On Complexity of Isoperimetric Problems on Trees*

Amir Daneshgar†
Department of Mathematical Sciences
Sharif University of Technology
P.O. Box 11155–9415, *Tehran, Iran*
daneshgar@sharif.ir

Ramin Javadi
Department of Mathematical Sciences
Sharif University of Technology
P.O. Box 11155–9415, *Tehran, Iran*
rjavadi@mehr.sharif.ir

Abstract

This paper is aimed to investigate some computational aspects of different isoperimetric problems on weighted trees. In this regard, we consider different connectivity parameters called *minimum normalized cuts/isoperimetric numbers* defined through taking minimum of the maximum or the mean of the normalized outgoing flows from a set of subdomains of vertices, where these subdomains constitute a *partition/subpartition*. We show that the decision problem for the case of taking k-partitions and the maximum (called the max normalized cut problem NCP^M) as well as the other two decision problems for the mean version (referred to as IPP^m and NCP^m) are NP-complete problems for weighted trees. On the other hand, we show that the decision problem for the case of taking k-subpartitions and the maximum (called the max isoperimetric problem IPP^M) can be solved in *linear time* for any weighted tree and any $k \geq 2$. Based on this fact, we provide polynomial time $O(k)$-approximation algorithms for all different versions of kth isoperimetric numbers considered.

Moreover, when the number of partitions/subpartitions, k, is a fixed constant, as an extension of a result of B. Mohar (1989) for the case $k = 2$ (usually referred to as the Cheeger constant), we prove that max and mean isoperimetric numbers of weighted trees as well as their max minimum normalized cut can be computed in polynomial time. We also prove some hardness results for the case of simple unweighted graphs and trees.

Key words: isoperimetric number, Cheeger constant, normalized cut, graph partitioning, computational complexity, approximation algorithms, weighted trees.

Subject classification: 05C85, 68Q25, 68R10.

1 Introduction

The classical isoperimetric problem is a well-known and well-studied subject in Riemannian geometry, while the analogous problems in discrete case have recently been at the center of attention. Different aspects of these problems have been extensively studied in the literature and variety of relations to many important concepts have been discovered. The significance of the isoperimetric problem is due to its relation to the central theoretical concepts and also its varied real world applications (e.g. see [2, 5, 14, 17–19, 24, 25] for motivations and the background).

Isoperimetric numbers can be considered as geometric tools to measure the connectivity of graphs. To begin, let us recall (e.g. see [19]) the definition of the classical isoperimetric

*A preliminary version of this article is available at arXiv:1009.0706.

†Correspondence should be addressed to daneshgar@sharif.ac.ir.

number (Cheeger constant) of a simple graph $G = (V, E)$ as

$$h(G) \stackrel{\text{def}}{=} \min_{|Q| \leq \frac{|V|}{2}} \frac{c(Q)}{|Q|} = \min_{Q \subseteq V} \max \left\{ \frac{c(Q)}{|Q|}, \frac{c(Q)}{|Q^c|} \right\},$$

where

$$c(Q) \stackrel{\text{def}}{=} |\{uv \in E : u \in Q \ \& \ v \notin Q\}|,$$

and the not so common mean version as follows

$$\iota(G) \stackrel{\text{def}}{=} \min_{Q \subseteq V} \frac{c(Q)}{|Q||Q^c|}. \tag{1}$$

Higher isoperimetric numbers, as generalizations of the classical isoperimetric numbers, have been defined for a general Markov chain on a directed base-graph and their properties has been studied extensively (e.g. see [7] and references therein). These problems deal with minimizing the max/mean of the normalized outgoing flows over all *subpartitions* (disjoint nonempty subsets) of the vertex set. One may also define similar parameters based on minimizing this value over all *partitions* of the vertex set usually known as the *minimum normalized cuts* (see e.g. [24, 25]). Following the main result of [7], it is known that the isoperimetric numbers can be described as $\{0, 1\}$-optimization programs which admit a relaxation to the reals, while this is not the case for the minimum normalized cuts. This fact can be considered as a clue that the normalized cut problem is likely to be harder than the isoperimetric problem, which is almost approved by the results of this article.

The main objective of this article is to investigate computational aspects of these parameters on weighted trees. Our motivations for this study are twofold. On the one hand, tree partitioning and in particular solving isoperimetric problems on weighted trees has its own importance due to the existence of many applications in the practical problems such as image segmentation and pattern recognition (e.g. see [3, 4, 11, 13, 16]). On the other hand, the study of isoperimetric problems on trees is important from a computational point of view, since they provide a universe in which by small perturbations of conditions, these problems change their computational hardness from simple (i.e. polynomial time) to hard (i.e. NP-complete) and vise versa. In this regard, our results provide compelling evidence to consider as a general belief that changing the problem from subpartitions to partitions or taking the mean instead of the maximum, usually makes the problem computationally harder.

Let us begin with a description of our general setup. Our framework is a *weighted graph* which is a simple graph $G = (V, E)$ along with two weight functions on the vertex and the edge sets as, $\omega : V \to \mathbb{Q}^+$ and $c : E \to \mathbb{Q}^+$, which is usually denoted by $G = (V, E, \omega, c)$. By an *unweighted graph* we mean a weighted graph where all the vertex and edge weights are equal to 1. For every nonvoid subsets $A, B \subseteq V$, we define

$$E(A, B) \stackrel{\text{def}}{=} \{e = uv \in E : \ u \in A, v \in B\},$$

$$\omega(A) \stackrel{\text{def}}{=} \sum_{u \in A} \omega(u), \quad c(A, B) \stackrel{\text{def}}{=} \sum_{e \in E(A,B)} c(e), \quad c(A) \stackrel{\text{def}}{=} c(A, A^c).$$

The *normalized outgoing flow* of the set A is defined as the quotient $c(A)/\omega(A)$. The set $\mathcal{D}_k(V)$ is defined to be the set of all *k-subpartitions* $\{A_1, \ldots, A_k\} \stackrel{\text{def}}{=} \{A_i\}_1^k$ of V, where A_i's are nonempty disjoint subsets of V. The set of all *k-partitions* of a set V, which is denoted by $\mathcal{P}_k(V)$, is the subclass of $\mathcal{D}_k(V)$ containing all k-sets $\{A_i\}_1^k$ for which $\cup_{i=1}^k A_i = V$. Also, for every positive integer n, the notation $[n]$ stands for the set $\{1, \ldots, n\}$.

Now, we define the mean and max isoperimetric numbers as well as the minimum normalized cuts as follows.

Definition 1. Given a weighted graph $G = (V, E, \omega, c)$, for each k, $1 \leq k \leq |V|$, the kth mean and max *isoperimetric numbers* of G, denoted by $\iota_k^m(G)$ and $\iota_k^M(G)$, respectively, are defined as

$$\iota_k^m(G) \stackrel{\text{def}}{=} \min_{\{A_i\}_1^k \in \mathcal{D}_k(V)} \frac{1}{k}\left(\sum_{i=1}^{k} \frac{c(A_i)}{\omega(A_i)}\right),$$

$$\iota_k^M(G) \stackrel{\text{def}}{=} \min_{\{A_i\}_1^k \in \mathcal{D}_k(V)} \max_{1\leq i\leq k} \frac{c(A_i)}{\omega(A_i)}.$$

Furthermore, considering the partitions, we define the following related constants as the kth (mean and max) *minimum normalized cuts* of G,

$$\tilde{\iota}_k^m(G) \stackrel{\text{def}}{=} \min_{\{A_i\}_1^k \in \mathcal{P}_k(V)} \frac{1}{k}\left(\sum_{i=1}^{k} \frac{c(A_i)}{\omega(A_i)}\right),$$

$$\tilde{\iota}_k^M(G) \stackrel{\text{def}}{=} \min_{\{A_i\}_1^k \in \mathcal{P}_k(V)} \max_{1\leq i\leq k} \frac{c(A_i)}{\omega(A_i)}.$$

We call a weighted graph G, mean (resp. max) *k-geometric*, if $\iota_k^m(G) = \tilde{\iota}_k^m(G)$ (resp. $\iota_k^M(G) = \tilde{\iota}_k^M(G)$). Also, G is called mean (resp. max) *supergeometric*, if it is mean (resp. max) k-geometric for all $2 \leq k \leq |V|$. We call a vertex $v \in V$, a (mean or max) *k-outlier*, if there exists a minimizing subpartition achieving $\iota_k(G)$, where v lies outside of the subpartition. It is well-known that $\iota_2 = \tilde{\iota}_2$ (see [7]) and the common value is usually called the *Cheeger constant* or *edge expansion* in the literature. ♠

In order to investigate computational complexity of these optimization problems, as is traditional for complexity results, we consider the corresponding decision problems. Furthermore, since the isoperimetric parameters as operators on weight functions preserve scalar multiplication, without loss of generality, we assume that the range of all weight functions is $\mathbb{Z}$ (instead of $\mathbb{Q}$). Moreover, for simplicity we use a couple of notations. The acronyms IPP and NCP stand, respectively, for the isoperimetric and normalized cut problems. As before, the superscripts m or M determine the mean or max version of these problems, respectively[1], and subscript k is used whenever k is a constant and does not appear as part of the input. For instance, NCP_k^M refers to the following problem,

$\mathbf{NCP_k^M}$

CONSTANTS: An integer k.

INSTANCE: A weighted graph $G = (V, E, \omega, c)$ and a positive rational number $N \in \mathbb{Q}^+$.

QUERY: Is it true that $\tilde{\iota}_k^M(G) \leq N$? In other words, is there a k-partition $\{A_i\}_1^k \in \mathcal{P}_k(V)$ such that $\max\limits_{1\leq i\leq k}\left\{\frac{c(A_i)}{\omega(A_i)}\right\} \leq N$?

By the following results, the equivalent problems IPP_2 and NCP_2 are known to be NP-complete.

Theorem A.

(i) [19] The problem NCP_2 is NP-complete for (unweighted) graphs with multiple edges.

(ii) [24] The problem NCP_2 is NP-complete for bipartite planar weighted graphs.

Note that, however, the planarity and the bipartiteness in Theorem A(ii) is not mentioned explicitly in [24], the above statement clearly follows from the reduction provided in the proof.

[1] Note that whenever the superscripts m and M are omitted, it means that the statement is true for both versions.

For a long time, it has been an open and challenging problem how well $\iota_2 = \tilde{\iota}_2$ can be approximated in polynomial time for general graphs. The best current known result is due to Arora *et al.* which gives a polynomial time approximation algorithm that computes ι_2 up to a factor of $O(\sqrt{\log n})$ for an n-vertex simple graph using semidefinite programming and geometric embedding (see [1, 23, 26]). Moreover, Wu *et. al.* present a polynomial time $((4 + o(1)) \log n)$-approximation algorithm for the minimum normalized cut on an n-vertex weighted graph [27].

