DOI: https://doi.org/10.1016/j.procs.2023.08.232

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 223 (2023) 223–230

Procedia Computer Science

www.elsevier.com/locate/procedia

XII Latin-American Algorithms, Graphs and Optimization Symposium (LAGOS 2023)

Five edge-independent spanning trees

Alonso Ali[a,*], Orlando Lee[a]

[a]Institute of Computing, State University of Campinas, Campinas, Brazil

Abstract

Let G be a graph and let r be a fixed vertex of G. Two spanning trees T_1 and T_2 of G rooted at r are *edge-independent* if for every vertex $v \in V(G)$, the paths from v to r in T_1 and from v to r in T_2 are edge-disjoint. Itai and Zehavi conjectured that for every k-edge-connected graph and any vertex $r \in V(G)$ there are k edge-independent spanning trees rooted at r (Edge-Independent Spanning Trees Conjecture). Itai and Rodeh proved the case $k = 2$, Schlipf and Schmidt proved the case $k = 3$, and Hoyer and Thomas proved the case $k = 4$ of the conjecture. In this paper, we prove the case $k = 5$.

Keywords: edge-independent spanning trees; edge-connectivity; graph decomposition

1. Introduction

A graph G is *k-edge-connected* if for any subset F of $E(G)$ with $|F| < k$, $G - F$ is connected. By Menger's theorem [1], this is equivalent to the statement that for every pair of vertices x, y if G, there are k edge-disjoint paths from x to y in G.

Let G be a graph and let r be a fixed vertex of G. Two spanning trees T_1 and T_2 of G rooted at r are *edge-independent* if for every vertex $v \in V(G)$, the paths from v to r in T_1 and from v to r in T_2 are edge-disjoint. A set of spanning trees rooted at r is *edge-independent* if the trees are pairwise edge-independent.

In 1984, Itai and Rodeh [2] introduced the concept of edge-independent spanning trees and proved that every 2-edge-connected graph has two edge-independent spanning trees rooted at any vertex r. In 1989, Itai and Zehavi conjectured the following:

Conjecture 1.1 (Edge-Independent Spanning Trees Conjecture, Itai and Zehavi). *Let G be a k-edge-connected graph and let r be a vertex of G. Then G contains k edge-independent spanning trees rooted at r.*

The case $k = 3$ of the conjecture was proven by Schlipf and Schmidt [3] and the case $k = 4$ was proven by Hoyer and Thomas [4]. Both proofs rely on a result of Mader [5] that describes how to build iteratively any k-edge-connected

* Corresponding author. Tel: +55 51 99807-6037.
E-mail address: alonso.goncalves@ic.unicamp.br

1877-0509 © 2023 The Authors. Published by Elsevier B.V.

10.1016/j.procs.2023.08.232

graph starting from a small graph. Using Mader's theorem, Schlipf and Schmidt (respectively, Hoyer and Thomas) showed that every 3-edge-connected (respectively, 4-edge-connected) graph admits a certain chain decomposition (rooted at any vertex r) similar to an ear decomposition. These decompositions are then used to define an *edge numbering* of the graph and to construct the required number of edge-independent spanning trees. In this paper, we prove the case $k = 5$ of Itai-Zehavi's conjecture. We use a similar approach to the one used by Hoyer and Thomas. First, using Mader's Theorem we prove that every 5-edge-connected graph G admits a certain chain decomposition rooted at any vertex r, and then we use this decomposition to construct five edge-independent spanning trees of G rooted at r. Due to space constraints, we only sketch most of the proofs.

In this section, we define our version of chain decomposition. Let G be a 5-edge-connected graph and fix a vertex $r \in V(G)$. In what follows, we introduce the types of "chains" that appear in the decomposition, and then we formally define what a chain decomposition is. There are seven types of chains that appear in the decomposition; we note that the first three types of chains (up chain, down chain and one-way chain) are essentially the ones used in the chain decomposition of Hoyer and Thomas [4] and the other four types (singular forward, semi-singular forward, singular backward and semi-singular backward chains) are introduced in this paper and we call them *singular chains*. A singular chain consists of an edge that has an "orientation". As we show in Section 6, the union of all singular chains is one of the five edge-independent spanning trees that we construct.

