Discrete Applied Mathematics 294 (2021) 152–166

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

Minimum Spanning Tree Cycle Intersection problem

Manuel Dubinsky [a,*], César Massri [b,1], Gabriel Taubin [c,2]

[a] Ingeniería en Informática, Departamento de Tecnología y Administración, Universidad Nacional de Avellaneda, Argentina
[b] Departamento de Matemática, Universidad de CAECE, CABA, Argentina
[c] School of Engineering, Brown University, Providence, RI, USA

ARTICLE INFO

Article history:
Received 24 December 2019
Received in revised form 21 January 2021
Accepted 28 January 2021
Available online 20 February 2021

Keywords:
Graphs
Spanning trees
Cycle bases

ABSTRACT

Consider a connected graph G and let T be a spanning tree of G. Every edge $e \in G - T$ induces a cycle in $T \cup \{e\}$. The intersection of two distinct such cycles is the set of edges of T that belong to both cycles. We consider the problem of finding a spanning tree that has the least number of such non-empty intersections. In this article we analyze the particular case of complete graphs, and formulate a conjecture for graphs that have a universal vertex.

1. Introduction

In this article we present what we believe is a new problem in graph theory, namely the Minimum Spanning Tree Cycle Intersection (*MSTCI*) problem which arose while investigating a (yet unpublished) method for *mesh deformation* in the area of *digital geometry processing*, see [2].

The problem can be expressed as follows. Let G be a graph and T a spanning tree of G. Every edge $e \in G - T$ induces a cycle in $T \cup \{e\}$. The intersection of two distinct such cycles is the set of edges of T that belong to both cycles. Consider the problem of finding a spanning tree that has the least number of such pairwise non-empty intersections.

The remainder of this section is dedicated to express the problem in the context of the theory of *cycle bases*, where it has a natural formulation, and to describe an application. Section 2 sets some notation and convenient definitions. In Section 3 the complete graph case is analyzed. Section 4 presents a variety of interesting properties, and a conjecture in the slightly general case of a graph (not necessarily complete) that admits a star spanning tree. Section 5 explores programmatically the space of spanning trees to provide evidence that the conjecture is well posed. Section 6 collects the conclusions of the article.

1.1. Cycle bases

The study of cycles of graphs has attracted attention for many years. To mention just three well known results consider *Veblen's theorem* [15] that characterizes graphs whose edges can be written as a disjoint union of cycles, *Maclane's planarity criterion* [11] which states that planar graphs are the only to admit a 2-basis, or the *polygon matroid* in Tutte's classical formulation of *matroid theory* [14].

* Corresponding author.
E-mail address: mdubinsky@undav.edu.ar (M. Dubinsky).

1 Massri was supported by Instituto de Investigaciones Matemáticas "Luis A. Santaló", UBA, CONICET, CABA, Argentina.
2 Taubin was partially supported by National Science Foundation, USA grant number IIS-1717355.

https://doi.org/10.1016/j.dam.2021.01.031
0166-218X/

The set of cycles of a graph has a vector space structure over $\mathbb{Z}_2$, in the case of undirected graphs, and over $\mathbb{Q}$, in the case of directed graphs [9]. A basis of such a vector space is denoted *cycle basis* and its dimension is the *cyclomatic number* $\nu = |E| - |V| + |CC|$ where E, V ad CC are the set of edges, vertices and connected components of the graph, resp. Given a cycle basis B we can define its *cycle matrix* $\Gamma \in K^{|E|\times\nu}$ where K is the scalar field (i.e.: $\mathbb{Z}_2$ or $\mathbb{Q}$), as the matrix that has the cycles of B as columns.

Different classes of cycle bases can be considered. In [10] the authors characterize them in terms of their corresponding cycle matrices and present a *Venn diagram* that shows their inclusion relations. Among these classes we can find the *strictly fundamental* class.

The *length* of a cycle is its number of edges. The *minimum cycle basis* (*MCB*) problem is the problem of finding a cycle basis such that the sum of the lengths (or edge weights) of its cycles is minimum. This problem was formulated by Stepanec [13] and Zykov [16] for general graphs and by Hubicka and Syslo [8] in the strictly fundamental class context. In more concrete terms this problem is equivalent to finding the cycle basis with the sparsest cycle matrix. In [9] a unified perspective of the problem is presented. The authors show that the *MCB* problem is different in nature for each class. For example in [3] a remarkable reduction is constructed to prove that the *MCB* problem is NP-hard for the strictly fundamental class, while in [7] a polynomial time algorithm is given to solve the problem for the undirected class. Some applications of the *MCB* problem are described in [1,3,7,9].

A related problem not covered in the literature (as far as we know) is to consider the sparsity of the *grammian matrix* of a cycle matrix. Let $B = (C_1, \ldots, C_\nu)$ be a cycle basis with corresponding cycle matrix Γ. The grammian of Γ is $\hat{\Gamma} = \Gamma^t\Gamma$. We will denote $\hat{\Gamma}$ the *cycle intersection matrix* of B. It is easy to check that the ij-entry of $\hat{\Gamma}$ is 0 if and only if the cycles C_i and C_j do not intersect (i.e.: they have no edges in common). It can be formulated as follows:

Problem. Let G be a (directed) graph, find a cycle basis B with corresponding cycle matrix Γ such that the grammian $\hat{\Gamma} = \Gamma^t\Gamma$ is sparsest.

In this context the MSTCI problem corresponds to the particular case of bases that belong to the strictly fundamental class.

1.2. An application

Let $G = (V, E)$ be a directed connected graph and $w : E \to \mathbb{R}$ be an edge function. We call w a *discrete 1-form* on G. Integrating w is the problem of finding a vertex function $x : V \to \mathbb{R}$ minimizing the error:

$$E(x) = \sum_{e_{ij}\in E} \|dx(e_{ij}) - w(e_{ij})\|^2.$$

where $dx : E \to \mathbb{R}^n$ is defined as

$$dx(e_{ij}) := x(v_j) - x(v_i).$$

on every directed edge $e_{ij} := v_i \to v_j$, and is called the *differential* of x. Note that w has the following property: $w(e_{ij}) = -w(e_{ji})$, where e_{ji} is the same underlying edge e_{ij} with opposite direction.

Given some consistent enumeration of the vertices and edges, the integration problem can be expressed in a compact form:

$$\underset{x}{\operatorname{argmin}} \|D\mathbf{x} - \mathbf{w}\|_2^2.$$

where $D \in \{0, 1, -1\}^{|E|\times|V|}$ is the *directed incidence matrix* of G, $\mathbf{w} \in \mathbb{R}^{|\mathbf{E}|}$ is the evaluation of w on the edges and $\mathbf{x} \in \mathbb{R}^{|\mathbf{V}|}$ is the solution. From a geometric perspective $D\mathbf{x}$ can be visualized as the orthogonal projection of $\mathbf{w}$ onto the subspace generated by D. The rank of the directed incidence matrix is $|V| - 1$, its kernel is generated by $\mathbb{1} \in \mathbb{R}^{|V|}$, the vector of all 1's. This degree of freedom can be geometrically interpreted as a rigid translation in the solution space. If we fix the value of the 0-form at some vertex, we can eliminate this degree of freedom. More precisely, we can fix the value of the first component of our solution vector: $x(v_1) = 0$. This will be equivalent to eliminating the first column of D. Let $\hat{D}$ be this new matrix of dimension $|E| \times |V| - 1$.