It is instructive to note that the *non-normalized* counterparts of the (mean) isoperimetric problem and the (mean) normalized cut problem are already known as the *minimum k-subpartition* problem and *minimum k-way cut* problem, respectively (e.g. see [21] for details and the background). Particularly, we know that there exists a polynomial time $2(1 - 1/k)$-approximation algorithm for the minimum k-way cut problem which is based on computation of the minimum k-subpartition problem [22]. In Section 2, along the same lines, we prove a couple of basic inequalities (Theorem 1) which show that the isoperimetric numbers can be considered as an approximation for the minimum normalized cuts. In Section 3 we consider the computational aspects of this approximation on weighted trees and we determine the computational complexity of the four main isoperimetric and normalized cut problems. There we prove that IPP^m, NCP^m, and NCP^M are all NP-complete for weighted trees, however, quite unexpectedly, it turns out that IPP^M is a linear time solvable problem in this case. This is used to provide polynomial time $O(k)$-approximation algorithms for the kth isoperimetric number and the kth minimum normalized cuts on weighted trees.

In Section 4 we focus on the case when the number of parts, k, is fixed and does not appear as part of the input. For $k = 2$, B. Mohar [19] has proved that there exists a linear time algorithm that computes ι_2 for trees. In this section as a generalization of Mohar's result we prove that, for each $k \geq 2$, all parameters ι_k^M, ι_k^m and $\tilde{\iota}_k^M$ can be computed in polynomial time for weighted trees. We also show that this fact can not be extended to weighted graphs with bounded tree-width (unless $P = NP$!) by proving that for every fixed $k \geq 2$, IPP$_k$ and NCP$_k$ (in both max and mean versions) are NP-complete for bipartite weighted graphs with tree-width 2.

In Section 5, we try to improve the hardness results to the case of unweighted (simple) graphs or trees. In this regard, we provide a general reduction method that can be used to improve any known strong NP-completeness result for weighted graphs to an NP-completeness result for unweighted graphs. Particularly, we use this reduction to prove the NP-completeness of NCP^M for unweighted trees and IPP$_k$ and NCP$_k$ for unweighted graphs.

Finally, throughout this article the *runtime of a graph algorithm* is the function describing the number of operations executed in terms of the number of vertices. Also, we assume that weighted trees are represented in a *succinct data structure* in which standard navigational operations, such as finding the parent, can be performed in constant time (e.g. see [20]).

2 A Basic Inequality

Our main result in this section is the following inequalities, which are counterparts of a similar result for the minimum k-way cut problem, that has already been proved in [22].

Theorem 1. *For every connected weighted graph G and every integer $3 \leq k \leq |V(G)|$,*

$$\iota_k^M(G) \;\leq\; \tilde{\iota}_k^M(G) \;<\; (k-1)\;\iota_k^M(G),$$

$$\iota_k^m(G) \;\leq\; \tilde{\iota}_k^m(G) \;<\; 2(1 - \frac{1}{k})\;\iota_k^m(G).$$

Note that, when $k = 2$, we have $\iota_2(G) = \tilde{\iota}_2(G)$ for both max and mean versions [7]. Moreover, the result shows that the parameters $\iota_k^m(G)$ and $\iota_k^M(G)$ can be seen as approximations of the parameters $\tilde{\iota}_k^m(G)$ and $\tilde{\iota}_k^M(G)$, respectively. Therefore, from this point of view, the isoperimetric numbers can be considered as approximations for the minimum normalized cuts. We shall elaborate the computational aspects of these approximations in the next section. To prove Theorem 1, we need the following lemma.

Lemma 1. *Given an integer $k \geq 1$ and nonnegative numbers λ, a_i, b_i $(1 \leq i \leq k)$, such that $0 < \lambda < k$ and $\sum_i a_i = 1$, the following inequality holds,*

$$\sum_{i=1}^{k} a_i b_i \leq \max_j \left(\lambda a_j b_j + (1 - \frac{\lambda}{k}) b_j \right). \tag{2}$$

Equality holds if and only if either for each i, $b_i = 0$, or for each i and some constant b, $a_i = 1/k$ and $b_i = b$.

Proof. Let $I := \sum_i a_i b_i$ and for every $1 \leq j \leq k$, let $c_j := \lambda a_j b_j + (1 - \lambda/k) b_j$ and $t_j := k^2 a_j/\lambda + k/(k - \lambda)$. Then

$$\begin{aligned}
\left(\sum_{j=1}^{k} t_j \right) \max_j (c_j - I) &\geq \sum_{j=1}^{k} t_j (c_j - I) \\
&= \left(\frac{k(k-\lambda)}{\lambda} + \frac{k\lambda}{k-\lambda} - \frac{k^2}{\lambda} - \frac{k^2}{k-\lambda} \right) I + \sum_j \left(k^2 a_j^2 b_j + b_j \right) \\
&= \sum_j \left(k^2 a_j^2 b_j + b_j - 2k a_j b_j \right) = \sum_j (k a_j - 1)^2 b_j \geq 0.
\end{aligned}$$

Thus, $\max_j (c_j - I) \geq 0$, as desired. Also, the equality conditions follow immediately from the proof. ∎

Proof of Theorem 1. Lower bounds are trivial from the definitions. To prove the upper bounds, let $\{A_i\}_1^k \in \mathcal{D}_k(V)$ be a k-subpartition of the vertices and define $A^* := V \backslash (\cup_i A_i)$. For simplicity let $w_i := \omega(A_i)$, $c_i := c(A_i)$ and $C := \sum_i c_i$. For a fixed j $(1 \leq j \leq k)$ define the k-partition $\pi^j := \{B_i^j\}_1^k$ as $B_i^j := A_i$ for all $i \neq j$ and $B_j^j := A_j \cup A^*$. Then, we have

$$c(B_j^j) \leq \sum_{i: i \neq j} c(A_i) = C - c_j.$$

Thus, for every $1 \leq j \leq k$,

$$\max_i \left(\frac{c(B_i^j)}{\omega(B_i^j)} \right) \leq \max_{i: i \neq j} \left(\frac{c_i}{w_i}, \frac{C - c_j}{w_j + \omega(A^*)} \right), \tag{3}$$

$$\sum_i \frac{c(B_i^j)}{\omega(B_i^j)} \leq \frac{C - c_j}{w_j + \omega(A^*)} + \sum_{i: i \neq j} \frac{c_i}{w_i}. \tag{4}$$

In order to prove the first inequality, assume that G is not k-geometric (if G is k-geometric the results are trivial) and let $\{A_i\}_1^k$ be a subpartition which achieves $\iota_k^M(G)$ and let $c_{j_0} = \max_i c_i$. By Inequality (3), we have

$$\tilde{\iota}_k^M(G) \leq \frac{C - c_{j_0}}{w_{j_0} + \omega(A^*)} < \frac{\sum_{i: i \neq j_0} c_i}{w_{j_0}} \leq (k-1) \frac{c_{j_0}}{w_{j_0}} \leq (k-1)\, \iota_k^M(G).$$

In order to prove the second inequality, assume that $\{A_i\}_1^k$ be a subpartition which achieves $\iota_k^m(G)$. By Inequality (4), we have

$$k\ \tilde{\iota}_k^m(G) \leq \min_j \left(\frac{C - c_j}{w_j + \omega(A^*)} + \sum_{i:i\neq j} \frac{c_i}{w_i} \right) < \min_j \left(\frac{C - 2c_j}{w_j} \right) + \sum_i \frac{c_i}{w_i}. \tag{5}$$

Now, let $C^* := \sum_i (c_i/w_i)$, then applying Lemma 1 with $a_j := \frac{c_j/w_j}{C^*}$, $b_j := w_j$ and $\lambda := 2$, yields

$$\frac{C}{C^*} \leq \max_j \left(\frac{2c_j}{C^*} + (1 - \frac{2}{k})w_j \right).$$

Therefore,

$$\min_j \left(\frac{C - 2c_j}{w_j} \right) \leq (1 - \frac{2}{k}) \sum_i \frac{c_i}{w_i}, \tag{6}$$

and the result follows from Inequalities (5) and (6). ∎

Example 1. In this example we show that the bounds in Theorem 1 are sharp, in the sense that for every fixed $k \geq 3$, there is a family of weighted graphs $\{G_t\}_{t\in\mathbb{N}}$ such that $\tilde{\iota}_k^M(G_t)/\iota_k^M(G_t)$ tends to $(k-1)$ and $\tilde{\iota}_k^m(G_t)/\iota_k^m(G_t)$ tends to $2(1-\frac{1}{k})$ as t tends to infinity. Let k be a constant. For every positive integer $t \geq k$, define the graph G_t as a star with a central vertex v of degree k and k vertices $v_1, \ldots, v_k$ each of degree 1. Also, define the weight functions ω and c as follows,

$$\omega(v) := k,\ \omega(v_i) := t,\ c(vv_i) := 1 \quad \forall\ 1 \leq i \leq k.$$

Then, by the definitions we have

$$\iota_k^M(G_t) = \frac{1}{t}, \qquad \tilde{\iota}_k^M(G_t) = \max\left(\frac{1}{t}, \frac{k-1}{t+k}\right) = \frac{k-1}{t+k},$$

$$\iota_k^m(G_t) = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{t} = \frac{1}{t}, \qquad \tilde{\iota}_k^m(G_t) = \frac{1}{k}\left(\frac{k-1}{t+k} + \sum_{i=1}^{k-1}\frac{1}{t}\right) = (1-\frac{1}{k})(\frac{1}{t+k} + \frac{1}{t}),$$

where $\iota_k^M(G_t)$ and $\iota_k^m(G_t)$ are achieved for the disjoint sets $A_i := \{v_i\}$, $1 \leq i \leq k$, and $\tilde{\iota}_k^M(G_t)$ and $\tilde{\iota}_k^m(G_t)$ are achieved for the k-partition $\{B_i\}_1^k$, with $B_i := \{v_i\}$, $1 \leq i \leq k-1$ and $B_k := \{v_k, v\}$. The claim immediately follows from the above equalities. ♣

3 Algorithms, Complexity and Approximation Results

In this section we consider IPP, NCP and their approximations for weighted trees. In this regard, we shall prove that NCP^M for weighted trees is NP-complete in the strong sense. Furthermore, as a bit of a surprise, we show that the corresponding problem on subpartitions, i.e. IPP^M, happen be solved in linear time using dynamic programming, where this can be used to obtain a polynomial time approximation for the minimum normalized cut of weighted trees.

Let us recall that a problem with numerical parameters is said to be *NP-complete in the strong sense*, when it remains NP-complete, even when all of its numerical parameters are bounded by a polynomial in terms of the length of the input. In other words, a strongly NP-complete problem remains NP-complete even when the input parameters are given in unary codes (instead of binary codes).

Theorem 2. *The problem* NCP^M *is NP-complete in the strong sense for weighted trees.*

Proof. Clearly, NCP^M is in NP. To prove the strong NP-completeness of the problem we prove a sequence of reductions as follows,

$$\text{3-PARTITION} \leq_m^p \text{SUBSET AVERAGE} \leq_m^p \text{NCP}^M,$$

where the well-known 3-PARTITION problem and the SUBSET AVERAGE problem are defined as,

3-PARTITION

INSTANCE: A positive integer $B \in \mathbb{Z}^+$ and $3m$ positive integers $x_1, \ldots, x_{3m} \in \mathbb{Z}^+$, such that $B/4 < x_i < B/2$, for each $1 \leq i \leq 3m$ and $\sum_{i=1}^{3m} x_i = mB$.

