European Journal of Operational Research 200 (2010) 28–35

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

A branch and cut algorithm for the hierarchical network design problem

Carlos Obreque [a,b], Macarena Donoso [a,c], Gabriel Gutiérrez [a,d], Vladimir Marianov [e,*]

[a] Graduate Program, Department of Systems Engineering, Pontificia Universidad Católica de Chile, Santiago, Chile
[b] Department of Industrial Engineering, Universidad del Bío-Bío, Concepción, Chile
[c] Department of Industrial Engineering, Universidad Diego Portales, Santiago, Chile
[d] Department of Industrial Engineering, Universidad Católica de la Santísima Concepción, Concepción, Chile
[e] Department of Electrical Engineering, Pontificia Universidad Católica de Chile, Santiago, Chile

ARTICLE INFO

Article history:
Received 18 July 2007
Accepted 12 December 2008
Available online 25 December 2008

Keywords:
Extensive facility location
Hierarchical networks
Branch and cut
Cutting planes

ABSTRACT

The Hierarchical Network Design Problem consists of locating a minimum cost bi-level network on a graph. The higher level sub-network is a path visiting two or more nodes. The lower level sub-network is a forest connecting the remaining nodes to the path. We optimally solve the problem using an *ad hoc* branch and cut procedure. Relaxed versions of a base model are solved using an optimization package and, if binary variables have fractional values or if some of the relaxed constraints are violated in the solution, cutting planes are added. Once no more cuts can be added, branch and bound is used. The method for finding valid cutting planes is presented. Finally, we use different available test instances to compare the procedure with the best known published optimal procedure, with good results. In none of the instances we needed to apply branch and bound, but only the cutting planes.

1. Introduction

The Hierarchical Network Design Problem (HNDP) consists of finding the minimum cost spanning bi-level network, in which the highest (or primary) level sub-network is a path connecting an origin and a destination node, and possibly visiting other nodes. All remaining nodes must be connected to this path through the lowest (or secondary) level sub-network, which is a forest. Usually, the highest level network is more expensive.

This type of network represents highways and secondary roads in a transportation setting; optical fiber and coaxial or twisted pair cables in telecommunications networks, or main pipes and secondary pipes in hydraulic networks.

The HNDP is NP-Hard, (Balakrishnan et al., 1994b; Obreque and Marianov, 2007), so efficient procedures are needed for finding its solution. Current et al. (1986) formulate the HNDP as an integer programming model, in which the number of constraints that avoid the formation of sub-tours is exponential in the number of nodes of the network. They propose an optimal iterative procedure, as well as a heuristic method, based on computation of the k shortest paths between origin and destination of the main path. Duin and Volgenant (1989), Koch and Martin (1998) and Chopra and Tsai (2002) reduce the size of the problem by identifying arcs that must be optimal and eliminating arcs and nodes that can not possibly belong to the solution. Obreque and Marianov (2007) also propose an efficient method to reduce the problem, and solve the reduced problem to optimality using a multicommodity flow approach. Duin and Volgenant (1990) provide a method for finding good upper and lower bounds for the solution; Pirkul et al. (1991) propose a heuristic based on a Lagrangean relaxation. Sancho (1995) finds a sub-optimal solution using a heuristic based on dynamic programming.

This problem can also be seen as a particular case of more general problems: Balakrishnan et al. (1994a,b), Chopra and Tsai (2002), Mirchandani (1996),Duin and Volgenant (1991), and Gouveia and Telhada (2001) address a bi-level network design in which the primary level is a tree instead of a path. Sancho (1996) solves a problem in which there are p primary paths, and Current (1988) and Current and Pirkul (1991) consider minimizing an additional objective consisting of fixed costs of transshipment facilities on the nodes of the main path.