An alternative way to solve the integration problem is to extend $\hat{D}$ to a basis of $\mathbb{R}^{|E|\times|E|}$ and solve a linear system:

$$M\begin{bmatrix}\hat{\mathbf{x}}\\ \mathbf{y}\end{bmatrix} = \left[\ \hat{D}\ \middle|\ \hat{D}^{\perp}\ \right]\begin{bmatrix}\hat{\mathbf{x}}\\ \mathbf{y}\end{bmatrix} = \left[\ \hat{D}\ \middle|\ \Gamma\ \right]\begin{bmatrix}\hat{\mathbf{x}}\\ \mathbf{y}\end{bmatrix} = \mathbf{w}.$$

where $\hat{\mathbf{x}}$ is the solution vector $\mathbf{x}$ without its first component (because $x(v_1) = 0$) and $\hat{D}^{\perp}$ is a set of generators of the orthogonal complement of $\hat{D}$. In this setting, a natural question is: how can we choose $\hat{D}^{\perp}$ such that M is as sparse as possible? An answer is given by considering the *cycle matrix* Γ of the *minimum cycle basis* B of G. It is easy to check that the columns of Γ are orthogonal to the columns of $\hat{D}$.

Yet another way of solving the integration problem is to consider the *Gram* matrix (i.e.: M^tM) of M. More precisely, we can left multiply by M^t in the previous equation:

$$M^tM \begin{bmatrix} \hat{\mathbf{x}} \\ \mathbf{y} \end{bmatrix} = \left[\begin{array}{c|c} \hat{D} & \Gamma \end{array}\right]^t \left[\begin{array}{c|c} \hat{D} & \Gamma \end{array}\right] \begin{bmatrix} \hat{\mathbf{x}} \\ \mathbf{y} \end{bmatrix} = \left[\begin{array}{c|c} \hat{L} & 0 \\ 0 & \hat{\Gamma} \end{array}\right] \begin{bmatrix} \hat{\mathbf{x}} \\ \mathbf{y} \end{bmatrix} = M^t\mathbf{w}.$$

where $\hat{L} = \hat{D}^t\hat{D}$ is the lower right $|V| - 1 \times |V| - 1$ submatrix of the *laplacian* matrix of G and $\hat{\Gamma} = \Gamma^t\Gamma$ is the cycle intersection matrix of B. The same question can be formulated in this setting: how can we choose B such that its corresponding cycle intersection matrix $\hat{\Gamma}$ is as sparse as possible? In the particular case where the cycle basis is in fact a strictly fundamental cycle basis, namely a cycle basis induced by a spanning tree, this is precisely the MSTCI problem.

2. Preliminaries

2.1. Overview

In the first part of this section we present some of the terms used in this article. Then, we define the notion of *closest-point* and *closest-point-set*. Finally, we show a convenient cycle partition.

2.2. Notation

Let $G = (V, E)$ be a graph and T a spanning tree of G. We will then refer to the edges $e \in T$ as *tree-edges* and to the $e \in G - T$ ones as *cycle-edges*.

Every cycle-edge e induces a cycle in $T \cup \{e\}$, which we will call a *tree-cycle*. We will name C_T to the set of tree-cycles of T.

The intersection of two tree-cycles is the set of edges of T that belongs to both cycles. We will define three functions concerning the intersection of tree-cycles.

The first is $\cap_T(\cdot,\cdot) : C_T \times C_T \to \{0, 1\}$

$$\cap_T(c_i, c_j) := \begin{cases} 1 & c_i \cap c_j \neq \varnothing \wedge c_i \neq c_j. \\ 0 & c_i \cap c_j = \varnothing \vee c_i = c_j. \end{cases}$$

As every tree-cycle intersects with itself, the case $c_i = c_j$ is excluded to focus on non-trivial intersections. This consideration will simplify future computations.

The second is $\cap_T(\cdot) : C_T \to \mathbb{N}$

$$\cap_T(c_i) := \sum_{c_j \in C_T} \cap_T(c_i, c_j).$$

We will call $\cap_T(c)$ the *cycle intersection number* of c. Given a tree-cycle c we will denote $\cap_{T,c}$ as the set of tree-cycles that have non-empty intersection with c. More precisely:

$$\cap_{T,c} \equiv \{c' \in C_T : \cap_T(c, c') = 1\}.$$

Note that $|\cap_{T,c}| = \cap_T(c)$.

In order to define the third function, consider $\mathscr{T}_G$ to be the set of spanning trees of G, therefore the definition will be as follows: $\cap_G : \mathscr{T}_G \to \mathbb{N}$

$$\cap_G(T) := \frac{1}{2} \sum_{c \in C_T} \cap_T(c).$$

We will call $\cap_G(T)$ the *tree intersection number* of T.

If the graph is clear from the context, we could remove the subindex and just write $\cap(T)$.

We shall call *star* spanning tree to one that has one vertex that connects to all other vertices, and K_n to the complete graph on n nodes. If $G = (V, E)$ we will say that $|V| = n$ is the number of vertices of G, $|c| = k$ is the length of the cycle c and $|p|$ is the length of the path p. Thus, uTv will denote the unique path between $u, v \in V$ in the spanning tree T; and $d_T(v)$ will be the degree of $v \in V$ relative to it. Whereas $N(v)$ will be the set of neighbor nodes of $v \in V$ and finally, the terms "node" and "vertex" will be used interchangeably.

2.3. Closest point

In this section we prove the following simple fact: if $G = (V, E)$ is a connected graph, T a spanning tree of G and $c \in C_T$ a tree-cycle, then for every node $v \in V$ there is a unique node $w \in c$ that minimizes the distance to v in T. We shall denote that node $closest - point(v, c)$.

Lemma 1. *Let $G = (V, E)$ be a connected graph, T a spanning tree of G and $c \in C_T$ a tree-cycle. Then for every node $v \in V$ there exists a unique node $w \in c$ such that*

$$|vTw| \leq |vTu| \ \forall u \in c.$$

Proof. The proof proceeds by contradiction. If $v \in c$, it is its own unique closest point. Suppose that $v \notin c$ and that there are two distinct nodes $w, w' \in c$ such that $|vTw| = |vTw'| \leq |vTu| \ \forall u \in c$, note that $w' \notin vTw$ and $w \notin vTw'$. We conclude that $vTw \cup wTw' \cup vTw'$ determines a cycle in T which contradicts the fact that T is a tree. □

The uniqueness of the *closest − point*(v, c) leads to the following definition.

Definition 2. Let $G = (V, E)$ be a connected graph, T a spanning tree of G and $c \in C_T$ a tree-cycle, then the set of closest points to a node $w \in c$ is defined as follows:

$$closest - point - set(w, c) := \{v \in V - c : closest - point(v, c) = w\}.$$

2.4. Tree cycle intersection partition

Now we define a partition of the set $\cap_{T,c}$. More precisely, let G be a connected graph, T a spanning tree of G and $c \in C_T$ a tree-cycle. As mentioned above, the set $\cap_{T,c}$ is the set of tree-cycles that have non-empty intersection with c.

Let us consider any tree-cycle $c' \in \cap_{T,c}$ induced by a cycle-edge $e = (v, w)$. In this setting we can define the following partition:

- *Internal tree-cycles*: c' is *internal* if $v, w \in c$.
- *External tree-cycles*: c' is *external* if $v \notin c$ and $w \in c$.
- *Transit tree-cycles*: c' is *transit* if $v, w \notin c$.

Let us denote the set of cycles of each type $\cap^i_{T,c}$, $\cap^e_{T,c}$, $\cap^t_{T,c}$, respectively. This partition will be convenient to simplify the computation of the *intersection number* of c.