QUERY: Is there an m-partition $\{S_i\}_1^m \in \mathcal{P}_m([3m])$ such that, for each $1 \leq j \leq m$, $\sum_{i \in S_j} x_i = B$?

SUBSET AVERAGE

INSTANCE: Positive integers $y_1, \ldots, y_n \in \mathbb{Z}^+$, where their average is an integer α along with a positive integer $m \leq n$.

QUERY: Is there an m-partition $\{T_i\}_1^m \in \mathcal{P}_m([n])$ such that, for each $1 \leq j \leq m$, average of the elements with indices in T_j is equal to α, i.e. $\sum_{i \in T_j} y_i = \alpha |T_j|$?

Note that the 3-PARTITION problem is known to be strongly NP-complete [12], and consequently, the claim follows from the above reductions.

Step 1. 3-PARTITION $\leq_m^p$ SUBSET AVERAGE.
In the first step, we show that SUBSET AVERAGE is NP-complete in the strong sense, by a reduction from 3-PARTITION. Given $3m$ positive integers $x_1, \ldots, x_{3m}$ as an instance of 3-PARTITION, define for $1 \leq i \leq 3m$, $y_i := x_i + B + 1$ and for $3m+1 \leq i \leq 4m$, $y_i := 1$. Now, consider $\{y_1, \ldots, y_{4m}\}$ together with the integer m as an instance of SUBSET AVERAGE. The average of y_i's is equal to $B+1$. If the answer to 3-PARTITION is yes, then there exists an m-partition $\{S_i\}_1^m \in \mathcal{P}_m([3m])$ such that, for each $1 \leq j \leq m$, $\sum_{i \in S_j} x_i = B$. Since $B/4 < x_i < B/2$, each S_j contains exactly 3 elements. Now, by defining $T_j := S_j \cup \{3m+j\}$, we have $\sum_{i \in T_j} y_i = 4B + 4 = (B+1)|T_j|$. Hence, the answer to SUBSET AVERAGE is also yes.

On the other hand, assume that the answer to SUBSET AVERAGE is yes, then there exists an m-partition $\{T'_i\}_1^m \in \mathcal{P}_m([4m])$ such that, for each $1 \leq j \leq m$, $\sum_{i \in T'_j} y_i = (B+1)|T'_j|$. Since x_i's are positive, each T'_j contains at least one of the elements $y_{3m+1}, \ldots, y_{4m}$ and since there are m disjoint subsets T'_j's, each T'_j contains exactly one of them. Thus, by defining $S'_j := T'_j \backslash \{3m+1, \ldots, 4m\}$, we have $\sum_{i \in S'_j} x_i = B$. Hence, the answer to 3-PARTITION is also yes. This completes the reduction.

Step 2. SUBSET AVERAGE $\leq_m^p$ NCP^M.
In the second step, we give a reduction from SUBSET AVERAGE to NCP^M on weighted trees, where all the edge weights are equal to 1. Consider positive integers $y_1, \ldots, y_n$ with the average α and a positive integer $m \leq n$ as an instance of SUBSET AVERAGE. Let l be an arbitrary positive fixed integer and construct a weighted tree $T = (V, E, \omega, c)$ as follows (see Figure 1).

$$V := \{u, u_i, v_{ij} \mid i = 1, \ldots, n, \ j = 1, \ldots, l-1\},$$
$$E := \{uu_i, u_i v_{ij} \mid i = 1, \ldots, n, \ j = 1, \ldots, l-1\},$$
$$\omega(u) := n\alpha, \ \ \omega(u_i) := l y_i, \ \ \omega(v_{ij}) := \alpha, \ \forall\, 1 \leq i \leq n, \ 1 \leq j \leq l-1.$$

Also, let all the edge weights be equal to 1. The weighted tree T together with the constants $k := n(l-1) + m + 1$ and $N := 1/\alpha$ constitute an instance of NCP^M. By assuming the

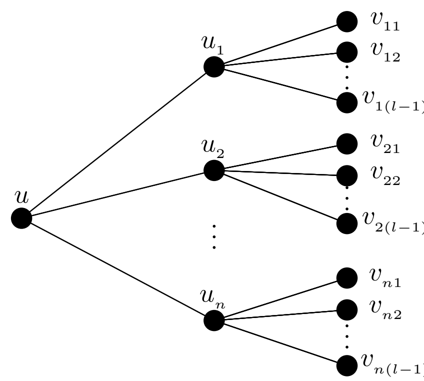


Figure 1: A weighted tree corresponding to an instance of SUBSET AVERAGE.

partition $\{T_i\}_1^m \in \mathcal{P}_m([n])$ as a positive answer to SUBSET AVERAGE, we define the k-partition

$$\{A_0\} \cup \{A_t\}_1^m \cup \{A_{ij} \ \ | \ \ 1 \le i \le n, 1 \le j \le l-1\} \in \mathcal{P}_k(V)$$

as follows,

$$A_0 := \{u\}, \ \ A_t := \{u_i | i \in T_t\}, \ \forall\ 1 \le t \le m, \ \ A_{ij} := \{v_{ij}\}, \ \forall\ 1 \le i \le n, 1 \le j \le l-1.$$

Now, we have

$$\frac{c(A_0)}{\omega(A_0)} = \frac{n}{n\alpha}, \ \ \frac{c(A_t)}{\omega(A_t)} = \frac{l|T_t|}{\sum_{i\in T_t} l y_i} = \frac{1}{\alpha}, \ \ \frac{c(A_{ij})}{\omega(A_{ij})} = \frac{1}{\alpha},$$

and consequently, the answer to NCP^M is also yes.

On the other hand, assume that $\{A'_i\}_1^k$ be a positive answer to NCP^M. We should find a positive answer to SUBSET AVERAGE. In this regard, we come up with a partition of $[n]$ into at least m subsets, each of which with an average equal to α (then, if it is necessary, we may merge some subsets and find an m-partition). Since $|V| = nl+1$, we have $|A'_j| \le n-m+1$ and there are at least m sets A'_j which has non-empty intersection with the set $\{u_i\}_1^n$. Now, define $T'_j := \{i |\ u_i \in A'_j\}$. Among T'_j's, the non-empty ones form a partition of the set $[n]$. We claim that the average of each set in this partition is equal to α. Fix j, where T'_j is non-empty and let $\sigma(T'_j) := \sum_{i\in T'_j} y_i$. Since $|A'_j| \le n-m+1$, we have

$$\frac{1}{\alpha} \ge \frac{c(A'_j)}{\omega(A'_j)} \ge \frac{l|T'_j| - (n-m)}{l\sigma(T'_j) + (2n-m-1)\alpha}. \tag{7}$$

Now, we choose l sufficiently larger than m, n, α, such that

$$\frac{|T'_j|}{\sigma(T'_j)} - \frac{l|T'_j| - (n-m)}{l\sigma(T'_j) + (2n-m-1)\alpha} < \frac{1}{n\alpha^2}. \tag{8}$$

Note that l depends only on n, m, α and does not depend on j and T'_j, because $|T'_j|$ and $\sigma(T'_j)$ are respectively bounded by n and $n\alpha$ (for instance one may choose $l = \alpha^3 n^2(3n-2m-1)$). Since $\sigma(T'_j) \le n\alpha$, Equations (7) and (8) yield

$$\frac{|T'_j|}{\sigma(T'_j)} < \frac{1}{\alpha} + \frac{1}{n\alpha^2} \le \frac{1}{\alpha} + \frac{1}{\alpha\ \sigma(T'_j)}.$$

Hence, $|T'_j|/\sigma(T'_j) \le 1/\alpha$ and this shows that the average of integers $(y_i\ :\ \ i \in T'_j)$ is at least α. Finally, since non-empty sets T'_j's form a partition of $[n]$, the average of integers $(y_i\ :\ \ i \in T'_j)$ is exactly equal to α. This completes the reduction and hence NCP^M is NP-complete in the strong sense. ■

Although Theorem 2 can be considered as an evidence for hardness of NCP^M for weighted trees, it turns out that the corresponding problem for subpartitions, i.e. IPP^M, is surprisingly a tractable problem. To prove this, we begin by the following lemma.

Lemma 2. *Given a weighted graph $G = (V, E, \omega, c)$ and integer $k \geq 2$, there exists a minimizing subpartition $\{A_i\}_1^k \in \mathcal{D}_k(V)$ attaining $\iota_k(G)$ such that the induced graph on each A_i is connected.*

Proof. Let $\{A_i\}_1^k$ be a minimizing subpartition achieving $\iota_k(G)$ and assume that the induced graph G on A_1 is not connected. Therefore $A_1 = A \sqcup B$, where there is no edges between A and B, we have

$$\min\left\{\frac{c(A)}{\omega(A)}, \frac{c(B)}{\omega(B)}\right\} \leq \frac{c(A) + c(B)}{\omega(A) + \omega(B)} = \frac{c(A_1)}{\omega(A_1)}.$$

Hence, we may remove one of the sets A or B from A_1, such that the resulting subpartition remains minimizing. By continuing this process, we can find a minimizing subpartition with connected components. ∎

Theorem 3. *There is a polynomial time algorithm that decides* IPP^M *for every weighted tree whose runtime is in $O(n)$.*

Proof. We prove a stronger version of the theorem. We assume that in addition to the vertex and the edge weight functions, ω, c, there exists another weight function $\gamma : V(T) \to \mathbb{Q}$ that intuitively can be considered as outgoing flows to the ground.[2] Therefore, for every $A \subset V$, we define the outgoing flow from A as $c(A) := \sum_{e \in E(A, A^c)} c(e) + \sum_{v \in A} \gamma(v)$ and we consider IPP^M for these new weighted trees. It is clear that when $\gamma(v) = 0$ for each $v \in V(T)$, the problem is the same as the classical IPP^M introduced before. Now, given a weighted tree $T = (V, E, \omega, c, \gamma)$ on n vertices, an integer $k \geq 2$ and a number N as the input of IPP^M, we perform the following algorithm on T to decide if $\iota_k^M(T) \leq N$ and to find a proof (affirmative subpartition) if there exists any.

Let $v \in V$ be an arbitrary vertex and consider the rooted tree T rooted at v. Sort the vertices of T as $v_1, \dots, v_n = v$, in a way that the vertices at level $i + 1$ precede the vertices at level i, for each i. This can be done in linear time by a breadth-first search.

