Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in DISCRETE MATHEMATICS
Electronic Notes in Discrete Mathematics 52 (2016) 285–292
www.elsevier.com/locate/endm

The Tree-Star Problem: A Formulation and a Branch-and-Cut Algorithm

Abilio Lucena [1]

Departamento de Administração / PESC-COPPE
Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil,
abiliolucena@cos.ufrj.br

Luidi Simonetti [2]

Instituto de Computação
Universidade Federal Fluminense
Niteroi, Brazil,
luidi@ic.uff.br

Alexandre Salles da Cunha [3]

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais,
Belo Horizonte, Brazil,
acunha@dcc.ufmg.br

Abstract

Let $G = (V, E)$ be a connected undirected graph and assume that an edge $e = \{i, j\} \in E$ may be priced differently, at the different spanning trees of G that contain it. A cost c_e applying when e is *leaf implying*, i.e., when e belongs to the spanning tree and i or j are spanning tree leaves. Otherwise, when e belongs to the tree but is not leaf implying, a cost d_e would apply. Under such a pricing of edges,

http://dx.doi.org/10.1016/j.endm.2016.03.038

the Tree-Star Problem is to find a least cost spanning tree of G. The problem does not appear to have been previously investigated in the literature and we show it to be NP-hard, formulate it as a mixed integer program, and propose and test a Branch-and-Cut algorithm for it. So far, the algorithm contains no primal heuristic. However, as it stands, it is already capable of solving to proven optimality tree-star instances with as many as 299 vertices.

Keywords: The Tree-Star Problem, Formulation, Branch-and-Cut Algorithm.

1 Introduction

Let $G = (V, E)$ be a connected undirected graph with a set of vertices V, a set of edges E, and two different sets of edge costs, $\mathbf{c}, \mathbf{d} \in \mathbb{R}^{|E|}$. It is assumed that E contains at least two edges. Additionally, for any spanning tree of G, say $T = (V, E_T)$, it is also assumed that E_T is partitioned into *leaf implying* and *internal* edges. An edge $e = \{i, j\} \in E_T$ being leaf implying when i or j are leaves of T. Otherwise, when neither i nor j are leaves of T, e is an internal edge. Assuming that $\mathbf{c}$ is used to price leaf implying edges and that $\mathbf{d}$ plays the same role for internal ones, the Tree-Star Problem (T-SP) is to find a minimum cost spanning tree of G. The problem, which does not appear to have been previously investigated in the literature, is addressed in this paper.

Quite clearly if one drops the leaves of a spanning tree of G, what remains is a non spanning tree of G, possibly containing a single vertex. Let us call that tree the *backbone* of the spanning tree. Conversely, leaf implying spanning tree edges are called *local access edges* to that backbone. A potential telecommunications application for T-SP is thus to find a spanning tree of G where the sum of backbone and local access edges is as small as possible. For such an application, the higher capacity backbone edges would typically cost more than their lower capacity local access counterparts.

T-SP is NP-hard since the Maximum Leaf Spanning Tree Problem (MLSTP) [3], which is known to be NP-hard, could be seen as a particular case of it. MLSTP being the problem of finding a spanning tree of G with as many leaves as possible. In order to reach the result, it suffices to take $\{c_e = -1 : e \in E\}$ and $\{d_e = 0 : e \in E\}$. Analogously, if one takes

[1] Abilio Lucena was partially funded by CNPq grant 307026/2013-2.
[2] Luidi Simonetti was partially funded by CNPq grant 304793/2011-6.
[3] Alexandre Salles da Cunha was partially funded by CNPq grants 200493/2014-0, 305423/2012-6, 471464/2013-9 and FAPEMIG CEX-PPM-00164-13.