2.5. Important remark

There is an alternative point of view that may clarify some of the proofs of this article. Instead of considering tree-cycles as the central object, this point of view considers paths in the spanning tree. More precisely let G be a connected graph, $T \in \mathscr{T}_G$ a spanning tree and $e_1, = (v_1, w_1), e_2 = (v_2, w_2) \in E$ two distinct cycle-edges with corresponding tree-cycles c_1, c_2. Then the following holds: $c_1 \cap c_2 = (v_1Tw_1) \cap (v_2Tw_2)$. Consequently, it is equivalent to consider tree-cycle intersections and intersection of paths in the spanning tree.

3. Tree cycles of complete graphs

3.1. Overview

In this section we analyze the complete graph case $G = K_n$ $(n \geq 3)$. First we deduce a formula to compute the cycle intersection number. Then we prove that the tree-cycles of a star spanning tree achieve the minimum cycle intersection number. Finally, we conclude that the star spanning trees are the unique solutions of the MSTCI problem.

3.2. Cycle intersection number formula

In this subsection we consider the problem of finding a formula to count tree-cycle intersections. More precisely, let $G = K_n$, T a spanning tree of G and c a tree-cycle, we intend to derive a formula to calculate $\cap_T(c)$.

The idea behind the formula is to consider the partition of $\cap_{T,c}$, defined in the previous section, and then by combinatorial arguments, compute the number of elements in each class.

We shall analyze in turn the three classes: $\cap^i_{T,c}$, $\cap^e_{T,c}$, $\cap^t_{T,c}$. In this section we will consider $c' \in \cap_T(c)$ to be a tree-cycle induced by a cycle-edge $e = (v, w)$.

The simplest case is the internal tree-cycles class: $\cap^i_{T,c}$. Let c' be an internal tree-cycle. By definition the nodes v and w belong to c, so the following holds: $(c' \cap T) \subset c$ because there is a unique path from v to w in T. So basically counting the number of internal tree-cycles reduces to counting the pairings on the nodes of c excluding some obvious cases such as the pairing of a node with itself and with its neighbors in c. Then the number of internal tree-cycles is:

$$|\cap^i_{T,c}| = \frac{(k-3)k}{2}.$$

where k is, as before, equal to $|c|$. The quotient is obviously due to the fact that every cycle is counted twice.

Next we consider the class of external tree-cycles. Now let c' be an external tree-cycle. In this case, by definition, either v or w belong to c. Without loss of generality (as we are considering undirected edges), suppose that $v \notin c$ and $w \in c$. Clearly $w \neq closest - point(v, c)$ because in that case $c' \cap c = \varnothing$ and consequently $c' \notin \cap_{T,c}$ which contradicts our hypothesis. Since the aforementioned is the only particular case that should be excluded, the number of external tree-cycles is:

$$|\cap^e_{T,c}| = (n-k)(k-1).$$

where $n = |V|$ is the number of vertices of G and $k = |c|$ is the length of c.

Last we consider the class of transit tree-cycles. In this case the key observation depends on the *closest − point − set* definition of the previous section. Let us define two classes of cycle-edges:

1. A cycle-edge $e = (v, w)$ is called *intraset cycle-edge* if both $v, w \in closest - point - set(u_i, c)$ for some $u_i \in c$
2. A cycle-edge $e = (v, w)$ is called *interset cycle-edge* if $v \in closest-point-set(u_i, c)$ and $w \in closest-point-set(u_j, c)$ where $u_i, u_j \in c$ and $u_i \neq u_j$

Then:

- Every *intraset cycle-edge* induces a tree-cycle c' such that $c' \cap c = \varnothing$
- Every *interset cycle-edge* induces a tree-cycle c' such that $c' \cap c \neq \varnothing$

So we should consider *interset cycle-edges* or equivalently, the pairing of the nodes that are in different sets. Let $q_i = |closest - point - set(w_i, c)|$ be defined for all $w_i \in c$, then the number of transit tree-cycles is:

$$|\cap^t_{T,c}| = \sum_{i<j} q_i q_j = \frac{1}{2}\sum_{i=1}^{k} q_i(n-k-q_i).$$

Finally, the intersection number formula is the aggregation of the three classes:

$$\cap_T(c) = |\cap_{T,c}| = |\cap^i_{T,c}| + |\cap^e_{T,c}| + |\cap^t_{T,c}| = \frac{(k-3)k}{2} + (n-k)(k-1) + \frac{1}{2}\sum_{i=1}^{k} q_i(n-k-q_i).$$

where n is the number of vertices of G, $k = |c|$ and $q_i = |closest - point - set(w_i, c)|$ for $w_i \in c$.

3.3. Main result

In this subsection we start by defining *transitless* tree-cycles. Then we prove two lemmas. The first one shows that for every cycle $c \in G = K_n$ we can build a spanning tree T such that c is a tree-cycle of T and the intersection number $\cap_T(c)$ is minimum. And the second one calculates the intersection number of tree-cycles of star spanning trees. Finally, we prove the main result of this section, namely that star spanning trees minimize $\cap(\cdot)$ in the case of complete graphs.

Definition 3. Let $G = (V, E)$ be a connected graph, T a spanning tree of G and $c \in C_T$ a tree-cycle, we call c a *transitless* tree-cycle if $|\cap^t_{T,c}| = 0$.

As an important remark, note that the number of elements in the internal and external classes of c are independent of the spanning tree because they depend exclusively on $n = |V|$ and $k = |c|$. Thus, two spanning trees, T_1 and T_2, which have c as a tree-cycle, induce an intersection number (for c) that only differs in the quantity of elements in their transit classes. We conclude that transitless tree-cycles have minimum intersection number.

Lemma 4. *Let $G = K_n$ and c a cycle of G. Then, the following construction leads to a spanning tree T that minimizes the intersection number of c:*

- *Let $e \in E$ such that $e \in c$.*
- *Let $v \in V$ such that $v \in c$.*
- *Define the set of edges of T as follows:*

$$E(T) = \{e' \in E : e' \in (c-e)\} \cup \{(v, w) \in E : w \in V \wedge w \notin c\}.$$

Proof. Note that T is a spanning tree of G, and c is a tree-cycle of T. Therefore, if we prove that $|\cap^t_{T,c}| = 0$ then the intersection number $\cap_T(c)$ is minimum. This is the case:

- $|closest - point - set(w, c)| = 0 \ \forall w \in c, w \neq v.$
- $|closest - point - set(v, c)| = n - k.$

So $|\cap^t_{T,c}| = \sum_{i<j} q_i q_j = 0.$ □

Lemma 5. *Let $G = K_n$ and let T_s be a star spanning tree of G. Then the following property holds*

$$\cap_{T_s}(c) = 2(n-3).$$

for any tree-cycle c of T_s.

Proof. Clearly the tree-cycles in T_s have the same intersection number (by symmetry). Let c be a tree-cycle of T_s. Note that c is a triangle ($|c| = 3$), so the corresponding internal tree-cycle class is empty: $|\cap^i_{T,c}| = 0$. Being c a transitless tree-cycle because its nodes are: the central node and two leaf nodes of T_s. Thus the external tree-cycle class is the only non-empty one:

$$\cap_{T_s}(c) = |\cap^e_{T_s,c}| = 2(n-3).$$

Proposition 6. *Let $G = K_n$ and let T_s be a star spanning tree of G. Then the following property holds*

$$\cap_{T_s}(\cdot) \leq \cap_T(\cdot).$$

where T is any spanning tree of G.

