A Tabu Search Approach to a Network Design Problem with Capacitated Facility Location

Sirlene R. Rolla[1], Geraldo R. Mateus[1] and Frederico R. B. Cruz[2]

[1]Departamento de Ciência da Computação, Universidade Federal de Minas Gerais
31270-901 - Belo Horizonte - MG, Brazil.
E-mails: {sirlene,mateus}@dcc.ufmg.br

[2]Departamento de Estatística, Universidade Federal de Minas Gerais
31270-901 - Belo Horizonte - MG, Brazil.
E-mail: fcruz@est.ufmg.br

1 Introduction

The capacitated facility location (CFL) problem we deals with here could be viewed as a generalized network design (ND) problem. In the practical point of view, the ND problem is a generic denomination encompassing important problems that have arisen in many different contexts such as telecommunication systems, energy distribution systems, and transportation networks, among others. The ND problems are also seen as very important in the theoretical terms because the appeal they have in the research fields of algorithms, data structures, complexity analysis and performance. In fact, they can be viewed as a generalization of several important theoretical problems that appear in the areas of network optimization, operational research, combinatorics and graph theory [1].

Classic models dealing both with ND and CFL aspects include the set covering location problem [19], the maximum covering location problem [3], and the warehouse location problem [13]. All of these models are concerned about locating facilities in a network whose topology is known beforehand. Probably motivated by the advances done in the area, it seems that researchers start to consider that disregarding topological aspects in CFL problems is too restrictive. Indeed, it was shown that the network topology plays a key hole in the final location [2]. The results reported in the literature usually do not consider fixed costs on the network arcs. In other words, usually, the models studied do not consider the ND aspect of the CFL problem. A simple formulation for a ND problem is the well known uncapacitated fixed-charge network flow (UFCNF) problem [15]. In such a problem, there is a source node producing flows to be sent to a set of demand nodes through a set of transshipment (or *Steiner*) nodes at minimum cost. The cost is composed by a fixed cost associated with the arcs that have been chosen and a variable cost dependent on the amount of flow the arcs carry. Besides the apparent simplicity of the UFCNF problem, it is a $\mathcal{NP}$-hard problem [6], and it generalizes well known $\mathcal{NP}$-hard problems. Disregarding the fixed costs, the problem is reduced to the simple source transshipment problem which is polinomially solvable by the Dijkstra's shortest path algorithm [8]. On the other hand, without variable costs, the UFCNF problem is a *Steiner* problem in graphs, well know to be $\mathcal{NP}$-hard [11].

It has been recognized that, usually, aspects of facility location and topological network design have been carelessly treated as separated issues by the papers [2]. By making certain changes in the network topology, it has been shown that the facility location could be improved [2]. In another paper, another group studied the relationship between the network configuration and the optimal solution of the p-median problem [16]. A model that could find the optimal facility location and the optimal topology has appeared recently [7], being solved by complete enumeration. The model we shall deals with here in this paper is an extension of the UFCNF problem. We think one should rather consider a *set* of

candidate source nodes each one with its own capacity and fixed charge associated. The problem shall be called as the network design with capacitated location (NDCL) problem. Concerning the NDCL model, many algorithms could be applied to it since these same algorithms were successful in solving some special case. Enumerative branch-and-bound based algorithms solved considerable large instances of the UFCNF problem [6], as well as cross decomposition algorithms [18], heuristic algorithms, such as greedy algorithms [12], tabu search algorithms [14], Lagrangean relaxation based algorithms [6, 10], and parallel algorithms [5].

In this paper, we shall present the results of a study concerning the use of the tabu search [17] in the NDCL problem, providing a comprehensive set of computational experiments. The paper is outlined as follows. In Section 2, we introduce the notation and present a mathematical programming formulation for the NDCL problem. In section 3, we describe the tabu search algorithm and some improvements were are proposing. Computational results are presented in Section 4. In Section 5, we close the paper with some open questions and future research directions.

