

k-Trails: recognition, complexity, and approximations

Journal Article**Author(s):**

Singh, Mohit; Zenklusen, Rico

Publication date:

2018-11

Permanent link:

<https://doi.org/10.3929/ethz-b-000128414>

Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

Originally published in:

Mathematical Programming 172(1), <https://doi.org/10.1007/s10107-017-1113-z>

Funding acknowledgement:

165866 - New Approaches to Constrained Submodular Maximization (SNF)

k -Trails: recognition, complexity, and approximations

Mohit Singh¹ · Rico Zenklusen² 

Received: 29 April 2016 / Accepted: 16 January 2017 / Published online: 25 January 2017
© Springer-Verlag Berlin Heidelberg and Mathematical Optimization Society 2017

Abstract The notion of degree-constrained spanning hierarchies, also called k -trails, was recently introduced in the context of network routing problems. They describe graphs that are homomorphic images of connected graphs of degree at most k . First results highlight several interesting advantages of k -trails compared to previous routing approaches. However, so far, only little is known regarding computational aspects of k -trails. In this work we aim to fill this gap by presenting how k -trails can be analyzed using techniques from algorithmic matroid theory. Exploiting this connection, we resolve several open questions about k -trails. In particular, we show that one can recognize efficiently whether a graph is a k -trail, and every graph containing a k -trail is a $(k + 1)$ -trail. Moreover, further leveraging the connection to matroids, we consider the problem of finding a minimum weight k -trail contained in a graph G . We show that one can efficiently find a $(2k - 1)$ -trail contained in G whose weight is no more than the cheapest k -trail contained in G , even when allowing negative weights. The above results settle several open questions raised by Molnár, Newman, and Sebő.

Keywords Matroids · Approximation algorithms · k -Trails

Mathematics Subject Classification 90C27

A short version of this paper appeared in the proceedings of IPCO 2016.

Rico Zenklusen was supported by Swiss National Science Foundation Grant 200021_165866.

✉ Rico Zenklusen
ricozen@math.ethz.ch
Mohit Singh
mohits@microsoft.com

¹ Microsoft Research, Redmond, WA, USA

² ETH Zurich, Zurich, Switzerland

1 Introduction

Motivated by applications in network routing, the notion of *degree-constrained spanning hierarchies* was introduced as a way to obtain lower-degree routing structures than what could be obtained with low-degree spanning subgraphs [8, 9]. These hierarchies, which, for brevity, have also simply been called *k-trails* [10, 13], describe how a given graph can be described as the homomorphic image of another low-degree graph. More precisely, the motivation for studying *k-trails* comes from network routing contexts where routing protocols with “branchings” of low degree are desirable. One simple routing possibility to achieve this, is to route messages over spanning trees with small degrees. This was a main motivation for the study of degree-constrained spanning trees (see [4, 11, 14, 15] and references therein). The idea of *k-trails* is to allow for more flexibility than a simple spanning tree protocol, while still maintaining a tree-like routing topology with branchings of low degree. Loosely speaking, a *k-trail* can be interpreted as a routing protocol described by a tree, where every edge corresponds to an edge of the original graph, but vertices of the original graph may appear several times. Hence, the possibility of using vertices multiple times leads to an additional flexibility compared to degree-bounded spanning trees. For more information, we refer the interested reader to [8, 9]. First results have been reported that show advantages of *k-trails* compared to traditional methods in network routing contexts [9].

However, many basic questions around *k-trails* remained open, like whether one can efficiently decide if a graph is a *k-trail*; we refer the interested reader to [13] for a nice overview of some open problems around *k-trails*. The goal of this work is to fill this gap by revealing and exploiting a connection to matroids.

Before giving a summary of our main results, we start by formally defining *k-trails* as well as some closely related notions. In particular, the notion of homomorphic images introduced below is the basis for defining *k-trails*. Throughout this paper, we focus on undirected connected graphs with possibly loops and parallel edges, and with at least 2 vertices to avoid trivial special cases.

Definition 1 (*Homomorphic image*) A graph $G = (V, E)$ is the *homomorphic image* of a graph $H = (W, F)$ if there is an onto function $\phi : W \rightarrow V$ such that for any two vertices $u, v \in V$ (with possibly $u = v$), the number of edges in G between u and v is equal to the number of edges in H whose endpoints get mapped by ϕ to $\{u, v\}$.

We would like to highlight that, contrary to some other definitions of graph homomorphisms found in the literature, our definition of a homomorphic image H of G preserves edge-multiplicities, and thus induces a bijection between the edges of G and H . Figure 1 shows an example graph and two homomorphic preimages of it.

In other words, a preimage $H = (W, F)$ of G corresponds to a graph obtained from G by splitting each of its vertices $v \in V$ into $|\phi^{-1}(v)|$ many copies. We therefore call $|\phi^{-1}(v)|$ the ϕ -multiplicity, or simply *multiplicity*, of v .

Definition 2 (*k-Trail*) A graph $G = (V, E)$ is a *k-trail* if it is the homomorphic image of a connected graph $H = (W, F)$ with maximum degree at most k .

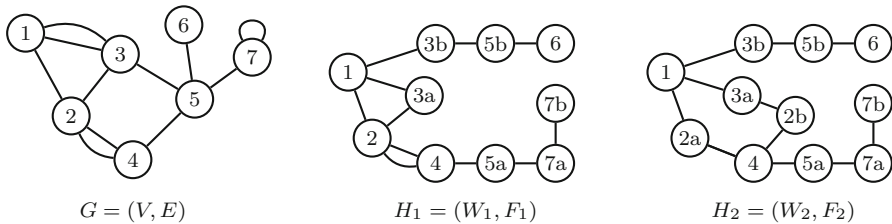


Fig. 1 The graph G is the homomorphic image of H_1 as well as H_2 . The naming of the vertices has been chosen to highlight the corresponding homomorphisms, e.g., the nodes $3a$ and $3b$ in H_1 are both mapped to node 3 in G . G is a 3-trail, because the homomorphic preimage H_2 has maximum degree 3

The graph G shown in Fig. 1 is a 3-trail since it is the homomorphic image of H_2 , which has maximum degree 3. It is not hard to see that k -trails can equivalently be defined as preimages of trees of degree at most k .

Definition 3 (k -Tree) A graph is a k -tree if it is a tree of maximum degree at most k .

Lemma 1 If G is a k -trail then it is a homomorphic image of a k -tree. More precisely, given a connected graph $H = (W, F)$ and onto function $\phi : W \rightarrow V$ such that G is the homomorphic image of H by ϕ , we can construct efficiently a tree $H' = (W', F')$ and onto function $\phi' : W' \rightarrow W$ such that H is the homomorphic image of H' by ϕ' . Thus, G is the $\phi \circ \phi'$ -homomorphic image of H' , and for $v \in V$, the $\phi \circ \phi'$ -multiplicity of v is at least the ϕ -multiplicity of v .

Proof Let G be a homomorphic image of the connected graph $H = (W, F)$ with maximum degree k . We show how vertices of H can be split step-by-step to arrive at H' . We only show a single splitting step transforming H into $\tilde{H} = (\tilde{W}, \tilde{F})$. Assume that H contains a cycle, say C ; otherwise we can set $H' = H$. Let $\{w_1, w_2\}$ be an edge in C . Let $\tilde{W} = W \cup \{x\}$ where x is a new vertex. Let $\tilde{H} = (\tilde{W}, \tilde{F})$ where $\tilde{F} = F \cup \{\{x, w_1\}\} \setminus \{\{w_1, w_2\}\}$ and $\tilde{\phi} : \tilde{W} \rightarrow W$ where $\tilde{\phi}(w) = w$ if $w \in W$ and $\tilde{\phi}(x) = w_2$. Clearly \tilde{H} is connected with maximum degree k and H is a homomorphic image of \tilde{H} by $\tilde{\phi}$. Since \tilde{H} has one more vertex than H , repeatedly applying this procedure will stop as soon as we split G into a connected graph with $|E| + 1$ vertices, in which case it has to be a tree H' . \square

Prior to this work, little was known regarding computational questions linked to k -trails. In particular, it was unknown whether one can efficiently determine if a graph is a k -trail for a given k , an open question raised in [13]. A nice result shown in [10, 13] is that every 2-edge-connected graph is a 3-trail. Further interesting open questions on k -trails that are motivated by routing applications are linked to the notion of whether a graph contains a k -trail, which is defined as follows.

Definition 4 (Containing a k -trail) We say that a graph $G = (V, E)$ contains a k -trail if there is a set $U \subseteq E$ such that $G' = (V, U)$ is a k -trail.

