On Tree-Star Network Design

Arnaud Knippel, *arnaud.knippel@insa-rouen.fr*
Viet Hung Nguyen, *hung.nguyen@lip6.fr*

Keywords: topological network design, integer programming.

1. Introduction

We consider a set of nodes that we have to link in a telecommunication network. In the design of this network, we have to distinguish between the core (or backbone network) and the extreme part. The core is a subset of nodes (called concentrators), and the other nodes (client nodes) are linked directly to concentrators. The topology of the core may vary depending on some constraints : connectivity, survivability... Typically, we ask the core at least to be connected, and if no other constraints are taken into account, the corresponding subgraph will be a tree. Some work has already been published on this topic [4], [5],[6], [7]. We distinguish two cases in these works:

- The node set V is partitioned in two subsets: the potential concentrators and the clinent nodes. All concentrators do not need to be selected, but every client has to be linked to one concentrator.
- All concentrators have to be linked in a Steiner tree (possibly includeing client nodes), or in a cycle.)

For more details, we refer to [3].

We study here a case where all nodes have to be connected by a tree, and the cost of the edges depend on these edges being pending or not. A pending edge ij means that either node i is a client node assigned to a concentrator located on node j, or node j is assigned to a concentrator located on node i. Such an edge has a cost c^1_{ij} or c^1_{ji} (cost can be symmetrical or not). A non-pending edge ij corresponds to two concentrators located on nodes i and j, linked in the core tree network. Its cost is c^2_{ij} (this cost is symmetrical).

We present more precisely this problem in section 2. A mixed integer formulation is proposed in section 3. We will also discuss numerical results during the presentation.

2. Notations and complexity results

Given a graph $G = (V, E)$ with $n = |V|$ and $m = |E|$. Each pair of nodes $i, j \in V$ is associated to three following costs

- the *edge cost* c^2_{ij}. If $ij \notin E$ then $c^2_{ij} = +\infty$;
- the *assignment cost* of i to j, c^1_{ij}. If such an assignment is impossible then $c^1_{ij} = +\infty$;
- the *assignment cost* of j to i, c^1_{ji}. If such an assignment is impossible then $c^1_{ji} = +\infty$;

By these definitions of costs, we can consider that G is a complete graph of n nodes. Given a spanning tree T of G and a set $C_T \subseteq L_T$ where L_T is the set of the leaves of T. The set C_T is called the *client set* and it's complementary $V \setminus C_T$ is called *core set*. Let us define the cost of the edges in T associated with the set C_T as follows:

- Each edge ji in T between a leaf $j \in C_T$ and his father i is evaluated by the c^1_{ij} the assignment cost of j to i.
- All the other edges in T are evaluated by the edge cost.

The cost of T is the sum of the cost of it's edges. We want to find the minimum cost spanning tree of G evaluated as above. We call this problem *the Tree-Star Network Design (TSND)* problem. When $c^1_{ij} = c^1_{ji}$ for all $i, j \in V$ the problem is called *the Symmetric Tree-Star Network Design (STSND)* problem. Otherwise, the problem is called the Asymmetric Tree-Star Network Design (ATSND) problem.

Remark 1. *A solution of the TSND problem is characterized by a couple (T, C_T) where T is a spanning tree and C_T is the client set whose nodes are among the leaves of T.*

Remark 2. *In the tree-star network design problem, for all $i, j \in V$, if the assignment of i to j is possible, i.e. c^1_{ij} is finite, we can suppose without loss of generality that $c^1_{ij} < c^2_{ij}$.*

Proof : Suppose that for some $i, j \in V$, we have c^1_{ij} finite and $c^1_{ij} \geq c^2_{ij}$. We have two possibles cases:

- There is no optimal spanning tree T with client set C_T such that $j \in C_T$ and the edge $ij \in T$. In this case, we can set $c^1_{ij} = +\infty$ which does not change anything on the optimal solutions of the problem. Therefore, we can obtain an equivalent problem without the assignment of i to j.
- Otherwise, i.e. there is an optimal spanning tree T with client set C_T such that $j \in C_T$ and the edge $ij \in T$. Thus the edge ij in T is evaluated with the cost c^1_{ij}. Let us consider the solution with the same tree T but with client set $C'_T = C_T \setminus \{j\}$. Since $c^1_{ij} \geq c^2_{ij}$, we obtain a new solution (T, C'_T) without the assignment of i to j which is as good as the solution (T, C_T).