Proof. We shall proceed by contradiction. Suppose that a spanning tree T and a tree-cycle c of T exist such that:

$$\cap_T(c) < \cap_{T_s}(\cdot) = 2(n-3).$$

We can assume that c is transitless because, if this were not the case, by Lemma 4 we could build a spanning tree T' such that $\cap_{T'}(c) < \cap_T(c)$. In this context the inequality can be expressed as

$$\cap_T(c) = |\cap^i_{T,c}| + |\cap^e_{T,c}| = \frac{(k-3)k}{2} + (n-k)(k-1) < 2(n-3).$$

Expanding and simplifying the expression we have

$$\frac{-1}{2}k^2 + (n - \frac{1}{2})k - 3n + 6 < 0.$$

The roots of this quadratic polynomial are: $r_1 = 3$ and $r_2 = 2(n-2)$. We should consider two cases depending on the relation of the roots:

1. $r_1 < r_2$.
2. $r_1 > r_2$.

The case $r_1 = r_2$ can be discarded because it leads to a fractional number of nodes ($n = \frac{7}{2}$). In the first case the inequality holds for $k < r_1 = 3$ or $k > r_2 = 2(n-2)$. The case $k < 3$ is an obvious contradiction since the size of the cycle must be $|c| = k \geq 3$. The case $k > 2(n-2)$ combined with the fact that $k \leq n$ induces the following inequality

$$r_1 = 3 < r_2 = 2(n-2) < k \leq n.$$

which implies a contradiction: $3 < n < 4$, because n is a positive integer.

The second case ($r_1 = 3 > r_2 = 2(n-2)$) implies $n < \frac{7}{2}$. So the only case that should be considered is $k = n = 3$ since $k \leq n$. However, this case makes the inequality false because $k = 3$ is a root of the quadratic polynomial. □

Corollary 7. *Let $G = K_n$ and let T_s be a star spanning tree of G. Then the following property holds*

$$\cap(T_s) \leq \cap(T).$$

where T is any spanning tree of G.

Proof. As expressed by Proposition 6, a tree-cycle of a star spanning tree has the minimum intersection number among all tree-cycles. Since any tree-cycle of a star spanning tree has the same intersection number, we conclude that the tree intersection number of a star spanning tree $\cap(T_s)$ is minimum among all spanning trees. □

This corollary can be further improved to a strict inequality. In other words: star spanning trees are the unique minimizers of $\cap(T)$.

Corollary 8. *Let $G = (V, E) = K_n$ where $|V| = n > 4$ and let T_s be a star spanning tree of G. Then, the following property holds*

$$\cap(T_s) < \cap(T).$$

where T is any non-star spanning tree of G.

Proof. A careful reading of Proposition 6 leads to the conclusion that the equality $\cap_{T_s}(c) = \cap_T(c)$ is achieved when k is either $r_1 = 3$ or $r_2 = 2(n-2)$ (the roots of the quadratic polynomial). If $k = r_2 = 2(n-2)$, and taking into account that $3 \le k \le n$, we conclude that $\frac{7}{2} \le n \le 4$; this case is explicitly excluded from our hypotheses (in fact, it is not difficult to check that the three non-isomorphic spanning trees of K_4 have all the same tree intersection number).

The other possibility is $k = r_1 = 3$. As all the tree-cycles of T_s fall into this category, it is enough to show that T has a tree-cycle c such that $|c| = k > 3$ to conclude our thesis. Let $w \in V$ be a node with maximum degree in T. And let $d_T(w)$ denote the degree of w in T and $N(w)$ to the set of neighbors of w in T. Since T is a non-star spanning tree then $2 \le d_T(w) < n-1$. So there is a node $u \in V$ such that $u \notin N(w)$ in T, and there is a node $k \in N(w)$ such that $k \notin wTu$. Notice that the edge $e = (u, k) \notin T$ (as it would induce a cycle). Hence, it is a cycle-edge. Note that the tree-cycle induced by e has length at least 4. □

This result can be summarized in the following way: star spanning trees are the unique solutions for the MSTCI problem for complete graphs.

4. Further generalization

4.1. Overview

Now we explore some aspects of a slightly more general case, namely: the MSTCI problem in the context of a graph (not necessarily complete) $G = (V, E)$ that admits a star spanning tree T_s. In the first part we present a formula to calculate $\cap(T_s)$. In the second one we show that $\cap(T_s)$ is a local minimum in the domain of what we refer to as the "spanning tree graph". In the third we prove a result that suggests a general observation: the fact that a spanning tree of a graph G is a solution for the MSTCI problem does not depend on an intrinsic property of T but on the particular embedding of T in G. Finally we conjecture a generalization of Corollary 7: $\cap(T_s) \le \cap(T)$ for every spanning tree T of G.

4.2. Formulas for star spanning trees

In this subsection we present two formulas for graphs $G = (V, E)$ that admit a star spanning tree T_s. Let us denote $v \in V$ to the central node of T_s.

The first formula corresponds to the cycle intersection number of a tree-cycle $c = (u, v, w) \in C_{T_s}$, namely $\cap_{T_s}(c)$. Recall from the previous section that c intersects neither transit nor internal tree-cycles: $|\cap^t_{T,c}| = 0$ and $|\cap^i_{T,c}| = 0$. So its non-empty intersections are the tree-cycles in the set $\cap^e_{T,c}$. Note that the remaining incident edges to u and w, are the only source of tree-cycles that have non-empty intersection with c. So the formula is straightforward:

$$\cap_{T_s}(c) = d(u) - 2 + d(w) - 2.$$

where $d(u)$ and $d(w)$ are the degrees of u and w, resp.

Now we shall deduce a formula for the tree intersection number $\cap(T_s)$. Each edge $(u, v) \in T$ is contained in $d(u) - 1$ tree-cycles, choosing two of them gives all the possible intersections equal to (u, v). The formula is as follows:

$$\cap(T_s) = \sum_{u \in V - \{v\}} \binom{d(u)-1}{2} = \frac{1}{2} \sum_{u \in V - \{v\}} (d(u)-1)(d(u)-2) = \frac{1}{2} \sum_{u \in V - \{v\}} d(u)^2 - 3d(u) + 2.$$

If we denote $\mathbf{d}$ as the degree vector of G, that is, a vector that has in the ith component the degree of the ith vertex. And taking into account that $\sum_{u \in V} d(u) = 2m$ then, the formula can be expressed as:

$$\cap(T_s) = \frac{1}{2}[\|\mathbf{d}\|_2^2 - 6m - (n-1)(n-6)].$$

4.3. Star spanning tree as a local minimum

In this subsection we prove that a star spanning tree is a local minimum respect to the tree intersection number in the domain of the *spanning tree graph*. We start by defining this second order graph of the original one $G = (V, E)$. Then we analyze the structure of the neighbors of a star spanning tree T_s. Finally we demonstrate the result by a bijection between tree-cycles to conclude that $\cap(T_s)$ is a local minimum.

Definition 9. Let $G = (V, E)$ be a graph, and S a subgraph of G. We denote as $e \leftrightarrow e'$ to the operation of replacing the edge $e \in S$ with the edge $e' \in G - S$. We call this operation *edge replacement on S*.

Definition 10. Let $G = (V, E)$ be a graph. We denote SP_G to the graph that has one node for every spanning tree of G and an edge between two nodes, if the corresponding spanning trees differ in exactly one edge replacement. We call this graph the *spanning tree graph of G*.

4.3.1. Neighborhood of T_s

Let $G = (V, E)$ be a graph that admits a star spanning tree T_s with $v \in V$ as its center. Let α_{T_s} be the node corresponding to T_s in SP_G and let α_T (with corresponding spanning tree T) be any neighbor of α_{T_s}. By definition T_s and T differ in exactly one edge replacement $e \leftrightarrow e'$ where $e = (v, w) \in T_s$ and $e' = (u, w) \in T$. Note that T is exactly the same as T_s except that the node w is no longer connected to the central node v but to the intermediate node u. This similar structure has direct consequences in the intersection numbers of both trees.