Notice that all k -trails are connected graphs, since they are homomorphic images of connected graphs. Hence, candidate edge sets $U \subseteq E$, for $G = (V, E)$ to contain a k -trail (V, U) , must be such that (V, U) is connected.

Also for the containment of k -trails, little was known prior to this work. In particular, it was conjectured in [13] that for any nonnegative edge weights $w : E \rightarrow \mathbb{Z}_{\geq 0}$ and any $k \geq 3$, there exists a polynomial algorithm returning a $(2k - 2)$ -trail in G whose cost is not larger than the minimum weight k -trail in G , if G contains a k -trail.

In this paper we are able to settle most of the above-mentioned open questions, by presenting a new viewpoint on k -trails in terms of matroids.

1.1 Our results

One of our main results, whose derivation will also be used to highlight a strong link between k -trails and matroids, is the fact that k -trails can be recognized efficiently.

Theorem 1 *Given a graph $G = (V, E)$ and $k \in \mathbb{Z}_{>0}$, one can check efficiently whether G is a k -trail. Moreover, if G is a k -trail, one can find efficiently a connected graph $H = (W, F)$ with degrees bounded by k and an onto function $\phi : W \rightarrow V$ such that G is the homomorphic image of H by ϕ .*

Contrary to the recognition problem, the containment problem is hard. Our hardness proof can be interpreted as a natural extension of a hardness proof shown in [8], for a weighted version of the problem. For completeness, we provide a full proof in Sect. 3.

Theorem 2 *For any $k \in \mathbb{Z}_{\geq 2}$, the problem of deciding whether a graph contains a k -trail is NP-complete.*

Despite the different complexity status of the containment and recognition question, the following theorem shows that they are closely related.

Theorem 3 *If G contains a k -trail then it is a $(k + 1)$ -trail.*

Using that recognition is polynomial time solvable, we can thus find the smallest k for which a given graph G is a k -trail, which then implies by Theorem 3 that the smallest k' for which G contains a k' -trail is either k or $k - 1$. Finally, we obtain the following result on the containment of weighted k -trails.

Theorem 4 *There exists a polynomial time algorithm that, given a graph $G = (V, E)$ with weight function $w : E \rightarrow \mathbb{Z}$ and an integer $k \geq 2$, either shows that there is no k -trail contained in G or returns a $(2k - 1)$ -trail contained in G whose total weight is at most the weight of any k -trail contained in G .*

Theorem 4 almost resolves a conjecture in [13], claiming that one can efficiently find a $(2k - 2)$ -trail in G of weight no more than the weight of any k -trail contained in G , assuming $k \geq 3$ and nonnegativity of the weights. Our result only implies the existence of a cheap $(2k - 1)$ -trail; however, it holds for arbitrary weights, and not just nonnegative ones. Furthermore, we can show that, even for nonnegative weights, the factor $2k - 1$ is optimal when comparing to a natural LP relaxation. More precisely, for any $k \geq 3$ there is a graph G such that the natural LP relaxation is feasible for k , i.e., there is a “fractional k -trail”, whereas G does not contain any $(2k - 2)$ -trail. Hence, a different approach is needed to resolve the conjecture on the existence of cheap $(2k - 2)$ -trails.

1.2 Organization of paper

Section 2 proves Theorem 1 and shows how k -trails can be studied using tools from algorithmic matroid theory. Section 3 discusses the hardness proof for Theorem 2 as well as an algorithm implying Theorem 3. In Sect. 3.1, we prove Theorem 4. Section 3.2 shows an integrality gap example with respect to a natural LP for finding a light k -trail contained in a given graph.

1.3 Basic terminology

The degree of a vertex $v \in V$ in a graph $G = (V, E)$ is denoted by $\deg_G(v)$, or simply $\deg(v)$ if there is no danger of confusion. If the graph is clear from context, we will also use the notation $\deg_U(v) := |\delta(v) \cap U|$ for a set $U \subseteq E$ and $v \in V$.

2 Recognition of k -trails

Let $G = (V, E)$ be an undirected graph and suppose we want to show that G is a k -trail for k as small as possible. Consider some connected graph $H = (W, F)$ such that G is the homomorphic image of H by some onto function $\phi : W \rightarrow V$. Let $v \in V$ and consider all vertices of H that get mapped to v , i.e., $\phi^{-1}(v) = \{w_1, \dots, w_\ell\}$, where $\ell = |\phi^{-1}(v)|$ is the multiplicity of v . Clearly, we have

$$\deg_G(v) = \sum_{i=1}^{\ell} \deg_H(w_i).$$

Knowing that v gets split into ℓ vertices by ϕ , for degrees to be low in H it would be best if all w_i for $i \in [\ell]$ have about the same degree. It turns out that starting with any H and corresponding ϕ , we can balance out the degrees of vertices in H that correspond to the same vertex in G , using a simple modification algorithm. The following lemma formalizes this statement.

Lemma 2 *Given two connected graphs $G = (V, E)$ and $H = (W, F)$, and an onto function $\phi : W \rightarrow V$ such that G is the homomorphic image of H , one can determine in polynomial time a connected graph $H' = (W, F')$ such that*

- (i) G is the homomorphic image of H' by ϕ .
- (ii) For any $v \in V$ and $w \in W$ such that $\phi(w) = v$, the degree of w in H' is either $\left\lfloor \frac{\deg_G(v)}{|\phi^{-1}(v)|} \right\rfloor$ or $\left\lceil \frac{\deg_G(v)}{|\phi^{-1}(v)|} \right\rceil$.

Proof Notice that condition (ii) of the lemma is equivalent to the property that for every $v \in V$ and $w, w' \in \phi^{-1}(v)$, we have $|\deg_{H'}(w) - \deg_{H'}(w')| \leq 1$. Suppose that $H = (W, F)$ does not satisfy this property. Thus, for some $v \in V$ and $w, w' \in \phi^{-1}(v)$, we have $\deg_H(w) \geq 2 + \deg_H(w')$. Consider an arbitrary w - w' path in H , and let u be a neighbor of w not lying on this path. Since the degree of w is at least three, such a neighbor always exists. Now, we construct a $\hat{H} = (W, \hat{F})$ where we let

$\hat{F} = F \cup \{\{u, w'\}\} \setminus \{\{u, w\}\}$. It is straightforward to see that \hat{H} is connected and G is also a homomorphic image of \hat{H} . We now repeat this operation as long as we can find such a pair of vertices. Observe that $\sum_{u \in W} \deg_{\hat{H}}(u)^2 < \sum_{u \in W} \deg_H(u)^2$ and thus a potential argument implies that the sequence is polynomially bounded. Thus, the graph obtained at the end of the sequence, $H' = (W, F')$, satisfies the conditions of the lemma. \square

We leverage the above lemma to rephrase the problem of whether a graph is a k -trail in terms of multiplicities.

Definition 5 (*Feasible multiplicity vector*) Let $G = (V, E)$ be a graph. A vector $\lambda \in \mathbb{Z}_{>0}^V$ is a *feasible multiplicity vector* (for G) if G is the homomorphic image of a connected graph with multiplicities given by λ ; more formally, if there is a connected graph $H = (W, F)$ such that G is the homomorphic image of H by some onto function $\phi : W \rightarrow V$, and $|\phi^{-1}(v)| = \lambda(v) \ \forall v \in V$.

Feasible multiplicity vectors fulfill the following down-monotonicity property.

Lemma 3 Let G be a graph and $\lambda \in \mathbb{Z}_{>0}^V$ be a feasible multiplicity vector. Then any vector $\lambda' \in \mathbb{Z}_{>0}^V$ with $\lambda' \leq \lambda$ (component-wise) is also a feasible multiplicity vector.

Furthermore, this result is constructive: Given a connected graph H and homomorphism ϕ such that G is the ϕ -homomorphic image of H and λ is the multiplicity vector corresponding to ϕ , we can efficiently construct for any $\lambda' \leq \lambda$ a connected graph H' and homomorphism ϕ' such that G is the ϕ' -homomorphic image of H' with corresponding multiplicity vector λ' .

Proof Given a feasible vector $\lambda \in \mathbb{Z}_{>0}^V$ with $\lambda(v) \geq 2$, we show that $\bar{\lambda}$ is also a feasible vector where $\bar{\lambda}(v) = \lambda(v) - 1$ and $\bar{\lambda}(u) = \lambda(u)$ for each $u \in V \setminus \{v\}$. The lemma then follows from induction.