A *decomposition* of a graph G is a sequence $(G_0, G_1, \ldots, G_m)$ of edge-disjoint subgraphs of G such that each edge of G belongs to exactly one G_i. Roughly speaking, a decomposition $(G_0, G_1, \ldots, G_m)$ of G is a *chain decomposition* if each G_i is a path or a cycle with certain connectivity constraints. In order to simplify the presentation, we describe each type of chain as a subgraph connecting $H := H_i = \bigcup_{j=0}^{i-1} G_j$ and $\overline{H} := \overline{H}_i = \bigcup_{j=i+1}^{m} G_j$.

For a vertex v, an edge incident to v is *non-valid* if it is a singular chain and v is its tail; otherwise, we say that such edge is a *valid edge* of v. This rather technical definition is necessary to guarantee that the five spanning trees are edge-independent. In the decomposition, every vertex other than the root r is the tail of exactly one singular chain.

Definition 1.2 (Up chain). Let $(H, \overline{H})$ be a pair of edge-disjoint subgraphs of G. An *up chain* of G with respect to $(H, \overline{H})$ is a subgraph of G, edge-disjoint from H and $\overline{H}$, which is

1. either a path with at least one edge in which every vertex is r or has at least two valid edges in $\overline{H}$, and its endpoints are r or have at least one valid edge in H,
2. or a cycle in which every vertex is r or has at least two valid edges in $\overline{H}$, and some vertex v is r or has at least two valid edges in H. In this case, we call v the *endpoint* of the up chain and the remaining vertices its *internal vertices*.

Definition 1.3 (Down chain). Let $(H, \overline{H})$ be a pair of edge-disjoint subgraphs of G. A *down chain* of G with respect to $(H, \overline{H})$ is an up chain of G with respect to $(\overline{H}, H)$.

Definition 1.4 (One-way chain). Let $(H, \overline{H})$ be a pair of edge-disjoint subgraphs of G. A *one-way chain* of G with respect to $(H, \overline{H})$ is a subgraph of G induced by an edge $e = uv \notin E(H \cup \overline{H})$ such that u is r or has at least two valid edges in H, and v is r or has at least two valid edges in $\overline{H}$. We call u the *tail* and v the *head* of the one-way chain and both are endpoints of it.

Definition 1.5 (Singular forward chain). Let $(H, \overline{H})$ be a pair of edge-disjoint subgraphs of G. A *singular forward chain* of G with respect to $(H, \overline{H})$ is a subgraph of G induced by an edge $e = uv \notin E(H \cup \overline{H})$ such that u is r or has at least two valid edges in H and at least two valid edges in $\overline{H}$, and v is r or has at least two valid edges in $\overline{H}$. We call u the *tail* and v the *head* of the singular forward chain and both are endpoints of it.

Definition 1.6 (Semi-singular forward chain). Let $(H, \overline{H})$ be a pair of edge-disjoint subgraphs of G. A *semi-singular forward chain* of G with respect to $(H, \overline{H})$ is a subgraph of G induced by an edge $e = uv \notin E(H \cup \overline{H})$ such that u is r or has at least one valid edge in H, and v is r or has at least two valid edges in $\overline{H}$. We call u the *tail* and v the *head* of the semi-singular forward chain and both are endpoints of it.

Definition 1.7 (Singular backward chain). Let $(H, \overline{H})$ be a pair of edge-disjoint subgraphs of G. A *singular backward chain* of G with respect to $(H, \overline{H})$ is a subgraph of G induced by an edge $e = uv \notin E(H \cup \overline{H})$ such that u is r or has at least two valid edges in H and at least two valid edges in $\overline{H}$, and v is r or has at least two valid edges in H. We call u the *tail* and v the *head* of the singular backward chain and both are endpoints of it.

Definition 1.8 (Semi-singular backward chain). Let $(H, \overline{H})$ be a pair of edge-disjoint subgraphs of G. A *semi-singular backward chain* of G with respect to $(H, \overline{H})$ is a subgraph of G induced by an edge $e = uv \notin E(H \cup \overline{H})$ such that u is r or has at least one valid edge in $\overline{H}$, and v is r or has at least two valid edges in H. We call u the *tail* and v the *head* of the singular backward chain and both are endpoints of it.

