Revisiting the Isoperimetric Graph Partitioning Problem

Sravan Danda, Aditya Challa, B S Daya Sagar, Laurent Najman

Revisiting the Isoperimetric Graph Partitioning Problem

Sravan Danda*, Aditya Challa†, B.S.Daya Sagar‡, and Laurent Najman§

Abstract. Isoperimetric graph partitioning which is also known the Cheeger cut is NP-Hard in its original form. In literature, multiple modifications to this problem have been proposed to obtain approximation algorithms for clustering applications. In the context of image segmentation, a heuristic continuous relaxation to this problem has yielded good quality results. This algorithm is based on solving a linear system of equations involving the Laplacian of the image graph. Further, the same algorithm applied to a maximum spanning tree (MST) of the image graph was shown to produce similar results at a much lesser computational cost. However, the data reduction step (i.e. considering a MST, a much sparser graph compared to the original graph) leading to a faster yet useful algorithm has not been analysed. In this article, we revisit the isoperimetric graph partitioning problem and rectify a few discrepancies in the simplifications of the heuristic continuous relaxation, leading to a better interpretation of what is really done by this algorithm. We then use the Power Watershed (PW) framework to show that is enough to solve the relaxed isoperimetric graph partitioning problem on the graph induced by Union of Maximum Spanning Trees (UMST) with a seed constraint. The UMST has a lesser number of edges compared to the original graph, thus improving the speed of sparse matrix multiplication. Further, given the interest of PW framework in solving the relaxed seeded isoperimetric partitioning problem, we discuss the links between the PW limit of the discrete isoperimetric graph partitioning and watershed cuts. We then illustrate with experiments, a detailed comparison of solutions to the relaxed seeded isoperimetric partitioning problem on the original graph with the ones on the UMST and a MST. Our study opens many research directions which are discussed in the conclusions section.

Key words. Image Segmentation, Isoperimetric Partitioning, Cheeger cut, Spectral Clustering, Power Watersheds.

AMS subject classifications. 90C05, 90C27, 94A08, 94A12

1. Introduction.

In this article, we consider the graph partitioning problem stated as - given an edge weighted graph $G = (V, E, w)$ with edge-weights reflecting similarity measure between adjacent nodes, find a 'suitable' partition of the finite set V into 2 subsets. There are, of course, several criteria to find a 'suitable' partition. One such criterion, which is the focus of this article is that of **isoperimetric partitioning**. This criterion arises from the classic isoperimetric problem - for a fixed area, find a region with minimum perimeter [10].

The isoperimetric graph partitioning problem also known the Cheeger cut problem is NP-Hard [25]. This problem is closely related to total variation (TV) minimization [24, 8] whose role is crucial to many inverse problems in computer vision. The Cheeger cut problem also has links with problems such as ratio cut [30] and normalized cut minimizatons [30] which belong to the family of spectral clustering methods [30]. In the recent past, the Cheeger cut

*Systems Science and Informatics Unit, Indian Statistical Institute (sravan8809@gmail.com, https://sites.google.com/site/sravandanda1988).

†Systems Science and Informatics Unit, Indian Statistical Institute (aditya.challa.20@gmail.com)

‡Systems Science and Informatics Unit, Indian Statistical Institute (bsdsagar@yahoo.co.uk)

§Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, Équipe A3SI, ESIEE Paris, France (laurentnajman@esiee.fr, http://www.laurentnajman.org/).

problem has been of interest to many researchers and multiple approximation algorithms have been proposed [24, 5, 18, 9, 21] .

One such approximation, which is the focus of this paper is a continuous relaxation to this problem proposed in [21]. Its application to image segmentation problem is discussed in [20]. In cases such as medical image segmentation, due to large data size, extremely fast algorithms are necessary. Thus, in [19], the authors propose a fast algorithm to obtain an approximate solution to the isoperimetric graph partitioning problem for medical image segmentation. A simple overview of the idea presented in [19] is - instead of solving the problem on the original graph, the authors in [19] construct a maximum spanning tree (MST), and solve the problem on this graph. This allows for orders of magnitude speed up in the algorithm.

However, a few questions remain - (i) What is the basis of the simplifications that led to the heuristic provided in [20, 21]. (ii) Why would solving the problem on a MST give similar results to the solution on the original graph. (iii) How close are these solutions? These questions were answered empirically in [19]. However, to our knowledge, in depth analysis into this was not done.

In this article, we aim to answer the questions highlighted in the previous paragraph by presenting a detailed analysis. In section 2, we review the isoperimetric graph partitioning problem and focus on the details of the algorithm provided in [20], rectifying a few discrepancies. This algorithm solves a seeded version of the isoperimetric partitioning problem, and we discuss the differences with the original formulation. In section 3, we calculate the limit of minimizers of the isoperimetric graph partitioning problem in the Power Watershed (PW) framework [27]. Specifically, to calculate the said limit of minimizers, we show that is enough to solve the isoperimetric graph partitioning problem on the graph induced by **Union of Maximum Spanning Trees (UMST)**. The UMST has a lesser number of edges compared to the original graph, thus improving the speed of sparse matrix multiplication[1].

Given the interest of the PW limit in solving the relaxed seeded isoperimetric graph partitioning problem, in section 4 we discuss the links between the PW limit of the discrete version (original formulation) and watershed cuts. In section 5, some experiments are performed to illustrate properties of the solutions to the relaxed seeded isoperimetric graph partitioning problem on each of the original, the UMST and MST graphs. In section 6, we provide some prospective research directions building on the ideas from this article.

Remark: For brevity and clarity, the proofs of all the results are moved to the appendix.

2. Isoperimetric Graph Partitioning Problem.

In this paper, $G = (V, E, w)$ denotes an edge-weighted graph where V denotes the set of vertices, E denotes the set of edges and $w : E \to \mathbb{R}^+$ denotes the weights assigned to each edge reflecting similarity between adjacent vertices. To simplify the notation, shorter expression w_{ij} is used instead of $w(e_{ij})$ to denote the weight of the edge between vertices i and j. Let $S \subset V$, then $\overline{S}$ denotes the complement of the set S.