Let $H = (W, F)$ and $\phi : W \rightarrow V$ certify the feasibility of λ . Since $\lambda(v) \geq 2$, there exists $w_1, w_2 \in \phi^{-1}(v)$ such that $w_1 \neq w_2$. Let $H' = (W', F')$ be obtained by merging w_1 and w_2 into a new vertex w . Thus $W' = W \cup \{w\} \setminus \{w_1, w_2\}$. Let $\phi' : W' \rightarrow V$ where $\phi'(w) = v$ and $\phi'(x) = \phi(x)$ for each $x \in W' \setminus \{w\}$. Then G is a homomorphic image of H' by ϕ' and H' certifies the feasibility of $\bar{\lambda}$, thus proving the lemma. \square

Lemmas 2 and 3 easily imply that the question of whether G is a k -trail for some given k can be reduced to the problem of deciding whether some multiplicity vector $\lambda \in \mathbb{Z}_{>0}^V$ is feasible.

Lemma 4 A graph $G = (V, E)$ is a k -trail if and only if the following multiplicity vector $\lambda \in \mathbb{Z}_{>0}^V$ is feasible:

$$\lambda(v) = \left\lceil \frac{\deg_G(v)}{k} \right\rceil \quad \forall v \in V.$$

Proof Let G be a k -trail witnessed by H and ϕ . Since H has maximum degree k , we must have $|\phi^{-1}(v)| \geq \left\lceil \frac{\deg_G(v)}{k} \right\rceil$ for all $v \in V$, proving the only if direction.

Now, let λ be feasible where $\lambda(v) = \left\lceil \frac{\deg_G(v)}{k} \right\rceil$ for each $v \in V$. From Lemma 2, it follows that G is a homomorphic image of a connected graph $H = (W, F)$ by $\phi : W \rightarrow V$ such that $\deg_H(w) \leq \left\lceil \frac{\deg_G(v)}{\lambda(v)} \right\rceil$ for each $w \in \phi^{-1}(v)$. Since $\left\lceil \frac{\deg_G(v)}{\lambda(v)} \right\rceil \leq k$, the lemma follows. \square

To finally provide an efficient recognition algorithm to decide whether a graph is a k -trail, we show that feasible multiplicity vectors are highly structured.

Notice that a feasible multiplicity vector is at least 1 in each coordinate. For simplicity, we introduce a shifted version of feasible multiplicity vectors, called *feasible split vector*; a vector $\mu \in \mathbb{Z}_{\geq 0}^V$ is a feasible split vector if $\mu + \mathbf{1}$ is a feasible multiplicity vector, where $\mathbf{1} \in \mathbb{Z}^V$ is the all-ones vector. Hence, a split vector tells us how many times a vertex is split.

Theorem 5 *Let G be an undirected graph. The set of feasible split vectors corresponds to the integral points of a polymatroid,¹ i.e.,*

$$P_G := \text{conv}(\{\mu \in \mathbb{Z}_{\geq 0}^V \mid \mu \text{ is a feasible split vector}\}),$$

is a polymatroid. Furthermore, we can efficiently optimize over P_G , and for any feasible split vector $\mu \in \mathbb{Z}_{\geq 0}^V$ we can efficiently find a connected graph $H = (W, F)$ and an onto function $\phi : W \rightarrow V$ such that G is the homomorphic image of H , and $|\phi^{-1}(v)| = \mu(v) \forall v \in V$.

Before proving the theorem, we start with a few observations and show that Theorem 5 implies Theorem 1. It is well-known that P_G being a polymatroid implies that P_G is given by

$$P_G = \{x \in \mathbb{R}_{\geq 0}^V \mid x(S) \leq f(S) \forall S \subseteq V\},$$

where $f : 2^V \rightarrow \mathbb{Z}_{\geq 0}$ is the submodular function defined by

$$f(S) = \max\{x(S) \mid x \in P_G\} \quad \forall S \subseteq V.$$

Many results on polymatroids typically assume that a polymatroid is given through a value oracle for the function f . Clearly, if we can efficiently optimize over P_G , we can also evaluate efficiently the submodular function f , which, as described above, corresponds to maximizing a $\{0, 1\}$ -objective over P_G .

We are particularly interested in checking whether some split vector $\mu \in \mathbb{Z}_{\geq 0}^V$ is feasible. Having an efficient evaluation oracle for f allows for checking whether $\mu \in P_G$ by standard techniques: It suffices to solve the submodular function minimization problem $\min\{f(S) - x(S) \mid S \subseteq V\}$; if the optimal value is negative then $\mu \notin P_G$, otherwise $\mu \in P_G$. We will later see that the link between k -trails and matroids that

¹ A polymatroid over a finite set N is a polytope $P \subseteq \mathbb{R}_{\geq 0}^N$ described by $P = \{x \in \mathbb{R}_{\geq 0}^N \mid x(S) \leq f(S) \forall S \subseteq N\}$, where $f : 2^N \rightarrow \mathbb{Z}_{\geq 0}$ is a submodular function, and $x(S) = \sum_{v \in S} x_v$. We refer the interested reader to [12, Volume B] for more information on polymatroids.

we establish implies an easy way to check whether $\mu \in P_G$ using a simple matroid intersection problem (without relying on submodular function minimization).

Combining the above results and observations, Theorem 1 easily follows.

Proof of Theorem 1 Let $\lambda \in \mathbb{Z}_{>0}^V$ be defined as in Lemma 4, and let $\mu = \lambda - \mathbf{1}$. Lemma 4—rephrased in terms of μ —states that G is a k -trail if and only if μ is a feasible split vector, which can be checked efficiently by Theorem 5. Furthermore, if μ is feasible, then Theorem 5 also shows that we can efficiently obtain a connected graph $\bar{H} = (W, \bar{F})$ and onto function $\phi : W \rightarrow V$ such that

- (i) G is the homomorphic image of \bar{H} by ϕ , and
- (ii) $|\phi^{-1}(v)| = \mu(v) + 1 \quad \forall v \in V$.

By Lemma 2 we can balance the degrees of \bar{H} for each vertex set $\phi^{-1}(v)$ efficiently, to obtain a connected graph $H = (W, F)$ with balanced degrees as stated in Lemma 2. It remains to observe that all degrees in H are bounded by k . This indeed holds: let $w \in W$ and $v = \phi(w)$; we thus obtain

$$\deg_H(w) \leq \left\lceil \frac{\deg_G(v)}{|\phi^{-1}(v)|} \right\rceil = \left\lceil \frac{\deg_G(v)}{\mu(v) + 1} \right\rceil = \left\lceil \frac{\deg_G(v)}{\left\lceil \frac{\deg_G(v)}{k} \right\rceil} \right\rceil \leq \left\lceil \frac{\deg_G(v)}{\frac{\deg_G(v)}{k}} \right\rceil = k ,$$

where the first inequality follows by Lemma 2 and the second equality by $\mu(v) + 1 = \lceil \deg_G(v)/k \rceil$. \square

2.1 Matroidal description of k -trails and proof of Theorem 5

We start by introducing an auxiliary graph $G' = (V', E')$ such that spanning trees in G' can be interpreted as graphs H such that G is a homomorphic image of H . Using this connection, we then derive that P_G is a polymatroid over which we can optimize efficiently, and show how to construct a homomorphic preimage of G corresponding to some split vector μ , as claimed by Theorem 5.

Hence, let $G = (V, E)$ be an undirected graph. The graph $G' = (V', E')$ contains a vertex for each of the two endpoints of each edge in E . More formally, for each $v \in V$, the Graph G' contains $\deg_G(v)$ many vertices $V'_v := \{v_e\}_{e \in \delta(v)}$; hence,

$$V' = \bigcup_{v \in V} V'_v.$$

We note that describing V'_v by $\{v_e\}_{e \in \delta(v)}$ is a slight abuse of notation, since for each loop at v we include two vertices in V'_v and not just one. Furthermore,

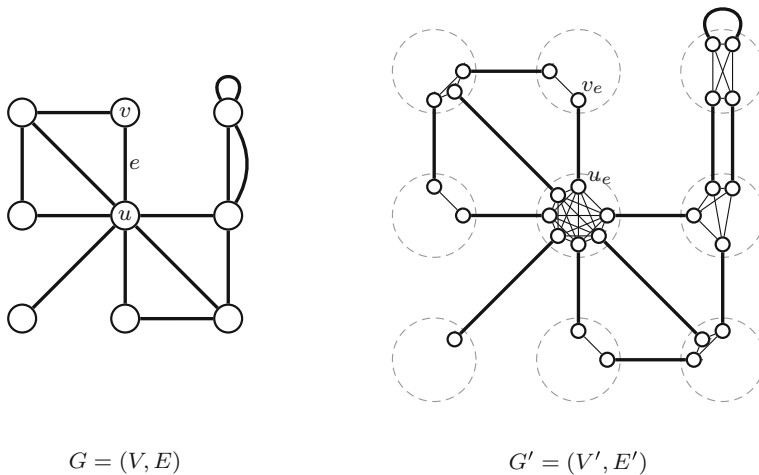


Fig. 2 An example for the construction of the auxiliary graph $G' = (V', E')$ from $G = (V, E)$. In G' the thick edges correspond to edges in \bar{E} and the thin ones to edges in K . The dashed gray circles correspond to the cliques (V'_v, K_v) and highlight the link between the vertices $v \in V$ in G and vertex sets V'_v in G' which correspond to v

$$\begin{aligned}
 E' &= \bar{E} \cup K, \quad \text{where} \\
 \bar{E} &= \{\{v_e, u_e\} \mid e = \{u, v\} \in E\}, \quad \text{and} \\
 K &= \bigcup_{v \in V} K_v, \quad \text{where} \\
 K_v &= \{\{v_e, v_f\} \mid v_e, v_f \in V'_v, v_e \neq v_f\} \quad \forall v \in V.
 \end{aligned}$$

See Fig. 2 for an example of the above construction.