Algorithm 1 Solve IPP^M

Initialize the set function $\eta : V \to P(V)$ by $\eta(v_i) := \{v_i\}$ for each $1 \leq i \leq n$.
Define $i = j := 1$.
while $j < k$ **and** $i \leq n$ **do**
  Let u be the unique parent of v_i and $e := uv_i \in E$ (if $i = n$, then define $c(e) := 0$)
  if $\gamma(v_i) + c(e) \leq N\omega(v_i)$ **then**
    $j \leftarrow j + 1$, $A_j \leftarrow \eta(v_i)$, $\omega(A_j) \leftarrow \omega(v_i)$, $c(A_j) \leftarrow c(e) + \gamma(v_i)$, $\gamma(u) \leftarrow \gamma(u) + c(e)$
  else if $\gamma(v_i) - c(e) < N\omega(v_i)$ **then**
    $\eta(u) \leftarrow \eta(u) \cup \eta(v_i)$, $\omega(u) \leftarrow \omega(u) + \omega(v_i)$, $\gamma(u) \leftarrow \gamma(u) + \gamma(v_i)$
  else {i.e. $\gamma(v_i) - c(e) \geq N\omega(v_i)$}
    $\gamma(u) \leftarrow \gamma(u) + c(e)$
  end if
end while
if $j = k$ **then**
  return YES and $\{A_1, \dots, A_k\}$
else
  return NO
end if

[2] It can also be considered as outgoing flows to the boundary in the setup of graphs with boundary (see Section 6).

Now, we prove the correctness of the algorithm. First, we adopt a couple of notions. We say two instances (G_1, k_1, N_1) and (G_2, k_2, N_2) are *equivalent* if the answer to IPP^M for both of them are the same. Given a weighted graph $G = (V, E, \omega, c, \gamma)$ and a vertex $v \in V$, $G\backslash v = (V', E', \omega', c', \gamma')$ denotes the weighted graph obtained from G by deleting the vertex v, where $\omega' := \omega|_{V'}$, $c' := c|_{E'}$ and for each $u \in V'$, $\gamma'(u) := \gamma(u) + \sum_{e=uv\in E} c(e)$. Furthermore, for an edge $e \in E$, G/e denotes the weighted graph obtained from G by contracting the edge e, where the weight of the new vertex is defined as sum of the weights of the two old vertices. (If it is necessary we put together multiple edges and sum up their weights to get a simple graph.) Let v be a leaf in $V(T)$ and $e = vu$ be the pendant edge.

1. If $\gamma(v) + c(e) \le N\omega(v)$, then (T, k, N) is clearly equivalent to $(T\backslash v, k-1, N)$.

2. If (1) is not the case and $\gamma(v) - c(e) < N\omega(v)$, then (T, k, N) is equivalent to $(T/e, k, N)$. To see this, let $\pi := \{A_i\}_1^k \in \mathcal{D}_k(V)$ be an affirmative answer for T, where the induced graph on each A_i is connected (see Lemma 2). If $u \notin \cup A_i$, then $v \notin \cup A_i$ (because A_i's are connected) and hence, π is also an affirmative answer for T/e. Now, assume that $u \in A_1$ and $v \notin \cup A_i$. Define $A_1' := A_1 \cup \{v\}$, then,

$$c(A_1') - N\omega(A_1') = c(A_1) - N\omega(A_1) + \gamma(v) - c(e) - N\omega(v) < 0.$$

   Thus, the answer to $(T/e, k, N)$ is also yes.

3. Finally, if $\gamma(v) - c(e) \ge N\omega(v)$, then (T, k, N) is equivalent to $(T\backslash v, k, N)$. To see this, as before let $\pi := \{A_i\}_1^k \in \mathcal{D}_k(V)$ be an affirmative answer for T, where the induced graph on each A_i is connected. If $v \notin \cup A_i$, there is nothing to prove. If $v \in A_1$, then $u \in A_1$ (because A_1 is connected). Define $A_1' := A_1 \backslash v$, then,

$$c(A_1') - N\omega(A_1') = c(A_1) - N\omega(A_1) - \gamma(v) + c(e) + N\omega(v) \le 0.$$

   Thus, the answer to $(T\backslash v, k, N)$ is also yes.

This shows that IPP^M for weighted trees is self-reducible. Moreover, note that the runtime of the algorithm is clearly of order $O(n)$. ■

For an optimization problem, a *fully polynomial time approximation scheme* (FPTAS) is an algorithm that takes an instance of the problem together with a number $\epsilon > 0$ and outputs a feasible solution within a factor $(1+\epsilon)$ of the optimal solution and its running time is bounded by a polynomial in the size of the instance and $1/\epsilon$. By using Algorithm 1 as well as a standard iterative method, we can find an FPTAS to approximate $\iota_k^M(T)$. Also, using Theorem 1, we can find polynomial time approximation algorithms for the parameters $\tilde{\iota}_k^M(T)$, $\iota_k^m(T)$ and $\tilde{\iota}_k^m(T)$.

Corollary 1. *Let T be a weighted tree and $2 \le k \le |V(T)|$ be an integer.*

(i) *There exists an* FPTAS *that approximates the parameter $\iota_k^M(T)$.*

(ii) *For every $\epsilon > 0$, there exists a polynomial time approximation algorithm that approximates the parameters $\tilde{\iota}_k^M(T)$, $\iota_k^m(T)$ and $\tilde{\iota}_k^m(T)$, within factors $k-1+\epsilon$, $k+\epsilon$ and $2k-2+\epsilon$, respectively.*

Proof. Given a weighted tree $T = (V, E, \omega, c)$, an integer $2 \le k \le |V|$ and a number $\epsilon > 0$, define $w_0 := \min_{v\in V} w(v)$, $W := \sum_{v\in V} w(v)$, $c_0 := \min_{e\in E} c(e)$ and $C := \sum_{e\in E} c(e)$. Therefore, $\iota_k^M(T)$ is within the interval $[2c_0/W, C/w_0]$. We start with this interval and do the following iteratively.

Let $[a_i, b_i]$ be the interval obtained in step i. Then, in step $i+1$, using Algorithm 1, check if $\iota_k^M(T) \le (a_i+b_i)/2$ and find an interval containing $\iota_k^M(T)$ whose length is $(b_i - a_i)/2$. We continue this process for t steps, where $t := \log(1/(2\epsilon)) + \log(CW/c_0w_0 - 2)$. Finally, we come to an interval $[a_t, b_t]$ containing $\iota_k^M(T)$ whose length is $(C/w_0 - 2c_0/W)/2^t = \epsilon\ 2c_0/W$. We output b_t as the approximation for $\iota_k^M(T)$. We have

$$\iota_k^M(T) \leq b_t = a_t + \epsilon \frac{2c_0}{W} \leq (1+\epsilon)\iota_k^M(T).$$

Also, the runtime of this algorithm is

$$O(nt) = O\left(n\left(\log(\frac{1}{2\epsilon}) + \log(\frac{CW}{c_0w_0} - 2)\right)\right),$$

and consequently, this is an FPTAS that approximates $\iota_k^M(T)$.
Part (ii) follows from Part (i), Theorem 1 and the fact that $\iota_k^m(T) \leq \iota_k^M(T) \leq k\ \iota_k^m(T)$. ∎

The next result (Theorem 4) shows that the approximation method previously used to approximate $\tilde{\iota}_k^M$ by ι_k^M can not be applied to approximate $\tilde{\iota}_k^m$ by ι_k^m, since, contrary to the max version, IPP^m appears to be an NP-complete problem for weighted trees. To prove this, first we need the following simple lemma that will also be used in the proof of Theorem 6.

Lemma 3. *Let $G = (V, E, \omega, c)$ be a connected weighted graph and $S = \{v_1, \ldots, v_s\} \subset V$ be a fixed subset of vertices. Define $W := \sum_{u \in V \setminus S} \omega(u)$, $C := \sum_{e \in E} c(e)$ and $c_0 := \min_{e \in E} c(e)$. If $s \leq k \leq |V|$ is an integer and for each $1 \leq i \leq s$, $\omega(v_i) \geq (2CW/c_0)$, then there exists a minimizing partition (resp. subpartition) achieving $\tilde{\iota}_k(G)$ (resp. $\iota_k(G)$) in which all the vertices $v_1, \ldots, v_s$ are in different parts. Also, none of the vertices in S are k-outlier.*

Proof. We prove the lemma for $\iota_k^m(G)$. The other cases are similar. Let $\{A_i\}_1^k$ be a minimizing subpartition achieving $\iota_k^m(G)$ and assume that A_1 contains two vertices in S, say v_1, v_2 and $\omega(v_1) \geq \omega(v_2) \geq (2CW/c_0)$. Then there is a subset, say A_2, which contains no vertex of S. Now, move v_2 from A_1 to A_2 and call the new subsets A_1' and A_2'. Thus

$$\frac{c(A_1')}{\omega(A_1')} + \frac{c(A_2')}{\omega(A_2')} \leq \frac{2C}{\omega(v_2)} \leq \frac{c(A_2)}{\omega(A_2)} < \frac{c(A_1)}{\omega(A_1)} + \frac{c(A_2)}{\omega(A_2)}.$$

This contradicts the fact that $\{A_i\}_1^k$ is a minimizing subpartition. Therefore, all of the vertices $v_1, \ldots, v_s$ are in different parts. Moreover, if a vertex v_i does not lie in the subpartition, we may add it to a subset A_j which has no intersection with S to find a new subpartition contradicting the minimality of $\{A_i\}_1^k$. Hence, no vertex in S can be a k-outlier. ∎

Theorem 4. *The problems* IPP^m *and* NCP^m *are NP-complete for weighted trees.*

Proof. We verify a reduction from the NP-complete problem EQUIPARTITION [12].

EQUIPARTITION

INSTANCE: $2n$ positive integers $x_1, \ldots, x_{2n}$ such that $\sum_{i=1}^{2n} x_i = 2B$.

QUERY: Is there a subset $I \subset [2n]$ such that $|I| = n$ and $\sum_{i \in I} x_i = B$?