Given a graph, its isoperimetric number is defined by

$$h_G = \inf_S \frac{|\partial S|}{min\{vol(S), vol(\overline{S})\}} \tag{1}$$

[1] Given a $n \times n$ matrix with m non-zero entries, the complexity of matrix-vector multiplication is $\mathcal{O}(m+n)$

where $S \subset V$, the boundary edges of the set S is denoted by $\partial S = \{e_{ij} \mid i \in S, j \in \overline{S}\}$. The sum of the edge weights on the boundary is denoted by $|\partial S|$. The volume of the subset S denoted by $vol(S)$ is given by the cardinality of the set S.

The indicator of a set $S \subset V$ is defined as the vector $\mathbf{x}(S) = (x_1, x_2, \cdots, x_{|V|})$ where

$$x_i = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Also, the Laplacian of an edge-weighted graph G is defined as

$$L_{ij} = \begin{cases} \sum_k w_{ik} & \text{if } i = j \\ -w_{ij} & \text{if } e_{ij} \in E \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Using the definitions above, the isoperimetric graph partitioning problem can be stated as the following optimization problem.

$$\begin{aligned} &\text{Find } \arg\min_{\mathbf{x}} && \frac{\mathbf{x}^t L \mathbf{x}}{\min\{\mathbf{x}^t \mathbf{1}, (\mathbf{1} - \mathbf{x})^t \mathbf{1}\}} \\ &\text{subject to} && x_i \in \{0, 1\} \quad \text{for all } i \end{aligned} \tag{4}$$

Here $\mathbf{1}$ indicates the column vector with all elements equal to 1.

Observe that $x_i = 0$ for all i is invalid since the denominator of the cost function is 0. Similarly, $x_i = 1$ for all i is also invalid. The optimization problem (4) is a NP-hard problem [25]. To address this, the authors in [20, 21] propose a continuous relaxation of the problem as in (5) and continue to solve the relaxed problem.

$$\begin{aligned} &\text{Find } \arg\min_{\mathbf{x}} && \frac{\mathbf{x}^t L \mathbf{x}}{\min\{\mathbf{x}^t \mathbf{1}, (\mathbf{1} - \mathbf{x})^t \mathbf{1}\}} \\ &\text{subject to} && x_i \in [0, 1] \quad \text{for all } i \end{aligned} \tag{5}$$

Then, every threshold of the solution $\mathbf{x}$ is examined and a partition (called an *optimal threshold*) with least isoperimetric ratio among them is chosen. The authors of [20, 21] directly proceed to a modification of problem (5), without discussing it. Before going to their proposal, we provide hereafter a formal analysis of (5).

An issue with optimization problem (5) is the fact that at $\mathbf{x} = t\,\mathbf{1}$, where $0 < t < 1$, the cost function takes the minimum value of 0. Thus, optimal solutions to optimization problem (5) are degenerate.

Adding to the above issue, given any partition of $V = S \cup \overline{S}$, one can find a suitable $\mathbf{x}$, which when thresholded results in this partition and is close to the optimal solution. That is, solutions to (5) are not robust. This implies that solving (5) cannot be used to obtain meaningful partitions. This is stated rigorously in proposition 2.2. Before stating the proposition, we need to define the notion of ϵ-optimal solution.

Definition 2.1. *Let P be the minimization problem with loss function L on the constraint set $S \subset \mathbb{R}^n$. Let $\mathbf{x}^* \in S$ be an optimal solution and $\epsilon > 0$ denote a constant. If $\mathbf{x} \in S$ satisfies $|L(\mathbf{x}) - L(\mathbf{x}^*)| < \epsilon$ then $\mathbf{x}$ is said to be ϵ-optimal for P.*

Proposition 2.2. *Let $\epsilon > 0$ be some constant and let $V = S \cup \overline{S}$ be any partition. Then, given the notation as before, one can find $\mathbf{x} \in [0,1]^{|V|}$ such that*

$$\frac{\mathbf{x}^t\, L\, \mathbf{x}}{\min\{\mathbf{x}^t\, \mathbf{1}, (\mathbf{1} - \mathbf{x})^t\, \mathbf{1}\}} < \epsilon \tag{6}$$

and an optimal threshold of $\mathbf{x}$ results in the partition $S \cup \overline{S}$.

One way to rectify this is to consider the seeded version of the problem, i.e. set the value of one of the vertices to be 0. In the context of image segmentation, this can be interpreted as setting one of the vertices to be in the background of the object. This is the approach proposed in [20, 21], however the authors of [20, 21] do not clearly state that this problem is actually different from the unseeded version.

Note that one cannot a-priori know a pixel which would belong to the background without extra knowledge. Thus, in practice, either a seed must be given or one can solve the problem for all possible seeds and pick the best solution (an approach not very practical).

The relaxed seeded isoperimetric graph partitioning problem is stated as

$$\begin{aligned} \text{Find } \arg\min_{\mathbf{x}} \quad & \frac{\mathbf{x}^t\, L\, \mathbf{x}}{\min\{\mathbf{x}^t\, \mathbf{1}, (\mathbf{1} - \mathbf{x})^t\, \mathbf{1}\}} \\ \text{subject to} \quad & x_j = 0 \text{ for some } j \\ & x_i \in [0,1], \text{ for all } i \neq j \end{aligned} \tag{7}$$

which can equivalently be stated as

$$\begin{aligned} \text{Find } \arg\min_{\mathbf{x}} \quad & \frac{\mathbf{x}^t\, L\, \mathbf{x}}{\mathbf{x}^t\, \mathbf{1}} \\ \text{subject to} \quad & x_j = 0 \text{ for some } j \\ & x_i \in [0,1] \text{ for all } i \neq j \\ & \mathbf{x}^t\, \mathbf{1} \leq \frac{|V|}{2} \end{aligned} \tag{8}$$