For any spanning tree $T \subseteq E'$ in G' that contains \bar{E} , we define a graph $H_T = (W_T, F_T)$ and an onto function $\phi_T : W_T \rightarrow V$, such that G is the homomorphic image of H_T by ϕ , as follows. For each $v \in V$ consider the connected components of $(V'_v, K_v \cap T)$. Let q_v be the number of these connected components and let

$$V'_v = C_v^1 \cup C_v^2 \cup \dots \cup C_v^{q_v}$$

be the partition of V'_v into vertex sets of the q_v connected components in $(V'_v, K_v \cap T)$.

We now define $H_T = (W_T, F_T)$ as the graph obtained from (V', T) by contracting all C_v^j for $v \in V$ and $j \in [q_v] := \{1, \dots, q_v\}$. For clarity, we call the vertices in H_T nodes. Contracting C_v^j corresponds to replacing C_v^j with a single node, which we identify with the set C_v^j for simplicity, thus leading to the following set of nodes for H_T :

$$W_T = \{C_v^j \mid v \in V, j \in [q_v]\}.$$

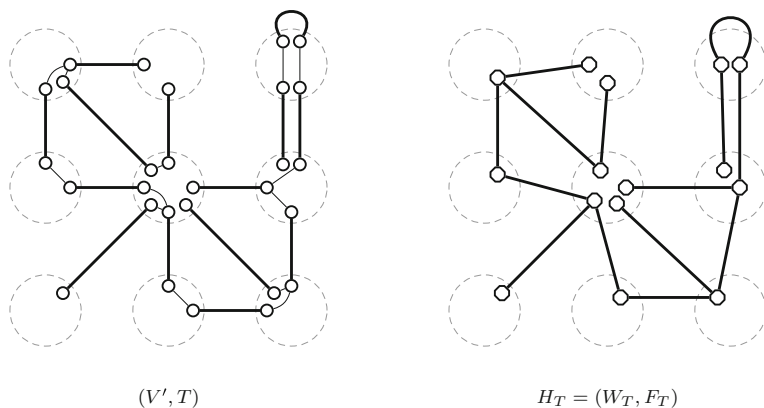


Fig. 3 On the *left-hand side*, a spanning tree T in the auxiliary graph $G' = (V', E')$ is highlighted, which corresponds to the graph G shown in Figure 2. On the *right-hand side*, the corresponding graph H_T is shown. The homomorphism $\phi_T : W_T \rightarrow V$ maps all vertices of H_T within the same dashed circle to the same vertex of G . For $v \in V$, the number q_v is equal to the number of vertices on the *right-hand side* lying inside the *dashed circle* corresponding to v

Furthermore, two nodes $C_v^j, C_w^\ell \in W_T$ are adjacent in H_T if and only if there is a pair of vertices, one in C_v^j and one in C_w^ℓ , that are connected by an edge in $T \cap \bar{E}$; formally, this corresponds to the existence of $e \in E$ such that the edge in \bar{E} that corresponds to e is contained in T , and $v_e \in C_v^j$ and $w_e \in C_w^\ell$. Moreover, $\phi_T : W_T \rightarrow V$ is defined by

$$\phi_T(C_v^j) = v \quad \forall C_v^j \in W_T.$$

Figure 3 shows an example construction of H_T from a spanning tree T that contains \bar{E} .

We start with some observations that follow immediately from the above construction:

- G is the homomorphic image of H_T by ϕ_T .
- The multiplicity of $v \in V$ is q_v .
- G' can be constructed efficiently from G .
- Several spanning trees T can lead to the same graph H_T and homomorphism ϕ_T ; indeed, for any component C_v^j , the edges of T with both endpoints in C_v^j form a spanning tree over C_v^j , which can be replaced by any other spanning tree without changing H_T or ϕ_T .

Conversely, every description of G as a homomorphic image of a tree can be obtained by the above construction:

Claim Let $H = (W, F)$ be a tree and $\phi : W \rightarrow V$ be an onto function such that G is the homomorphic image of H by ϕ . Then there exists a spanning tree T in G' such that $H = H_T$ and $\phi = \phi_T$.

Proof of claim The spanning tree T can be chosen as follows. Starting with $T = \emptyset$, we first add to T all edges in \bar{E} . For each $w \in W$, we do the following: Let $v = \phi(w)$, and let $e_1, \dots, e_h \in E$ be the ϕ -images of the edges $\delta_H(w)$; in particular, $e_1, \dots, e_h \in \delta_G(v)$. We add to T an arbitrary set of $h - 1$ edges of G' that form a spanning tree over the vertices v_{e_1}, \dots, v_{e_h} . One can now easily observe that the constructed tree T has the desired properties. \square

The equivalence between (ii) and (iii) in the following statement summarizes the above discussion. The equivalence between (i) and (ii) follows from Lemma 1 (implying (i) \Rightarrow (ii)) and Lemma 3 (implying (ii) \Rightarrow (i)). Furthermore, we highlight that the equivalences in Property 1 are all constructive.

Property 1 Let $G = (V, E)$ be an undirected graph and $\mu \in \mathbb{Z}_{\geq 0}^V$. The following three statements are equivalent:

- (i) G is the homomorphic image of a connected graph $H = (W, F)$ by an onto function $\phi : W \rightarrow V$ such that $|\phi^{-1}(v)| = \mu(v) + 1 \forall v \in V$.
- (ii) G is the homomorphic image of a tree $H = (V, W)$ by an onto function $\phi : W \rightarrow V$ such that $|\phi^{-1}(v)| \geq \mu(v) + 1 \forall v \in V$.
- (iii) There is a spanning tree T in the auxiliary graph $G' = (V', E')$ such that $\bar{E} \subseteq T$ and $|T \cap K_v| \leq |\delta(v)| - 1 - \mu(v) \forall v \in V$.

Using the above connection between the auxiliary graph G' and homomorphic preimages of G , Theorem 5 can now be derived as follows. For brevity, we use the following notation. For any spanning tree T in G' such that $\bar{E} \subseteq T$, we define $\alpha_T \in \mathbb{Z}_{\geq 0}^V$ by $\alpha_T(v) := |T \cap K_v| \forall v \in V$. Furthermore, let $\deg \in \mathbb{Z}_{\geq 0}^V$ be the degree vector of G , i.e., $\deg(v)$ is the degree of v as usual. The equivalence between point (i) and point (iii) of Property 1 can thus be rephrased as follows.

$$\mu \in \mathbb{Z}_{\geq 0}^V \text{ is a feasible split vector} \Leftrightarrow \begin{array}{l} \exists \text{ spanning tree } T \text{ in } G' \text{ with } \bar{E} \subseteq T \\ \text{and } \alpha_T \leq \deg - \mathbf{1} - \mu, \end{array} \quad (1)$$

where $\mathbf{1} \in \mathbb{Z}^V$ is the all-ones vector.

The equivalence highlighted by (1) directly leads to an efficient way to check whether a given vector $\mu \in \mathbb{Z}_{\geq 0}^V$ is a feasible split vector, and if so, obtain a homomorphic preimage of G that certifies it. Indeed, finding a spanning tree T in G' that contains \bar{E} and satisfies $\alpha_T \leq \deg - \mathbf{1} - \mu$ is a matroid intersection problem. More precisely, the task is to find a spanning tree in G'/\bar{E} (the graph G' after contracting \bar{E})—such spanning trees are the bases of the graphic matroid on G'/\bar{E} —whose edges are simultaneously an independent set in the partition matroid on the partition $K = \cup_{v \in V} K_v$, requiring that no more than $\deg(v) - 1 - \mu(v)$ edges are selected within K_v for each $v \in V$. If this matroid intersection problem has a solution, then we get the desired spanning tree T fulfilling the conditions of point (iii) in Property 1, which is equivalent to point (i), and this equivalence is constructive, thus leading to the desired homomorphic preimage H of G that corresponds to the split vector μ .