An interesting property of the HNDP was found by Duin and Volgenant (1989). This property can be easily applied no matter what solving procedure is being used. They show that for the HNDP, all arcs in the secondary forest must belong to a minimum spanning tree (MST) of the original problem, computed using secondary arc costs. Thus, once an MST is found, all secondary arcs not belonging to the MST can be eliminated, and the number of variables reduced. They later find optimal or near-optimal solutions using a Lagrangean relaxation technique. The same property is shown and used by Balakrishnan et al. (1994a). A multicommodity flow model and a dual-based algorithm are used to solve the

* Corresponding author. Tel.: +56 2 3544974; fax: +56 2 5522563.
E-mail addresses: cobreque@ubiobio.cl (C. Obreque), macarena.donoso@udp.cl (M. Donoso), ggutierrez@ucsc.cl (G. Gutiérrez), marianov@ing.puc.cl (V. Marianov).

0377-2217/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2008.12.022

problem, not necessarily to optimality. The same property is again used in Mirchandani (1996), and in Obreque and Marianov (2007), where the problem is reduced (considering only the case in which primary and secondary costs are proportional) and later solved with conventional techniques, and Obreque et al. (2008), who solve the HNDP with unknown origin and destination nodes to optimality, using a two-stage procedure.

The procedure we present here is complementary to that in Obreque and Marianov (2007) and Obreque et al. (2008). We propose a Branch and Cut (B&C) method to optimally solve the HNDP, a complementary procedure to that in Obreque and Marianov (2007) and Obreque et al. (2008). B&C is a combination of the Cutting Planes method (Dantzig et al., 1954) with the Branch and Bound method. The cutting planes method solves a sequence of continuous relaxations of the integer programming formulation, adding, at each iteration, cutting planes (or cuts) that exclude the current non-integer optimal solution without excluding any feasible integer solutions. We also use a property of the HNDP found by Duin and Volgenant (1989), that allows precluding secondary arcs that do not belong to a previously found MST.

The paper is organized as follows. In Section 2 we present the integer programming formulation and briefly discuss the structure and properties of the HNDP. Section 3 is devoted to the description of the valid cuts. In Section 4, we describe the full procedure. In Section 5, we propose a separation algorithm and in Section 6, we show the results of the numerical experiments. Finally, we draw some conclusions.

2. Integer programming formulation of the HNDP

Let $G = (N,A)$ be a graph consisting of a non-empty set of nodes N, and a set A of directed arcs connecting the nodes in N. An arc (i,j) starts at node i and ends at node j. $|N|$ is the cardinality of set N, i.e. the number of nodes in N. Similarly, $|A|$ is the number of arcs in A. All of the nodes of the network to be built must coincide with nodes of the graph, and its arcs must be built along the arcs of the graph. The network arcs can have two possible costs: $c_{ij} = c_{ji}$ is the cost of a primary network arc built along the arc (i,j) of the graph, while $d_{ij} = d_{ji}$ is the cost of a secondary arc of the resulting network built along the arc (i,j) of the graph. We assume that $c_{ij} \geqslant d_{ij}$ for all $(i,j) \in A$, since the primary arcs are more expensive to build.

The HNDP consists in finding the least cost bi-level spanning network, in which the primary network is a path starting at an origin node O and reaching a destination node D, while possibly visiting other nodes of the network. These arcs have costs c_{ij}. The secondary network consists of a forest (one or more trees) connecting all remaining nodes to the primary path. All arcs composing this forest have costs d_{ij}.

We use a formulation similar to that in Current et al. (1986), except for the fact that secondary arcs are defined only over arcs of a previously found minimum spanning tree (MST).