Note that using the constraint $x_j = 0$ and slack variables, the above problem can be further simplified to

$$\begin{aligned} \text{Find } \arg\min_{x} \quad & \mathbf{x}_{-j}^t L_{(-j,-j)} \mathbf{x}_{-j} \\ \text{subject to} \quad & (\mathbf{x}_{-j})_i \in [0,1] \text{ for all } i \\ & \mathbf{x}_{-j}^t\, \mathbf{1} = \frac{|V|}{2} \end{aligned} \tag{9}$$

where $L_{(-j,-j)}$ is the Laplacian of the graph with j^{th} column and row removed, and $\mathbf{x}_{-j}$ is the vector with j^{th} entry removed. Optimization problem (9) is henceforth referred to the *relaxed seeded isoperimetric paritioning problem.*

Using the idea of Lagrange multipliers, one can find the solution to the above problem as proposed in [21] by solving

$$L_{(-j,-j)} \mathbf{x}_{-j} = \mathbf{1} \tag{10}$$

The constants are ignored since, only relative values of the solution are of interest. Thus, finding a solution to the relaxed seeded isoperimetric partitioning problem is reduced to solving (10) [2].

Remark: An important property of the solution to seeded isoperimetric partitioning problem is a *continuity* property (discussed in detail in [21]). It states that, for any vertex v, there exists a path to the seed g (say) - $< v = v_0, v_1, \cdots, g >$, such that the solution x satisfies

$$x(v_i) \geq x(v_{i+1}) \tag{11}$$

i.e. there exists a descending path from the vertex v to the seed ($x(v)$ denotes the value of the solution at the vertex v). This property implies that the optimal component containing the seed is connected, which is important for practical purposes.

3. Calculating the limit of minimizers.

In this section, we are going to compute the limit of the minimizers of (9). The maximum spanning tree is instrumental in doing so. Recall that a maximum spanning tree (MST) of a graph $G = (V, E, w)$ is a connected subgraph of G spanning V, with no cycles such that

$$\text{weight of the MST} = \sum_{e_{ij} \in MST} w_{ij} \tag{12}$$

is maximized. The UMST is the weighted graph induced by the union of all the maximum spanning trees. In [19] the authors claim that instead of solving (10) on the Laplacian of the original graph, it is sufficient to solve the problem using the Laplacian of a MST of the graph. As a MST does not have any cycles, this allows for obtaining fast solution to (10). This was verified empirically in [19] but a detailed analysis of that claim is currently missing. We are going to undertake such an analysis using the Power Watershed framework.

Given $G = (V, E, w)$, a finite edge weighted graph, define an *exponentiated graph* by $G^{(p)} = (V, E, w^{(p)})$, where $w^{(p)}(e_{ij}) = (w(e_{ij}))^p$. In the rest of the article we assume that G has k distinct weights $w_1 < w_2 < \cdots < w_k$. Also assume that G is connected. G^{umst} denotes the graph (weighted) induced by the UMST.

3.1. Power Watershed Framework.

Let $\{Q_i(.)\}$ be a set of cost functions on $\mathbb{R}^n$ and $0 < \lambda_1 < \lambda_2 < \cdots < \lambda_k$ a set of constants. Define

$$Q(\mathbf{x}) = \sum_{i=1}^{k} \lambda_i Q_i(\mathbf{x}) \tag{13}$$

and

$$Q^{(p)}(\mathbf{x}) = \sum_{i=1}^{k} \lambda_i^p Q_i(\mathbf{x}) \tag{14}$$

[2] Note that for the unseeded version, the equivalent of (10) is $L\,\mathbf{x} = \mathbf{1}$. In [20] it was stated that the reason to consider the seeded problem is - $L\,\mathbf{x} = \mathbf{1}$ has several solutions. However, multiplying with $\mathbf{1}^t$ on both sides of $L\,\mathbf{x} = \mathbf{1}$, one can easily see that the LHS is equal to 0, while the RHS is greater than 0. This implies that the system of equations $L\,\mathbf{x} = \mathbf{1}$ has no solutions. Here we provide a better understanding for solving the relaxed seeded isoperimetric partitioning problem.

Let $\mathbf{x}_p^*$ denote a minimizer of $Q^{(p)}$. We are interested in computing a limit $\mathbf{x}^*$ of minimizers $(\mathbf{x}_p^*)_{p>0}$ as $p \to \infty$.

In [27], the author provides a theory to calculate a limit of minimizers. In simple terms, given a specific structure of the cost function, one can compute a limit of minimizers by iteratively calculating the set of minimizers at every scale starting from the highest scale λ_k (λ_i provides the notion of scale here). Formally, the following result holds.

Theorem 3.1. *[27] Let $Q^{(p)} := \sum_{i=i}^{k} \lambda_i^p Q_i$, where $(\lambda_i)_{1\leq i\leq k} \in \mathbb{R}^k$ is such that $0 < \lambda_1 < \lambda_2 < \cdots < \lambda_k \leq 1$, and $(Q_i)_{1\leq i\leq k}$ are real-valued continuous functions defined on $\mathbb{R}^n$. Let M_k be the set of minimizers of Q_k, and for $1 \leq i < k$, M_i be recursively defined as follows:*

$$M_k = argmin_{\mathbf{x}\in\mathbb{R}^n} Q_k(\mathbf{x}) \tag{15}$$

$$\forall 1 \leq i < k,\ M_i = argmin_{\mathbf{x}\in M_{i+1}} Q_i(\mathbf{x}) \tag{16}$$

Any convergent sequence $(\mathbf{x}_p)_{p>0}$ of minimizers of $Q^{(p)}$ converges to some point of M_1. In particular, if for all $p > 0$, $(\mathbf{x}_p)_{p>0}$ is bounded (i.e. if there exists $C > 0$ such that for all $p > 0$, $||\,\mathbf{x}_p\,||_\infty \leq C$), then, up to a subsequence $(\mathbf{x}_p)_{p>0}$ is convergent to a point in M_1. Further, we can estimate the minimum of $Q^{(p)}$ as follows:

$$min_{\mathbf{x}\in\mathbb{R}^n} Q^{(p)}(\mathbf{x}) = \sum_{1\leq i\leq k} \lambda_i^p m_i + o(\lambda_1^p) \tag{17}$$

where $m_i = min_{\mathbf{x}\in M_i} Q_i(\mathbf{x})$ and $o(\lambda_1^p)$ is the Landau notion of negligibility.

