Minimum Weight Tree Spanner Problem*

Hugo Braga

[1]Instituto de Matemática e Estatística – Universidade de São Paulo
05508-090 – São Paulo – SP – Brazil

hbraga@ime.usp.br

Abstract. *Let (G, w, t) denote a triplet consisting of a connected graph $G = (V, E)$ with a nonnegative weight function w defined on E, and a real number $t > 1$. A tree t-spanner of G is a spanning tree H of G such that for each pair of vertices u, v, the distance between u and v in H is at most t times the distance between u and v in G. We address the* Minimum Weight Tree Spanner Problem (MWTS)*, defined as follows. Given a triplet (G, w, t), find a tree t-spanner in G that has the smallest possible weight. It is known that* MWTS *is* NP*-hard for every fixed $t \geq 4$. We propose two ILP formulations for* MWTS*, based on arborescences, of polynomial size, and present some preliminary results on the computational experiments with these formulations.*

Resumo. *Seja (G, w, t) uma tripla formada por um grafo conexo $G = (V, E)$ com uma função peso w definida sobre E, e um número real $t > 1$. Uma árvore t-spanner de G é uma árvore geradora H de G tal que para quaisquer pares de vértices u, v, a distância entre u e v em H é no máximo t vezes a distância entre u e v em G. Estudamos o* problema da árvore spanner de custo mínimo*, denotado por* MWTS *(acrônimo de* Minimum Weight Tree Spanner*): dada uma tripla (G, w, t), encontrar em G uma árvore t-spanner que tenha o menor peso possível. Sabe-se que* MWTS *é* NP*-difícil para todo $t \geq 4$, fixo. Propomos duas formulações lineares inteiras para o* MWTS*, baseadas em arborescências, de tamanho polinomial, e apresentamos resultados preliminares sobre os experimentos computacionais realizados com essas formulações.*

1. Introduction

In this work, (G, w, t) denotes a triplet consisting of a connected graph $G = (V, E)$ with nonnegative weight w_e for each edge $e \in E$, and a real number $t > 1$. For two distinct vertices u, v in V, the *distance between u and v* in G, denoted by $\mathrm{dist}_G(u, v)$, is the length of a shortest path (w.r.t. w) from u to v in G.

A *t-spanner* of G is a spanning subgraph H of G such that $\mathrm{dist}_H(u, v) \leq t \cdot \mathrm{dist}_G(u, v)$ for all vertices u, v in V. The integer t is called the *strech factor*. A *tree t-spanner* is a t-spanner that is a tree. We study the *Minimum Weight Tree Spanner Problem* (MWTS), defined as follows: given a a triplet (G, w, t), find a tree t-spanner in G of minimum weight.

*Preliminary results of an ongoing research carried out by the author in his PhD program at IME-USP under the supervision of Professor Yoshiko Wakabayashi. This work also benefited from helpful discussions with Professor Manoel Campêlo (UFC). Research supported by FAPESP fellowship, Project (Proc. 2013/22875-9) and MaCLinC project of NUMEC/USP.

The cardinality version of this problem is basically a feasibility (or existence) problem. Even in this case, the problem is known to be NP-complete [Cai and Corneil 1995] for every fixed $t \geq 4$. It is known to be solvable in polynomial time for $t = 2$ [Bondy 1989, Cai and Corneil 1995]. The computational complexity status when $t = 3$ is unknown.

The more general problem, in which the objective is to find a t-spanner, has been largely investigated. It arises in distributed computing scenarios [Awerbuch 1985], communication networks [Peleg and Upfal 1988] and robotics. The concept of *spanners* was introduced in 1987 (in a conference) by Peleg and Ullman [Peleg and Upfal 1988], in connection with construction of synchronizers. Heuristics as well as a column-generation approach have been proposed for the graph spanner (but not for the tree spanner) problem.

In this work we focus on two ILP formulations for MWTS. In these formulations, we use the result proved by [Cai and Corneil 1995] that guarantees that the stretch factor condition can be simplified to $\text{dist}_H(u, v) \leq t \cdot w(u, v)$ for all $uv \in E$. To our knowledge, no approximation algorithms or ILP formulations have been proposed for MWTS.