In particular, to check whether G is a k -trail, we know by Lemma 4 that G is a k -trail if and only if the split vector $\mu \in \mathbb{Z}_{\geq 0}^V$ given by $\mu(v) = \left\lceil \frac{\deg(v)}{k} \right\rceil - 1$ is feasible.

Together with (1), this leads to the following characterization of G being a k -trail in terms of a matroid intersection problem.

Corollary 1 *The following two statements are equivalent:*

- (i) G is a k -trail.
- (ii) *There exists a spanning T in the auxiliary graph G' such that*

$$|T \cap K_v| \leq \deg(v) - \left\lceil \frac{\deg(v)}{k} \right\rceil \quad \forall v \in V.$$

We finish by proving the claims about P_G in Theorem 5. For this, we start by observing that the vectors α_T are integral base vectors of a polymatroid.

Lemma 5 *The polytope*

$$\bar{B}_G = \text{conv}(\{\alpha_T \mid T \text{ is a spanning tree in } G' \text{ with } \bar{E} \subseteq T\})$$

is the base polytope of a polymatroid. Furthermore, we can optimize efficiently over \bar{B}_G .

Proof Let $M = (K, \mathcal{T})$ be the graphic matroid obtained by starting with the graphic matroid on G' and contracting the edges \bar{E} . Notice that the ground set of M is indeed K , i.e., all edges with both endpoints in one of the set V'_v for $v \in V$. Spanning trees T in G' that contain \bar{E} are precisely the bases of M . Thus the vectors α_T are obtained by taking a basis in M , considering the partition $K = \cup_{v \in V} K_v$ of K , and counting the number of elements that T has in each part of this partition. This way of constructing vectors out of a matroid is often called the aggregation of a matroid (see, e.g., [3, page 54]), and is well known to lead to the integral points of a base polytope of a polymatroid (see [3] and [12, Volume B]). This shows that \bar{B}_G is indeed a polymatroid base polytope. Furthermore, optimization over \bar{B}_G can simply be done by optimizing over M , using the greedy algorithm. More precisely, for a weight function $w \in \mathbb{Z}^V$, a maximum weight point in \bar{B}_G is obtained by finding a maximum weight spanning tree in G' after contracting \bar{E} and by assigning, for $v \in V$, to each edge in K_v a weight equal to $w(v)$. \square

Consider the polymatroid \bar{P}_G that corresponds to the base polytope \bar{B}_G , i.e.,

$$\bar{P}_G = \{x \in \mathbb{R}_{\geq 0}^V \mid \exists \alpha \in \bar{B}_G \text{ with } x \leq \alpha\}.$$

We finish the proof of Theorem 5 by showing that P_G is the (polymatroidal) dual of \bar{P}_G . More precisely, McDiarmid [7] (see also [12, Volume B, Section 44.6f]) introduced the following notion of a dual of a polymatroid, say $\bar{P}_G \subseteq \mathbb{R}^V$. Consider a vector $y \in \mathbb{Z}^V$ such that \bar{P}_G is contained in the box $[0, y]$; we choose $y = \deg - \mathbf{1}$. Then the set of all points $y - \alpha$ for $\alpha \in \bar{P}_G$ correspond to the bases of a polymatroid, which is called the *dual* of \bar{P}_G with respect to y . By (1), the vectors $\mu \in \mathbb{Z}^V$ obtained by taking any integral point $\alpha \in \bar{B}_G$ and setting $\mu = \deg - \mathbf{1} - \alpha$ correspond precisely to the maximal vertices of P_G as defined in Theorem 5. Hence, P_G is the dual of \bar{P}_G with

respect to y , and thus a polymatroid. Moreover, analogous to matroid duality, we can efficiently optimize over P_G because we can efficiently optimize over \tilde{P}_G (see [12, Volume B] for details).

3 Containment of k -trails

We first prove the hardness result claimed in Theorem 2.

Proof of Theorem 2 We give a reduction to the Hamiltonian path problem in cubic graphs which is known to be NP-complete. First we prove the theorem for $k = 2$. We claim a cubic graph G contains a 2-trail if and only if G contains a Hamiltonian path. Indeed if G contains a Hamiltonian path $G' = (V, E')$, then the Hamiltonian path G' certifies that G contains a 2-trail with ϕ given by the identity map between the vertices of G' and G .

Now, suppose that G contains a 2-trail and let $G' = (V, U)$ be a 2-trail contained in G with *minimum* number of edges. Let $H = (W, F)$ be a 2-tree, i.e., a Hamiltonian path, and $\phi : W \rightarrow V$ certify that G' is a 2-trail. Let w be one of the two leaves in H and $v = \phi(w)$ and $\{w, w'\} \in F$. If $|\phi^{-1}(v)| > 1$, then we can consider $\hat{H} = H \setminus \{w\}$ and restrict ϕ to $W \setminus \{w\}$. It is straightforward to see that $\hat{G} = (V, U \setminus \{\phi(w), \phi(w')\})$ is a 2-trail with possible preimage \hat{H} , thus contradicting the minimality of G' . Thus for each $v \in V$, if $|\phi^{-1}(v)| \geq 2$ then $\phi^{-1}(v)$ does not contain a leaf of H . But the degree of v is three and if $|\phi^{-1}(v)| \geq 2$ then at least one of the vertices in $\phi^{-1}(v)$ must be a leaf. This implies that ϕ is one to one and G' is a Hamiltonian path as well.

Consider any $k \geq 3$. We now reduce it to the $k = 2$ case. Again consider any cubic graph $\tilde{G} = (\tilde{V}, \tilde{E})$. For each vertex $v \in \tilde{V}$, we introduce a new set of vertices v_i , $1 \leq i \leq k - 2$ and connect them to v via the edge $\{v, v_i\}$. We let the new graph be $G = (V, E)$ where $V = \tilde{V} \cup X$, with X denoting the new vertices introduced. We now claim that G has a k -trail if and only if \tilde{G} contains a 2-trail. Let $H = (W, F)$ denote the k -tree and $\phi : W \rightarrow V$ be such that $\tilde{G} = (V, U)$ be the homomorphic image of H by ϕ and $U \subseteq E$. Since every vertex $v \in X$ is a leaf in G , it must also be a leaf in \tilde{G} . Thus $|\phi^{-1}(v)| = 1$ and if $\{w\} = \phi^{-1}(v)$, then w is also a leaf in H . Thus deleting $\phi^{-1}(X)$ from H keeps it connected. Let $H' = (W', F')$ be the connected graph obtained by deleting $\phi^{-1}(X)$ from H . Restricting ϕ to W' , we obtain a map $\phi' : W' \rightarrow \tilde{V}$. Let $G' = (\tilde{V}, U')$ denote the image of H' by ϕ' where $U' \subseteq \tilde{E}$. We claim that H' is a Hamiltonian path. Suppose for sake of contradiction that there exists a vertex $w \in W'$ of degree at least three in H' . Let $v = \phi'(w) \in \tilde{V}$. Since v has degree three in \tilde{G} , it must have degree exactly three in G' and moreover, $|\phi^{-1}(v)| = 1$. But then in H , the sole vertex in $\phi^{-1}(v_i)$ must be connected to w as well since w is the only vertex in $\phi^{-1}(v)$. Since there are $k - 2$ such vertices v_i , we obtain that the degree of w in H is at least $3 + k - 2 = k + 1$ which is a contradiction. Thus we obtain that H is a Hamiltonian path and therefore, \tilde{G} contains a 2-trail, completing the proof. \square

To complement the hardness result, we prove Theorem 3 which implies that we can approximate the minimum k such that G contains a k -trail by an additive one.

Proof of Theorem 3 Let $G' = (V, U)$ denote a k -trail contained in G with maximum number of edges. Let $H = (W, F)$ denote a k -tree and let $\phi : W \rightarrow V$ be an onto function such that G' is the ϕ -homomorphic image of H .

Property 2 $G \setminus U$ is acyclic.