Now we prove the result of this section.

Theorem 11. *Let $G = (V, E)$ be a graph that admits a star spanning tree T_s with $v \in V$ as its center. Then, T_s is a local minimum with respect to the tree intersection number in the domain of SP_G.*

Proof. Let T be a spanning tree corresponding to a neighbor of T_s in SP_G. We want to prove that $\cap(T_s) \le \cap(T)$. Therefore, we shall proceed by defining a bijection between the tree-cycles of both trees $\{c \leftrightarrow d : c \in C_{T_s} \wedge d \in C_T\}$ such that $\cap_{T_s}(c) \le \cap_T(d)$, this strategy clearly implies the thesis since by definition:

$$\cap(T_s) = \frac{1}{2}\sum_c \cap_{T_s}(c) \le \frac{1}{2}\sum_d \cap_T(d) = \cap(T).$$

Let $e_{T_s} \leftrightarrow e_T$ with $e_{T_s} = (v, w) \in T_s$ and $e_T = (u, w) \in T$ be the edge replacement in SP_G. Consider the following simple facts:

- e_T is a cycle-edge in T_s, with corresponding tree-cycle c.
- e_{T_s} is a cycle-edge in T, with corresponding tree-cycle d.
- Except for e_{T_s} and e_T, T_s and T have the same set of cycle-edges. For every $e \in E - T_s - T$ we denote c_e and d_e to the corresponding tree-cycles in T_s and T, resp.

According to this naming convention, we can define the following "natural" bijection between tree-cycles:

$$\{c \leftrightarrow d\} \cup \{c_e \leftrightarrow d_e : e \in E - T_s - T\}.$$

In order to compare the intersection numbers of the bijected pairs it is convenient to distinguish the following partition:

- Case 1: the pair induced by the edge replacement, $\{c \leftrightarrow d\}$.
- Case 2: pairs induced by cycle-edges non-incident to u nor to w, $\{c_e \leftrightarrow d_e : e \in E - T_s - T \wedge u \notin e \wedge w \notin e\}$.
- Case 3: pairs induced by cycle-edges incident to u or w, $\{c_e \leftrightarrow d_e : e \in E - T_s - T \wedge (u \in e \vee w \in e)\}$.

Case 1 is the easiest: note that c and d are the same tree-cycle (u, v, w), which is a transitless triangle, so its intersection number is determined by its external intersections:

$$\cap_{T_s}(c) = d(u) - 2 + d(w) - 2 = \cap_T(d).$$

Case 2 is similar, let $e = (h, k)$ be a cycle-edge non-incident to u or to w and $c_e \leftrightarrow d_e$ its corresponding pair of bijected tree-cycles. Clearly e determines the transitless triangle (h, v, k) both in T_s and T and as $d_{T_s}(h) = d_{T_s}(k) = d_T(h) = d_T(k) = 1$, then every other edge incident to h or k induces a tree-cycle that intersects (h, v, k). We conclude that:

$$\cap_{T_s}(c_e) = d(h) - 2 + d(k) - 2 = \cap_T(d_e).$$

Case 3 is the one that should be analyzed more carefully. As we already know how to calculate intersection numbers of tree-cycles in T_s, we will focus on the tree-cycles of T and divide this partition in two sub-partitions:

- Case 3.1: pairs induced by cycle-edges incident to u, $\{c_e \leftrightarrow d_e : e \in E - T_s - T \wedge u \in e\}$.
- Case 3.2: pairs induced by cycle-edges incident to w, $\{c_e \leftrightarrow d_e : e \in E - T_s - T \wedge w \in e\}$.

In case 3.1 the situation is as follows: the cycle-edge $e = (u, k)$ defines the tree-cycle $c_e = d_e = (u, v, k)$ (both in T and T_s). The important details are:

- $d_T(u) = 2$: u induces $d(u) - 3$ intersections.
- $d_T(k) = 1$: k induces $d(k) - 2$ intersections.
- $d_T(w) = 1$: w induces $d(w) - 1$ intersections.
- $d(w) \ge 2$ since it is connected at least to u and v in G.
- w may have an incident cycle-edge connecting it to k, so we should avoid counting twice that intersection.

Now we claim that

$$\cap_T(d_e) \ge d(u) - 3 + d(k) - 2 + d(w) - 1 - \epsilon(w, k) \ge d(u) - 2 + d(k) - 2 = \cap_{T_s}(c_e).$$

where

$$\epsilon(w, k) = \begin{cases} 1 & (w, k) \in E. \\ 0 & otherwise. \end{cases}$$

The inequality follows since $d(w) - 1 - \epsilon(w, k) \geq 1$.

In case 3.2 the situation is as follows: the cycle-edge $e = (w, h)$ defines the tree-cycle $d_e = (w, u, v, h)$ in T and $c_e = (w, v, h)$ in T_s. The important details are:

- $d_T(u) = 2$: u induces $d(u) - 2$ intersections.
- $d_T(h) = 1$: h induces $d(h) - 2$ intersections.
- $d_T(w) = 1$: w induces $d(w) - 2$ intersections.
- u may have an incident cycle-edge connecting it to h, so we should avoid counting twice that intersection.

And we claim that

$$\cap_T(d_e) \geq d(w) - 2 + d(h) - 2 + d(u) - 2 - \epsilon(u, h) \geq d(w) - 2 + d(h) - 2 = \cap_{T_s}(c_e).$$

The inequality follows since $d(u) - 2 - \epsilon(u, h) \geq 0$. □

4.4. Intrinsic tree invariants

In this subsection we consider the following question: is there any correlation between an *intrinsic tree invariant* and the tree intersection number of the spanning trees for every graph? If so we could formulate an alternative characterization of the MSTCI problem expressed in terms of the invariant.

By *intrinsic tree invariant* we denote a map $f : \mathscr{T} \to \mathbb{R}$ on the set of all trees. Of particular interest are the degree-based topological indices [6]. The topological index that motivated our question is the *atom-bond connectivity* (ABC) index [4]. As shown by [5] the star trees are maximal among all trees respect to the ABC index. In the previous section we proved that in the complete graph the star spanning trees are minimal respect to the tree intersection number. Consequently we can formulate a natural question: is there a negative correlation between the ABC index of the spanning trees and their corresponding intersection numbers?

We will prove that the answer to our question is negative. Without loss of generality we will consider positive correlation (negative correlation is analogous). The underlying idea of the proof is as follows: suppose that there exists an intrinsic tree invariant $f : \mathscr{T} \to \mathbb{R}$ such that for every graph G the intersection number $\cap(\cdot)$ is positively correlated with f. This can be expressed as:

$$f(T_1) \leq f(T_2) \iff \cap_G(T_1) \leq \cap_G(T_2), \forall G, T_1, T_2.$$

According to this property if we consider two trees T_1 and T_2 and two graphs G and H such that $T_1, T_2 \in \mathscr{T}_G$ and $T_1, T_2 \in \mathscr{T}_H$, then this equivalence follows:

$$\cap_G(T_1) \leq \cap_G(T_2) \iff \cap_H(T_1) \leq \cap_H(T_2).$$

So it suffices to show that there exist T_1, T_2, G and H such that the equivalence is not satisfied to answer the question negatively.

First we prove a simple lemma regarding the tree intersection number of a spanning tree T under the removal of a cycle-edge. Namely, if a cycle-edge e is removed from G then the tree intersection number of T decreases exactly in the intersection number of its corresponding tree-cycle.