□

Despite of it's neighbour aspect to the minimum spanning tree problem, we will show below that the TSND problem is NP-hard.

Theorem 1. *The ATSND problem is NP-hard.*

Proof : We will reduce the Minimum Steiner Tree (MST) problem to TSND problem. Given any instance I^1 of MST consisting of a weighted graph $G^1 = (V^1, E^1)$ and a set S of terminal nodes. Each edge $e \in E^1$ is associated to the weight c_e. The MST problem is to find a minimum weighted tree which covers all nodes of S. We transform I^1 to an instance I of TSND as follows. The graph G in TSND is the complete graph whose node set is V^1.

- For all pairs $s_1, s_2 \in S$, $c^1_{s_1 s_2} = c^1_{s_2 s_1} = +\infty$, if $s_1 s_2 \in E^1$ then $c^2_{s_1 s_2} = c_{s_1 s_2}$, otherwise, $c^2_{s_1 s_2} = +\infty$.
- For all pairs $s \in S$ and $i \in V^1 \setminus S$, $c^1_{si} = +\infty$ and $c^1_{is} = 0$ and if $si \in E^1$ then $c^2_{si} = c_{si}$, otherwise $c^2_{si} = +\infty$.
- For all pairs $i, j \in V^1 \setminus S$, $c^1_{ij} = c^1_{ji} = 0$ and if $ij \in E^1$ then $c^2_{ij} = c_{ij}$, otherwise $c^2_{ij} = +\infty$.

Note that in this transformation, setting a cost to $+\infty$ is equivalent to say there is no such an assignment (if it is a assigment cost) or such an edge (if it is an edge cost) in the instance of ATSND. Thus there is no problem of how to represent $+\infty$ numerically.
We will show the followings:

1. A Steiner tree ST in G^1 correspond to a tree-star network TS of same cost in G.

2. A finite cost tree-star network TS in G correspond to a Steiner tree ST of same cost in G^1.

To prove (1)., we take the same tree as ST in G. We set $V(ST)$ as the core set and $V^1 \setminus V(ST)$ as client set. Now we complet the tree ST to the tree-star network TS by assign each client node $j \notin V(ST)$ to any node in $V(ST)$. It is easy to see that TS has the same cost as ST.
To prove (2)., we can see that the core set of TS should contain all nodes of S because otherwise the cost of TS should not be finite. Restricting the tree TS on the core set, we obtain a Steiner tree of the same cost as TS in G^1. $\square$

The STSND can be also proved to be NP-hard by a reduction this time from a NP-hard subproblem of the Steiner Tree problem.

Theorem 2. *The STSND problem is NP-hard.*

Proof : Let us consider the MST problem. We can see that a Steiner tree is

- either a spanning tree of the subgraph induced by S, the set of terminal nodes.
- or a tree containing all terminal nodes and at least a Steiner node (a non-terminal node).

Thus we can divise the MST problem into 2 following subproblems:

1. Finding a minimum panning tree of the subgraph induced by S.
2. Finding a minimum tree containing all terminal nodes and at least a Steiner node.

Problem (1). can be solved in polynomial time and as the MST problem is NP-hard, we deduce that Problem (2). is NP-hard. We transform $I^1 = (G^1 = (V^1, E^1), S, c)$ of this problem to an instance I of STSND as follows. The graph G in STSND is the complete graph whose node set is V^1.

- For all pairs $s_1, s_2 \in S$, $c^1_{s_1 s_2} = c^1_{s_2 s_1} = +\infty$, if $s_1 s_2 \in E^1$ then $c^2_{s_1 s_2} = c_{s_1 s_2}$, otherwise, $c^2_{s_1 s_2} = +\infty$.
- For all pairs $s \in S$ and $i \in V^1 \setminus S$, $c^1_{si} = c^1_{is} = +\infty$ and if $si \in E^1$ then $c^2_{si} = c_{si}$, otherwise $c^2_{si} = +\infty$.
- For all pairs $i, j \in V^1 \setminus S$, $c^1_{ij} = c^1_{ji} = 0$ and if $ij \in E^1$ then $c^2_{ij} = c_{ij}$, otherwise $c^2_{ij} = +\infty$.