For the remainder of the text, we refer to these structures as chains. Vertices in chains that are not its endpoints are called its *internal vertices*. We refer to a chain as a *singular chain* if it is one of the chains described from Definitions 1.5 to 1.8. We call a chain *short* if it is a one-way chain or a singular chain.

Let $(G_0, G_1, \ldots, G_m)$ be a sequence of subgraphs of G. For i in $\{0, 1, \ldots, m\}$, let $H_i = G_0 \cup \ldots \cup G_{i-1}$ and $\overline{H}_i = G_{i+1} \cup \ldots \cup G_m$. Note that H_0 and $\overline{H}_m$ are empty graphs. A *chain decomposition of G rooted at r* is a decomposition $(G_0, G_1, \ldots, G_m)$ of G such that

- every subgraph G_i is a chain with respect to $(H_i, \overline{H}_i)$,
- every vertex other than r is the tail of exactly one singular chain.

Note that G_0 is either a closed up chain with endpoint r or a one-way chain with tail r. Similarly, G_m is either a closed down chain with endpoint r or a one-way chain with head r.

An up chain G_i is *minimal* if none of its internal vertices are in $V(H_i) \cup \{r\}$. Analogously, a down chain G_i is minimal if none of its internal vertices are in $V(\overline{H}_i) \cup \{r\}$. A chain decomposition is *minimal* if every up chain and down chain in it is minimal. A non-minimal up or down chain can be divided into minimal chains by breaking it at its offending internal vertices, as long as the resulting chains are inserted sequentially in the decomposition and substituting the divided chain; the resulting sequence of chains also gives rise to a chain decomposition. This remark is relevant since it allows us to suppose that any chain decomposition is minimal; if it is not, then we can apply the previous operation.

Another important property is the *symmetry* of the chain decomposition. If $D = (G_0, G_1, \ldots, G_m)$ is a chain decomposition of G rooted at r, then so is $D' = (G_m, G_{m-1}, \ldots, G_0)$. Moreover,

- every up chain in D is a down chain in D' and vice versa,
- every one-way chain in D is a one-way chain in D' with the roles of its head and tail swapped,
- every (semi-)singular backward chain in D is a (semi-)singular forward chain in D' with the same head and tail, and vice versa.

We now state the two main results of this paper which imply the case $k = 5$ of the Edge-Independent Spanning Trees Conjecture.

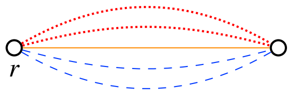
Theorem 1.9. *Let G be a 5-edge-connected graph and let $r \in V(G)$. Then G has a chain decomposition rooted at r.*

Theorem 1.10. *Let G be a 5-edge-connected graph and let $r \in V(G)$. If there is a chain decomposition of G rooted at r, then there are five edge-independent spanning trees of G rooted at r.*

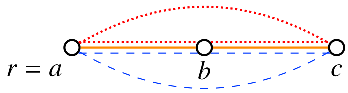
2. Proof of Theorem 1.9

We present an outline of the proof of Theorem 1.9. In 1978, Mader proved that any k-edge-connected graph can be constructed as follows: starting with a k-edge-connected graph with two or three vertices, we apply repeatedly a sequence of operations called Mader operations [5]. A Mader operation always preserves k-edge-connectivity, i.e., if a graph G is k-edge-connected, then so is a graph resulting from an application of a Mader operation to G. In what follows, we describe the operations specifically for the case $k = 5$.

Definition 2.1. Let G be a 5-connected graph. A *Mader operation* on G is one of the following three procedures applied on G:



(a) The chain decomposition of K_2^5. Red dotted edges are in the up chain G_0, the orange straight edge is the singular forward chain G_1 with tail different than r, and blue dashed edges are in the down chain G_2.



(b) The chain decomposition of K_3^5. Red dotted edges are in the up chain G_0, the orange straight edges are singular forward chains G_1 and G_2 with tail c and b, respectively, and blue dashed edges are in the down chain G_3.