2 Problem Formulation

The NDCL problem is defined on a graph $G = (N, A)$, where N is the set of nodes and A, the set of arcs. We now shall define the remaining of the notation used:

- S - set of candidate supply nodes;
- T - set of transhipment or *Steiner* nodes;
- D - set of demand nodes;
- s_k - capacity of supply node $k \in S$;
- d_k - demand of the demand node $k \in D$;
- c_{ij} - per unit cost on arc $(i, j) \in A$;
- f_{ij} - fixed cost for installing arc $(i, j) \in A$;
- f_k - allocation fixed cost for candidate supply node $k \in S$;
- x_{ij} - flow through arc $(i, j) \in A$;
- y_{ij} - binary variable which assumes the value 1 and 0 whether or not the arc $(i, j) \in A$ is chosen or not to support flow;
- z_k - binary variable which assumes the value 1 and 0 whether or not the node $k \in S$ is chosen or not to provide flow;
- U - a "large" number, say $U = \sum_{k \in D} d_k$, or the capacity of all arcs $(i, j) \in A$.

The mathematical programming formulation describing the NDCL problem is:

(M)**:**

$$\min \left[\sum_{(i,j) \in A} (c_{ij} x_{ij} + f_{ij} y_{ij}) + \sum_{k \in S} f_k z_k \right], \tag{1}$$

s.t.:

$$\sum_{(k,j) \in A} x_{kj} - \sum_{(i,k) \in A} x_{ik} \leq s_k z_k, \quad \forall \quad k \in S, \tag{2}$$

$$\sum_{(k,j) \in A} x_{kj} - \sum_{(i,k) \in A} x_{ik} = 0, \quad \forall \quad k \in T, \tag{3}$$

$$\sum_{(k,j) \in A} x_{kj} - \sum_{(i,k) \in A} x_{ik} = -d_k, \quad \forall \quad k \in D, \tag{4}$$

$$x_{ij} \leq U y_{ij}, \quad \forall \quad (i, j) \in A, \tag{5}$$

$$x_{ij} \geq 0, \quad \forall \quad (i,j) \in A, \tag{6}$$
$$y_{ij} \in \{0,1\}, \quad \forall \quad (i,j) \in A, \tag{7}$$
$$z_k \in \{0,1\}, \quad \forall \quad k \in S. \tag{8}$$

The objective function (1) minimizes the variable and fixed costs associated with the arcs as well as the location costs of the facilities located. Constraints (2) ensure that the flows originate only from installed facilities ($z_k = 1$) and constraints (4) that they will go to all demand nodes. The flow conservation in the *Steiner* nodes is expressed in constraints (3). Constraints (5) express the fact that the flow through an arc must be zero if this arc is not included in the design. Finally, (6)–(8) are the usual non-negativity and integrality constraints.

3 Algorithms

Concerning the NDCL problem, one could easily figure out that these are two decision variables. The z_k's are related to the facilities to be located and y_{ij}'s are related to the arcs that form the routes. A reasonable initial approach should be to consider the tabu search technique to solve the problem in S. Once the variables from S were chosen, the resulting problem in y_{ij} and x_{ij} could be solved *e.g.* by Lagrangean relaxation as it was done before [6], or else by a linear programming relaxation. A feasible solution could be derived from any of the relaxations mentioned above by merely rounding the non-integer y_{ij}'s. The algorithm we propose may be stated as follows:

Initialization Phase: An initial solution is identified for nodes in the set S which is set up by choosing those nodes with lower location costs up to that point the installed capacity is enough to supply the total demand.

Improvement Phase: Initially, one pair of variables in S, in which one of them is set to 0 and the other is set to 1, are interchanged if this change improves the current solution in respect to S, disregarding the problem feasibility. If there is no such a candidate pair, the algorithm tries to find two pairs, and so on, up to the limit of three pairs. Once a pair is found, the algorithm resolve the overall problem and get another feasible solution. Otherwise, the algorithm goes to the next step.