Problem P:

$$\text{Minimize} \quad \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{(i,j)\in T} d_{ij}y_{ij} \tag{1}$$

$$\text{s.t.} \quad \sum_{j\in N:(O,j)\in A} x_{Oj} = 1, \tag{2}$$

$$\sum_{i\in N:(i,D)\in A} x_{iD} = 1, \tag{3}$$

$$\sum_{i\in N\setminus\{D\}:(i,j)\in A} x_{ij} = \sum_{h\in N\setminus\{O\}:(j,h)\in A} x_{jh} \quad \forall j \in N,\ j\neq O, D \tag{4}$$

$$\sum_{i\in N:(i,j)\in A} x_{ij} + \sum_{i\in N:(i,j)\in T} y_{ij} = 1 \quad \forall j \in N,\ j\neq O, \tag{5}$$

$$\sum_{i\in S, j\in S:(i,j)\in A} x_{ij} + \sum_{i\in S, j\in S:(i,j)\in T} y_{ij} \leqslant |S| - 1 \quad \forall S \subseteq N \setminus \{O\} \text{ such that } |S| \geqslant 2, \tag{6}$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A, \tag{7}$$

$$y_{ij} \in \{0,1\} \quad \forall (i,j) \in T, \tag{8}$$

where c_{ij} is the cost of a primary arc connecting node i to node j; d_{ij} is the cost of a secondary arc connecting node i to node j;

$$x_{ij} = \begin{cases} 1 & \text{if a primary arc connects node } i \text{ to node } j \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ij} = \begin{cases} 1 & \text{if a secondary arc connects node } i \text{ to node } j \\ 0 & \text{otherwise} \end{cases}$$

O is the origin node; D is the destination node; N is the set of nodes; A is the set of arcs; T is the set of the arcs in the MST. $T \subseteq A$. S is a non-empty subset of $N \setminus \{O\}$.

Constraints (2)–(4) are the shortest route constraints that force a main path to be built. Constraints (5) require all nodes, except for the origin node, to be reached by a primary or a secondary arc. There is no need to write one of these constraints for the destination node, since constraint (3) takes care of that. Constraints (6) are the sub-tour eliminating constraints. Constraints (7) and (8) require all variables to be binary.

Note that variables y_{ij} need to be defined only for arcs of a previously found MST. Besides reducing the number of variables, this fact implies that secondary arcs alone cannot form sub-tours, because the MST, by definition, does not contain any tours. Thus, there are no possible tours with only secondary arcs in it. Any tour must have some primary arc. Note also that constraint (5) forces all nodes, except for the origin, to be the end node of a directed arc.

Since arcs are directed, we need to preclude sub-tours formed by the two arcs going in opposite directions between nodes i and j, so we add the following constraint:

$$x_{ij} + x_{ji} + y_{ij} + y_{ji} \leqslant 1 \quad \forall (i,j) \in T, i < j. \tag{9}$$

There are exactly $|N| - 1$ of these constraints. These constraints are a subset of constraints (6). We explicitly use them because they not only exclude two-arc sub-tours, but they also act as an upper bound on the values of all the variables corresponding to the arc (i,j).

This formulation is our base model. The procedure requires that constraints (7) and (8) are relaxed. Also, it is clear that, for large instances, it is impossible to include the full set of constraints (6) in the formulation, since they are exponential in number: as many as the number of subsets of the set of nodes. Consequently, we also relax these constraints, and later use them as cutting planes.

3. Additional cutting planes

Adequate cutting planes need to be found, that help excluding infeasible or fractional solutions. These cutting planes are added as needed at each step of the procedure. We now formulate a new cut and show that is equivalent to (6).

Theorem 1. *The following cut is equivalent to (6):*

$$\sum_{i\in S^c, j\in S:(i,j)\in A} x_{ij} + \sum_{i\in S^c, j\in S:(i,j)\in T} y_{ij} \geqslant 1 \quad \forall S \subseteq N \setminus \{O\}. \tag{10}$$

Proof. Constraints (5) must hold for all nodes except the node O. Then, for each subset S of N not containing the origin node, it must hold that:

$$\sum_{i\in N:(i,j)\in A} x_{ij} + \sum_{i\in N:(i,j)\in T} y_{ij} = 1 \quad \forall j \in S, \text{with } S \subseteq N \setminus \{O\}. \tag{11}$$

Since $N = S \cup S^C$, where $S^C = N \setminus S$, we can rewrite (11) as