The following algorithm is readily derived from the theorem.

Algorithm 1 Calculating limit of minimizers [27]

```
Set i = k and M_{i+1} is the entire space.
while i > 0 do
    Compute the set of minimizers M_i = arg min_{x∈M_{i+1}} Q_i(x)
end while
return Some x ∈ M_1.
```

3.2. Limit of Minimizers of Relaxed Seeded Isoperimetric Partitioning Problem.

As we are working with finite graphs, each of the edge weights can take values in a finite set. Hence, there are k distinct weights $w_1 < w_2 < w_3 < \cdots < w_k$, the cost function of the isoperimetric partitioning problem can be written as

$$Q(\mathbf{x}) = \mathbf{x}^t L \mathbf{x} = \sum_{i=1}^{k} w_i \left(\mathbf{x}^t L_i \mathbf{x}\right) \tag{18}$$

where L_i denotes the Laplacian of the graph induced by edges with weight exactly equal to w_i. Observe that non-diagonal entries of L_i are either 0 or 1. It is easy to see that the cost function of the isoperimetric partitioning problem for the exponentiated graphs is equal to

$$Q^{(p)}(\mathbf{x}) = \mathbf{x}^t L \mathbf{x} = \sum_{i=1}^{k} w_i^p \left(\mathbf{x}^t L_i \mathbf{x}\right) \tag{19}$$

This allows us to use theorem 3.1 to calculate the limit of minimizers as $p \to \infty$. The following theorem holds.

Theorem 3.2. *Let G denote a finite edge-weighted graph and G^{umst} the weighted graph induced by the UMST. The limit of minimizers of the relaxed seeded isoperimetric partitioning problem on $G^{(p)}$ is equal to the limit of minimizers of the relaxed seeded isoperimetric partitioning problem on $G^{(p),umst}$ as $p \to \infty$.*

The above theorem provides an initial step to explain the MST approximation in [19]. It has been shown earlier [7, 15, 11, 6, 14] that the limit of minimizers preserves the essential properties of solutions, thus giving useful results. Theorem 3.2 states that computing the limit of the minimizers of the relaxed seeded isoperimetric partitioning problem on UMST is same as on the original graph. Thus, assuming that the limit of minimizers yields useful solutions, theorem 3.2 allows us to solve the relaxed seeded isoperimetric partitioning problem on a smaller[3] UMST graph instead of the original graph. While currently there is no formal statement justifying such approximation, section 5 provides some empirical evidences for this claim.

Note that algorithm 1 provides only a heuristic to calculate the limit of minimizers, which may not be implementable in practice. Theorem 3.3 provides a method of calculating the limit of minimizers of the relaxed seeded isoperimetric partitioning problem.

Theorem 3.3. *Let $\mathbf{x}^*$ be a limit of minimizers of the relaxed seeded isoperimetric partitioning problem on $G^{(p)}$ as $p \to \infty$. Assuming that x_j denotes the seed, $L^{umst}_{(-j,-j)}$ denotes the Laplacian of the UMST with j^{th} row and column removed. Then for some $\lambda \in \mathbb{R}$,*

$$L^{umst}_{(-j,-j)}\mathbf{x}^* = \lambda \mathbf{1} \tag{20}$$

Since $L^{umst}_{(-j,-j)}$ is non-singular, solving the equation

$$L^{umst}_{(-j,-j)}\,\mathbf{x} = \mathbf{1} \tag{21}$$

yields a solution to the relaxed seeded isoperimetric partitioning problem.

3.3. Going from UMST to MST?

The above theorems exhibit that in the limiting case, solving the relaxed seeded isoperimetric partitioning problem on the UMST is same as solving the problem on the original graph. However, in [19] the authors consider an arbitrary MST to solve the problem.

In general one cannot assure that solving the relaxed seeded isoperimetric partitioning problem on UMST and MST provide the same solution. Examples demonstrating this are discussed in the next section. However, there are few cases (not encountered often in practice) where it holds true.

- When all edges have distinct weights, UMST and MST are identical and hence they yield the same solution.
- Note that the final partition is obtained by thresholding the solution to the seeded isoperimetric partitioning problem. In the case when the ideal partition exists as a

[3] smaller in the sense of number of edges

threshold of the original graph, it is assured that both UMST and MST give the correct the partition[4]. This is because all the three structures MST, UMST and the original graph have the same components when thresholded.

Intuitively, UMST removes the ambiguity of choosing an arbitrary MST and considers a 'union' instead. And hence, it results in a more deterministic behavior. In the following part we answer the question - How different can the solutions of UMST and MST be?

Before proceeding, we need some more notions. Recall that it is assumed that there exists k distinct weights on the graph - $w_1 < w_2 < \cdots w_k$. For any subgraph one can assign a *weight-distribution* vector of length k - $[l_1, l_2, \cdots l_k]$, where l_i is an integer denoting the number of edges with weight w_i. Thus, the UMST and MST graphs also have such weight distributions which is denoted by $[u_1, u_2, \cdots, u_k]$ and $[m_1, m_2, \cdots, m_k]$ respectively. The following proposition holds.

Proposition 3.4. *Given an edge weighted graph $G = (V, E, w)$, all the MST's have the same weight distribution.*

Recall that the solution is obtained by solving the following linear equation

$$L_{(-j,-j)}\mathbf{x}_{-j} = \mathbf{1} \tag{22}$$

where $L_{(-j,-j)}$ is the reduced Laplacian. This implies that for each $i \neq j$ ($x_j = 0$ corresponds to the seed) the following equation holds.

$$x_i = \sum_l \frac{w_{il}}{d_i} x_l + \frac{1}{d_i} \tag{23}$$

where $d_i = \sum_l w_{il}$ denotes the degree of the vertex i. Let D denote the matrix $diag(1/d_1, 1/d_2, \cdots, 1/d_n)$, and assume W to indicate the adjacency matrix, hence W_{il} denotes the weight w_{il}. Also let f indicate the vector $[1/d_1, 1/d_2, \cdots, 1/d_n]$. Using these notations, the solution to the relaxed seeded isoperimetric partitioning problem satisfies

$$T(\mathbf{x}) = D^{-1}W\,\mathbf{x} + f = \mathbf{x} \tag{24}$$

In other words, the solution is a fixed point of the linear operator $T(.)$.