Tabu Phase: In this phase, using a similar idea, some movements that may even worsen the solution are done in order to move the search out of local optimal points toward another directions. Movements recently or frequently done, called tabu movements, are prohibited. The algorithm stops after a pre-specified number of iterations between the improvement and the tabu phase or else there is no possible movement at all. Otherwise, the algorithm goes back to the improvement phase.

We shall also consider a modification on the tabu phase inspired in our practical experience with the algorithm above. We have noticed that the algorithm generates to much cycling solutions. In order to avoid such cycles, we should create more tabu movements, corresponding to these cycles.

4 Computational Aspects and Results

In this paper, the AMPL [9] was used to implement all mathematical formulations which were solved by means of the CPLEX [4]. All tests were performed in a workstation Sun Ultra 1 Model 140 with 128 MB RAM and the operating system SunOS 5.5.1. We have tested problems randomly generated with 20, 50 and 100 nodes each of them with Γ incident arcs in average and respectively 5, 10 and

20 supply candidate nodes. Additionally, we consider different fixed and variable costs ratio in the arcs, randomly generated location costs between 1,000 and 10,000, and 10,000 as the total demand. The results are reported in Table 4.1. All CPU times reported are in seconds and exclude all I/O operations and consider that there was only one process running at that time.

From these results, it is remarkable that the *gaps* are considerably small, mainly in the version 2, attesting the good quality of our heuristic solution. In the 100 node series with fixed and variable cost ratio equal to 10, it is noticeable the success of the heuristic approach over the exact approach. In spite of their fastness, the heuristic solutions proved to be very good approximations for the cases tested.

Table 4.1 : Computational Results

			Exact		Tabu Algorithm 1							Tabu Algorithm 2						
f_{ij}/c_{ij}	P	Γ	$\|S\|$	CPU(s)	$\|S\|$	CPU(s)	GAP†	TP‡	IP§	nB††	nT‡‡	$\|S\|$	CPU(s)	GAP†	TP‡	IP§	nB††	nT‡‡
1	20	3	3	0.03	3	0.07	0.00	2	3	5	6	3	0.05	0.00	2	0	3	3
	20	4	4	0.04	3	0.19	0.44	3	5	7	9	3	0.08	0.44	2	1	4	4
	50	3	7	0.18	7	3.60	32.00	10	50	26	61	7	0.46	12.30	3	4	8	8
	50	4	7	0.14	7	2.60	1.70	10	50	4	61	7	0.31	1.40	2	4	6	7
	100	4	14	7.40	15	10.00	21.00	7	93	44	104	15	4.00	2.09	7	25	36	36
	100	2	13	3.00	15	12.00	11.00	7	93	49	104	15	4.10	1.46	8	23	9	35
0.1	20	3	3	0.03	3	0.11	0.00	2	3	5	6	3	0.10	0.00	2	0	3	3
	20	4	4	0.04	3	0.20	0.46	3	5	7	9	3	0.11	0.46	2	1	4	4
	50	3	7	0.15	7	3.60	31.00	10	50	26	61	7	0.49	12.30	3	4	8	8
	50	4	7	0.12	7	2.50	1.70	10	50	4	61	7	0.29	1.41	2	4	6	7
	100	4	14	1.20	15	10.00	21.00	7	93	44	104	15	3.50	2.06	7	25	36	36
	100	2	13	0.67	15	12.00	11.00	7	93	50	104	15	4.00	1.45	8	23	9	35
10.0	20	3	3	0.03	3	0.09	0.00	2	3	5	6	3	0.05	0.00	2	0	3	3
	20	4	4	0.04	3	0.22	0.31	3	5	7	9	3	0.08	0.31	2	1	4	4
	50	3	7	0.36	7	3.60	31.00	10	50	26	61	7	0.51	12.30	3	4	8	8
	50	4	8	0.15	7	2.50	1.90	10	50	4	61	7	0.31	1.54	2	4	6	7
	100	4	14	900*	15	10.00	21.00	7	93	44	104	15	4.00	2.42	7	25	25	36
	100	2	13	900*	15	12.00	11.00	7	93	49	104	15	4.10	1.60	8	23	9	35