Proof of Property 2 Suppose for sake of contradiction, there is a cycle C in $G \setminus U$. We will augment G' to a k -trail which also includes all edges of C . Let $C = (v_1, \dots, v_r)$ where $\{v_i, v_{i+1}\} \in E$ for each $1 \leq i \leq r$ where we let $v_{r+1} = v_1$. Let $w_i \in \phi^{-1}(v_i)$ for each $1 \leq i \leq r$. Introduce a new vertex u_i for each $1 \leq i \leq r$, and let $W' = W \cup \{u_1, \dots, u_r\}$ and $F' = F \cup \{\{w_i, u_{i+1}\} : 1 \leq i \leq r\}$. Moreover, we extend ϕ to ϕ' by letting $\phi'(u_i) = v_i$ for each i . It is easy to see that $G'' = (V, U \cup C)$ is an homomorphic image of $H' = (W', F')$ by $\phi' : W' \rightarrow V$. While H' is not necessarily a k -tree, Lemma 2 implies that there is another $\tilde{H} = (W', \tilde{F})$ such that the degree of any vertex $w \in \phi^{-1}(v)$ for $v \in C$ is at most

$$\left\lceil \frac{\deg_{G'}(v) + 2}{|\phi'^{-1}(v)|} \right\rceil \leq \left\lceil \frac{k|\phi^{-1}(v)| + 2}{|\phi^{-1}(v)| + 1} \right\rceil \leq k.$$

For any other vertex w , the degree does not change and therefore, remains at most k . This proves Property 2. \square

Now we will augment U to \hat{U} inductively one edge at a time while maintaining two hypotheses. Firstly, $\hat{G} = (V, \hat{U})$ will be a $(k+1)$ -trail. Moreover, only vertices in $\hat{H} = (\hat{W}, \hat{F})$ —the preimage of \hat{G} —of degree $k+1$ will be isolated nodes in $G \setminus \hat{U}$. Clearly, the conditions are satisfied initially when $\hat{U} = U$ and $\hat{G} = G'$ since there are no vertices of degree $k+1$ in $\hat{H} = H$. Suppose it is true for some $\hat{U} \supseteq U$ such that $E \setminus \hat{U} \neq \emptyset$. Since $G \setminus \hat{U}$ is a forest, there exists a leaf vertex u with the only edge incident being $\{u, v\}$. We include $\{u, v\}$ in \hat{U} , introduce another a new vertex v' in \hat{W} such that $\phi(v') = v$. Moreover, we include the edge $\{u', v'\}$ in \hat{F} where $u' \in \phi^{-1}(u)$. By the induction hypothesis, u' must have had degree k initially and after the introduction of the edge $\{u', v'\}$, its degree increases to $k+1$. Since u is now isolated in $G \setminus \hat{U}$ the induction hypothesis continues to hold.

Thus, after inclusion of all edges we obtain a $(k+1)$ -tree such that G is the homomorphic image of the tree, proving the theorem. \square

3.1 Containment of minimum weight k -trails

Now we consider the problem of finding the minimum weight k -trail contained in $G = (V, E)$ and prove Theorem 4. Our goal is to use the auxiliary graph $G' = (V', E')$ described in the proof of Theorem 1 for the recognition algorithm. Recall that edges in E are in one-to-one correspondence with $\bar{E} \subseteq E'$. We extend the weight function $w : E \rightarrow \mathbb{Z}$ to all edges in E' , where $e \in \bar{E}$ gets the same weight as the corresponding edge in E . The rest of the edges in $E' \setminus \bar{E}$ are assigned weight 0. Recall, V'_v denotes the set of vertices introduced for vertex v , and $K_v = E'(V'_v)$ denotes all edges with both endpoints in V'_v . We start by rephrasing the problem of finding a light k -trail in

terms of the auxiliary graph G' . This reformulation of the problem in terms of G' is very similar in spirit to the way we rephrased the recognition problem on G' . The key difference is that, since a k -trail contained in G may not use all edges of G , the spanning trees we consider in G' do not need to contain all of the edges in \bar{E} . We provide a brief proof of the lemma below for completeness.

Lemma 6 *The following two statements are equivalent:*

- (i) $G = (V, E)$ contains a k -trail of weight $\alpha \in \mathbb{Z}$.
- (ii) There is a spanning tree T with $w(T) = \alpha$ in the auxiliary graph $G' = (V', E')$ such that $\frac{1}{k} \deg_T(v) + |T \cap K_v| \leq \deg_E(v)$ for $v \in V$. In this case, the graph $(V, E \cap T)$ is a k -trail contained in $G = (V, E)$ of weight α .

Proof (ii) \Rightarrow (i): Analogous to the discussion in Sect. 2 about the auxiliary graph, we can derive from T a homomorphic preimage of a subgraph of G . We quickly recall the main steps. In the auxiliary graph G' we first contract all edges in $T \cap K_v$ for $v \in V$. We then define a homomorphism ϕ from this resulting graph to a subgraph of G by mapping each vertex u of the contracted version of G' —notice that u represents a set of contracted vertices in some V'_v —to the vertex v . The average degree of the preimages of v is equal to $\frac{\deg_T(v)}{\deg_E(v) - |T \cap K_v|}$, which is at most k by assumption. Hence, because the degrees can be balanced out by Lemma 2, this implies that $(V, E \cap T)$ is indeed a k -trail in G . Clearly, the weight $w(T \cap E)$ of $(V, E \cap T)$ is equal to the weight of T in G' .

(i) \Rightarrow (ii): Let (V, U) with $U \subseteq E$ be a k -trail contained in G of weight $w(U) = \alpha$. Consider the auxiliary graph $G'' = (V'', E'')$ that corresponds to (V, U) . Analogous to the discussion in Sect. 2, there is a spanning tree $T'' \subseteq E''$ in G'' that contains all edges corresponding to U and corresponds—when contracting all edges of T'' in each blob V''_v for $v \in V$ —to a k -tree that is a preimage of (V, U) . Notice that each edge in T'' also exists in the auxiliary graph $G' = (V', E')$. Also observe, that G' may contain more vertices than G'' , namely one vertex for each endpoint of $E \setminus U$. We conclude by observing that T'' can be extended to a spanning tree T in the auxiliary graph $G' = (V', E')$ by adding edges within each V'_v for $v \in V$ as follows. For each $v \in V$, there are $|V'_v| - |V''_v|$ vertices in V'_v that are not connected by T'' . For each such vertex, we add an edge of $E'(V'_v)$ to T'' connecting it to T'' . Thus a tree T in $G' = (V', E')$ is obtained with $w(T) = w(U)$ that corresponds to the same splitting vector as T'' . Hence, T represents a homomorphic preimage of the k -trail (V, U) , and thus fulfills (ii). \square

We will prove Theorem 4 through an algorithm based on the iterative relaxation paradigm. Iterative relaxation and related techniques have previously been applied to degree-constrained spanning tree problems [1, 2, 5, 14, 15], and we refer the reader to [6] for further details and examples related to this technique.

Our algorithm is based on a natural linear programming relaxation for the problem, highlighted below as **LPA**. Since we will delete edges, or more precisely fix their value to zero, and drop degree constraints throughout our algorithm, we define **LPA** for an arbitrary subset of the edge $E^* \subseteq E'$, which corresponds to edges that have not been set to zero yet and is initially set to $E' = E^*$, and we impose degree constraints only

on a subset $Q \subseteq V$, which is initially set to $Q = V$. For any set $S \subseteq V'$ and set of edges $F \subseteq E'$, we denote by $F(S) = \{\{u, v\} \in F : u, v \in S\}$ all edges of F with both endpoints in S . Notice that the first two lines of constraints in **LPA** together with the nonnegativity constraints described the spanning tree polytope over G' . Hence, by Lemma 6, a $\{0, 1\}$ -solution $x \in \{0, 1\}^{E'}$ to **LPA** for $E^* = E'$ and $Q = V$ corresponds precisely to a k -trail contained in G . Thus, **LPA** for $E^* = E'$ and $Q = V$ is indeed a relaxation of the problem of finding a minimum weight k -trail in G .

$$\begin{array}{ll}
 \min \sum_{e \in E^*} w_e x_e & \\
 x(E^*) & = |V'| - 1 \\
 x(E^*(S)) & \leq |S| - 1 \quad \forall S \subseteq V', |S| \geq 2 \\
 x(\delta_{E^*}(V'_v)) + k x(E^*(V'_v)) & \leq k \cdot \deg_E(v) \quad \forall v \in Q \\
 x_e & \geq 0 \quad \forall e \in E' \\
 x_e & = 0 \quad \forall e \in E' \setminus E^*
 \end{array} \quad (\text{LPA})$$

Algorithm 1 describes our iterative relaxation procedure to find a light $(2k - 1)$ -trail.

Algorithm 1: Obtaining a light $(2k - 1)$ -trail through iterative relaxation.

1. Initialize $Q \leftarrow V$, $E^* \leftarrow E$.
2. While $Q \neq \emptyset$
 - (a) Let x denote the optimal extreme point solution to **LPA**.
 - (b) If there exists an edge $e \in E^*$ such that $x_e = 0$, then $E^* \leftarrow E^* \setminus \{e\}$.
 - (c) If there exists a vertex $v \in V$ such that one of the following is satisfied

$$\begin{aligned}
 \deg_{E^*}(v) + (2k - 1) \cdot |E^*(V'_v)| &\leq (2k - 1) \cdot \deg_E(v), \text{ or} \\
 \deg_{E^*}(v) &\leq 2k - 1,
 \end{aligned}$$

then $Q \leftarrow Q \setminus \{v\}$.