Observe that, each adjacency matrix gives a different operator (the matrix D depends on the adjacency matrix). Thus, there are two operators - T_{umst} and T_{mst}, corresponding to the UMST and MST graphs respectively. To characterize the difference between the solutions of seeded isoperimetric partitioning problem on UMST and MST, it is enough to consider the distance between these two operators. In particular, the following theorem holds.

Theorem 3.5. *Let T_{umst} and T_{mst} denote the operators on UMST and MST respectively, as defined above. Then there exists two positive constants K_1 and K_2 such that*

$$K_1 \sum_{i=1}^{k} (u_i - m_i)^2 w_i^2 \leq \|T_{umst} - T_{mst}\| \leq K_2 \sum_{i=1}^{k} (u_i - m_i)^2 w_i^2 \tag{25}$$

[4] Assuming that the ideal partition is the one which minimizes the isoperimetric ratio.

The significance of theorem 3.5 is - it gives bounds on how different the solutions of UMST and MST can be for the seeded isoperimetric partitioning problem in terms of their weight distributions. As a consequence of proposition 3.4, these bounds can be calculated from the raw data without resorting to solving the linear equation or using an explicit structure of a MST. Note that in the case of all edge weights being distinct, the following holds true

$$\sum_{i=1}^{k}(u_i - m_i)^2 w_i^2 = 0 \tag{26}$$

since $u_i = m_i$ for all i. This implies that the bounds in theorem 3.5 are attained.

4. Limit of Minimizers of Discrete Isoperimetric Partitioning Problem. Given the interest of PW framework in solving the relaxed seeded isoperimetric partitioning problem, we explore the limit of the discrete isoperimetric partitioning problem (4) in the PW framework. In this section we characterize the limit of minimizers to the discrete problem (4) in the Power Watershed framework and establish links with other existing methods.

Theorem 4.1 shows the most important property of the limit of minimizers of the discrete isoperimetric partitioning problem. Recall the assumption that the edge weights can attain one of the k distinct weights $w_1 < w_2 < w_3 < \cdots < w_k$. Also let $G_{\geq w}$ indicate the graph induced by the edges in G whose weight is at least w.

Theorem 4.1. *Let x^* be a limit of minimizers of the discrete isoperimetric partitioning problem. If $G_{\geq w}$ is disconnected, i.e. it has at least two connected components, then x^* is constant on each of these components.*

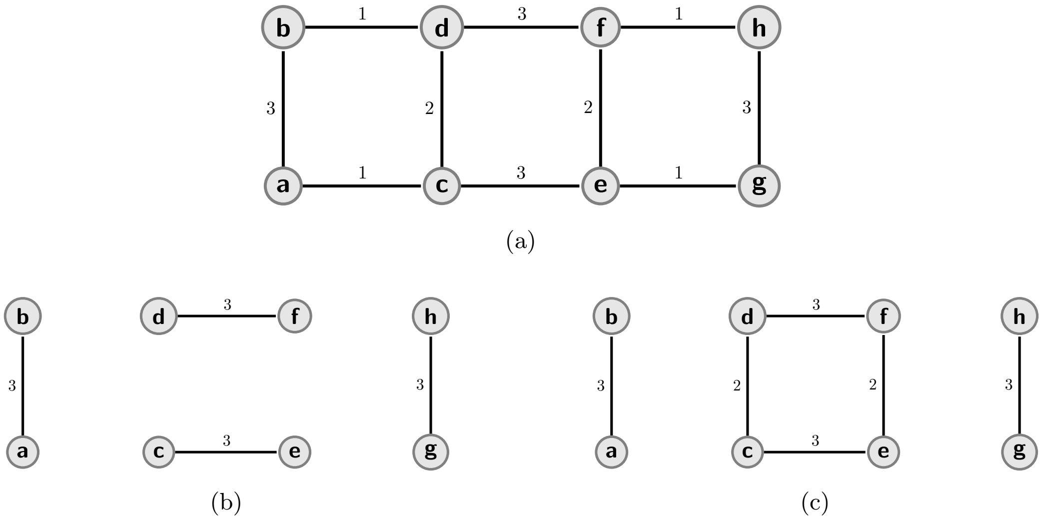


Figure 1. *(a) A synthetic graph, G (b) Graph in (a) thresholded at 3, $G_{\geq 3}$. (c) Graph in (a) thresholded at 2, $G_{\geq 2}$. Note that the graph thresholded at 1 is the original graph.*

To illustrate theorem 4.1, consider the graph G as shown in figure 1. The theorem implies that x^* (solution to limit of minimizers of the discrete isoperimetric partitioning problem) has constant values within each of the connected components of $G_{\geq 3}$ i.e. $x^*(a) = x^*(b)$, $x^*(c) = x^*(e)$, $x^*(d) = x^*(f)$, $x^*(h) = x^*(g)$. Further, theorem 4.1 applied to $G_{\geq 2}$ implies that the equalities $x^*(c) = x^*(d) = x^*(e) = x^*(f)$ hold.

The above illustration indicates that one can define a *critical weight*, which is the largest weight w such that $G_{>w}$ (the subgraph induced by edges of G with weights greater than w) is disconnected while $G_{\geq w}$ is connected. Theorem 4.1 implies that , x^* attains a constant value on each component of $G_{>w}$.

Recall that a subgraph induced by a subset of edges $E_1 \subset E$ of an edge-weighted graph $G = (V, E, w)$ is said to be a *maximum* if: every edge in E_1 has the same weight; any edge in $E \setminus E_1$ adjacent to an edge in E_1 has strictly lesser weight; and E_1 induces a connected subgraph. The *watershed cut* [12, 13] of an edge-weighted graph is a maximum spanning forest relative to its maxima (when the edge weights represent similarity measure). This allows us to make the following interesting observation:

> *The partition corresponding to the PW limit of minimizers of the discrete isoperimetric partitioning problem can be obtained by successively adding edges to a watershed cut in decreasing order of their weights until the resulting graph contains two connected components.*

In the case where $G_{>w}$ has exactly two components, the limit of the discrete isoperimetric partitioning problem is exactly the same as a watershed cut. For further details on watershed cuts, the reader may refer to [12, 13].