Fig. 1: The chain decomposition of K_2^5 and K_3^5

- Add an edge between any two vertices of G.
- Let $e_1 = xy$ and $e_2 = wz$ be distinct edges of $E(G)$ and let $u \in V(G)$. Remove e_1 and e_2 from $E(G)$, add a new vertex v and add the edges xv, yv, wv, zv and uv. We say that u was *pinched* and we refer to this operation as *pinch* for the rest of the text.
- Let $e_1 = xy$ and $e_2 = wz$ be distinct edges of $E(G)$. Remove e_1 and e_2 from $E(G)$, add a new vertex v and add the edges xv, yv, wv and zv; let G' denote the resulting graph. Now select two distinct edges $e_3 = ab$ and $e_4 = cd$, not all incident to v. Remove e_3 and e_4 from $E(G')$, add a new vertex v' and add the edges av', bv', cv' and dv'. Finally, add an edge between the two new vertices v and v'. We refer to this operation as *double pinch* for the rest of the text.

Mader made the following claim. Let K_2^5 be the graph shown in Figure 1a and let K_3^5 be the graph shown in Figure 1b. Every 5-edge-connected graph can be obtained through repeated applications of Mader operations starting from K_2^5 or K_3^5 [5].

We use the same approach of Schlipf and Schmidt [3] for $k = 3$, and Hoyer and Thomas [4] for $k = 4$. First, we show that both graphs K_2^5 and K_3^5 have a chain decomposition. Next, we show that the application of a Mader operation on a graph with a chain decomposition results in a graph that also has a chain decomposition. This implies that every 5-edge-connected graph has a chain decomposition. Figures 1a and 1b illustrate the chain decomposition of K_2^5 and K_3^5, respectively.

To prove Theorem 1.9, it suffices to show that if G has a chain decomposition D, then the graph G' obtained by an application of a Mader operation has a chain decomposition D'. To achieve this, we show that a chain decomposition D' of G' rooted at r can be obtained from D through local modifications, depending on the applied Mader operation and which chains of D were affected by the (double) pinch operations.

The proof of Theorem 1.9 is divided into three sections, one dedicated to each Mader operation. The proof is done by a case-by-case analysis for every possible combination of types of chains being pinched. Due to space constraints, we omit the proofs, but we outline the main ideas. The following lemma is useful in our proof.

Lemma 2.2. *Let $v \in V(G)$. If v has less than two valid edges in H_i, then v has at least two valid edges in $\overline{H}_i$. Analogously, if v has less than two valid edges in $\overline{H}_i$, then v has at least two valid edges in H_i.*

In the next sections, we use the following approach. We assume that we have a chain decomposition $D = (G_0, G_1, \ldots, G_m)$ of a graph G and then we apply a Mader operation to G to obtain a new graph G'. We describe how we modify D to obtain a chain decomposition D' of G'. The modifications consist of inserting one or more new chains in some position or exchanging one or more chains of D for new ones in D'.

3. Edge addition

Suppose an edge was added between vertices x and y. Let i be the smallest index of a chain in D containing x or y. Without loss of generality, assume that $x \in V(G_i)$.

If G_i is an up chain, then x is an internal vertex since there are no previous chains containing it. Insert xy as a one-way chain with tail x immediately after G_i to obtain a chain decomposition D' of G'.

If G_i is not an up chain, it must be a one-way chain or a singular forward chain with head x. In any case, x is incident to two edges in $\overline{H}_i$. Let $G_{i'}$ be the chain with the smallest index $i' > i$ containing x. If $G_{i'}$ is an up chain, then

(a) Edges xy, wz and vertex u in G, before the pinch operation.

(b) Result of the pinch operation.

Fig. 2: Example of the pinch operation on edges xy, wz and vertex u.

x is one of its endpoints since every chain is minimal. If $G_{i'}$ is not an up chain, it is a one-way chain or a singular forward chain with head x, so it has two edges in $\overline{H}_{i'}$. Note that x also has two edges in $H_{i'+1}$. Now consider vertex y. Lemma 2.2 guarantees that y has two edges in $H_{i'}$ or in $\overline{H}_{i'}$. If y has two edges in $H_{i'}$, insert xy as a one-way chain with tail y immediately after $G_{i'}$ in D'; otherwise, insert it as a one-way chain with tail x.

