00	369	18.00	17.00	147	7.00
	B_2	9	59	2.50	29.00	59	3.90	27.00	133	8.30	20.00	31	2.20
	B_3	8	25	1.60	32.00	25	2.50	22.00	45	2.90	13.00	3	0.48
	B_4	7	33	1.10	22.00	33	1.70	15.00	29	1.30	12.00	9	0.73
	B_5	8	77	2.90	32.00	77	4.50	27.00	77	4.20	23.00	49	2.40
0.1	B_1	10	7	0.52	4.30	7	0.77	3.70	11	0.96	2.60	5	0.55
	B_2	9	61	1.70	5.60	61	2.70	4.60	71	3.10	3.40	25	1.30
	B_3	8	1	0.20	5.10	1	0.29	3.50	1	0.28	2.10	1	0.27
	B_4	7	9	0.46	4.00	9	0.70	2.80	9	0.66	2.50	9	0.70
	B_5	8	1	0.22	5.60	1	0.32	5.30	13	0.82	4.70	1	0.35
0.01	B_1	10	1	0.20	0.45	1	0.28	0.39	1	0.28	0.27	1	0.31
	B_2	8	25	0.73	0.73	25	1.10	0.63	25	1.10	0.55	1	0.31
	B_3	8	1	0.20	0.55	1	0.29	0.38	1	0.28	0.22	1	0.27
	B_4	7	1	0.19	0.42	1	0.26	0.30	1	0.26	0.27	1	0.26
	B_5	8	1	0.20	0.65	1	0.29	0.57	1	0.28	0.50	1	0.28

‡Gap (in the first node) = 100% × [(Best Upper Bound) − (Best Lower Bound)]/(Best Upper Bound)

Table 2. Analysis of Variance (Balanced Design).

Source	Degrees of Freedom	Sum of Squares	Mean Square	F	p-value
Problems	4	268.195	67.049	17.45	0.000
Blocks (Problems)	20	76.835	3.842	42.37	0.000
Algorithms	3	8.261	2.754	30.37	0.000
Problem × Algorithm	12	1.494	0.125	1.37	0.204
Error	60	5.440	0.091	-	-
Total	99	360.226	-	-	-

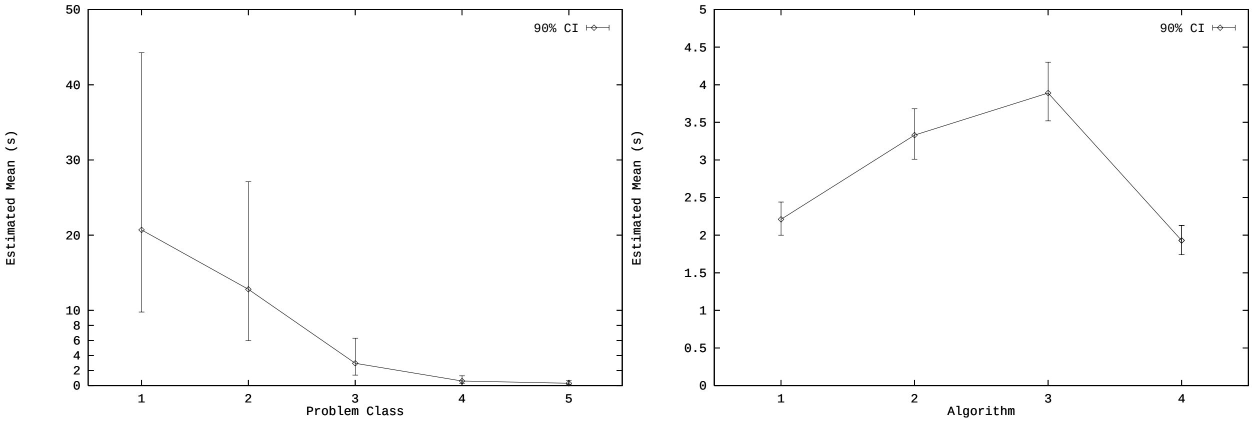
5.5.1. All test problems are single-supply-node Euclidean graphs randomly generated using a procedure similar to one presented by Aneja (1980). According to this procedure, node positions, arc extremities, arc weights Ω_{ij}, supplying node, and demand nodes are randomly chosen using a uniform distribution. Interested readers are encouraged to look at Aneja's (1980) paper for further details.