Remark: Note that the discrete isoperimetric partitioning problem is NP-hard. This property holds when computing the limit of minimizers as well, i.e. there does not exist a polynomial time algorithm to calculate the limit of minimizers to the discrete isoperimetric partitioning problem in general.

5. Empirical Analysis.

To recap, we have shown in the Power Watershed framework that the limits of the relaxed seeded isoperimetric partitioning problem on the original graph and UMST are identical. Also, we have analyzed the situation if MST was used in place of UMST to solve the relaxed seeded isoperimetric partitioning problem as proposed in [19]. In this section we provide several examples and experimental results to further understand the relation between the solutions of the relaxed seeded isoperimetric partitioning problem on the original graph, UMST and MST.

Remark: Note from earlier that only the relative values of the solution are of interest, since after the calculation of the solution, each threshold is evaluated to obtain the optimal partition (See [20] for details). Thus, in this section two solutions $\mathbf{x}$ and $\mathbf{y}$ are considered to be *equivalent* if they have the same order, that is

$$x_i \leq x_j \Leftrightarrow y_i \leq y_j \text{ for all } i, j \tag{27}$$

A sufficient condition to achieve the same partition is provided by this equivalence condition, and the thresholding step is not considered hereafter.

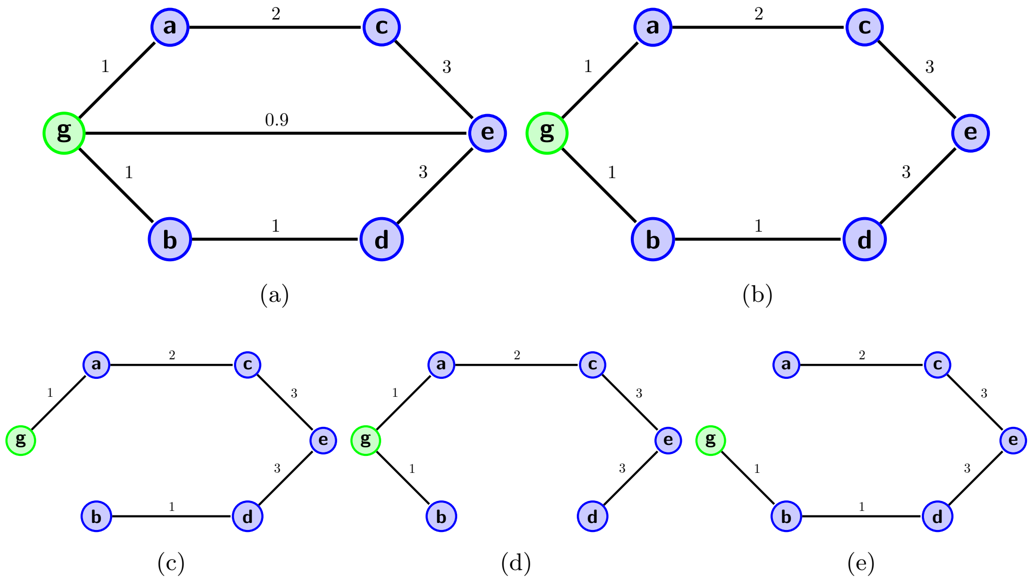


Figure 2. *Example to illustrate the differences between the solutions to the relaxed seeded isoperimetric partitioning problem on the original graph, UMST and MST. (a) Original graph. (b) UMST. (c) MST_1 (d) MST_2 (e) MST_3. (c) - (e) shows all possible MST's for graph in (a). vertex labelled 'g' indicates the seed vertex in all cases. The edge weights are as shown on the edges.*

Table 1

Solutions to relaxed seeded isoperimetric partitioning problem with graphs in figure 2

Node	Original	UMST	MST_1	MST_2	MST_3
g	0.00	0.00	0.00	0.00	0.00
a	1.69	2.68	5.00	4.00	11.16
b	1.54	2.32	9.66	1.00	5.00
c	2.04	3.52	7.00	5.50	10.66
d	2.09	3.64	8.66	6.50	9.00
e	1.94	3.74	8.00	6.16	10.00

5.1. Finding a suitable MST?. One question which naturally arises is - Does there exist a MST on which the solution of the relaxed seeded isoperimetric partitioning problem is equivalent to the solution obtained with the original graph? What about UMST?

In general it is not assured that such a MST exists. Consider a simple graph as shown in figure 2. Corresponding UMST and all possible MST's are also shown. Vertex 'g' denotes the seed. The solution to the relaxed seeded isoperimetric partitioning problem for each of these graphs is given in table 1.

The following conclusions can be drawn from the results:

1. Note that the relative ordering of the co-ordinates of solution to the seeded isoperimetric partitioning problem on original graph does not match with any of MST's. Hence this provides a counter example.
2. The relative ordering of the co-ordinates of the solutions for UMST is different from

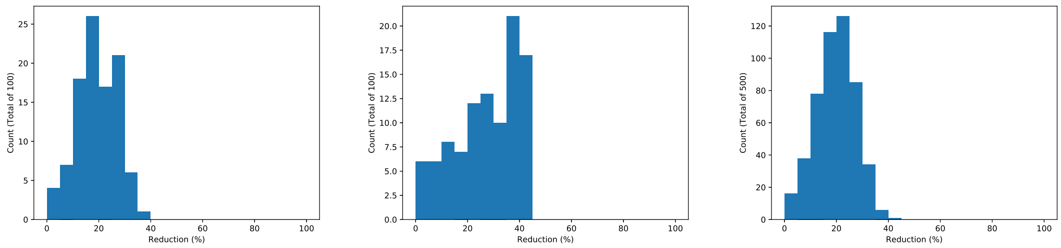
those obtained from any of the MST's.

