Proceedings of the VIII Latin-Iberian-American Congress on Operations Research and System Engineering (VIII CLAIO)
Rio de Janeiro, pp. 519–524, August, 1996

A REDUCTION TEST BASED ON LAGRANGEAN RELAXATION APPLIED TO UNCAPACITATED FIXED-CHARGE NETWORK FLOW PROBLEMS

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

The Lagrangean Relaxation is a well-know technique that has been applied successfully to many combinatorial optimization problems which is normally used to provide the bounds that are needed by the branch-and-bound algorithms. In this paper, we present another use for the Lagrangean Relaxation developing a reduction test for the uncapacitated fixed-charge network flow (UFNF) problem. Computational results are provided and they show that the technique is an effective way of solving the problem.

Keywords: Graphs and Network Flows, Combinatorial Optimization, Mixed-integer Programming.

1 Introduction

The uncapacitated fixed-charge network flow (UFNF) problem represents an important class of mixed-integer programming problems. The problems are defined 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 supply nodes to a collection of demand nodes, possibly using intermediate *Steiner* or transshipment nodes.

This is clearly an $\mathcal{NP}$-hard optimization problem since it generalizes the Steiner problem in graphs, $\mathcal{NP}$-hard [9], among others. This generic model has applications for problems of

[†]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.

distribution, transportation and communication. It is also useful for certain routing problems where the network is already in existence. Besides being an important model by itself, several special cases of the UFNF problem are of substantial interest. A simple way to obtain special cases is to restrict the network structure, *e.g.* as in the transportation problem.

Some experimental work concerning exact and approximate solutions for the UFNF problems and their special cases have been done previously. The special case without *Steiner* nodes was treated by [10] and by [12]. 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 was developed. In [2, 4], some special cases were solved using a branch-and-bound algorithm with fractional cutting-planes. Previously, we have studied *ADD* and *DROP* heuristic approaches [13] as well as simplified branch-and-bound algorithms [6] to solve the general UFNF problems and these algorithms have performed well in practice.

A natural generalization of the UFNF problem is the capacitated fixed-charge network flow (CFNF) problem. Although the CFNF problem is very difficult to be solved, it can be represented by a surprisingly compact mathematical programming formulation [14]:

(M)**:**

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

s.t.:

$$\sum_{j\in\delta^+(i)} x_{ij} - \sum_{j\in\delta^-(i)} x_{ji} = b_i, \qquad \forall i \in N, \tag{2}$$

$$x_{ij} \le u_{ij}y_{ij}, \qquad \forall (i,j) \in A, \tag{3}$$

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

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

where $\mathcal{D} = (N, A)$ is a digraph, N is the set of nodes, A is the set of arcs, $\delta^+(i) = \{j|(i,j) \in A\}$, $\delta^-(i) = \{j|(j,i) \in A\}$, $b_i > 0$ (< 0) is the supply (demand) at node i, f_{ij} is the fixed cost of having flow on arc (i,j), c_{ij} is the variable cost per unit of flow on arc (i,j), and u_{ij} is the capacity of arc (i,j). It is noticeable that the only difference between the CFNF problem and the linear minimum-cost network flow (MCNF) problem is that, in the former, if the flow is positive, *i.e.* $x_{ij} > 0$, then its cost is $c_{ij}x_{ij} + f_{ij}$. The capacity constraints (3) ensure that characteristic. That simple difference transforms the polynomially solvable MCNF problem into the $\mathcal{NP}$-hard CFNF problem.

As pointed out by [14], a necessary condition for feasibility, assumed throughout this work, is that $\sum_{i\in N} b_i = 0$. Additionally, it is assumed that all problems are single-supply-node. The fixed cost f_{ij} must be non-negative for all arcs (i,j), since otherwise one could set y_{ij} to 1 and eliminate it from the problem. On the other hand, the variable cost c_{ij} is unrestricted. However, to ensure that the objective function is bounded from below, it is assumed that there are no negative-cost directed cycles with respect to c_{ij}. An important simplification we are considering here concerns the capacity constraints. If the u_{ij} is sufficiently large, say $u_{ij} \ge \frac{1}{2}\sum_{i\in N} |b_i|$, the capacity constraints only force the inclusion of the fixed cost in the objective function when the flow is positive. The problems under such assumption are called *uncapacitated* and these are the only problems treated in this work.