The problems actually solved were the directed version of the graph generated, each edge being substituted by two opposite arcs with the same weight. All demands were considered unitary. The costs f_{ij} and c_{ij} were derived from the weights Ω_{ij} using the constant factors 1, 10, and 100. All CPU times reported are the elapsed time, in seconds, to solve the instance, excluding all I/O operations and considering that only a single process was running on the machine.

Table 1 presents the results of all computational experiments. For each of the 25 different problems tested, Table 1 shows the ratio f_{ij}/c_{ij}, the number of arcs present in the optimal solution, which was denoted by $|Y|^*$. It also shows, for each of the 4 algorithms tested, the gap in the first node of the branch-and-bound search tree (the gaps for the first two algorithms are the same), the total number of explored nodes and the CPU time, in seconds, spent by the algorithm before optimality was reached. The first class uses the ratio $f_{ij}/c_{ij} = 100$, resulting in almost *Steiner* problems, which are known to be $\mathcal{NP}$-hard. On the other hand, the last class uses the ratio $f_{ij}/c_{ij} = 0.01$, being almost MCNF problems which are polynomially solvable. However, we remind the reader that both classes *still* are UFNF problems which are $\mathcal{NP}$-hard!

For each of these 25 problems, 4 different algorithms were considered. The first one corresponds to solve the problem using model (LR_2), meaning that the cardinality of set Y was let free. The second uses model (LR_2') and also let the cardinality free[b]. These two cases are meant only to demonstrate, in practice, the impact of model (LR_2') in the overall computation time. We remind the

[b]To ensure free cardinality, we must set $\gamma = 0$.



a) Problem Classes b) Algorithms

Figure 2. 90% Confidence Intervals.

reader that (LR'_2) is a problem computationally harder than (LR_2). In the remaining cases, the minimum cardinality was fixed to different values up to the value from which constraint (24) is no longer redundant and could change the optimal solution of the original UFNF problem[c].

An immediate minimum cardinality for set Y is $|D|$ because at least one arc must enter into each demand node. However, as we will shortly show in the statistical analysis of the data, this bound is not effective because it does not reduce substantially the number of nodes or the CPU times. Finally, using the tightest bound possible (in Table 1, see the last three columns) the best performance was found. We now move on to the statistical analysis of the data.

4 Statistical Analysis

Looking at Table 1 and considering that the experiments were properly randomized, it is possible to recognize a two-factor experiment (problems and algorithms) with repeated measures on one factor (algorithms). For additional details on this matter, we recommend *e.g.* Neter et al. (1990). In this analysis of variance, both factors, problems and algorithms, are of equal interest, as is the possibility that these factors interact.

Now, let us consider a mathematical model that might describe such a design. The model assumes that the response variable Y_{ijk} (CPU time) may be represented by the sum of the general mean $\mu_{\ldots}$, the factor A (problem) main effect α_j, the factor B (algorithm) main effect β_k, the interaction effect $(\alpha\beta)_{jk}$, and the subject (block) main effect, denoted by $\rho_{i(j)}$, recognizing that in this design the subject effect is nested within factor A. The model is as follows:

$$Y_{ijk} = \mu_{\ldots} + \rho_{i(j)} + \alpha_j + \beta_k + (\alpha\beta)_{jk} + \varepsilon_{(ijk)}, \quad (25)$$

where $\mu_{\ldots}$ is a constant, $\rho_{i(j)}$ are independent $N(0, \sigma_\rho^2)$, α_j are constants subject to $\sum \alpha_j = 0$, β_k are constants subject to $\sum \beta_k = 0$, $(\alpha\beta)_{jk}$ are constants subject to $\sum_j (\alpha\beta)_{jk} = 0$ for all k and $\sum_k (\alpha\beta)_{jk} = 0$ for all j, $\varepsilon_{(ijk)}$ are independent $N(0, \sigma^2)$, and $i = 1, \ldots, n$, $j = 1, \ldots, a$, $k = 1, \ldots, b$.

The response variable actually taken is the $\log_e$ of the CPU time which is the transformation usually applied to situations in which the response is the time until occurrence of some event of interest (Lawless, 1982). Adjusting the model (25) and considering that all assumptions associated with this analysis were validated, we may proceed with the analysis of variance. With the help of the package *MINITAB*[d] *for WINDOWS*[e], we have obtained the results presented in Table 2. Regarding the *p*-value column, we remark that the data show strong evidence that the problems and the algorithms have a significant effect in the response variable and that there is no interaction between these two factors for the problems tested.