3. Moreover the relative ordering of the co-ordinates of solutions of UMST and the original graph also do not match.

Example in figure 2 conclusively shows that, in general one cannot expect a relation between the solutions of the relaxed seeded isoperimetric partitioning problem on the original graph, UMST and MST. However, in several practical cases, one can expect them to be 'close'. One such application is that of image segmentation. In this case, most of the edges in the UMST and MST are within the object and hence might give similar results. This is discussed in detail in the next part of the section.

Another important observation from the above example is that the values of solutions on MST are widely fluctuating. That is, the solution changes with respect to the choice of MST. This ambiguity is not present when considering the UMST.

5.2. Results in Practice. In this part we focus on how the solutions of the relaxed seeded isoperimetric partitioning problem on the original graph, UMST and MST behave in practice. Let $\mathbf{x}$, $\mathbf{x}^{umst}$, $\mathbf{x}^{mst}$ indicate the solution to the original graph, the UMST solution and MST solution respectively. As the seed, a vertex in the interior of an object is randomly picked, and the same seed is used for all three solutions. The datasets considered are the Weizmann 1-Object and 2-Object datasets [2] and BSDS500 dataset [3]. We select to use the classic 4-adjacency graph constructed from the image [5].



(a) (b) (c)

Figure 3. *Histograms indicating the amount of reduction in number of edges obtained when constructing the UMST. x-axis represents the percentage reduction obtained. y-axis represents the number of images achieving the given amount of reduction. The results are computed on (a) Weizmann 1-Object dataset, (b) Weizmann 2-Object dataset and (c) BSDS500 dataset.*

Implementation Note: Recall the assumption that there exists k distinct values for the edge weights. In practice, the edge weights are represented by floating point numbers and hence 'equality' cannot be judged. To overcome this, we consider an ϵ-*precision* where the weights, w_{ij} are modified as below.

$$w_{ij} \to int(w_{ij}/\epsilon) \times \epsilon \tag{28}$$

[5]The isoperimetric graph partitioning problem is no longer NP-hard on a 4-adjacency graph. However, the number of partitions of the vertex set V into two subsets is $\mathcal{O}(|V|^{\frac{3}{2}})$ [1]. Hence, solving the discrete isoperimetric graph partitioning directly is inefficient.

Intuitively, this operation restricts the precision of a floating point number.

Firstly, observe that the reduction in the complexity is by reducing the number of non-zero entries of the matrix L in (10). Thus one question to ask is - How much reduction in the number of edges is achieved when considering the reduction to UMST or MST? In the case of using a MST instead of the original graph, the number of edges is simply $n-1$ where n indicates the number of vertices in the graph (which is $\approx$ 50% reduction on a 4-adjacency graph). In the case of UMST, in general it is not possible to predict the amount of reduction in number of edges. Figure 3 shows the histogram of the percentage of reduction achieved on the Weizmann and BSDS datasets. Observe that, on average we achieve 20% reduction, which can go up to 40%. Intuitively, UMST only removes 'non-informative' edges from the graph. This is dependent on the image under consideration.

5.2.1. Accuracy of $\mathbf{x}$, $\mathbf{x}^{umst}$, $\mathbf{x}^{mst}$. We now inspect how the different solutions affect the accuracy of segmentation. For the results to be as precise as possible we consider *recursive partitioning* - that is, each of the components of the partition is further partitioned, until a stopping criterion is met. We consider the stopping criterion to be when isoperimetric ratio crosses a given threshold.

Table 2
Accuracy Measures used in figure 4

Measure	Description
Adjusted Rand Index (ARI) [23]	Rand Index adjusted for chance.
Adjusted Mutual Information (AMI) [29]	Mutual Information adjusted for chance.
Precision (P_r in [28])	Reflects the probability that a pair of pixels predicted to have same label does indeed have same label.
Recall (R_r in [28])	Reflects the probability that a pair of pixels having the same label is predicted to have same label.
F-Score (F_r in [28])	Summary Measure given by $(2 * \text{Precision} * \text{Recall})/(\text{Precision} + \text{Recall})$

The accuracy measures considered are described in table 2. For each image in Weizmann 1-Object dataset, we compare the recursive partition obtained using the solution to the relaxed seeded isoperimetric partitioning problem on UMST/MST with the solution to the relaxed seeded isoperimetric partitioning problem on original graph. These results are plotted as a scatter plot in figure 4. Note that the results on UMST and original graph are almost similar. However, when considering MST, sometimes the results are better and sometimes worse. This can, once again be attributed to the previous observation that MST loses information.

Relative ordering of the co-ordinates in $\mathbf{x}$, $\mathbf{x}^{umst}$, $\mathbf{x}^{mst}$. Recall that two solutions are considered equivalent if they have the same order (see (27)). Here we consider how different are the orders of $\mathbf{x}^{umst}$, $\mathbf{x}^{mst}$ with respect to $\mathbf{x}$. In figures 5a, 5b the scatter plot is used to demonstrate the differences between $\mathbf{x}^{umst}$ and $\mathbf{x}^{mst}$. The scatter plot is between the values of the solutions $\mathbf{x}^{umst}$ and $\mathbf{x}^{mst}$, with respect to $\mathbf{x}$, at several random vertices across few random images. In the ideal case of the order being perfectly preserved, we expect the plot to follow a strictly increasing function. The size of deviation from the increasing function reflects how different the orders of the solution are. In figure 5a, observe that the UMST preserves the order quite well, while figure 5b suggests that MST does not preserve the order so well. This