2. Integer Linear Formulations for MWTS

The polyhedron of the tree spanner problem associated with (G, w, t) is defined as $P(G, t) := \text{conv}\{\mathcal{X}^F \in \mathbb{R}^{|E|} \mid G[F] \text{ is a tree } t\text{-spanner}\}$, where $\mathcal{X}^F$ denotes the incidence vector of F. In what follows we present two ILP formulations. With respect to the two polyhedra defined by the convex hull of the integer points satisfing these formulations, $P(G, t)$ is a projection on the space $\mathbb{R}^{|E|}$.

2.1. Formulation 1: finding distances between vertices

Given a graph $G = (V, E)$, let $D = (V, A)$ be the digraph obtained from G, where $A = \{(u, v), (v, u) : uv \in E\}$. Let $\mathbb{B} := \{0, 1\}$. Take a vertex $v \in V$ (root of a v-rooted arborescence) and consider the variable $z^v = (z^v_{ij})_{ij \in A}$, associated with v. This variable tells which arcs are in the v-rooted arborescence. In the formulation we deal with aborescences in D, rooted at different vertices of this digraph. They allow us to find paths between vertices on the given arborescences in a easy way. Then, using these arborescences, we construct a spanning tree (defined by the variables x) satisfying the stretch factor condition.

The decision variable $x \in \mathbb{B}^{|E|}$ has the following meaning: for each edge e, $x(e) = 1$ if and only if e is part of the solution. For each $f \in E$, consider $y^f = (y^f_e)_{e \in E}$. The variable $y \in \mathbb{B}^{|E| \times |E|}$ has the following meaning: for each edge e, and edge $f = uv$, $y^f_e = 1$ if and only if e is in the path between u and v in the solution tree.

$$\min \sum_{e \in E} w_e x_e$$

s.t.

$$\sum_{e \in E} x_e = |V| - 1 \tag{1}$$

$$\sum_{i \in \delta^-(j)} z^r_{ij} = 1 \qquad \forall r \in V, \forall j \in V \setminus \{r\} \tag{2}$$

$$\sum_{i \in \delta^-(r)} z^r_{ir} = 0 \qquad \forall r \in V \tag{3}$$

$$x_e = z^r_{ij} + z^r_{ji} \qquad \forall r \in V,\ \forall e = \{i,j\} \in E \tag{4}$$

$$z^u_{ij} - z^v_{ij} \leq y^{uv}_e \leq z^u_{ij} + z^v_{ij} \qquad \forall uv \in E,\ \forall e = \{i,j\} \in E \tag{5}$$

$$z^u_{ji} - z^v_{ji} \leq y^{uv}_e \leq z^u_{ji} + z^v_{ji} \qquad \forall uv \in E,\ \forall e = \{i,j\} \in E \tag{6}$$

$$\sum_{e \in E} w_e\, y^{uv}_e \leq t \cdot w(u,v) \qquad \forall uv \in E \tag{7}$$

$$x \in \mathbb{B}^{|E|}, y \in \mathbb{B}^{|E|\times|E|}, z^v \in \mathbb{B}^{|E|\times|E|} \qquad \forall v \in V \tag{8}$$

2.2. Formulation 2: with variables representing distances between pairs of vertices

For this formulation, we consider the same setting as the previous formulation. Given $G = (V, E)$, we consider similarly the digraph $D = (V, A)$. The variables $z \in \mathbb{R}^{|V|\times|E|\times|E|}$ and $x \in \mathbb{R}^{|E|}$ are also defined similarly. Additionally, now let $u \in \mathbb{R}^{|V|\times|V|}$ be a variable, such that for $i, j \in V$, when $u^j_i = u^i_j$, then u^i_j represents the distance between i and j. In the formulation, M_{ij} is a given upper bound on the distance between vertices i and j in G.

$$\min \sum_{e \in E} w_e x_e$$

s.t.