Consider positive integers $x_1, \ldots, x_{2n}$ with the sum $2B$ as an instance of EQUIPARTITION. Define $Q := (1/2)\sum_{i=1}^{2n} x_i^2$ which is an integer and construct a weighted tree $T = (V, E, \omega, c)$, where $V := \{v_0, v_1, \ldots, v_{2n}, u_1, \ldots, u_{2n}\}$ and $E := \{v_0u_i, u_iv_i,\ i = 1, \ldots, 2n\}$. Also, let $k := 3n + 1$ and for arbitrary positive integers d, D, define the weight functions as follows.

$$\omega(v_0) := 2dB,\quad \omega(v_i) := 2D,\quad \omega(u_i) := 2x_i,\ \forall\ 1 \leq i \leq 2n,$$

$$c_i := c(v_0v_i) = c(u_iv_i) = x_i\left((d+1)^2B^2 + Q - Bx_i\right).$$

Suppose that d, D be sufficiently larger than B. Then by Lemma 3, none of the vertices $v_0, v_1, \ldots, v_{2n}$ are k-outlier. Also, if for some i, the vertex u_i is k-outlier, then we can move u_i to the set containing v_i, without increasing the normalized outgoing flow of that set. Thus, the tree T is k-geometric (i.e. $\iota_k(T) = \tilde{\iota}_k(T)$) and in every minimizing k-partition, the vertices $v_0, v_1, \ldots, v_{2n}$ lie in different parts. Moreover, suppose that D is sufficiently larger than d, then there exists a minimizing k-partition in which each vertex v_i forms a single part in the partition. Thus, the minimizing partition which achieves $\iota_k^m(T) = \tilde{\iota}_k^m(T)$ is of the form

$$\pi_I := \{ \{v_1\}, \ldots, \{v_{2n}\}, \{v_0, u_i, i \in I\}, \{u_j\}, j \notin I \},$$

for some subset $I \subset [2n]$ with $|I| = n$. Therefore, $k\ \iota_k^m(T) = k\ \tilde{\iota}_k^m(T) \le N$ if and only if there exists an n-subset $I \subset [2n]$, where

$$\sum_{i=1}^{2n} \frac{c_i}{\omega(v_i)} + \frac{\sum_{i=1}^{2n} c_i}{\omega(v_0) + \sum_{i \in I} \omega(u_i)} + \sum_{i \notin I} \frac{2c_i}{\omega(u_i)} \le N. \tag{9}$$

On the other hand,

$$\sum_{i=1}^{2n} c_i = \big((d+1)^2 B^2 + Q\big) \sum_i x_i - B \sum_i x_i^2 = 2(d+1)^2 B^3.$$

Consequently, Inequality (9) is equivalent to

$$\frac{(d+1)^2 B^3}{D} + \frac{(d+1)^2 B^3}{dB + \sum_{i \in I} x_i} + \sum_{i \notin I} \big((d+1)^2 B^2 + Q - B x_i\big) \le N.$$

If we define

$$N := n(d+1)^2 B^2 + nQ + dB^2 + \frac{(d+1)^2 B^3}{D}, \tag{10}$$

then by substituting N from (10) and simplifying, we have the following inequality.

$$(d+1)^2 B^2 \le \left(dB + \sum_{i \in I} x_i \right) \left(dB + \sum_{i \notin I} x_i \right).$$

Now, since $\sum_{i=1}^{2n} x_i = 2B$, we have $(dB + \sum_{i \in I} x_i)(dB + \sum_{i \notin I} x_i) \le (d+1)^2 B^2$ and equality holds if and only if there exists some I such that $\sum_{i \in I} x_i = \sum_{i \notin I} x_i = B$. Hence, $\iota_k^m(T) = \tilde{\iota}_k^m(T) \le N/k$ if and only if there exists some subset I with $|I| = n$, where $\sum_{i \in I} x_i = B$. This completes the proof. ∎

4 The Case of Fixed k

In this section we concentrate on the computation of the isoperimetric parameters when k is assumed to be a constant. In fact the main theorem that we shall prove in this section is the following.

Theorem 5. *Let $k \ge 2$ be a constant integer.*

(i) *There exists a polynomial time algorithm that computes parameters $\iota_k^M(T)$ and $\iota_k^m(T)$ for every weighted tree T, whose runtime is in $O(n^{\lfloor (3k-3)/2 \rfloor})$.*

(ii) *There exists a polynomial time algorithm that computes $\tilde{\iota}_3^M(T)$ for every weighted tree T, whose runtime is in $O(n^2)$.*

(iii) *If $k \ge 4$, there exists a polynomial time algorithm that computes $\tilde{\iota}_k^M(T)$ for every weighted tree T, whose runtime is in $O(n^{(2k^2-6k-3)})$.*

Note that the runtimes of the algorithms presented in Theorem 5 are exponential in k, but polynomial in n, when k is a constant. Nevertheless, this exponential inefficiency is likely to be unavoidable duo to Theorems 2 and 4.

In order to prove this theorem we go through two basic stages. Firstly, by proving Lemmas 4, 5 and 6 we restrict the search space of all k-subpartitions (or k-partitions) to a space of partitions with connected parts whose number of parts is bounded by a polynomial of k. Secondly, we provide a search procedure that generates all these partitions and for each partition computes the normalized outgoing flows of its parts in constant time (see lemma 7). This is done through adopting a succinct tree representation that allows constant time navigation operations on the corresponding tree.

To begin, we introduce the concept of the quotient of a graph $G = (V, E)$ with respect to a k-partition of V.

Definition 2. Given a weighted graph $G = (V, E, \omega, c)$ and a k-partition $\pi = \{A_i\}_1^k \in \mathcal{P}_k(V)$, for each $1 \leq i \leq k$, let $\{A_i^1, \ldots, A_i^{n_i}\}$ be the set of connected components of the induced graph of G on A_i. The *quotient graph* of G with respect to π, denoted by G/π, is defined to be a weighted graph $G/\pi = (V', E', \omega', c')$, where

$$V' := \{v_i^r : \ 1 \leq i \leq k, 1 \leq r \leq n_i\},$$

$$E' := \{v_i^r v_j^s : \ E(A_i^r, A_j^s) \neq \emptyset\},$$

$$\omega'(v_i^r) := \omega(A_i^r), \quad c'(v_i^r v_j^s) := \sum_{e \in E(A_i^r, A_j^s)} c(e).$$

It is clear that the quotient graph G/π is a minor of G as a graph. Thus, if G is planar, then G/π is planar as well. Moreover, if G is acyclic, then G/π is also acyclic. For a subset $F \subseteq E$, the graph obtained from G by deleting the edges in F, is denoted by $G \backslash F$. ♠

Lemma 4. *Let $G = (V, E, \omega, c)$ be a weighted graph and $\pi = \{A_i\}_1^k \in \mathcal{D}_k(V)$ be a minimizing subpartition for $\iota_k(G)$. Define the $(k+1)$-partition $\overline{\pi} := \{A_i\}_1^{k+1}$, where $A_{k+1} = V \backslash (\cup_1^k A_i)$ and let $G/\overline{\pi}$ be the quotient graph of G with respect to $\overline{\pi}$. Then, we have $\iota_k(G) = \iota_k(G/\overline{\pi})$. (Similar statements are also true for the other parameters $\tilde{\iota}_k^m$ and $\tilde{\iota}_k^M$.)*

Proof. We prove the lemma for ι_k^m. The other cases follow similarly. Let V', c', ω' be as in Definition 2 and for every $1 \leq i \leq k$, define $A_i' := \{v_i^r : \ 1 \leq r \leq n_i\}$. Then,

$$\iota_k^m(G) = \frac{1}{k} \sum_{i=1}^{k} \frac{c(A_i)}{\omega(A_i)} = \frac{1}{k} \sum_{i=1}^{k} \frac{c'(A_i')}{\omega'(A_i')} \geq \iota_k^m(G/\overline{\pi}).$$

Also, if $\pi' = \{B_i'\}_1^k \in \mathcal{D}_k(V')$ is a minimizing subpartition for $\iota_k^m(G/\overline{\pi})$ and $B_i := \cup\{A_j^s : v_j^s \in B_i'\}$, then,

$$\frac{1}{k} \sum_{i=1}^{k} \frac{c'(B_i')}{\omega'(B_i')} = \frac{1}{k} \sum_{i=1}^{k} \frac{c(B_i)}{\omega(B_i)} \geq \iota_k^m(G).$$

■

Lemma 5. *Let $G = (V, E, \omega, c)$ be a weighted graph and $2 \leq k \leq |V|$ be an integer. Then, there exists a subpartition $\pi = \{A_i\}_1^k \in \mathcal{D}_k(V)$ attaining $\iota_k(G)$ such that the number of connected components of $G \backslash \cup_i E(A_i, A_i^c)$ is*
(i) *at most $\lfloor (3k-1)/2 \rfloor$, if G is acyclic.*
(ii) *at most $3k - 4$, if G is planar.*

Proof. Consider the nonempty set $\mathcal{C}_k(V)$ of all the minimizing subpartitions $\{A_i\}_1^k \in \mathcal{D}_k(V)$ where the induced graph on each A_i is connected (see Lemma 2), and for each such subpartition, let $\{A_{k+1}^1, \ldots, A_{k+1}^d\}$ be the set of all connected components of the induced graph on $A_{k+1} := V \backslash (\cup_1^k A_i)$. Now, choose an extremal subpartition $\pi = \{A_i\}_1^k \in \mathcal{C}_k(V)$ for which d is minimized. Let $\overline{\pi} := \{A_i\}_1^{k+1}$ and $V(G/\overline{\pi}) = \{v_1, \ldots, v_k, v_{k+1}^1, \ldots, v_{k+1}^d\}$ as in Definition 2. First, we claim that $\deg(v_{k+1}^p) \geq 3$ for each $1 \leq p \leq d$. By contradiction, assume that $\deg(v_{k+1}^p) \leq 2$. Then, A_{k+1}^p is connected to at most two subsets in π, say A_1, A_2. Without loss of generality, assume that $c(A_{k+1}^p, A_1) \geq c(A_{k+1}^p, A_2)$. Define $B_1 := A_1 \cup A_{k+1}^p$ and $B_i := A_i$ for all $2 \leq i \leq k$. Therefore, $\pi' = \{B_i\}_1^k \in \mathcal{D}_k(G)$ is a subpartition and

$$\frac{c(B_1)}{\omega(B_1)} = \frac{c(A_1) - c(A_1, A_{k+1}^p) + c(A_{k+1}^p, A_2)}{\omega(A_1) + \omega(A_{k+1}^p)} < \frac{c(A_1)}{\omega(A_1)},$$

that contradicts the minimality of π. Hence, $\deg(v_{k+1}^p) \geq 3$, for each $1 \leq p \leq d$ and the set of vertices of $G/\overline{\pi}$ with degree less than 3 is a subset of $\{v_1, \ldots, v_k\}$.

Let G' be the graph obtained from $G/\overline{\pi}$ by deleting all the edges $e = v_iv_j \in E(G/\overline{\pi})$, for every $1 \leq i, j \leq k$. Then,

$$|E(G')| = \sum_{p=1}^{d} \deg(v_{k+1}^p) \geq 3(|V(G')| - k). \tag{11}$$

On the other hand, if G is acyclic, then G' is also acyclic and $|E(G')| \leq |V(G')| - 1$. This fact along with (11) yields $|V(G/\overline{\pi})| = |V(G')| \leq (3k-1)/2$.
Now, if G is planar, then G' is also planar. Furthermore, G' is bipartite with independent parts $\{v_1, \ldots, v_k\}$ and $\{v_{k+1}^1, \ldots, v_{k+1}^d\}$. Therefore, G' is a bipartite planar graph and $|E(G')| \leq 2|V(G')| - 4$. This fact along with Inequality (11) yields $|V(G/\overline{\pi})| = |V(G')| \leq 3k - 4$. ■

Lemma 6. *Let $T = (V, E, \omega, c)$ be a weighted tree and $3 \leq k \leq |V|$ be an integer. Then, there exists a minimizing partition $\pi = \{A_i\}_1^k \in \mathcal{P}_k(V)$ for $\tilde{\imath}_k^M(T)$ such that the number of connected components of $T \backslash \cup_i E(A_i, A_i^c)$ is at most* $\max\{2k^2 - 6k - 2, k\}$.