3. Return the optimal extreme point solution x^* to **LPA**.
-

Observe that when $Q = \emptyset$, then **LPA** optimizes over the convex hull of spanning trees of G' , and therefore, x^* is an indicator of a spanning tree T of G' . Moreover, the weight of x^* is at most the weight of the initial linear programming solution since we either delete edges with zero fractional value or remove constraints. Consider a vertex $v \in V$, and let E^* be the support of the fractional solution x^* at the moment of the algorithm when v was removed from Q . If $\deg_{E^*}(v) \leq 2k - 1$ then it follows that $\deg_T(v) \leq 2k - 1$ since the returned spanning tree T satisfies $T \subseteq E^*$. Moreover, because $|T(V'_v)| \leq |V'_v| - 1 = \deg_E(v) - 1$, following by T being a spanning tree, we have that $\deg_T(v) + (2k - 1) \cdot |T(V'_v)| \leq (2k - 1) \cdot \deg_E(v)$, showing that the constraint for v will be fulfilled for $2k - 1$ even after dropping the constraint in **LPA** that corresponds to v . In the other case, we also have that $\deg_T(v) + (2k - 1) \cdot |T(V'_v)| \leq (2k - 1) \cdot \deg_E(v)$ since $T \subseteq E^*$. Thus, from Lemma 6 it follows that if Algorithm 1 terminates, then the returned spanning tree T (defined through

$\chi(T) = x^*$) corresponds to a $(2k - 1)$ -trail of G of weight at most the optimal value of the LP relaxation at the beginning of the algorithm, i.e., **LPA** with $E^* = E'$ and $Q = V$, proving the theorem. It remains to prove that the algorithm is able to make progress in each iteration, which is shown in the following lemma.

Lemma 7 *Let x denote an extreme point solution to **LPA** for some sets $E^* \subseteq E'$ and $Q \subseteq V$. Then either there is an edge $e \in E^*$ with $x_e = 0$ or a vertex $v \in Q$ such that one of the following is satisfied*

$$\deg_{E^*}(v) + (2k - 1) \cdot |E^*(V'_v)| \leq (2k - 1) \cdot \deg_E(v), \text{ or} \\ \deg_{E^*}(v) \leq 2k - 1.$$

The proof of Theorem 4 now follows from Lemma 7. Later in Sect. 3.2, we show a matching integrality gap example. It remains to prove Lemma 7.

Proof of Lemma 7 Suppose for sake of contradiction that the conditions in the lemma are not satisfied by some extreme point solution x . We consider the set of tight linear constraints defining x . Let $\tau = \{S \subseteq V' : |S| \geq 2, x(E^*(S)) = |S| - 1\}$ be the tight constraints among the spanning tree constraints and let $Q_t \subseteq Q$ denote the tight degree constraints. Standard uncrossing arguments imply the following lemma (see Lau et al. [6] Chapter 4 for details).

Lemma 8 *There exists a laminar family $\mathcal{L} \subseteq \tau$ and $Q' \subseteq Q_t$ such that the following hold.*

- (i) $|\mathcal{L}| + |Q'| = |E^*|$.
- (ii) The vectors $\{\chi(E^*(S)) : S \in \mathcal{L}\} \cup \{\chi(\delta_{E^*}(V'_v)) + k\chi(E^*(V'_v)) : v \in Q'\}$ are linearly independent.
- (iii) $\{\chi(E^*(S)) : S \in \mathcal{L}\}$ span all the tight constraints corresponding to sets in τ , and $\{\chi(E^*(S)) : S \in \mathcal{L}\} \cup \{\chi(\delta_{E^*}(V'_v)) + k\chi(E^*(V'_v)) : v \in Q'\}$ span all the tight constraints corresponding to sets in τ and vertices in Q_t .

We now show a contradiction by a counting argument. Observe that we have $|\mathcal{L}| \leq |V'| - 1$ since \mathcal{L} is a laminar family over a ground set of size $|V'|$ and contains only sets of size at least two. Let $z \in \mathbb{R}_{\geq 0}^{E^*}$ be the slack vector defined by $z_e = 1 - x_e$ for any edge $e \in E^*$. Hence, using the fact that $x(E^*) = |V'| - 1$, the total slack is bounded by

$$z(E^*) = |E^*| - x(E^*) \leq |\mathcal{L}| + |Q'| - |V'| + 1 \leq |Q'|.$$

Since we assume the conditions of the lemma are not satisfied and constraints for vertices in Q' are tight, the following inequality and equation follow for any $v \in Q'$:

$$\frac{k}{2k - 1} (\deg_{E^*}(v) + (2k - 1) \cdot |E^*(V'_v)|) \geq \frac{k}{2k - 1} ((2k - 1) \cdot \deg_E(v) + 1), \\ - (x(\delta_{E^*}(V'_v)) + k \cdot x(E^*(V'_v))) = -k \cdot \deg_E(v).$$

Summing up the left-hand sides and right-hand sides of the above relations, adding to both sides $\frac{k-1}{2k-1} \deg_{E^*}(v)$, and exploiting that $z_e + x_e = 1$ for each edge $e \in E^*$, we obtain the following inequality:

$$z(\delta_{E^*}(V'_v)) + k \cdot z(E^*(V'_v)) \geq \frac{k-1}{2k-1} \deg_{E^*}(v) + \frac{k}{2k-1} \quad \forall v \in Q'.$$

Since we assume that none of the two conditions of Lemma 7 holds, we have $\deg_{E^*}(v) \geq 2k$, and thus

$$z(\delta_{E^*}(V'_v)) + k \cdot z(E^*(V'_v)) \geq \frac{k-1}{2k-1} \cdot 2k + \frac{k}{2k-1} = k.$$

Summing over all vertices $v \in Q'$, we get

$$\sum_{v \in Q'} (z(\delta_{E^*}(V'_v)) + k \cdot z(E^*(V'_v))) \geq k \cdot |Q'|. \quad (2)$$

Since $E^*(V_v)$ are disjoint for different vertices v , every edge is counted at most k times in the left-hand side and therefore it is at most $k \cdot z(E^*)$. Moreover, we have

$$z(E^*) = |E^*| - x(E^*) = |E^*| - |V'| + 1 \leq |E^*| - |\mathcal{L}| = |Q'|. \quad (3)$$

We thus obtain

$$k|Q'| \geq k z(E^*) \geq k|Q'|,$$

where the first inequality is due to (3) and the second one is due to (2). Thus we must have equality everywhere and, unless $k = 2$, we must have $z(\delta_{E^*}(V'_v)) = 0$ for each $v \in Q'$ since these edges were counted at most twice in (2).

First assume $k \geq 3$ and thus we have $z(E^*(V'_v)) = 1$ for each $v \in Q'$. Since $E^*(V'_v)$ are disjoint, this implies that $z(E^*) \geq z(\cup_{v \in Q'} E^*(V'_v)) = |Q'|$ which is a contradiction unless $z_e = 0$ for each $e \in E^* \setminus (\cup_{v \in Q'} E^*(V'_v))$. Now, we show a linear dependence. Since $x_e = 1$ for each $e \in E^* \setminus (\cup_{v \in Q'} E^*(V'_v))$, we have that $\chi(\{e\})$ is in the span of tight constraints in \mathcal{L} . Subtracting these vectors from $\chi(E^*)$ we obtain that $\chi(\cup_{v \in Q'} E^*(V'_v))$ is in the span of tight constraints in τ which includes the vector $\chi(E^*)$. But this vector is also in the span of $\{\chi(\delta_{E^*}(V'_v)) + k\chi(E^*(V'_v)) : v \in Q'\}$ and $\{\chi(\{e\}) : e \in \cup_{v \in Q'} E^*(V'_v)\}$ which contradicts point (ii) of Lemma 8.