During this paper, we shall consider an alternative formulation of model (M), more convenient for our purposes in developing the reduction test. Let us define $K_0 \subseteq A$, the set of arcs that have been positively identified as not-present in an optimal solution by some reduction method, $K_1 \subseteq A$, the set of arcs that have been positively identified as present in an optimal solution, and $K = A \setminus K_0 \setminus K_1$, the set of free or undefined arcs. So, the model (M) can be alternatively represented by the following formulation:

(M')**:**

$$\min \sum_{(i,j)\in A} (c_{ij}x_{ij} + f_{ij}y_{ij}), \tag{6}$$

s.t.:

$$\sum_{j\in\delta^+(i)} x_{ij} - \sum_{j\in\delta^-(i)} x_{ji} = b_i, \qquad \forall i \in N, \tag{7}$$

$$x_{ij} \le u_{ij}y_{ij}, \qquad \forall (i,j) \in K, \tag{8}$$

$$x_{ij} \ge 0, \qquad \forall (i,j) \in K, \tag{9}$$

$$y_{ij} \in \{0,1\}, \qquad \forall (i,j) \in K, \tag{10}$$

$$x_{ij} \le u_{ij}, \qquad \forall (i,j) \in K_1, \tag{11}$$

$$x_{ij} \ge 0, \qquad \forall (i,j) \in K_1, \tag{12}$$

$$y_{ij} = 1, \qquad \forall (i,j) \in K_1, \tag{13}$$

$$x_{ij} = 0, \qquad \forall (i,j) \in K_0, \tag{14}$$

$$y_{ij} = 0, \qquad \forall (i,j) \in K_0. \tag{15}$$

In a previous work [6], we developed a branch-and-bound algorithm to solve the UFNF problem. This is a well-known procedure to solve $\mathcal{NP}$-hard problems, computationally inefficient because of its exponential worst-case time complexity, $O(2^{|K|})$, but acceptable in practice for small sized problem instances ($K < 124$) [6]. The Lagrangean relaxation technique was applied to provide the lower bounds. Dropping the constraints (8) by means of the dual variables $w_{ij} \ge 0$, the Lagrangean dual problem $L(\mathbf{w}, K, K_0, K_1)$ results. In this work, we plan to enlarge the size of manageable instances proposing a reduction test based on the Lagrangean relaxation for the UFNF problem.

The remaining of this work will be as follows. In Section 2, we shall present our reduction algorithm. The algorithm was implemented and our experimental results are reported in Section 3. Section 4 closes the paper with final remarks, open questions and the presentation of some possible extensions.

2 Reduction Technique

The new lower bound that would result from forcing the arcs *in* or *out of* the solution can be easily estimated from the Lagrangean relaxation. If the lower bound resulting from imposing some condition to the Lagrangean relaxation is above the best upper bound, then the condition in consideration cannot be satisfied in the optimum. This idea is inspired by the reduction procedures developed in [5] to solve the p-median problem, with very good results in practice. In [5], some terms of the corresponding Lagrangean function have been used to estimate the increment in the lower bound under the imposed condition. We propose the following reduction procedure that uses estimated lower bounds computed by means of a complete resolution of the Lagrangean dual problem, $L(\mathbf{w}, K, K_0, K_1)$, but without subgradient optimizations. The reduction algorithm is depicted in Figure 1.