Proof. Let $\pi = \{A_i\}_1^k \in \mathcal{P}_k(V)$ be a minimizing k-partition achieving $\tilde{\imath}_k^M(T)$ for which the number of vertices of T/π is minimal. Let $\{A_i^1, \ldots, A_i^{n_i}\}$ be the set of connected components of the induced graph T on A_i and V' be the set of vertices of T/π as in Definition 2. For each i, partition the set $[n_i]$ into two subsets L_i and L_i^c, where $L_i := \left\{r \mid 2 \leq r \leq n_i,\ \frac{c(A_i^r)}{\omega(A_i^r)} \geq \frac{c(A_i)}{\omega(A_i)}\right\}$.

Firstly, we prove that for each $r \in L_i$, $\deg(v_i^r) \geq 3$. By contradiction, assume that $r \in L_i$ and $\deg(v_i^r) \leq 2$. Therefore, A_i^r is connected to at most two sets, say A_j^s, A_l^t. Without loss of generality assume that $c(A_i^r, A_j^s) \geq c(A_i^r, A_l^t)$. Now, let $B_i := A_i \backslash A_i^r$, $B_j := A_j \cup A_i^r$ and $B_h := A_h$ for $h \neq i, j$. Thus, $\{B_i\}_1^k$ is a k-partition of V and since $r \in L_i$,

$$\begin{aligned}
\frac{c(B_i)}{\omega(B_i)} &= \frac{c(A_i) - c(A_i^r)}{\omega(A_i) - \omega(A_i^r)} \leq \frac{c(A_i)}{\omega(A_i)}, \\
\frac{c(B_j)}{\omega(B_j)} &= \frac{c(A_j) - c(A_i^r, A_j^s) + c(A_i^r, A_l^t)}{\omega(A_j) + \omega(A_i^r)} < \frac{c(A_j)}{\omega(A_j)}.
\end{aligned}$$

Hence, $\max_i \left(\frac{c(B_i)}{\omega(B_i)}\right) \leq \tilde{\imath}_k^M(T)$ that contradicts the minimality of $|V(T/\pi)|$. Therefore, $\deg(v_i^r) \geq 3$, as long as $r \in L_i$. Moreover, the above argument shows that if $k = 3$, then for each i, $L_i = \emptyset$ and $|V(T/\pi)| = k = 3$.

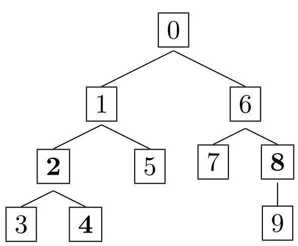
Secondly, provided $k \geq 4$, we prove that for each i, $|L_i^c| \leq k-3$. By contradiction, assume that $|L_1^c| \geq k-2$ and define $A' := A_2$ and $A'' := \cup_{3 \leq i \leq k} A_i$. For each $r \in L_1$, as before, we transfer the vertices in A_1^r into A' or A'', without increasing the normalized outgoing flow of these subsets. Call the new subsets as B' and B''. Hence, $\pi' := \{B', B'', A_1^r \mid r \in L_1^c\}$ is a k-partition that achieves $\tilde{\imath}_k^M(T)$ whereas $|V(T/\pi')| < |V(T/\pi)|$. This contradicts the minimality of $|V(T/\pi)|$ and therefore, $|L_i^c| \leq k-3$, for each i, whenever $k \geq 4$.

These facts show that the number of vertices in $V' = V(T/\pi)$ whose degrees are less than 3, is at most $k(k-3)$. Hence,

$$2(|V'|-1) = 2|E(T/\pi)| = \sum_{v \in V'} \deg(v) \geq 3(|V'| - k(k-3)) + k(k-3),$$

and consequently, $|V(T/\pi)| \leq 2k(k-3) - 2$, as long as $k \geq 4$. ■

Given a rooted tree $T = (V, E)$, one can represent T by a string of $2n$ balanced parentheses, ordered from 1 to $2n$, in which the matched pairs of these parentheses are in one to one correspondence with the vertices. Based on this correspondence, one may define a labeling of the vertex set[3] in a way that the ith open parenthesis corresponds to the vertex with the label $i-1$. Therefore, 0 is the root of the tree and we add this labeling to the string of parentheses along with its ordering to form a representation called the *balanced parenthesis representation* (or the BP representation) of T. An example of such a representation is depicted in Figure 2. The (induced) BP representation of a subset $X \subset V \backslash \{0\}$ is defined to be the subarray of the BP representation of T consisting of columns corresponding to the vertices in $X \cup \{0\}$. It is easy to check, using a DFS algorithm, that one may extract the BP representation of a rooted tree in linear time.



positition	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
parenthesis	(	(	(	(	)	(	)	)	(	)	)	(	(	)	(	(	)	)	)	)
label	0	1	2	3	3	4	4	2	5	5	1	6	7	7	8	9	9	8	6	0

positition	1	3	6	7	8	15	18	20
parenthesis	(	(	(	)	)	(	)	)
label	0	2	4	4	2	8	8	0

Figure 2: The BP representations of a rooted tree and the subset $\{2, 4, 8\}$.

In [20] it is proved that for every $2n$ balanced parentheses one may effectively construct a succinct representation, using $2n + o(n)$ bits, in such a way that the following navigational operations can be performed in constant time. Given the position of an open parenthesis,

- find the position of the its matched closing parenthesis, and vise versa.
- find the number i where this parenthesis is the ith open parenthesis of the sequence.
- find the position of next open parenthesis.

Henceforth, by abuse of language, we may assume that in any given BP representation of a rooted tree the above navigational operations can be performed in constant time (this clearly can be done using the succinct version of the given BP representation).

Lemma 7. *Let $\mathcal{R}(T)$ and $\mathcal{R}(X)$ be, respectively, the BP representation of the rooted tree $T = (V, E)$ on n vertices and the subset of vertices $X = \{a_1, \ldots, a_t\} \subset \{1, \ldots, n-1\}$ with $a_1 \leq a_2 \leq \ldots \leq a_t$. Define i_0 to be the largest number i such that $a_i + 1 \notin X$. Then, we can find the BP representation of the set $\{a_1, \ldots, a_{i_0-1}, a_{i_0}+1, a_{i_0}+2, \ldots, a_{i_0}+t-i_0+1\}$ in time $O(t)$.*

[3]Hereafter, each vertex is identified with its label.

Proof. The proof is a direct consequence of applying constant time navigational operations. In particular, we find the position of the vertex a_{i_0} in $\mathcal{R}(X)$, the position of the next open parenthesis with the label $a_{i_0}+1$ in $\mathcal{R}(T)$ and the position of the its corresponding closed parenthesis in $\mathcal{R}(T)$. Then in $\mathcal{R}(X)$, we exclude the columns with label a_{i_0} and insert the columns of $\mathcal{R}(T)$ with labels $a_{i_0}+1$ according to the prescribed ordering. The claim is proved by application of a series of this procedure to the representation $\mathcal{R}(X)$. ■

Proof of Theorem 5. Now, we are ready to provide algorithms to compute $\iota_k^m(T)$, $\iota_k^M(T)$ and $\tilde{\iota}_k^M(T)$ for a weighted tree T and to find the corresponding minimizing partitions and subpartitions. We use Lemmas 5, 6, 7 and the fact that removing t edges from a tree yields a forest with exactly $t+1$ connected components.

Let $T=(V,E,\omega,c)$ be a weighted rooted tree on the vertex set $V=\{0,1,\ldots,n-1\}$ rooted at 0. For each vertex $v\neq 0$, let $\mathtt{epar}(v)$ be the edge uv where u is the parent of v in T and for every $v\in V$ let T_v stand for the subtree of T rooted at v. Also, define

$$\overline{\omega}(v)\stackrel{\text{def}}{=}\omega(V(T_v)),\qquad \overline{c}(v)\stackrel{\text{def}}{=}\sum_{u\text{ is a child of }v}c(vu).$$

By traversing the vertices upwards, we find and save the quantities $\overline{\omega}(v)$ and $\overline{c}(v)$ for every $v\in V$ in time $O(n)$. Our algorithm is as follows.

Algorithm 2 Compute $\iota_k^m(T)$, $\iota_k^M(T)$

1: Let $I^M=I^m:=\sum_{e\in E}c(e)$ and $a_0:=0$.
2: **for** $t:=k-1$ **to** $\lfloor(3k-3)/2\rfloor$ **do**
3: Let $(a_1,a_2,\ldots,a_t)\leftarrow(1,2,\ldots,t)$, generate the BP representation of the subset $X=\{a_1,\ldots,a_t\}$ and its corresponding tree T' on the vertex set $X\cup\{0\}$.
4: **while** $(a_1,a_2,\ldots,a_t)\neq((n-t+1),\ldots,(n-2),(n-1))$ **do**
5: For each $0\le i\le t$, compute $\omega_i:=\overline{\omega}(a_i)-\sum_{a_j}\overline{\omega}(a_j)$ and $c_i:=c(ua_i)+\sum_{a_j}\overline{c}(a_j)$, where the sums run over all children a_j of a_i in T' and u is the parent of a_i in T. Also, compute the quantities $f_i:=c_i/\omega_i$ for each $0\le i\le t$.
6: Sort $f_0,\ldots,f_t$ in increasing order, say $f_{n_0}\le f_{n_1}\le\ldots\le f_{n_t}$ and define $J^M:=f_{n_{k-1}}$ and $J^m:=(f_{n_0}+\ldots+f_{n_{k-1}})/k$.
7: **if** $J^M<I^M$ (resp. $J^m<I^m$) **then**
8: $I^M\leftarrow J^M$ (resp. $I^m\leftarrow J^m$) and define $\pi^M:=(a_{n_0},\ldots,a_{n_t})$ (resp. $\pi^m:=(a_{n_0},\ldots,a_{n_t})$).
9: **end if**
10: Define i_0 to be the largest number i such that $a_i+1\notin X$ and set

$$(a_1,a_2,\ldots,a_t)\leftarrow(a_1,\ldots,a_{i_0-1},a_{i_0}+1,a_{i_0}+2,\ldots,a_{i_0}+t-i_0+1)$$

11: Using Lemma 7, find the BP representation of the new subset and its corresponding tree T'.
12: **end while**
13: **end for**
14: Define $F\subset E$, as $F:=\cup_{0\neq v\in\pi^M}\mathtt{epar}(v)$ and compute the connected components of $T\backslash F$. For each $0\le i\le k-1$, let A_i be the set of vertices of the connected component containing a_{n_i} to form the k-subpartition $\{A_0,\ldots,A_{k-1}\}$. Do the same thing for π^m to obtain the k-subpartition $\{B_0,\ldots,B_{k-1}\}$.
return I^M and $\{A_i\}_0^{k-1}$ and also I^m and $\{B_i\}_0^{k-1}$

We can also compute $\tilde{\iota}_k^M(T)$ by a slight modification of Lines 2, 6–9 and 14 in Algorithm 2 as follows.