In the case when $k = 2$, we let $R \subseteq E^*$ be all edges $e \in E^* \setminus (\cup_{v \in Q'} E^*(V'_v))$ such that the two endpoints $u, v \in V$ of the edge e in G are not both contained in Q' . Because (2) must hold with equality and the total spare is bounded by $|Q'|$, there is no spare on edges in R , and hence $x_e = 1$ for each $e \in R$. But summing $\{\chi(\delta_{E^*}(V'_v)) + 2\chi(E^*(V'_v)) : v \in Q'\}$ we obtain $2\chi(E^* \setminus R) = 2\chi(E^*) - 2\chi(R)$. Since $\chi(\{e\})$ for each $e \in R$, and $\chi(E^*)$ is in the span of tight constraints in τ , and therefore also in \mathcal{L} , we again obtain a contradiction with respect to point (ii) of Lemma 8. This completes the proof of the lemma and Theorem 4. \square

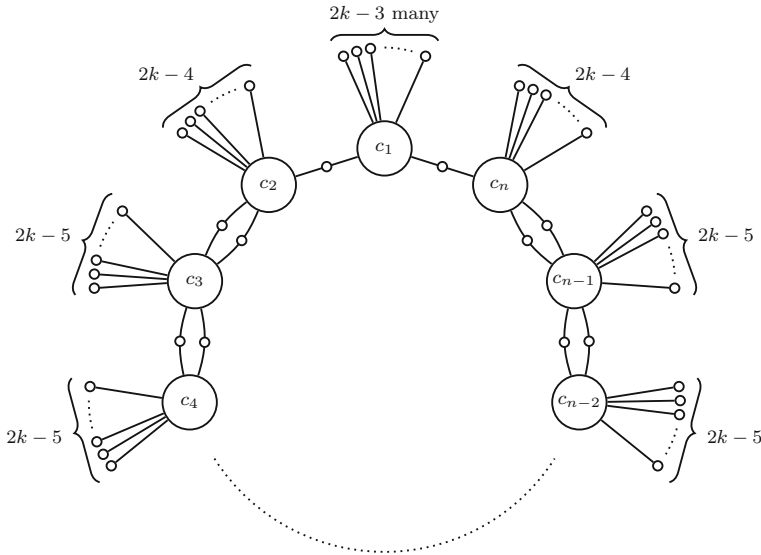


Fig. 4 The construction of our integrality gap example. Each vertex c_i for $1 \leq i \leq n$ has degree $2k - 1$. All vertices in C apart from c_1 , c_2 , and c_n have the same number of attached pending edges, namely $2k - 5$

3.2 Integrality gap example

In this section, we give an integrality gap example for the linear program **LPA**, for $E^* = E'$ and $Q = V$. First observe that for $k = 2$, Theorem 4 returns a 3-trail and therefore is optimal. Now consider any integer $k \geq 3$. We show that there is a graph $G = (V, E)$ such that G contains no $(2k - 2)$ -trail but the linear program is feasible for k .

Consider the cycle $C = \{c_1, \dots, c_n\}$ where we have two parallel edges between c_i and c_{i+1} for each $2 \leq i \leq n - 1$. Thus, c_1 has degree two, c_2 and c_n have degree three and every other vertex has degree four. Now we subdivide each of these $2n - 2$ edges by introducing a vertex of degree two on each edge. Moreover, for each $1 \leq i \leq n$, we add $2k - 1 - \deg(c_i)$ additional distinct vertices, say $R_i = \{w_i^1, \dots, w_i^{b_i}\}$, and there is an edge (w, v_i) for each $w \in R_i$. Thus, the degree of each vertex is now exactly $2k - 1$. Figure 4 summarizes the construction of our instance $G = (V, E)$.

Observe that any split vector μ with a single zero and $n - 1$ ones among the vertices C , and zeros everywhere else, is feasible. Taking the uniform convex combination of all these split vectors, we obtain a fractional split vector, which splits each vertex in C precisely $1 - \frac{1}{n}$ times. This shows that there is an LP solution x to **LPA** (with $E^* = E'$ and $Q = V$)—defined as usual over the auxiliary graph G' —such that $x(e) = 1$ for all edges $e \in E$ and

$$x(E'(V'_c)) = \deg_E(c) - 1 - \left(1 - \frac{1}{n}\right) = 2k - 2 - \left(1 - \frac{1}{n}\right) \quad \forall c \in C.$$

If $n \geq k$, this LP solution is indeed feasible for k because for any $c \in C$,

$$\begin{aligned}
 x(\delta_{E'}(V'_c)) + k x(E'(V'_c)) &= \deg_E(v) + k \left(2k - 2 - 1 - \left(1 - \frac{1}{n} \right) \right) \\
 &= 2k - 1 + k \left(2k - 2 - \left(1 - \frac{1}{n} \right) \right) \\
 &= k(2k - 1) + \frac{k}{n} - 1 \\
 &\leq k(2k - 1) \\
 &= k \deg_E(v).
 \end{aligned}$$

Hence, in terms of the LP, the graph G is a fractional k -trail.

We now consider (integral) trails contained in G . For any $k' \geq 2$, to show that G contains a k' -trail, we have to choose a subset $U \subseteq E$ of the edges in G and show that there is a homomorphism ϕ from a k' -tree $H = (W, F)$ to (V, U) . We start by deriving a basic relation between the total number of splits $\alpha = |W| - |V|$, and $\beta = |E| - |U|$, which is the number of edges to be removed from G to obtain the k' -trail contained in G .

Notice that $|F| = |U|$ since edges between H and G are in one-to-one correspondence. Moreover, because H is a tree, we must have

$$|W| = |F| + 1 = |U| + 1.$$

Thus,

$$\alpha + \beta = |W| - |U| + |E| - |V| = 1 + |E| - |V|.$$

Note that the total number of vertices and edges of our instance G are

$$\begin{aligned}
 |V| &= (2k - 1) \cdot n - n + 2, \text{ and} \\
 |E| &= (2k - 1) \cdot n,
 \end{aligned}$$

respectively. Thus

$$\alpha + \beta = n - 1.$$

In words, when constructing a homomorphism from a tree to a subgraph of G , then the sum of the total number of times vertices get split and the total number of edges removed from G to obtain the subgraph is equal to $n - 1$. Since we have n vertices on our cycle, and each edge is incident with precisely one vertex of C , we have by the pigeonhole principle that there is at least one vertex $c \in C$ such that c does not get split and no edge incident with c got removed. However, this implies that the constructed subgraph has a vertex of degree $2k - 1$ that does not get split. Hence, G does not contain any k' -trail for $k' < 2k - 1$.

Acknowledgements We are grateful to Michel Goemans, Anupam Gupta, Neil Olver, and András Sebő for inspiring discussions, and to the anonymous reviewers for many helpful comments. This research project started while both authors were guests at the Hausdorff Research Institute for Mathematics (HIM) during the 2015 Trimester on Combinatorial Optimization. Both authors are very thankful for the generous support and inspiring environment provided by the HIM and the organizers of the trimester program.

References

1. Bansal, N., Khandekar, R., Könemann, J., Nagarajan, V., Peis, B.: On generalizations of network design problems with degree bounds. *Math. Program.* **141**(1), 479–506 (2013)
2. Bansal, N., Khandekar, R., Nagarajan, V.: Additive guarantees for degree-bounded directed network design. *SIAM J. Comput.* **39**(4), 1413–1431 (2009)
3. Fujishige, S.: *Submodular Functions and Optimization*. Elsevier, Amsterdam (2005)
4. Goemans, M.X.: Minimum bounded degree spanning trees. In: *Proceedings of the 47th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 273–282 (2006)
5. Király, T., Lau, L.C., Singh, M.: Degree bounded matroids and submodular flows. *Combinatorica* **32**(6), 703–720 (2012)
6. Lau, L.C., Ravi, R., Singh, M.: *Iterative Methods in Combinatorial Optimization*, vol. 46. Cambridge University Press, Cambridge (2011)
7. McDiarmid, C.J.H.: Rado’s theorem for polymatroids. *Math. Proc. Camb. Philos. Soc.* **78**, 263–281 (1975)
8. Molnár, M., Durand, S., Merabet, M.: Approximation of the degree-constrained minimum spanning hierarchies. In: *Proceedings of 21st International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pp. 96–107 (2014)
9. Molnár, M., Durand, S., Merabet, M.: A new formulation of degree-constrained spanning problems. In: *Proceedings of 9th International Colloquium on Graph Theory and Combinatorics (ICGT)* (2014). <http://oc.inpg.fr/conf/icgt2014/>
10. Molnár, M., Newman, A., Sebő, A.: Travelling salesmen on bounded degree trails (2015). Hausdorff report (in preparation)
11. Ravi, R., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Hunt III, H.B.: Approximation algorithms for degree-constrained minimum-cost network-design problems. *Algorithmica* **31**(1), 58–78 (2001)
12. Schrijver, A.: *Combinatorial Optimization, Polyhedra and Efficiency*. Springer, Berlin (2003)
13. Sebő, A.: Travelling salesmen on bounded degree trails. Presentation at HIM Connectivity Workshop in Bonn. Presentation is accessible online at <https://www.youtube.com/watch?v=5Do2JMhgrCM> (2015)
14. Singh, M., Lau, L.C.: Approximating minimum bounded degree spanning trees to within one of optimal. In: *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 661–670 (2007)
15. Zenklusen, R.: Matroidal degree-bounded minimum spanning trees. In: *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1512–1521 (2012)