The reduction procedure stops after the examination of each arc exactly once, $O(|A|)$. Some variations are immediate, *e.g.* passing through each arc twice, etc. Each iteration involves at most two lower bound calculations which are $O(|N||A|)$ each. Recall that no subgradient optimization is performed here. Additionally, at most four set insertions and deletions are involved which are $O(|A|)$, but the computation of function $L(\mathbf{w}, K, K_0, K_1)$ dominates. Then, the procedure will run with worst-case time complexity $O(|N||A|^2)$.

```
procedure Reduce(M')
            /* initialize set of arcs recently fixed */
    F ← ∅
            /* proceed with reduction */
    for all (i, j) ∈ A do
        if (i, j) ∈ K then
            K ← K \ (i, j); K_1 ← K_1 ∪ (i, j)
            if L(w, K, K_0, K_1) > U_BEST then
                K_1 ← K_1 \ (i, j); K_0 ← K_0 ∪ (i, j)
                F ← F ∪ (i, j)
            else
                K_1 ← K_1 \ (i, j); K_0 ← K_0 ∪ (i, j)
                if L(w, K, K_0, K_1) > U_BEST then
                    K_0 ← K_0 \ (i, j); K_1 ← K_1 ∪ (i, j)
                    F ← F ∪ (i, j)
                else
                    K_0 ← K_0 \ (i, j); K ← K ∪ (i, j)
                end if
            end if
        end if
    end for
end procedure
```

Figure 1: Reduction Algorithm

3 Experimental Results

A preliminary version of the algorithm coded in the *C* programming language was developed and is available upon request. All tests presented were performed using a DECstation 3100 running the operating system ULTRIX V4.2A (Rev. 47). All test problems came from Euclidean graphs randomly generated using a procedure similar to one presented in [1] that has been extensively applied for creating testing instances [15, 3].

Table 1 presents the results of all computational experiments. The results presented for the first node of the branch-and-bound search tree are the best upper bound, the gap, and the CPU time spent in seconds. The total number of branch-and-bound nodes explored and the CPU time in seconds spent after the first node are presented using and not using the reduction. All CPU times reported are the clock time excluding all I/O operations and considering only a single process running on the machine. For each graph, three instances with different $\frac{f_{ij}}{c_{ij}}$ ratios were considered. The problems with ratio $1:10$ ($f_{ij} = \Omega_{ij}$ and $c_{ij} = 10\Omega_{ij}$) form a class approaching the MCNF problem which is polynomially solvable. On the other hand, the problems with ratio $10:1$ ($f_{ij} = 10\Omega_{ij}$ and $c_{ij} = \Omega_{ij}$) form a class of almost *Steiner* problems which is $\mathcal{NP}$-hard. However, both cases are still $\mathcal{NP}$-hard.

It may be seen the remarkable effect on processing time caused by the reduction algorithm. In sparse instances, the reduction algorithm kept the number of explored branch-and-bound node surprisingly low. The branch-and-bound algorithm using the reduction technique was able to solve quickly dense networks if the number of demand nodes was low and the problems were closer to the MCNF problems. The problems closer to *Steiner* problems are really harder.