Similarly as in the ATSND problem that in this transformation, setting a cost to $+\infty$ is equivalent to say there is no such an assignment (if it is a assigment cost) or such an edge (if it is an edge cost) in the instance of STSND. Thus there is no problem of how to represent $+\infty$ numerically.
We will show the followings:

1. A Steiner tree ST containing at least a Steiner node in G^1 correspond to a tree-star network TS of same cost in G.
2. A finite cost tree-star network TS in G correspond to a Steiner tree ST containing at least a Steiner node of same cost in G^1.

To prove (1)., we take the same tree as ST in G. We set $V(ST)$ as the core set and $V^1 \setminus V(ST)$ as client set. Now we complet the tree ST to the tree-star network TS by assign each client node $j \notin V(ST)$ to some Steiner node in $V(ST)$. It is easy to see that TS has the same cost as ST.
To prove (2)., we can see that the core set of TS should contain all nodes of S and at least a Steiner node because otherwise the cost of TS should not be finite. Restricting the tree TS on the core set, we obtain a Steiner tree containing at least a Steiner node of the same cost as TS in G^1. $\square$

3. A MIP formulation

We start from the caracterization of a tree as a connected graph with $n-1$ edges. The connectivity can be enforced with $n-1$ flows of value 1 having node 1 as common sink and all other nodes as target. In our formulation, these flows are replaced by a single generalized flow (constraints$(1)(2)$), so that we have far less variables and constraints. An edge is selected (either as an edge of the core tree network, or as an assignement of a client to a node of the core network) if there is a non-zero flow value on it.

For each edge ij not incident to node 1, we select capacity $x^1_{(ij)} - (n-2)x^2_{(ij)}$, with binary variables x^1_{ij}, x^2_{ij}. A value 1 for x^1_{ij} means that ij corresponds to an assignment of i to j or j to i. A value 1 for x^2_{ij} means that ij belongs to the core tree network.

The choice of node 1 as source of the flows is an arbitrary decision, and this node does not necessarily have to be in the optimal core tree network. Instead of solving $n-1$ different mixt linear integer programs, we use one additional binary variable x^3_{1j} for every edge $1j$ adjacent to node 1, having value 1 if this edge is the only edge adjacent to node 1, meaning that node 1 is a leaf and that there is a flow value $n-1$ on that edge.

To avoid considering cycles or cuts in the graph, we add a penalization to the cost function so that the flow values are minimized. We have to do so here because the cost function is not monotonous, due to the x^3_{ij} variables.

In the resulting formulation (P), the constraints (1) and (2) express the flow conservation, and the constraints (3) and (4) are capacity constraints.

The constraints (6) force a numbr of $n-1$ edges.

The exclusion constraints (8) force a choice for each edge to be in the core network ($x^2_{ij} = 1$) or to represent an assignment of a client to a concentrator ($x^1_{ij} = 1$ or $x^3_{ij} = 1$), or nothing.

The constraint (5) makes sure that x^3_{ij} take value 1 only in the particular case where node 1 is a leaf and $1i$ a pending edge.

The constraint (7) is not strictly necessary, it means that there is at most one pending edge incident to node 1.

$$
(P)\left\{
\begin{array}{llll}
\min z = \sum\limits_{(ij)\in E} (c^1_{(ij)} x^1_{(ij)} + c^2_{(ij)} x^2_{(ij)}) + \sum\limits_{(1i)\in E} c^1_{(1i)} x^3_{(1i)} \quad +\alpha \sum\limits_{(ij)\in E} (f_{ij} + f_{ji}) & & & \\
\text{s.t.} & & & \\
\sum\limits_{i\neq 1} f_{1i} - \sum\limits_{i\neq 1} f_{i1} & = n-1 & & (1) \\
\sum\limits_{j\neq i} f_{ij} - \sum\limits_{j\neq i} f_{ji} & = -1 & \forall i \neq 1 & (2) \\
f_{ij} + f_{ji} - x^1_{(ij)} - (n-2)x^2_{(ij)} & \leq 0 & \forall (ij) \in E - \delta(1) & (3) \\
f_{1j} + f_{j1} - x^1_{(1j)} - (n-2)x^2_{(1j)} - (n-1)x^3_{(1j)} & \leq 0 & \forall (1j) \in E & (4) \\
(n-1)x^3_{(1i)} - f_{1i} - f_{i1} & \leq 0 & \forall i \in N - \{1\} & (5) \\
\sum\limits_{k=1}^{3} \sum\limits_{(1i)\in E} x^k_{(1i)} & = n-1 & & (6) \\
\sum\limits_{(1i)\in E} x^3_{(1i)} & \leq 1 & & (7) \\
\sum\limits_{k=1}^{3} x^k_{(ij)} & \leq 1 & & (8) \\
x^1_{(ij)}, x^2_{(ij)}, x^1_{(1k)} \in \{0,1\} & \forall (ij) \in E & \forall (1k) \in E & \\
f_{ij} \geq 0 & \forall (ij) \in E & &
\end{array}
\right.
$$