$$\sum_{e \in E} x_e = |V| - 1 \tag{9}$$

$$\sum_{i \in \delta^-(j)} z^r_{ij} = 1 \qquad \forall r \in V,\ \forall j \in V \setminus \{r\} \tag{10}$$

$$\sum_{i \in \delta^-(r)} z^r_{ir} = 0 \qquad \forall r \in V \tag{11}$$

$$x_e = z^r_{ij} + z^r_{ji} \qquad \forall r \in V,\ \forall e = \{i,j\} \in E \tag{12}$$

$$u^r_i - u^r_j + (M_{ij} + w_{ij}) z^r_{ij} + (M_{ij} - w_{ij}) z^r_{ji} \leq M_{ij} \qquad \forall r \in V, \forall ij \in A, j \neq r \tag{13}$$

$$u^r_i + (M_{ir} - w_{ir}) z_{ri} \leq M_{ir} \qquad \forall r \in V, \forall ri \in A \tag{14}$$

$$u^r_r = 0 \qquad \forall r \in V \tag{15}$$

$$u^j_i = u^i_j \leq t \cdot w_{ij} \qquad \forall ij \in E \tag{16}$$

$$x \in \mathbb{B}^{|E|}, z^r \in \mathbb{B}^{|E|\times|E|}, u^r \in \mathbb{R}^{|V|} \qquad \forall r \in V \tag{17}$$

The idea behind constraint (13) was used by [Miller et al. 1960] for the TSP.

3. Computational experiments

We carried out some computational experiments to compare the two formulations and to have some idea of the strength of these formulations. We focused the unweighted case (cardinality version) and the case in which the weights represent Euclidean distance. For the cardinality version, we implemented a polynomial-time algorithm (showed by Nadiradze, 2013) for the *Bounded Diameter Minimum Spanning Tree* problem to use in the preprocessing phase. We also added two other valid inequalities for this version.

Our implementation was coded in C++, using CPLEX as the ILP solver. The code was run on a machine with 65 GB of RAM and 1.6 GHz (using single core). We present only results obtained with Formulation 2, as it consistently outperformed Formulation 1. Table 1 shows results for the cardinality version. We considered three parameters, and for each combination, we generated 10 random instances. *Density* means the percentage of edges in the graph compared to the complete graph. *Time* is the average time in seconds (for the instances solved with ILP); TLE (Time Limit Exceeded) indicates the number of instances not solved within 1 hour of CPU time.

t	$\|V\|$	Density	# Solved in prep. phase	TLE	# Solved with ILP	Time (s)
	10	40	5	0	5	0.02
	10	60	10	0	0	-
	20	20	0	0	10	0.07
	20	40	0	0	10	7.03
	20	60	7	0	3	87.61
3	30	20	0	0	10	0.32
	30	40	0	0	10	1250.94
	30	60	7	3	0	-
	40	20	0	0	10	83.30
	40	40	0	10	0	-
	40	60	0	10	0	-

Table 1. Computational results for the cardinality version

For $t = 3$, instances with density 40 and 60 are the hardest ones. For this case, almost all instances (of order up to 60) with density 20 could be solved quickly. Similarly, for $t \in \{3, 4\}$, all instances (of order up to 60) with density 80 were solved quickly (all of them with preprocessing). For $t = 4$, all instances with density 40 or 60 were solved quickly (with positive answer).

For the Euclidean case, we considered instances of order in the range from 10 to 120 (subgraphs of graphs from public libraries). For $t = 3$, all instances were solved. However, few of these instances admit a tree 3-spanner, and these are all of low order. For $t = 4$, the greater the density, the smaller the number of instances solved. In this case, for instances with density 80, only instances of order up to 40 were solved. For the instances considered, finding tree spanners for $t = 3$ was faster than for $t = 4$. We have to perform more computational experiments, but it seems that the random instances, cardinality version, are harder to be solved.

References

[Awerbuch 1985] Awerbuch, B. (1985). Complexity of network synchronization. *J. ACM*, 32(4):804–823.

[Bondy 1989] Bondy, J. A. (1989). Trigraphs. *Discrete Math.*, 75(1-3):69–79.

[Cai and Corneil 1995] Cai, L. and Corneil, D. G. (1995). Tree spanners. *SIAM J. Disc. Math.*, 8(3):359–387.

[Miller et al. 1960] Miller, C. E., Tucker, A. W., and Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *J. ACM*, 7(4):326–329.

[Peleg and Upfal 1988] Peleg, D. and Upfal, E. (1988). A tradeoff between space and efficiency for routing tables. In *STOC '88*, pages 43–52, New York, USA.