Table 1: Effect of the Reduction Algorithm

								Branch-and-Bound			
					First Node			No Reduction		Reduction	
$\|N\|$	$\|A\|$	$\|D\|$	$\frac{f_{ij}}{\Omega_{ij}}$	$\frac{c_{ij}}{\Omega_{ij}}$	SOL[1]	GAP[2]	CPU	Nodes	CPU	Nodes	CPU
16	30	4	1	10	1.0000	1.50	0.20	29	2.40	3	0.03
			1	1	1.0000	12.00	0.20	29	2.40	3	0.03
			10	1	1.0000	44.00	0.22	29	2.50	3	0.03
		8	1	10	1.0000	2.10	0.21	43	3.70	3	0.03
			1	1	1.0000	19.00	0.21	43	3.80	3	0.03
			10	1	1.0000	94.00	0.22	43	4.00	3	0.03
	60	4	1	10	1.0000	3.40	0.52	189	45.00	5	1.28
			1	1	1.0000	24.00	0.53	245	61.00	35	14.90
			10	1	1.0000	68.00	0.56	375	110.00	45	20.40
32	62	4	1	10	1.0000	4.20	0.64	1,923	510.00	3	0.14
			1	1	1.0000	34.00	0.64	1,923	510.00	3	0.14
			10	1	1.0000	120.00	0.68	1,923	540.00	3	0.14
		8	1	10	1.0000	3.70	0.65	3,635	960.00	3	0.15
			1	1	1.0000	33.00	0.63	3,635	960.00	3	0.15
			10	1	1.0000	170.00	0.67	3,635	1,000.00	3	0.15
		16	1	10	1.0000	2.70	0.68	2,071	610.00	3	0.16
			1	1	1.0000	26.00	0.67	2,071	610.00	3	0.16
			10	1	1.0000	170.00	0.71	2,071	640.00	3	0.16
		31	1	10	1.0000	2.20	0.74	63	21.00	1	0.18
			1	1	1.0000	22.00	0.74	63	21.00	1	0.18
			10	1	1.0000	170.00	0.78	63	22.00	1	0.18
	124	4	1	10	1.0000	5.40	2.10	379	330.00	5	5.09
			1	1	1.0024	44.00	2.00	4,663	4,200.00	195	270.00
			10	1	1.0011	120.00	2.20	16,817	17,000.00	1,283	1,720.00
		8	1	10	1.0000	4.80	2.00	2,285	1,990.00	7	4.93
			1	1	1.0001	38.00	2.00	11,999	11,200.00	1,543	2,050.00
		16	1	10	1.0010	6.60	2.00	**	**	27	47.00
			1	1	1.0070	34.00	2.00	**	**	4,951	6,650.00
	248	4	1	10	1.0000	6.60	6.70	**	**	31	123.00

** Not available (time overflow).

4 Final Remarks

The UFNF problem is a challenging intractable problem ($\mathcal{NP}$-hard) with many applications for the real word. The generic model also encompasses many other special cases with remarkable importance in practice. A mathematical programming formulation for the problem was presented. A new Lagrangean relaxation based reduction test was introduced. The algorithm was implemented performing very well in practice as the computational experiments have shown.

An interesting conclusion after reviewing all computational experiments is that in some cases it may be valuable to increase the complexity of node explorations in branch-and-bound algorithms. The reduction algorithm increased the computational complexity of each node exploration but it yielded an overall gain because it made possible an early identification of dead ends in the branch-and-bound tree, saving unnecessary explorations.

Some open questions remains such as whether or not it would be possible to improve even more the branch-and-bound algorithm using Lagrangean relaxations that provide tight lower bound. Future work may include investigation of this question. It may also include the development of additional reduction tests that eliminate arcs and/or nodes from the original problem

[1] $\text{SOL} = \frac{\text{best upper bound}}{\text{optimal solution}}$

[2] $\text{GAP} = \frac{\text{(best upper bound)} - \text{(best lower bound)}}{\text{best lower bound}} * 100\%$

in a preprocessing stage, similar to those tests presented in [11] and [7] for the *Steiner* problem in graphs. It is also of interest to investigate how the techniques proposed here can be adapted for solving some special cases of the UFNF problem (*e.g.* the fixed-charge transportation problem [2] and the uncapacitated facility location problem [8]) taking advantage of their particular network structure.

References

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

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

[3] J.E. Beasley. An SST-based algorithm for the Steiner problem in graphs. *Networks*, 19:1–16, 1989.

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

[5] N. Christofides and J.E. Beasley. A tree search algorithm for the p-median problem. *European Journal of Operational Research*, 10:196–204, 1982.

[6] 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. (submitted to *Computers & Operations Research*).

[7] C.W. Duin and A. Volgenant. Reduction tests for the Steiner problem in graphs. *Networks*, 19:549–567, 1989.

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

[9] 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.

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

[11] N. Maculan, P. Souza, and A.C. Vejar. An approach for the Steiner problem in directed graphs. *Annals of Operations Research*, 33:471–480, 1991.

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

[13] 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.

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

[15] R.T. Wong. A dual ascent algorithm for the Steiner problem in directed graphs. *Mathematical Programming*, 28:271–287, 1984.