The previous formulation can be changed by using incremental variables $y^k_{(ij)}$ instead of multiple choice variables $x^k_{(ij)}$. We then obtain the following formulation :

$$
(P') \left\{ \begin{array}{llll}
\min \quad z = \sum\limits_{(ij)\in E} (c^1_{(ij)} y^1_{(ij)} + (c^2_{(ij)} - c^1_{(ji)}) y^2_{(ij)}) \quad + \sum\limits_{(1i)\in E} (c^1_{(1i)} - c^2_{(1i)}) y^3_{(1i)} \quad + \alpha \sum\limits_{(ij)\in E} (f_{ij} + f_{ji}) & & & \\
\text{s.t.} & & & \\
(1), (2) & & & \\
f_{ij} + f_{ji} - y^1_{(ij)} - (n-3) y^2_{(ij)} & \leq 0 & \forall (ij) \in E/\delta(1) & (3') \\
f_{1j} + f_{j1} - y^1_{(1j)} - (n-3) y^2_{(1j)} - y^3_{(1j)} & \leq 0 & \forall (1j) \in E & (4') \\
(n-1) y^3_{(1i)} - f_{1i} - f_{i1} & \leq 0 & \forall i \in N - \{1\} & (5') \\
\sum\limits_{(1i)\in E} y^1_{(1i)} & = n-1 & & (6') \\
\sum\limits_{(1i)\in E} y^3_{(1i)} & \leq 1 & & (7') \\
y^2_{(ij)} - y^1_{(ij)} & \leq 0 & \forall (ij) \in E & (8') \\
y^3_{(1i)} - y^2_{(1i)} & \leq 0 & \forall (1i) \in E & (8'') \\
y^1_{(ij)}, y^2_{(ij)}, y^3_{(1k)} \in \{0,1\} & \forall (ij) \in E & \forall (1k) \in E & \\
f_{ij} \geq 0 & \forall (ij) \in E & &
\end{array} \right.
$$

4. Numerical results

In the presentation, we will discuss numerical results based on the formulations presented in the previous section. The instances are generated starting from the network problems used in [8] and [9], where 12 to 70 nodes are distributed randomly in a 1000×1000 square. The edge costs are taken equal to the rounded distances between nodes, and the assignment costs are a fraction of the edge costs. In a first series of instances, all edges can be selected for the core network (i.e.: they all have finite cost), whereas in a second series of instances, only the edges present in the graph of the initial instance have finite cost. This corresponds to cases where edges have to be chosen among the edges of a previously existing network).

Some preliminary results obtained with XPressMP 2006-B on a 3 GHz PC computer are shown in table 1. We give for each instance the number n of nodes, the number m of possible links, the value z^* of an optimal solution and its number of links in the core network.