Lemma 12. *Let $G = (V, E)$ be a graph, $T \in \mathscr{T}_G$ a spanning tree, $e \in G - T$ a cycle-edge, and c the corresponding tree-cycle, then the following holds:*

$$\cap_{G-e}(T) = \cap_G(T) - \cap_T(c).$$

Proof. As the spanning tree T is the same in both G and $G - e$, the remaining cycle-edges define the same tree-cycles so their pairwise intersection relations are identical. As c is not a cycle in $G - e$ then the equality follows. □

Theorem 13. *There is no intrinsic tree invariant $f : \mathscr{T} \to \mathbb{R}$ positively correlated with the intersection number $\cap_G(\cdot)$ for every graph G.*

Proof. We will proceed by contradiction: let f be such an intrinsic tree invariant. Then by definition for arbitrary graphs G and H the following equivalences hold

$$f(T_1) \leq f(T_2) \iff \cap_G(T_1) \leq \cap_G(T_2).$$

$$f(T_1) \leq f(T_2) \iff \cap_H(T_1) \leq \cap_H(T_2).$$

where $T_1, T_2 \in \mathscr{T}_G$ and $T_1, T_2 \in \mathscr{T}_H$. This in turn implies that

$$\cap_G(T_1) \leq \cap_G(T_2) \iff \cap_H(T_1) \leq \cap_H(T_2).$$

The proof will be based on showing two graphs and two spanning trees such that the latter equivalence is not valid.

- Let G be the complete graph K_n.
- Let H be the graph $K_n - \{e_{i,1}, \dots, e_{i,n-3}\}$. where the edges $e_{i,1}, \dots, e_{i,n-3}$ are $n-3$ edges incident to some arbitrary node v_i. We will refer to v_i as the *almost disconnected* node of H. Note that $d(v_i) = 2$.
- Let T_1 be the star spanning tree T_s.
- Let T_2 be the spanning tree defined as $T_s - \{e_i\} \cup \{e_{i,j}\}$, where e_i is the edge that connects some arbitrary node v_i (in H this role will be played by the almost disconnected node) to the center of the star and $e_{i,j}$ is an edge that connects v_i to a different node v_j.

It is easy to check that T_1 and T_2 are spanning trees of both G and H. If we also suppose that $|V| = n > 4$ then by Corollary 8

$$\cap_G(T_1) < \cap_G(T_2).$$

By the previous equivalence it is expected that $\cap_H(T_1) < \cap_H(T_2)$ as well. But we will show that this is not the case.

By a suitable labeling of the nodes of H we can refer to: the center of the star spanning tree as v_1, the almost disconnected node of H as v_2 and the other neighbor of v_2 as v_3. By Lemma 12 we have that

$$\cap_H(T_1) = \cap_{H-e_{2,3}}(T_1) + \cap_{T_1}(c_{2,3}).$$

$$\cap_H(T_2) = \cap_{H-e_{1,2}}(T_2) + \cap_{T_2}(c_{1,2}).$$

where $c_{2,3}$ and $c_{1,2}$ are the tree-cycles induced by $e_{2,3}$ and $e_{1,2}$ in T_1 and T_2, resp. The remaining tree-cycles corresponding to both trees are the same then

$$\cap_{H-e_{2,3}}(T_1) = \cap_{H-e_{1,2}}(T_2).$$

And this implies the following

$$\cap_H(T_1) - \cap_H(T_2) = \cap_{T_1}(c_{2,3}) - \cap_{T_2}(c_{1,2}).$$

It is an easy exercise to check that

$$\cap_{T_1}(c_{2,3}) = \cap_{T_2}(c_{1,2}) = d(v_3) - 2 = n - 3.$$

At this point we can conclude that

$$\cap_H(T_1) = \cap_H(T_2).$$

Contradicting the fact that f is positively correlated with the tree intersection number for every graph. □

The underlying key fact of this result is that a spanning tree T that solves the MSTCI problem for a graph G does not depend on intrinsic properties of T but on the embedding of T in G.

Note that as an interesting side effect this demonstration shows that a star spanning tree is not necessarily a strict local minimum in the spanning tree graph (see previous subsection).

4.5. Intersection number conjecture

In this subsection we present the conjecture $\cap(T_s) \le \cap(T)$ for every spanning tree T which generalizes Theorem 11. Then we explore two ideas to simplify a hypothetical counterexample of the conjecture. The first is based on the notion of *interbranch* cycle-edge. We show that if a non-star spanning tree T exists such that $\cap(T) < \cap(T_s)$, then the inequality must hold if we remove the interbranch cycle-edges. The second is based on the notion of *principal subtree*. In this case we show that the inequality must hold for some principal subtree of T. These ideas will be of practical use in the next section.

4.5.1. The conjecture statement

We present below the conjecture that generalizes the case of complete graphs.

Conjecture 14. *Let $G = (V, E)$ be a graph that admits a star spanning tree T_s, then*

$$\cap(T_s) \le \cap(T).$$

for every spanning tree $T \in \mathscr{T}_G$.

As an important remark, a demonstration of this result seems difficult if approached by a local-to-global strategy as in the complete graph case exposed previously.

4.5.2. Counterexample simplification

In this part we consider some ideas to simplify a hypothetical counterexample of Conjecture 14.

Below we define the notion of *interbranch* cycle-edge.

Definition 15. Let $G = (V, E)$ be a graph that admits a star spanning tree T_s and let $v \in V$ be the center of T_s. Let $T \in \mathscr{T}_G$ be a spanning tree. We call *interbranch cycle-edge of T* to any cycle-edge of T, $e = (u, w)$, such that $closest-point(v, c) \neq u, w$, where c is the induced tree-cycle of e in T.

The intuition behind this definition is that the paths vTu and vTw belong to different branches with respect to v, more precisely, $vTu \not\subset vTw$ and $vTw \not\subset vTu$. The following lemmas show that if we can find a counterexample to Conjecture 14 (i.e.: $\cap(T) < \cap(T_s)$) then we can build a simpler one removing the interbranch cycle-edges of T from G.

Lemma 16. *Let $G = (V, E)$ be a graph that admits a star spanning tree T_s with $v \in V$ as its center. Let $T \in \mathscr{T}_G$ be a spanning tree and $e = (u, w) \in \Delta_T$ an interbranch cycle-edge of T, then e is a cycle-edge of T_s.*

Proof. Since $v \neq u, w$ by definition of interbranch cycle-edge, then u and w are leaves of T_s and consequently e is a cycle-edge of T_s. □

Lemma 17. *Let $G = (V, E)$ be a graph that admits a star spanning tree T_s with $v \in V$ as its center. Let $T \in \mathscr{T}_G$ be a spanning tree and $e = (u, w) \in \Delta_T$ an interbranch cycle-edge of T, then*

$$\cap_{T_s}(c) \leq \cap_T(c').$$

where c and c' are the tree-cycles induced by e in T_s and T, resp.

Proof. By the intersection number formula we have that $\cap_{T_s}(c) = d(u) - 2 + d(w) - 2$.

In order to prove that $\cap_{T_s}(c) = d(u) - 2 + d(w) - 2 \leq \cap_T(c')$ we have to consider the set of neighbors of u and w in G. We will consider only the set $N(w)$ because the same argument is valid for $N(u)$. Below we will show that $N(w)$ contributes with at least $d(w) - 2$ tree-cycles to $\cap_T(c')$.