Algorithm 3 Compute $\tilde{\iota}_k^M(T)$

$\vdots$

for $t := k-1$ **to** $\max\{2k^2-6k-3, k-1\}$ **do**

$\quad\vdots$

$\quad$ Consider all possible proper k-coloring of the tree T' by colors $0, 1, \ldots, k-1$ with color classes $\mathcal{C} = \{C_0, \ldots, C_{k-1}\}$ and define $g_i(\mathcal{C}) := (\sum_{j \in C_i} c_j)/(\sum_{j \in C_i} \omega_j)$, $0 \leq i \leq k-1$ and $J^M := \min_{\mathcal{C}} \max_i g_i(\mathcal{C}) = \max_i g_i(\mathcal{C}_0)$ for some $\mathcal{C}_0$.

$\quad$ **if** $J^M < I^M$ **then**

$\quad\quad$ $I^M \leftarrow J^M$ and define $\mathcal{C}^M := \mathcal{C}_0$.

$\quad$ **end if**

$\quad\vdots$

end for

For $\mathcal{C}^M = \{C_0^M, \ldots, C_{k-1}^M\}$, define $F \subset E$, as $F := \cup_{i=0}^{k-1} \cup_{0 \neq v \in C_i^M} \texttt{epar}(v)$ and compute the connected components of $T \backslash F$ as $A_0, \ldots, A_t$. For each $0 \leq i \leq k-1$, define $B_i := \cup_{j \in C_i^M} A_j$.

return I^M and $\{B_i\}_0^{k-1}$

Now, it is easy to verify that by Lemma 7 all computations appearing within the **while** loop are performed in constant time (for constant k). Also, all computations outside the **for** loop are performed in linear time. Hence, the runtime of Algorithm 2 is of order $O(n^{\lfloor (3k-3)/2 \rfloor})$ and the runtime of Algorithm 3 is of order $O(n^2)$ when $k = 3$ and $O(n^{2k^2-6k-3})$ when $k \geq 4$. ∎

Theorem 5 as a generalization of B. Mohar's result for $k = 2$ [19] shows that for every fixed integer $k \geq 2$, computing the mentioned kth isoperimetric parameters is polynomially solvable for weighted trees. However, the following theorem shows that this result can not be generalized to the case of weighted graphs with bounded tree-width (for the general background and definition of tree-width, see e.g. [9] and references therein).

Theorem 6. *For every fixed integer $k \geq 2$, IPP_k and NCP_k (in both max and mean versions) are NP-complete for bipartite weighted graphs with tree-width two.*[4]

Proof. First we show that it is enough to prove the theorem for $k = 2$. For this, assume that $k > 2$ is an integer and G is a weighted graph. Add $k-2$ new isolated vertices of weight 1 to obtain a new weighted graph G'. For every k-subpartition of $V(G')$, there are two subsets completely included in $V(G)$. Thus, solving IPP_2 (equivalently NCP_2) for the graph G is equivalent to solving IPP_k and NCP_k for the graph G'. Henceforth, we concentrate on NCP_2^M, mentioning that the proof of the mean version is similar.

Consider the following NP-complete problem in the class of KNAPSACK problems, known as the PARTITION problem [12].

PARTITION

INSTANCE: n positive integers $x_1, \ldots, x_n$ such that $\sum_{i=1}^n x_i = 2B$.

QUERY: Is there a subset $I \subset [n]$ such that $\sum_{i \in I} x_i = B$?

We shall propose a polynomial reduction from PARTITION to NCP_2^M. Let $x_1, \ldots, x_n$ be n positive integers where $\sum_{i=1}^n x_i = 2B$. Then, define the bipartite weighted graph G as follows.

$$V(G) := \{u_1, u_2, v_1, \ldots, v_n\}, \quad E(G) := \{u_1 v_i, u_2 v_i, 1 \leq i \leq n\},$$

[4] Note that tree-width at most two implies planarity.

$$\omega(u_1) = \omega(u_2) = M, \quad \omega(v_i) := x_i, \ \forall\ 1 \leq i \leq n,$$

where M is an arbitrary positive integer. Also, let all the edge weights be equal to 1. It is clear that the graph G has tree-width equal to 2. Assume M is sufficiently larger than B, then by Lemma 3, there exists a minimizing 2-partition (A_1, A_2) achieving $\tilde{\iota}_2(G)$ where u_1 and u_2 are in different parts. Thus

$$\tilde{\iota}_2^M(G) = \max\left\{\frac{c(A_1)}{\omega(A_1)}, \frac{c(A_2)}{\omega(A_2)}\right\} = \max\left\{\frac{n}{M + \sum_{v_i \in A_1} x_i}, \frac{n}{M + \sum_{v_i \in A_2} x_i}\right\}.$$

Hence, $\iota_2^M(G) \leq n/(M + B)$ if and only if $\sum_{v_i \in A_1} x_i = B$. This completes the proof. ■

5 The Unitarization Process

In this section we establish a machinery to convert the hardness results from weighted graphs to unweighted (simple) graphs, i.e. graphs whose all the vertex and edge weights are equal to 1. In fact this method that we call the *unitarization* process, is a polynomial reduction and will be used to prove some hardness results for unweighted graphs and trees. Define the class $\mathcal{ISO}$ to be the set of all problems IPP, NCP, IPP$_k$ and NCP$_k$ for the maximum and the mean version.

Proposition 1. If P is a problem in the class $\mathcal{ISO}$ which is NP-complete in the strong sense for weighted graphs, then it is NP-complete for unweighted (simple) graphs as well.

Proof. We prove the proposition for NCP^M and the other cases are similar. Assume that NCP^M is NP-complete in the strong sense and let $G = (V, E, \omega, c)$ together with the integer $k \geq 2$ and the number $N = M/L$ be an instance of NCP^M, where all the weights and integers M, L are given in unary codes. We apply a *unitarization process* on G which is a polynomial reduction to obtain a simple graph G' with all the weights equal to 1 and a constant N', such that for NCP^M, (G, k, N) is a positive instance if and only if (G', k, N') is a positive instance. This implies the NP-completeness of NCP^M for unweighted graphs. The process is described in two steps.

Step 1. Unitarization of the vertex weights.
In this step, we propose a method to make all the vertex weights equal to 1. First, multiply all the vertex weights by a sufficiently large constant χ such that for every vertex $u \in V$, $\chi\omega(u) \geq \sum_{e=uv\in E} c(e)$. Then, for every $A \subset V$, we have $c(A)/\omega(A) \leq \chi$. Now, to construct the graph $G' = (V', E', c)$ from G, for each vertex $u \in V$, add a set W_u of exactly $\chi\omega(u) - 1$ new vertices and join all of the vertices in W_u to u (see Figure 3). Furthermore, let the new edges ux, $x \in W_u$, have weights equal to 1. We claim that $\tilde{\iota}_k^M(G) = \chi\tilde{\iota}_k^M(G')$. Let $\{A_i\}_1^k$ be a minimizing partition for $\tilde{\iota}_k^M(G)$. Then, by defining $A_i' := A_i \cup (\cup_{u\in A_i} W_u)$, it is clear that $c(A_i')/|A_i'| = (1/\chi)c(A_i)/\omega(A_i)$. Therefore, $\tilde{\iota}_k^M(G') \leq (1/\chi)\tilde{\iota}_k^M(G)$. To prove the equality, let $\{B_i'\}_1^k$ be a minimizing partition achieving $\tilde{\iota}_k^M(G')$. For a vertex $u \in V$, assume $u \in B_i'$, for some i. If there exists $x \in W_u$, such that $x \in B_j'$, for some $j \neq i$, then we transfer x from B_j' to B_i' and define $B_i'' := B_i' \cup \{x\}$ and $B_j'' := B_j' \backslash \{x\}$. Since $c(B_j')/|B_j'| \leq \tilde{\iota}_k^M(G') \leq (1/\chi)\tilde{\iota}_k^M(G) \leq 1$, we have

$$\frac{c(B_j'')}{|B_j''|} = \frac{c(B_j') - 1}{|B_j'| - 1} \leq \frac{c(B_j')}{|B_j'|}, \qquad \frac{c(B_i'')}{|B_i''|} = \frac{c(B_i') - 1}{|B_i'| + 1} \leq \frac{c(B_i')}{|B_i'|}.$$

By continuing this process, we get a minimizing partition $\{B_i''\}_1^k$ achieving $\tilde{\iota}_k^M(G')$ with the property that for every vertex $u \in V$, u and the vertices in W_u all are in the same part. Thus, by defining $B_i := V \cap B_i''$, we have $c(B_i)/\omega(B_i) = \chi c(B_i'')/|B_i''|$. Hence, $\tilde{\iota}_k^M(G) \leq \chi\tilde{\iota}_k^M(G')$.

It remains to let $N' := N/\chi$.

Step 2. Unitarization of the edge weights.
Let $n := |V|$ and assume that all the vertex weights are equal to 1 and replace every edge $e \in E$ by exactly $c(e)$ multiple edges. Then subdivide all the edges to obtain a simple graph G' and let the new edge weight function c' be the constant function 1 (see Figure 3). For each edge $e \in E$, let the set of new vertices obtained from the subdivisions be denoted by S_e and define $S := \cup_{e\in E} S_e$. Also, for a constant ψ, define the vertex weight function ω' to be equal to 1 on the set S and equal to ψ on the set V.

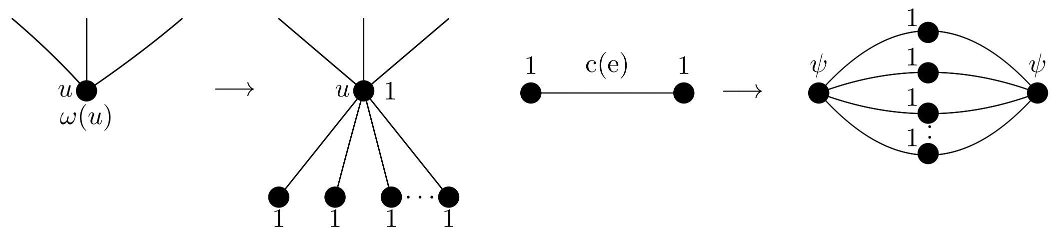


Figure 3: The vertex and edge gadgets used in the unitarization process.

We claim that if ψ is sufficiently larger than n, L and $|S|$, then $\tilde{\iota}_k^M(G) \le N$ if and only if $\tilde{\iota}_k^M(G') \le (N/\psi)$. For this, first assume that $\tilde{\iota}_k^M(G) \le N$ and let $\{A_i\}_1^k$ be a minimizing k-partition for $\tilde{\iota}_k^M(G)$. Define $A'_i := A_i \cup (\cup_{e\in E(A_i,A_j),1\le j\le i} S_e)$. It is clear that $c'(A'_i) = c(A_i)$ and $\omega'(A'_i) \ge \psi|A_i|$. Therefore, $\tilde{\iota}_k^M(G') \le (N/\psi)$. On the other hand, assume that $\tilde{\iota}_k^M(G') \le (N/\psi)$ and let $\{B'_i\}_1^k$ be a minimizing partition achieving $\tilde{\iota}_k^M(G')$. By defining $B_i := B'_i \cap V$, we have $c(B_i) \le c'(B'_i)$. Moreover, if ψ is sufficiently larger than n, L and $|S|$, then

$$\frac{c(B_i)}{|B_i|} < \frac{\psi c(B_i)}{\psi|B_i| + |S|} + \frac{1}{nL} \le \frac{\psi c'(B'_i)}{\omega'(B'_i)} + \frac{1}{nL}.$$

Thus,

$$\max_{1\le i\le k} \left\{\frac{c(B_i)}{|B_i|}\right\} = \frac{c(B_{i_0})}{|B_{i_0}|} < \psi\ \tilde{\iota}_k^M(G') + \frac{1}{nL} \le \frac{M}{L} + \frac{1}{nL} \le \frac{M}{L} + \frac{1}{L|B_{i_0}|}.$$

And consequently, $\tilde{\iota}_k^M(G) \le \max_i(c(B_i)/|B_i|) \le M/L = N$. This completes the second step.