	n	m	z*	links	T1	nodes	T1’	nodes	T2	nodes	T2’	nodes
d12-1	12	66	820	0	1,6	211	1,3	75	2,5	465	2,3	303
d12-2	12	66	976	0	2,3	633	1,6	393	5,2	1283	3,7	787
d12-3	12	66	828	0	1,0	73	1,2	149	1,3	95	1,8	209
d12-4	12	66	968	0	1,7	477	2,1	473	2,8	599	2,8	667
d12-5	12	66	764	0	1,3	269	1,3	217	1,6	283	1,6	113
d15-1	15	105	974	1	3,8	489	4 2	439	6,3	1125	10,1	2141
d15-2	15	105	1110	1	10,2	2171	15,1	3227	33,8	7493	23,9	5713
d15-3	15	105	1031	1	7,3	1445	6,1	1161	25,3	4997	16,2	3085
d15-4	15	105	1173	0	15,7	3689	15,4	3485	36,2	8327	28,6	5991
d15-5	15	105	1085	1	6,7	1193	9,9	1833	15,4	3019	13,5	2171
sp12-1	12	21	1314	4	4,4	1327	2,6	591	3,0	587	3,1	687
sp12-2	12	20	1305	4	0,8	67	1,0	89	1,7	161	4,1	717
sp12-3	12	22	1185	4	1,7	277	3,5	997	1,5	185	1,6	189
sp12-4	12	21	1179	4	1,1	75	1,0	65	1,3	57	1,3	87
sp12-5	12	20	1086	5	1,3	305	1,5	393	1,3	105	1,4	183
sp15-1	15	26	1542	7	32,6	11585	21,1	7809	11,4	2685	12,1	2629
sp15-2	15	27	1293	4	4,4	983	1,9	205	4,5	903	4,4	885
sp15-3	15	26	1441	5	4,3	933	3,6	775	3,4	354	4,4	515
sp15-4	15	27	1433	5	6,2	1753	4,3	1043	2,8	379	3,3	585
sp15-5	15	26	1422	4	3,9	867	5,4	1195	4,0	609	7,0	1599

Table 1: Table 1

For each instance, we give the computing time and the number of branching nodes for four different formulations. T_1 and T_2 refer to problems (P) and (P'). T'_1 and T'_2 refer to problem (P) and (P') with additional valid inequalities:

$$\sum_{(1i)\in E} x^1_{1i} + \sum_{(1i)\in E} x^2_{1i} + 2 \sum_{(1i)\in E} x^3_{1i} \geq 2 \qquad (9)$$

$$f_{1i} + f_{i1} - x^3_{1i} \leq n-2 \qquad \forall (1i) \in E \qquad (10)$$

$$\sum_{(1i)\in E} y^1_{1i} + \sum_{(1i)\in E} y^3_{1i} \geq 2 \qquad (9')$$

$$f_{1i} + f_{i1} - y^3_{1i} \leq n-2 \qquad \forall (1i) \in E \qquad (10')$$

The constraint (9) and $(9')$ mean that from node 1, there must be at least one edge with all the flow (there is at most one indeed) or two edges with only a part of the flow on each of them.
The constraints (10) (respectively $(10')$) force the variables x^3_{1i} (respectively y^3_{1i}) to value 1 whenever the total flow value on edge $1i$ is $n-1$.

On these small instances, it appears that the formulation (P) seems more efficient for instances with dense graph, whereas for sparse graphs, the formulation (P') gives better results.

References

[1] Dijkstra, E. W., "A Note on Two Problems in Connexion with Graphs," *Numer. Mathematik*, pp. 269–271, 1955.

[2] Ahuja, R. K., T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[3] E. Gourdin, M. Labbé, and H. Yaman, Telecommunication and Location. In Facility Location: Applications and Theory, Z. Drezner and H.W. Hamacher (Eds.), Springer, pp. 275-305, 2002.

[4] Y. Lee, B.H. Lim, and J.S. Park, A hub location problem in designing digital data service networks: lagrangian relaxation approach. *Location Science*, Vol.4, No. 3, pp. 185-194, 1996.

[5] C. Swamy and A. Kumar, Primal-Dual Algorithms for Connected Facility Location Problems. *Algorithmica*, Vol. 40, No. 4, pp. 245-269, 2004.

[6] V.H. Nguyen and S. Kedad-Sidhoum, New formulation for the Ring-Star problem. Technical report. 2006.

[7] M. Labbé, G. Laporte, I. Rodriguez Martin and J. J. Salazar Gonzalez, The ring star problem: polyhedral analysis and exact algorithm. *Networks*, Vol. 43, No. 3, pp. 177-189, 2004.

[8] V. Gabrel, A. Knippel and M. Minoux, Exact solution of multicommodity network optimization problems with general step cost functions. *Operations Research Letters* Vol. 25, pp. 15-23, 1999.

[9] V. Gabrel, A. Knippel and M. Minoux, A comparison of heuristics for the discrete cost multicommodity network optimization problem. *Journal of Heuristics*, Vol. 9, No 5, pp. 429-445, 2003.