A survivable variant of the Tree-Star problem

Louis Kurdyk, André Rossi, Sonia Toubaline

Université Paris Dauphine - PSL, CNRS, LAMSADE UMR 7243, 75016 Paris, France

firstname.lastname@dauphine.psl.eu

Keywords : *integer programming, network design, survivable networks, resilience*

1 Introduction

Given a complete mixed-weighted graph $G = (V, E \cup A)$, the Tree-Star (TS) problem is a network design problem where the goal is to construct a structure consisting of two interconnected sections. The first section, called the *backbone network*, is a tree spanning a subset of vertices known as *hubs*, using edges from E. The second section, called the *tributary network*, connects the remaining vertices, referred to as *terminals* (non-hub vertices), to the hubs in the backbone network using arcs from A. The objective of the TS problem is to design such a structure at minimum cost, with respect to a cost function $c : E \to \mathbb{R}^+$ for the edges and a cost function $d : A \to \mathbb{R}^+$ for the arcs. Additionally, d_{ii} represents the cost of opening a hub at vertex $i \in V$.

This problem is introduced by Lucena *et al.* in [1] where a mathematical model for solving it using a Branch-and-Cut algorithm is proposed, and its NP-hardness is established in [2].

Given the importance of resilience in network design, we study the *1-Survivable Tree-Star* (1-S-TS) problem, which is an extension of the TS problem. In this variant, we take into account one potential vertex failure, which may cause a loss in connectivity in one or both of the network sections. For example, if a non-leaf hub i in the backbone network fails, the tree could be split into a number of connected components equal to the degree of i, breaking the connectivity of the backbone network and potentially isolating any terminals attached to i. The 1-S-TS problem can be seen as a particular case of the *k-Survivable Tree-Star* (k-S-TS) problem in which up to k vertex failures can happen simultaneously. To prevent service disruptions due to hub breakdowns, we propose a model that ensures connectivity in the network despite the failure of any vertex in the graph, still at minimum cost. In order to achieve this, we consider the addition of extra edges and arcs, referred to as *backup edges* and *backup arcs*.

2 Problem formulation

The Integer Linear Programming (ILP) model we developed to solve the 1-S-TS problem can be interpreted as solving the nominal TS problem while ensuring survivability by adding backup edges. Specifically, we define the following binary decision variables:

- $x_{ij} = 1$ if edge ij connects two hubs in the backbone tree, for all ij in E
- $y_{ij} = 1$ if arc (i, j) is used to connect terminal i to hub j, for all $(i, j) \in A$. $y_{ii} = 1$ means that node i is selected as a hub, and $y_{ii} = 0$ indicates that node i is a terminal
- $x'_{ij} = 1$ if edge ij in E is used as a backup edge, *i.e.*, it is used to fix the backbone tree and maintain hub connectivity when some hub fails
- $x'^{i}_{jk} = 1$ if edge jk in $E \backslash \delta(i)$ is used as a backup edge to maintain connectivity when hub i is down, for all i in V. It is recalled that $\delta(i) = \delta(\{i\})$ is the set of all the edges of E that are incident to i.

With those variables, the ILP model is as follows:

$$\text{Minimize} \quad \sum_{ij \in E} c_{ij}(x_{ij} + x'_{ij}) + \sum_{(i,j) \in A} d_{ij} y_{ij}$$

$$\sum_{i \in V} y_{ii} = 1 + \sum_{ij \in E} x_{ij} \tag{1}$$

$$\sum_{ij \in E(S)} x_{ij} \leq \sum_{i \in S \setminus \{s\}} y_{ii} \quad \forall S \subseteq V, \forall s \in S \tag{2}$$

$$\sum_{jk \in E(S) \setminus \delta(i)} x_{jk} + \sum_{jk \in E(S)} x'^{i}_{jk} \leq \sum_{j \in S \setminus \{s\}} y_{jj} \quad \forall i \in V, \forall S \subseteq V \setminus \{i\}, \forall s \in S \tag{3}$$

$$\sum_{jk \in E \setminus \delta(i)} x'^{i}_{jk} = \sum_{jk \in \delta(i)} x_{jk} - y_{ii} \quad \forall i \in V \tag{4}$$

$$\sum_{i \in V} y_{ii} \geq 2 \tag{5}$$

$$\sum_{j \in V} y_{ij} = 2 - y_{ii} \quad \forall i \in V \tag{6}$$

$$x_{ij} + x'_{ij} + y_{ij} \leq y_{jj} \quad \forall ij \in E \tag{7}$$

$$x_{ij} + x'_{ij} + y_{ji} \leq y_{ii} \quad \forall ij \in E \tag{8}$$

$$x'^{i}_{jk} \leq x'_{jk} \quad \forall i \in V, \forall jk \in E \setminus \delta(i) \tag{9}$$

The objective function minimizes the cost of constructing the backbone network, including the setup costs of the hubs, the tributary network, and the backup edges. Constraint (1) ensures that the backbone has enough edges to form a tree. Furthermore, to guarantee that the backbone network is indeed a tree, constraint (2) prevents the formation of cycles. Constraints (3) and (4) ensure the connectivity of the backbone network even in the event of a hub failure, by selecting the appropriate number of backup edges and preventing these edges from forming cycles when the corresponding vertex i fails. While this approach preserves connectivity in the event of any hub failure, the main edges of the backbone network and the backups may jointly form cycles. The connection between the terminals and the rest of the network is enforced by constraint (6). Linking each terminal to two hubs ensures it remains connected to the backbone network even if one of the hubs it is linked to fails. To support this, constraint (5) imposes that there must be at least two hubs in the solution. Constraints (7) and (8) ensure that edges are only used to connect two hubs and that an edge cannot simultaneously serve as a main edge and a backup edge. These constraints also prevent direct connections between two terminals. Finally, constraint (9) ensures that any edge selected to address the failure of a vertex i is designated as a backup edge.

Different options are explored to solve some instances of 1-S-TS to optimality. Because the number of subtour elimination constraints (4) and (5) is exponential in number, we resort to a branch-and-cut approach, dynamically adding these constraints only when deemed necessary. Additionally, valid inequalities have been identified for solutions containing more than two hubs. Experimentally, using these inequalities has led to significant performance improvements, enabling the solution of larger problem instances. Ongoing research focuses on further enhancing performance to tackle larger problem instances. Numerical results obtained using Julia paired with Gurobi will be presented at the conference.

References

[1] A. Lucena, L. Simonetti and A. Salles da Cunha, *The Tree-Star Problem: A Formulation and a Branch-and-Cut Algorithm*, Electronic Notes in Discrete Mathematics, 52, 285–292, 2016.

[2] G. N. Frederickson and J. Ja'Ja, *Apporoximation algorithms for several graph augmentation problems*, SIAM Journal on Computing, 10, 270-283, 1981.