4. Pinch

Let $e_1 = xy$ and $e_2 = wz$ be the edges and u the vertex being pinched. We denote by v the new vertex and by e_x, e_y, e_w, e_z and e_u the new edges connecting v to the respective vertices. Let G_i be the chain containing xy and let G_j be the chain containing wz. Henceforth, without loss of generality, we assume that $i \leq j$. Let P_x be the path from an endpoint of G_i to x such that xy is not in P_x. Define P_y, P_z and P_w analogously. Figure 2 illustrates the pinch operation.

We analyze now all possible types of chains that G_i and G_j can be. For example, if G_i is an up chain, then every vertex in the chain has two valid edges in later chains and the endpoints have a valid edge in earlier chains, by the definition of an up chain. Since the pinch operation removes the edge xy from the graph and adds the edges e_x and e_y, we modify the chain G_i so the chain decomposition remains valid. Note that $P_x e_x e_y P_y$ is similar to the chain G_i in D, with the exception that the vertex v is added between x and y. Note as well that it is also an up chain: the endpoints have a valid edge in earlier chains – since P_x and P_y remain intact, they are the same as G_i in D –, every vertex other than v has two valid edges in later chains – since every vertex other than v was in the up chain G_i in D–, and v has two valid edges that may be added in later chains (e_u, e_w and e_z). This example illustrates the idea of the proof. Knowing the type of the chain that the deleted edges were in, we are able to use that information to insert the new edges e_x, e_y, e_w, e_z and e_u into old or new chains making only local changes in the decomposition.

Although many details are omitted, each possible case follows the outline of the example: e_x and e_y are used to "fix" the chain G_i broken by the pinch, e_w and e_z are used to "fix" G_j and e_u is generally added as a one-way chain or a singular forward or singular backward chain. In the next section, we show that one of the five trees we want is built from the property that every vertex in $V(G) - \{r\}$ is the tail of exactly one singular chain; so we must ensure that the new chain decomposition satisfies this property as well. Since the pinch and the double pinch operations are the only operations that create a new vertex, it is necessary to make one of the new edges e_x, e_y, e_w, e_z or e_u a singular chain such that v is its tail in every case. This suffices to guarantee this property.

5. Double pinch

Let us recall the double pinch operation. Let $e_1 = xy$ and $e_2 = wz$ be distinct edges of $E(G)$. Remove e_1 and e_2 from $E(G)$, add a new vertex v and add the edges xv, yv, wv and zv in G'. After this, select two distinct edges $e_3 = ab$ and $e_4 = cd$, not all incident to v. Remove e_3 and e_4 from $E(G')$, add a new vertex v' and add the edges av', bv', cv' and dv'. After this, add an edge between the two new vertices v and v'. This operation results in a graph with two new vertices, each with five new edges incident to it. Figure 3 illustrates the edges incident to each new vertex of this operation.

This part of the proof follows the same strategy as the last section. We modify the decomposition of the graph prior to the pinch to accommodate every new edge in a new chain. Note that a double pinch operation can be described as two applications of a pinch operation but without adding the edge uv, and then adding an edge between the new

Fig. 3: Double pinch operation applied to edges xy, wz, ab and cd in G.

vertices v and v'. So although a double pinch operation is more complex than a pinch operation, we can modify the chains and insert the new edges using the same kind of strategy. Due to space constraints, we omit these details.

6. Constructing the five trees

In this section, we outline the proof of Theorem 1.10. Assume that we have a chain decomposition of G (rooted at r) and let us show how to produce five edge-independent spanning trees rooted at r. We define the *chain index* $CI(e)$ of an edge e as the index of the chain containing e.

First, using the chain decomposition we define two partial edge numberings f and g that are then used to assign edges to subgraphs T_1, T_2, T_3, T_4, and T_5; afterward, we show that those subgraphs are five edge-independent spanning trees. Trees T_1 and T_2 are associated with the edge numbering f, while T_3 and T_4 are associated with the edge numbering g. For each numbering, we show that the pair of trees associated with it have different "trends" regarding the paths leading to r. One of the trees follows a strictly decreasing path with respect to the edge numbering while the other follows a strictly increasing path with respect to the same numbering. From this, it is straightforward to show that both trees from the same edge numbering are edge-independent. To show that any two trees created from different numberings are edge-independent, we argue that the difference in the numbering procedure guarantees that no same edge is chosen when assigning them to one of the trees. The fifth tree is built in a different and novel way, by taking all the edges which are part of a singular chain. This is the motivation for the requirement that every vertex other than r must be the tail of a singular chain.