Let $h \neq u \in N(w)$ be the other neighbor of w in c'(i.e. $(h, w) \in c'$). Note that the edge $(h, w) \in vTw$ because the definition of interbranch cycle-edge requires that $closest - point(v, c') \neq w$. More concretely, h is the immediate predecessor of w in vTw. We intend to show that for every vertex $k \in N(w) - \{h, u\}$ there is a distinct tree-cycle in T with non-empty intersection respect to c', thus achieving the claimed bound. We will consider the following cases:

1. $vTw \subset vTk$.
2. $vTk \subset vTw$.
3. $vTw \not\subset vTk$ and $vTk \not\subset vTw$.

In the first case vTw is a subpath of vTk, then the edge (v, k) is a cycle-edge of T that determines a tree-cycle that contains the edge (h, w).

In the second case vTk is a subpath of vTw, then the edge (w, k) is a cycle-edge of T that determines a tree-cycle that contains the edge (h, w).

In the third case there is no proper inclusion between vTw and vTk, then the edge (w, k) is a cycle-edge of T that determines a tree-cycle that contains the edge (h, w).

To check that the tree-cycles induced in this way by $N(w)$ and $N(u)$ are all distinct, note that in cases 2 and 3 the corresponding cycle-edges are incident either to u or to w. And case 1 cannot occur simultaneously ($vTw \subset vTk$ and $vTu \subset vTk$) because $closest - point(v, c') \neq u, w$.

So the claimed inequality follows. □

Lemma 18. *Let $G = (V, E)$ be a graph that admits a star spanning tree T_s with $v \in V$ as its center. Let $T \in \mathscr{T}_G$ be a spanning tree such that $\cap_G(T) < \cap_G(T_s)$ and let Δ_T be the set of interbranch cycle-edges of T, then*

$$\cap_{G-\Delta_T}(T) < \cap_{G-\Delta_T}(T_s).$$

Proof. Let $e = (u, w) \in \Delta_T$, c and c' the tree-cycles induced by e in T_s and T, resp. By Lemma 12 the following holds:

$$\cap_{G-e}(T_s) = \cap_G(T_s) - \cap_{T_s}(c).$$

$$\cap_{G-e}(T) = \cap_G(T) - \cap_T(c').$$

and by Lemma 17 we have that:

$$\cap_{T_s}(c) \leq \cap_T(c').$$

we conclude that

$$\cap_{G-e}(T) = \cap_G(T) - \cap_T(c') < \cap_G(T_s) - \cap_{T_s}(c) = \cap_{G-e}(T_s).$$

Applying this edge removal for every edge in Δ_T, the claimed inequality follows. □

Definition 19. Let $T = (V, E)$ be a rooted tree graph with root $v \in V$. Let $w \in N(v)$ then we call *principal subtree with respect to w* to the subtree spanned by v and the nodes $u \in V$ such that $w \in vTu$.

The next lemma expresses the intersection number of a spanning tree (without interbranch cycle-edges) as the sum of the intersection number of its principal subtrees.

Lemma 20. *Let $G = (V, E)$ be a graph that admits a star spanning tree T_s with $v \in V$ as its center. Let T be a spanning tree of G without interbranch cycle-edges (i.e.: $\Delta_T = \varnothing$), then the following holds*

$$\cap_G(T) = \sum_{w \in N(v)} \cap_{G_w}(T_w).$$

where T_w is the principal subtree of $w \in N(v)$ considering T as a rooted tree with v as its root. And G_w is the subgraph spanned by T_w.

Proof. As $\Delta_T = \varnothing$ there are no cycle-edges connecting any two such principal subtrees. This implies that the non-empty intersections between tree-cycles of T must occur inside each subtree. This determines a partition of C_T and the claimed expression follows. □

The following corollary, in line with Lemma 18, further simplifies a hypothetical counterexample of Conjecture 14.

Corollary 21. *Let $G = (V, E)$ be a graph that admits a star spanning tree T_s with $v \in V$ as its center. Let T be a spanning tree of G without interbranch cycle-edges (i.e.: $\Delta_T = \varnothing$) such that $\cap(T) < \cap(T_s)$ then*

$$\cap(T_w) < \cap(G_w \wedge T_s).$$

for some G_w, where T_w is the principal subtree of $w \in N(v)$ considering T as a rooted tree with v as its root; G_w is the subgraph of G spanned by T_w; $G_w \wedge T_s$ is the subtree of T_s restricted to G_w, namely the intersection between G_w and T_s.

Proof. First note that the G_w's are edge disjoint since $\Delta_T = \varnothing$. This partition of the edges of G also determines a partition of T_s such that $\cap(T_s) = \sum_{w \in N(v)} \cap(G_w \wedge T_s)$. As the parts are in a natural bijective relation because they are the subtrees of T and T_s restricted to each G_w, we can express the intersection number of T and T_s as follows

$$\cap(T) = \sum_{w \in N(v)} \cap(T_w) < \sum_{w \in N(v)} \cap(G_w \wedge T_s) = \cap(T_s).$$

And from the bijection we can deduce that $\cap(T_w) < \cap(G_w \wedge T_s)$ for some G_w. □

5. Programmatic exploration

5.1. Overview

In this section we present some experimental results to reinforce Conjecture 14. We proceed by trying to find a counterexample based on our previous observations. In the first part, we focus on the complete analysis of small graphs, that is: graphs of at most 9 nodes. In the second part, we analyze larger families of graphs by random sampling instances.

5.2. General remarks

In the previous section we showed that the space of candidate counterexamples of Conjecture 14 can be reduced. The general picture is as follows:

- Let $G = (V, E)$ be a graph that admits a star spanning tree T_s with $v \in V$ as its center.
- In the case that we can find some non-star spanning tree T of G such that $\cap(T) < \cap(T_s)$ then, we can "simplify" the instance by removing the interbranch cycle-edges with respect to T in G without affecting the inequality (see Lemma 18).
- We can further reduce the instance by focusing on the case where $d_T(v) = 1$, that is: the degree of v restricted to T is 1 (see Corollary 21).

These considerations can be used to implement algorithms to explore the space of spanning trees more efficiently, since the algorithms will generate instances in this 'reduced' form instead of a brute force approach.

Table 1
Results for small instances.

Nodes	Instances (approx.)
4	5
5	33
6	251
7	4200
8	125000
9	7900000

5.3. Complete analysis of small graphs

In this subsection we present an algorithm to explore the spanning tree space. The algorithm proceeds by exhaustively analyzing all the reduced graphs of a given number of nodes. The size of the space increases exponentially with respect to the number of nodes, so it has a major limitation: it can only be used to analyze small graphs. The main part is sketched in Algorithm 1.

The details of the algorithm are the following:

- The input parameter n is the number of nodes of the graphs to explore.
- *GenerateAllTrees*$(n-1)$ is a function that returns the list of all trees of $n-1$ nodes.
- *GenerateGraph*(w, T') is a function that builds a graph G. Based on the tree T', it adds a new node (v), which will play the role of the central node of a star spanning tree, and then the edge (v, w), to define our candidate tree counterexample T. Finally adds all the other edges that link v to the rest of the nodes to obtain G. It returns the graph G and $(\bar{\Delta})$ the set of "possible" non-interbranch cycle edges.
- *IntersectionNumber*(ϕ, G) is a function that calculates the intersection number of T in $G \cup \phi$, where $\phi \subset \bar{\Delta}$ is a subset of supplementary edges of G.
- *StarIntersectionFormula*(ϕ, G) is a function that calculates the intersection number of the star spanning tree in $G \cup \phi$.
- The algorithm finds a counterexample of the conjecture if *IntersectionNumber*$(\phi, G) <$ *StarIntersectionFormula*(ϕ, G).

Note that the analyzed graphs are reduced in the sense previously explained. The cycle-edges are non-interbranch by construction and $d_T(v) = 1$ since v is only connected to w in T (i.e. there is a single principal subtree). As the algorithm iterates over all possible spanning subtrees T' and all the combinations of possible non-interbranch cycle-edges, every instance is guaranteed to be explored at least once.