*time overflow; ‡number of tabu phase iterations; §number of improvement phase iterations;
††iteration in which the best solution was found; ‡‡overall number of iterations;
†GAP = 100% $\times$ (Optimal Solution $-$ Heuristic Solution) $\div$ Optimal Solution

5 Final Remarks

We have introduced a new model that integrates topological network design aspects and capacitated facility location and stressed the importance such a model might have in practice. We have proposed a tabu search based algorithm to tackle the problem as well some improvements that were based on the problem structure. Preliminary results seem to indicate that the approach is quite satisfactory. In special, we mention that definitely heuristic algorithms must be applied to solve large instances. The time to solve them exactly was too large. A topic for future research includes more computational experiments in other types of testing problems, as well as a "fine tuning" of the tabu search based algorithm that would probably reduce even more the tabu movements. We have seen that a simple procedure to avoid cycling solutions leaded to improvements, as well as reduced the total computational time and did not jeopardize the solution quality.

Bibliography

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

[2] O. Berman, D. I. Ingco, and A. R. Odoni. Improving the location of minimum facilities through network modification. *Annals of Operations Research*, 40:1–16, 1992.

[3] R. Church and C. ReVelle. The maximal covering location problem. *Regional Science Association Papers*, 32:101–118, 1974.

[4] CPLEX Optimization, Inc. *Using the CPLEX™ Callable Library and CPLEX™ Mixed Integer Library*. CPLEX Optimization, Inc., Incline Village, NV, 1993.

[5] F. R. B. Cruz, G. R. Mateus, and C. D. Randazzo. Randomized parallel branch-and-bound algorithms applied to a networkdesign problem. In *Proceedings of the VIII Latin American Congress on Automatic Control - VIII CLCA*, pages ?–?, Viña del Mar, Chile, 1998. ACCA.

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

[7] M. S. Daskin and S. Melkote. An integrated model for facility location and network design. Working paper, The Transportation Center, Northwestern University, Evanston, IL, USA, 1995.

[8] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerical Mathematics*, 1:269–271, 1959.

[9] R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. The Scientific Press, 1993.

[10] K. Holmberg and J. Hellstrand. Solving the uncapacitated network design problem by a Lagrangean heuristic and branch-and-bound. *Operations Research*, 46(2):247–259, 1998.

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

[12] G. R. Mateus and C. T. Bornstein. Dominance criteria for the capacitated warehouse location problem. *Journal of the Operational Research Society*, 42:145–149, 1991.

[13] G. R. Mateus and J. C. P. Carvalho. O problema de localização não capacitado: Modelos e algoritmos. *Investigación Operativa*, 2:297–317, 1992.

[14] F. M. Miller. Tabu search in the solution of zero-one problems. Master's thesis, Departamento de Engenharia de Sistemas, UNICAMP, Campinas, SP, 1990. (in Portuguese).

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

[16] D. Peeters and I. Thomas. The effect of spatial structure on location-allocation results. In *ISOLDE VI Conference*, Lesvos, Greece, 1993.

[17] C. R. Reeves. *Modern Heuristic Techniques for Combinatorial Problems*. John Wiley & Sons, Inc., New York - Toronto, 1993.

[18] T. J. Van Roy. A cross decomposition algorithm for the capacitated facility location. *Operations Research*, 34:145–163, 1986.

[19] C. Toregas, R. Swain, C. ReVelle, and L. Bergmann. The location of emergency service facilities. *Operations Research*, 19:1363–1373, 1971.