Let us provide some details of the proof. We define two edge-numberings f and g as follows. The numbering f assigns real values to edges in up chains, one-way chains, singular forward chains, and semi-singular forward chains, while g assigns real values to edges in down chains, one-way chains, singular backward chains, and semi-singular backward chains. Note that edges in one-way chains are numbered by both f and g. We refer to *consecutive edges* in a chain as two edges that are incident to the same internal vertex of the chain. We note that the two edges incident to an endpoint of a closed chain are not consecutive, since the endpoint is not an internal vertex of the chain.

For each vertex v distinct from r, we call *f-edges of v* the two edges incident to v with the two smallest chain indexes. Analogously, we call *g-edges of v* the two edges incident to v with the two largest chain indexes. We note that an edge in a down chain can never be an f-edge since it would mean that v has two edges incident to it with a lower chain index. Similarly, an edge in an up chain can never be an g-edge.

We now state the numbering process of f. Let $(G_0, G_1, \ldots, G_m)$ be a chain decomposition of G rooted at r. We remark that f only numbers the edges in up chains, one-way chains, and (semi-)singular forward chains. First, we number the edges of G_0, and then proceed to G_1, and so on. The numbering procedure for the edges in a chain G_i is described next; one important detail which we omit in the description is that no two edges are assigned the same value by f (this is possible because we assign real values).

- If G_i is a closed up chain containing r, then number each consecutive edge in the chain so that the values increase monotonically, starting with an edge incident to r. The specific values do not matter as long as it keeps the monotone property.

- If G_i is a closed up chain not containing r, then note that both f-edges of the endpoint are already numbered since they are in earlier chains of the decomposition. Let α and β be the values assigned by f to those edges. Number the edges in $E(G_i)$ consecutively with monotonically increasing values between α and β.

- If G_i is an open up chain containing r, then one of the endpoints of the chain G_i is r and the other is some vertex $v \neq r$. Note that at least one f-edge of v is already numbered with some value, say α. Number the edges in $E(G_i)$ consecutively with monotonically increasing values greater than α from v to r.

- If G_i is an open up chain not containing r, then let x and y be the endpoints of the chain. Note that at least one f-edge of x and one f-edge of y are already numbered. Let α and β be the values assigned by f to those edges, respectively. Without loss of generality, assume that $\alpha < \beta$. Number the edges in $E(G_i)$ consecutively with monotonically increasing values between α and β from x to y.

- If G_i is a one-way chain with tail r, then number the edge in G_i arbitrarily.

- If G_i is a one-way chain with tail $v \neq r$, then both f-edges of v are already numbered. Let α and β be the values assigned by f to these edges. Number the edge in G_i with a value between α and β.

- If G_i is a singular forward chain with tail v, then both f-edges of v are already numbered. Let α and β be the values assigned by f to these edges of the chain. Number the edge in G_i with a value between α and β.

- If G_i is a semi-singular forward chain with tail v, then an f-edge of v is already numbered. If the head of G_i does not have a valid edge in an earlier chain, then let α be the value assigned by f to the f-edge of v. Number the edge in G_i with a value greater than a. Otherwise, if the head of G_i has a valid edge in a previous chain, then let β be the value assigned to it by f and number the edge in G_i with a value between α and β.

This concludes the numbering procedure f. The g-numbering is analogous to the f-numbering, but applied to the reversed chain decomposition.

Now, let us describe how the subgraphs T_1, T_2, T_3, T_4, and T_5 are built. For every vertex $v \neq r$, consider its two f-edges and assign the edge with the smallest f-numbering to T_1 and the other one to T_2. Similarly, consider its two g-edges and assign the one with the smallest g-numbering to T_3 and the other one to T_4. It follows from Lemma 6.1 that all edges added to T_1 are distinct; moreover, there are no cycles in T_1, and a path in T_1 towards r is (non-necessarily monotonically) decreasing with respect to the chain indexes and strictly decreasing with respect to the f-values of its edges. From these observations and the fact that T_1 is created by choosing an edge from every vertex other than r, it follows that T_1 is a spanning tree of G rooted at r.