Algorithm 1 CounterexampleSearch(n)

```
𝒯 ← GenerateAllTrees(n − 1)
for each tree T′ ∈ 𝒯 do
    for each node w ∈ T′ do
        G, Δ̄ ← GenerateGraph(w, T′)
        for each subset φ ⊂ Δ̄ do
            check (IntersectionNumber(φ, G) <
            StarIntersectionFormula(φ, G))
```

In order to generate all non-isomorphic trees of $|V|-1$ nodes, we used the package *nauty* [12].

The proposed algorithm did not find a counterexample of the intersection conjecture. Table 1 shows the size of the experiments. Column *Nodes* is the number of nodes of the graph family, i.e.: $|V|$. Column *Instances* is the number of instances processed.

5.4. Random sampling of large graphs

In this section we present another algorithm to explore the spanning tree space. The strategy in this case is to sample reduced graphs of a given number of nodes. The main part is sketched in Algorithm 2.

The details of the algorithm are the following:

- The input parameters are: n the number of nodes of the graphs and k the size of the sample.
- *GenerateRandomTree*(n) is a function that returns a random tree T of n nodes, where the node v, that will play the role of center of the star, has degree 1 restricted to T.
- *GenerateGraph*(T) is a function that builds a reduced graph G. Based on the tree T, it adds all the edges that link v to the rest of the nodes to obtain G. It returns the graph G and $(\bar{\Delta})$ a random set of non-interbranch cycle edges.

Table 2
Results for random instances.

Nodes	Instances
25	3000000
50	300000
100	30000
200	15000
400	300

- *IntersectionNumber*(ϕ, G) same as Algorithm 1.
- *StarIntersectionFormula*(ϕ, G) same as Algorithm 1.
- The algorithm finds a counterexample of the conjecture if: *IntersectionNumber*(ϕ, G) < *StarIntersectionFormula*(ϕ, G).

Algorithm 2 CounterexampleRandomSearch(n, k)

for `i := 1..k` **do**
  $T \leftarrow$ *GenerateRandomTree*(n)
  $G, \bar{\Delta} \leftarrow$ *GenerateRandomGraph*(T)
  check (*IntersectionNumber*(ϕ, G) <
  StarIntersectionFormula(ϕ, G))

We used a uniformly distributed random number generator. To generate trees we used a simple algorithm that randomly connects a new node to an already connected tree. The non-interbranch cycle-edge set is built by associating a *Bernoulli* trial to each possible edge. To achieve some diversity for each tree we built three different sets to obtain sparse, medium and dense ones based on corresponding probabilities 0.1, 0.5, 0.9.

The proposed algorithm did not find a counterexample of the intersection conjecture. Table 2 shows the size of the experiments. Column *Nodes* is the number of nodes of the graph family, i.e.: $|V|$. Column *Instances* is the number of instances processed.

6. Conclusion

In this article we introduced the *Minimum Spanning Tree Cycle Intersection* (MSTCI) problem.

We proved by enumerative arguments that the star spanning trees are the unique solutions of the problem in the context of complete graphs.

We conjectured a generalization to the case of graphs (not necessarily complete) which admit a star spanning tree. In this sense we showed that such tree is a local minimum in the domain of the *spanning tree graph*. We deduced a closed formula for the tree intersection number of star spanning trees in this setting. We proposed two ideas to reduce the search space of a counterexample of the conjecture. Those ideas were the basis of two strategies to programmatically explore the space of solutions in the pursuit of a counterexample. The negative result of the experiments suggests that the conjecture is well posed. Unlike the complete graph context, in this slightly more general case star spanning trees are not unique; there are other spanning trees T such that $\cap(T_s) = \cap(T)$.

We proved a general result that shows that spanning trees that solve the MSTCI problem do not depend on some intrinsic property but on their particular embedding in the ambient graph.

An interesting direction of research is to consider the MSTCI problem for other families of graphs, i.e.: graphs that do not admit a star spanning tree. Of particular interest for us is the class of triangular meshes, i.e.: graphs that model the immersion of compact surfaces in the 3D Euclidean space.

Another interesting line of research is to analyze the complexity class of the MSTCI problem. In case of belonging to the NP-hard class, it will be necessary to find approximate, probabilistic and heuristic algorithms.

In the introduction of this article we mentioned that the MSTCI problem is a particular case of finding a cycle basis with sparsest cycle intersection matrix. Another possible analysis would be to consider this in the context of the cycle basis classes described in [10].

Acknowledgments

The authors are grateful to the anonymous reviewers for their careful reading, constructive corrections and valuable comments on this paper, which have considerably improved its presentation.

References

[1] F. Berger, P. Gritzmann, S. de Vries, Minimum cycle bases for network graphs, Algorithmica 40 (2004) 51–62, http://dx.doi.org/10.1007/s00453-004-1098-x.
[2] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, B. Lévy, Polygon Mesh Processing, A. K. Peters, 2010.
[3] N. Deo, M.S. Krishnomoorthy, G. Prabhu, Algorithms for generating fundamental cycles in a graph, ACM Trans. Math. Software 8 (1) (1982) 26–42.
[4] E. Estrada, L. Torres, L. Rodriguez, I. Gutman, An atom-bond connectivity index: Modelling the enthalpy of formation of alkanes, Indian J. Chem. (1998) 849–855.
[5] B. Furtula, A. Graovac, D. Vukičević, Atom–bond connectivity index of trees, Discrete Appl. Math. 157 (13) (2009) 2828–2835, http://dx.doi.org/10.1016/j.dam.2009.03.004.
[6] I. Gutman, Degree-based topological indices, Croat. Chem. Acta 86 (4) (2013) 351–361, http://dx.doi.org/10.5562/cca2294.
[7] J. Horton, A polynomial-time algorithm to find the shortest cycle basis of a graph, SIAM J. Comput. 16 (2) (1987) 358–366.
[8] E. Hubicka, M.M. Sysło, Minimal bases of cycles of a graph, in: Recent Advances in Graph Theory, Academia Praha, 1975, pp. 283–293.
[9] T. Kavitha, C. Liebchen, K. Mehlhorn, D. Michail, R. Rizzi, T. Ueckerdt, K.A. Zweig, Cycle bases in graphs characterization, algorithms, complexity, and applications, Comp. Sci. Rev. 3 (4) (2009) 199–243, http://dx.doi.org/10.1016/j.cosrev.2009.08.001.
[10] C. Liebchen, R. Rizzi, Classes of cycle bases, Discrete Appl. Math. 155 (3) (2007) 337–355, http://dx.doi.org/10.1016/j.dam.2006.06.007.
[11] S. Maclane, A combinatorial condition for planar graphs, Fund. Math. 28 (1937) 22–32.
[12] B.D. McKay, A. Piperno, Practical graph isomorphism, II, J. Symbolic Comput. 60 (2014) 94–112, http://dx.doi.org/10.1016/j.jsc.2013.09.003.
[13] G.F. Stepanec, Basis systems of vector cycles with extremal properties in graphs, Uspekhi Mat. Nauk 19 (1964) 171–175 (in Russian).
[14] W.T. Tutte, Lecture on matroids, Math. Math. Phys. 69B (1 and 2) (1965).
[15] O. Veblen, An application of modular equations in analysis situs, Ann. Mat. 14 (1/4) (1912) 86, http://dx.doi.org/10.2307/1967604.
[16] A.A. Zykov, Theory of Finite Graphs, Nauka, Novosibirsk, 1969, in Russian.