Finally, by repeating Step 1, we may find a simple graph all of whose edge and vertex weights are equal to 1. Note that since the edge and vertex weights of G and also M, L are given in unary codes, the obtained simple graph is polynomial time computable. ■

By Theorem A (i), we know that NCP_2 is NP-complete for graphs with multiple edges. Thus, NCP_2 is NP-complete in the strong sense for weighted graphs. The following corollary is deduced from this fact along with Theorem 2 and Proposition 1. Part (i) can be seen as a generalization of B. Mohar's result (Theorem A (i)). Also, note that for Part (ii) we do not use Step 2 of the unitarization process, because within the reduction in the proof of Theorem 2, all the edge weights are equal to 1. Hence, this process preserves the property of being acyclic.

Corollary 2.
(i) *For every fixed* $k \ge 2$, IPP_k *and* NCP_k (*in both max and mean versions*) *are* NP*-complete for unweighted* (*simple*) *graphs.*
(ii) *The problem* NCP^M *is* NP*-complete for unweighted trees.*

6 Concluding Remarks

Our results show that the study of isoperimetric numbers and minimum normalized cuts on weighted trees is not only important because of its wide range of applications, but also the scope of weighted trees provide a vast arena to test the computational complexity of these problems in which these isoperimetric problems change their computational behavior by a very slight perturbation of conditions. This fact, on the one hand, is quite interesting from a complexity theoretic point of view, where one is quite eager to investigate problems close to the borders of the classes P and NP-complete, and on the other hand, is also interesting from the point of view of approximation algorithms for applications. In this regard, according to our results, *intuitively*, passing from taking the maximum to the mean or restricting the space of subpartitions to partitions will generally make the problem computationally harder. These observations provide enough evidence for the fact that the study of the following open problems ought to be challenging.

- Does there exist a polynomial time algorithm that given the number $k \geq 2$ and a weighted tree T, computes the parameter $\iota_k^M(T)$?
- Given a constant number $k \geq 4$, does there exist a polynomial time algorithm that computes the parameter $\tilde{\iota}_k^m$ for weighted trees? (It can be verified that an argument similar to what has appeared in the proof of Lemma 6 implies that this problem is solvable for $k = 3$ in time $O(n^2)$.)
- Determine the computational complexity of IPP^m and NCP^m for unweighted trees.
- Determine the computational complexity of IPP$_k$ and NCP$_k$ for bipartite planar unweighted graphs.

Also, it should be noted that from a parameterized complexity point of view Theorem 5 does not imply that the corresponding computational problems are in the class FPT with respect to the parameter k. Hence, the following question also seems to be interesting.

- Do the computational problems discussed in Theorem 5 fall in the class FPT as parameterized problems with respect to the parameter k?

Moreover, one may consider a number of different variants of isoperimetric problems on graphs and study their computational properties. As a couple of these variants we propose the following setups.

Firstly, we may consider all isoperimetric numbers and problems in the more general framework of graphs with boundary. For instance, the proof of Theorem 3 is presented in this framework where there is an extra weight function γ on vertices that represents the outgoing flows to the boundary. Therefore, Theorem 3 is also valid for the Dirichlet version of the problem IPP^M. In this regard, the study of computational aspects of the *Dirichlet isoperimetric problems* is an area to be explored (e.g. see [6, 8, 10]).

Secondly, considering the maximum and mean versions of the introduced parameters as $\|.\|_\infty$ and $\|.\|_1$ counterparts of the isoperimetric problem, respectively, it is interesting to study the $\|.\|_p$ versions of these parameters and the computational complexity of the corresponding problems. In this setting, it is important to try to characterize the properties that are responsible for the change of hardness from NP-completeness of IPP^m to the tractability of IPP^M in the limit.

Thirdly, the semisupervised variant of these partitioning problems can be formulated as the *multiterminal isoperimetric problems*, in which given a weighted graph along with k specified vertices $v_1, \ldots, v_k$, we look for a k-subpartition (k-partition) such that v_i's appear in different parts and the corresponding cost functions (see Definition 1) are minimized. For instance, using a similar argument as in the proof of Theorem 6, one may prove that for any $k \geq 2$ the multiterminal versions of IPP_k and NCP_k are NP-complete for weighted trees [15].

As another variant of these problems, one may focus on the approach through (k, b)-subpartitions (see [21, 22]) that can be considered as a combination of the max and the mean approach and follow the same line of study.

References

[1] Sanjeev Arora, Satish Rao, and Umesh Vazirani, *Expander flows, geometric embeddings and graph partitioning*, J. ACM **56** (2009), no. 2, Art. 5, 37. MR MR2535878

[2] Catherine Bandle, *Isoperimetric inequalities and applications*, Monographs and Studies in Mathematics, vol. 7, Pitman (Advanced Publishing Program), Boston, Mass., 1980. MR 572958 (81e:35095)

[3] Ronald I. Becker and Yehoshua Perl, *The shifting algorithm technique for the partitioning of trees*, Discrete Appl. Math. **62** (1995), no. 1-3, 15–34, Partitioning and decomposition in combinatorial optimization. MR 1355571 (96h:68080)

[4] Frédéric Chataigner, Liliane R. B. Salgado, and Yoshiko Wakabayashi, *Approximation and inapproximability results on balanced connected partitions of graphs*, Discrete Math. Theor. Comput. Sci. **9** (2007), no. 1, 177–192 (electronic). MR 2342251 (2008g:68039)

[5] Fan R. K. Chung, *Spectral graph theory*, CBMS Regional Conference Series in Mathematics, vol. 92, Published for the Conference Board of the Mathematical Sciences, Washington, DC, 1997. MR 1421568 (97k:58183)

[6] Fan R. K. Chung and Kevin Oden, *Weighted graph Laplacians and isoperimetric inequalities*, Pacific J. Math. **192** (2000), 257–273.

[7] Amir Daneshgar, Hossein Hajiabolhassan, and Ramin Javadi, *On the isoperimetric spectrum of graphs and its approximations*, J. Combin. Theory Ser. B **100** (2010), 390–412.

[8] Amir Daneshgar, Ramin Javadi and Laurent Miclo, *On nodal domains of finite reversible Markov processes and spectral decomposition of cycles*, Availabe at: http://hal.archives-ouvertes.fr/docs/00/59/85/89/PDF/cycles.pdf

[9] Reinhard Diestel, *Graph Theory*, Graduate Texts in Mathematics, 173, Springer, Heidelberg, 2010.

[10] Jozef Dodziuk, *Difference equations, isoperimetric inequality and transience of certain random walks*, Trans. Amer. Math. Soc. **284** (1984), 787–794.

[11] Gary William Flake, Robert E. Tarjan, and Kostas Tsioutsiouliklis, *Graph clustering and minimum cut trees*, Internet Math. **1** (2004), no. 4, 385–408. MR 2119992 (2005m:05210)

[12] Michael R. Garey and David S. Johnson, *Computers and intractability, a guide to the theory of np-completeness*, W. H. Freeman and Co., San Francisco, Calif., 1979. MR 519066 (80g:68056)

[13] Leo Grady, *Fast, quality, segmentation of large volumes isoperimetric distance trees*, Computer Vision, Lecture Notes in Comput. Sci., vol. 3953, Springer, Berlin / Heidelberg, 2006, pp. 449–462.

[14] L. H. Harper, *Global methods for combinatorial isoperimetric problems*, Cambridge Studies in Advanced Mathematics, vol. 90, Cambridge University Press, Cambridge, 2004. MR 2035509 (2005c:05002)

[15] Ramin Javadi, *Isoperimetric problems on graphs*, Ph.D. thesis, Department of mathematical sciences, Sharif university of technology, 2010, manuscript.

[16] Isabella Lari, Maurizio Maravalle, and Bruno Simeone, *Computing sharp bounds for hard clustering problems on trees*, Discrete Appl. Math. **157** (2009), no. 5, 991–1008. MR 2500717 (2010c:05131)

[17] Imre Leader, *Discrete isoperimetric inequalities*, Probabilistic combinatorics and its applications (San Francisco, CA, 1991), Proc. Sympos. Appl. Math., vol. 44, Amer. Math. Soc., Providence, RI, 1991, pp. 57–80. MR 1141923 (93j:60010)

[18] Alexander Lubotzky, *Discrete groups, expanding graphs and invariant measures*, Progress in Mathematics, vol. 125, Birkhäuser Verlag, Basel, 1994, With an appendix by Jonathan D. Rogawski. MR 1308046 (96g:22018)

[19] Bojan Mohar, *Isoperimetric numbers of graphs*, J. Combin. Theory Ser. B **47** (1989), no. 3, 274–291. MR 1026065 (90m:05087)

[20] J. Ian Munro and Venkatesh Raman, *Succinct representation of balanced parentheses and static trees*, SIAM J. Comput. **31** (2001/02), no. 3, 762–776 (electronic). MR 1896457 (2003b:68035)

[21] Hiroshi Nagamochi and Toshihide Ibaraki, *Algorithmic aspects of graph connectivity*, Encyclopedia of Mathematics and its Applications, vol. 123, Cambridge University Press, Cambridge, 2008. MR 2442690 (2009j:05002)

[22] Hiroshi Nagamochi and Yoko Kamidoi, *Minimum cost subpartitions in graphs*, Inform. Process. Lett. **102** (2007), no. 2-3, 79–84. MR 2292963 (2007k:05205)

[23] J. Sherman, *Breaking the multicommodity flow barrier for $O(\sqrt{\log n})$-approximations to sparsest cut*, arXiv:0908.1379 (2009).

[24] Jianbo Shi and Jitendra Malik, *Normalized cuts and image segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (1997), 888–905.

[25] P. Soundararajan and S. Sarkar, *An in-depth study of graph partitioning measures for perceptual organization*, IEEE Trans. Pattern Anal. Mach. Intell. **25** (2003), no. 6, 642–660.

[26] Vijay V. Vazirani, *Approximation algorithms*, Springer-Verlag, Berlin, 2001. MR 1851303 (2002h:68001)

[27] Xiaodong Wu, Danny Z. Chen, James J. Mason, and Steven R. Schmid, *Efficient approximation algorithms for pairwise data clustering and applications*, Internat. J. Comput. Geom. Appl. **14** (2004), no. 1-2, 85–104. MR 2067051 (2005b:68219)