Lemma 6.1. *For any $v_1 \neq r$, consider the edge e_1 incident to v_1 and assigned to the subgraph T_1. Let v_2 be the other end of the edge e_1. If $v_2 \neq r$, let e_2 be the edge incident to v_2 and assigned to T_1. Then $CI(e_2) \leq CI(e_1)$ and $f(e_2) < f(e_1)$.* □

Although omitted, the same argument used to prove Lemma 6.1 can be used to prove that a path in T_2 towards r is (non-necessarily monotonically) decreasing with respect to the chain indexes and monotonically *increasing* with respect to the f-values of its edges. Hence, the spanning trees T_1 and T_2 are edge-independent due to the different trends of f-values in paths towards r.

Analogously, Lemma 6.1 can be replicated to show that T_3 and T_4 are spanning trees of r and are edge-independent because paths to the root in both of them are non-strictly increasing in chain index, but are strictly decreasing and increasing in g-values, respectively. So, it has been shown that T_1 and T_2 are pair-wise edge-independent and so are T_3 and T_4. To show that T_1 is edge-independent with T_3 and T_4, note that the paths to the root in T_1 are decreasing in chain index while paths to the root in T_3 and T_4 are increasing, but not strictly. To see that an f-edge and an g-edge of the same vertex are never in the same chain, remember that edges in down chains can not be f-edges and edges in up chains can not be g-edges. The remaining possibilities are chains induced by a single edge, so an f-edge can not have the same chain index as an g-edge of the same vertex. This concludes the argument that T_1 is edge-independent with T_3 and T_4 and the same argument can be made to show that it is true that T_2 is edge-independent with T_3 and T_4.

It remains to show that T_5 is a spanning tree of G and pairwise edge-independent with $\{T_1, T_2, T_3, T_4\}$. Since T_5 is created by assigning exactly one edge from every vertex other than r to it, it is a spanning subgraph of G. To see that T_5 is a tree, consider the proof of Theorem 1.9 where every new vertex created with a pinch or a double pinch is assigned to be the tail of a singular chain, and whenever a singular chain is removed from the graph a new one is assigned to the former tail. Intuitively, every pinch or double pinch is always 'hanging' new vertices as leaves in T_5.

This guarantees that for every chain decomposition, a spanning tree is induced by its singular chains. Finally, note that the tail of a singular chain has two edges in previous chains and two edges in later chains. This means that a singular chain is never an f- or an g-edge of its tail. Hence, T_5 is edge-independent in relation to $\{T_1, T_2, T_3, T_4\}$ since every edge incident to a vertex $v \neq r$ and assigned to T_1, T_2, T_3 or T_4 is different from the edge assigned to T_5. This concludes the proof of Theorem 1.10.

7. Conclusions

We were able to show that every 5-edge-connected graph has a chain decomposition and that five edge-independent spanning trees can be constructed from a chain decomposition. This answers the case $k = 5$ of the Edge-Independent Spanning Trees Conjecture.

8. Acknowledgments

This work is funded by grant 2020/11118-6, São Paulo Research Foundation (FAPESP) and grant 2015/11937-9, São Paulo Research Foundation (FAPESP).

References

[1] K. Menger, Zur allgemeinen Kurventheorie, Fundamenta Mathematicae 10 (1) (1927) 96–115. URL http://eudml.org/doc/211191

[2] A. Itai, M. Rodeh, The multi-tree approach to reliability in distributed networks, Inf. Comput. 79 (1) (1988) 43–59. doi:10.1016/0890-5401(88)90016-8.

[3] L. Schlipf, J. M. Schmidt, Edge-orders, Algorithmica 81 (5) (2019) 1881–1900. doi:10.1007/s00453-018-0516-4.

[4] A. Hoyer, R. Thomas, Four edge-independent spanning trees, SIAM Journal on Discrete Mathematics 32 (1) (2018) 233–248. doi:10.1137/17M1134056.

[5] W. Mader, A reduction method for edge-connectivity in graphs, in: B. Bollobás (Ed.), Advances in Graph Theory, Vol. 3 of Annals of Discrete Mathematics, Elsevier, 1978, pp. 145–164. doi:10.1016/S0167-5060(08)70504-1.