Further, using the Duncan method for multiple comparison of means Neter et al. (1990), we formed the 90% confidence intervals for the estimated means presented in Figure 2. The data show some evidence that the algorithm with $\gamma = |Y|^*$ performs better than the others. The data also confirm what we suspected that problems with higher f_{ij}/c_{ij} ratio, which are closer to the *Steiner* problems, are significantly harder to tackle, no matter which algorithm is applied.

[c]Unfortunately, at the present stage of our research, we are only able to determine these values by the computation of the original UFNF problem optimal solution.

[d]Copyright © 1996, Minitab Inc.

[e]Copyright © 1981–1996, Microsoft Corp.

5 Conclusions and Final Remarks

Computational experiments concerning the use of the alternative model (LR_2') to solve the UFNF problem to optimality were presented. This model fixes the minimum number of arcs that will be present in the optimum solution of the UFNF problem. Although the solution process can be significantly accelerated by model (LR_2') when good bounds for γ are in use, it was shown that it is not effective to use poor bounds for γ. Significant improvements were reached using tight values for γ. It is remarkable that the improvements are closely related to the reduction in the gap in the first node of the branch-and-bound search tree. However, some research questions remain open. Would it be *easy* to determine a value for γ tight enough to yield further reductions in processing time[f]? Does the behavior observed for the networks tested here occur in different topologies with more and less demand points, sparseness, *etc.*? A possible extension of this work might include the investigation of these research questions.

Acknowledgments

The research of Frederico Rodrigues Borges da Cruz, Geraldo Robson Mateus, and Enrico Antônio Colosimo is partially founded by CNPq and FAPEMIG, Brazil.

References

Aneja, Y. P. (1980). An integer linear programming approach to Steiner problem in graphs, *Networks*, **10**: 167–178.

Barr, R. S., Glover, F., and Klingman, D. (1981). A new optimization method for large scale fixed charge transportation problems, *Operations Research*, **29**:448–463.

Beasley, J. E. (1984). An algorithm for the Steiner problem in graphs, *Networks*, **14**:147–159.

Cabot, A. V. and Erenguc, S. S. (1984). Some branch-and-bound procedures for fixed-cost transportation problems, *Naval Research Logistics Quarterly*, **31**:145–154.

Cruz, F. R. B., Almeida, J. A., and Mateus, G. R. (1994). Análise do problema de planejamento de redes telefônicas de alimentação, *Proceedings of the 10th Congresso Brasileiro de Automática*, Vol. I, Rio de Janeiro, pp. 572–577.

Cruz, F. R. B., MacGregor Smith, J., and Mateus, G. R. (1998). Solving to optimality the uncapacitated fixed-charge network flow problem, *Computers & Operations Research*, **25**(1):67–81.

Fisher, M. L. (1985). An application oriented guide to Lagrangean relaxation, *Interfaces*, **15**:10–21.

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

Geoffrion, A. M. (1974). Lagrangean relaxation and its uses in integer programming, *Mathematical Programming Study*, **2**:82–114.

Glover, F., Klingman, D., and Phillips, N. (1985). A new polynomially bounded shortest path algorithm, *Operations Research*, **33**(1):65–73.

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

Lawless, J. F. (1982). *Statistical Models and Methods for Lifetime Data*, John Wiley & Sons, New York, USA.

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

Magnanti, T. L., Mireault, P., and Wong, R. T. (1986). Tailoring Benders decomposition for uncapacitated network design, *Mathematical Programming Study*, **26**:112–154.

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

Mateus, G. R., Pádua, C. I. P. S., and Luna, H. P. L. (1990). Integrated network models for local access network design, *Proceedings of the International Telecommunications Symposium 1996*, Acapulco, Mexico, pp. 6–10.

Neter, J., Wasserman, W., and Kutner, M. H. (1990). *Applied Linear Statistical Models: Regression, Analysis of Variance, and Experimental Design*, Richard D. Irwin, Homewood, Illinois, 4edn.

Rothfarb, B., Frank, H., Rosembaun, D. M., and Steiglitz, K. (1970). Optimal design of offshore natural-gas pipeline systems, *Operations Research*, **18**:992–1020.

Suhl, U. (1985). Solving large scale mixed integer programs with fixed charge variables, *Mathematical Programming*, **32**:165–182.

[f] Usually in such context *easy* is used in reference to polynomially solvable problems.