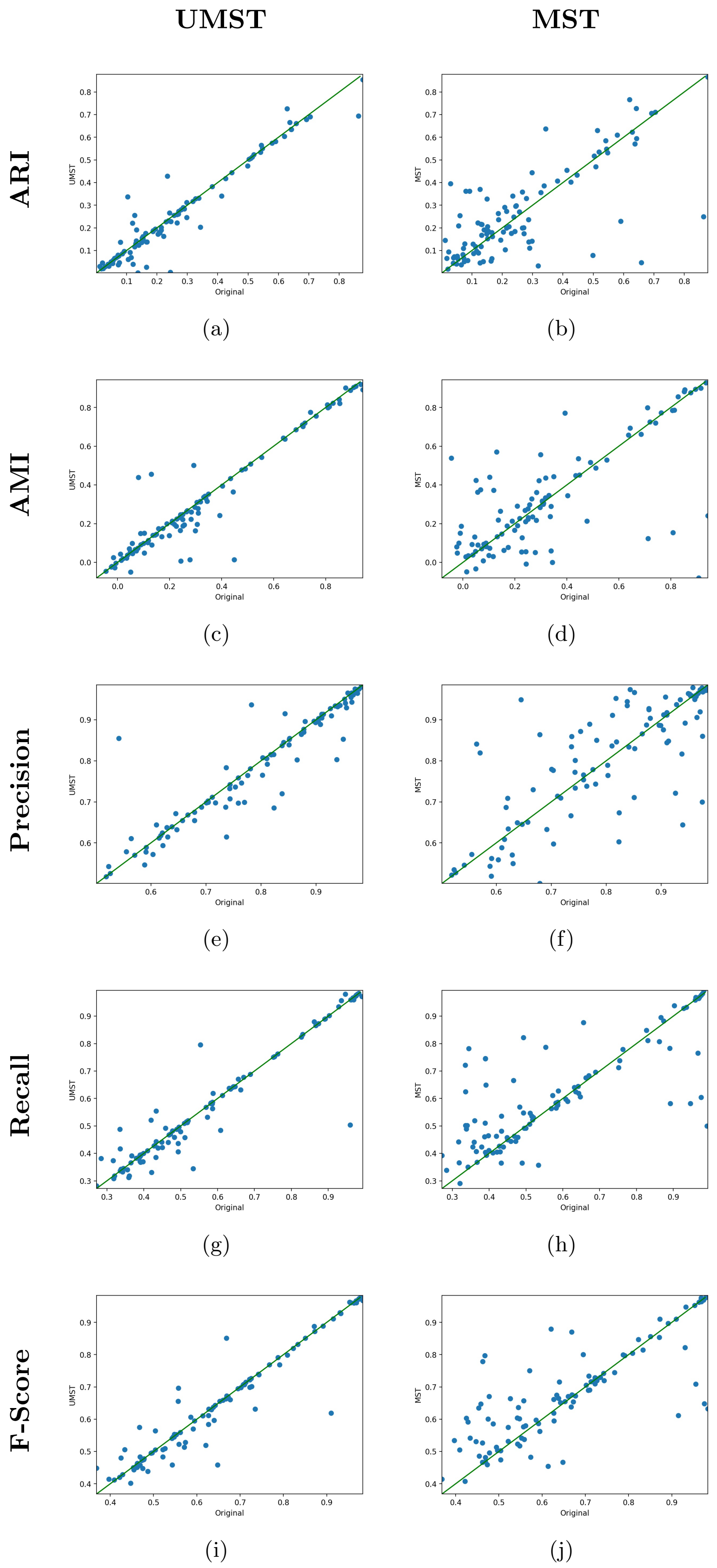
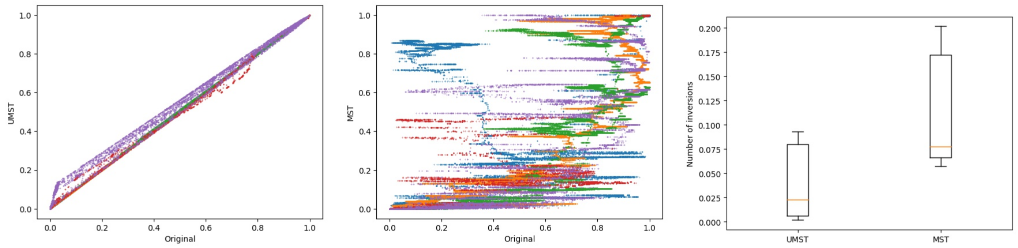


Figure 4. *Scatter plots between measures obtained using UMST/MST and the original graph, on images from Weizmann 1-Object dataset, for several different measures described in table 2. Observe that the results for UMST (first column) are very close to the original graph, while results obtained using MST (second column) have large deviations. This is especially evident when considering the measure 'Precision'.*



(a) (b) (c)

Figure 5. *Relative ordering of the solutions of seeded isoperimetric partitioning problem on the UMST and MST with respect to the original graph. (a) Shows the scatter plot of the solutions at the vertices between* $\mathbf{x}^{umst}$ *and* $\mathbf{x}$ *for several images. Observe that the ordering relation* (27) *is preserved well. (b) Shows the scatter plot of the solutions at the vertices between* $\mathbf{x}^{mst}$ *and* $\mathbf{x}$ *for several images. Observe that the ordering relation* (27) *is* ***not*** *preserved as well as UMST. (c) Box plot indicating the number of inversions obtained, normalized with number of pairs, for* $\mathbf{x}^{umst}$ *and* $\mathbf{x}^{mst}$ *with respect to* $\mathbf{x}$*.*

indicates that MST loses much more relevant information with respect to the original graph than UMST.

Another metric to measure the amount of difference is by calculating the *number of inversions* between the solutions. This is calculated as follows - Order the solution $\mathbf{x}^{umst}$ or $\mathbf{x}^{mst}$ with respect to the order $\mathbf{x}$. Then count the number of inverted pairs - (x_i, x_j) such that $i < j$ and $x_i > x_j$. Normalize with respect to the total number of pairs possible, to obtain consistency across differently sized images. This is measured on each image of Weizmann 1-Object dataset and a box plot is plotted in figure 5c. This substantiates the evidence that MST loses a lot of relevant information while UMST preserves it.

6. Conclusions and Perspectives.

In this article we have revisited the NP-hard isoperimetric graph partitioning problem. We have presented a detailed analysis of the continuous relaxation of the problem, clarifying the construction followed in [20, 21]. In [19] the author exhibited empirically that - solving the relaxed seeded isoperimetric partitioning problem on a much smaller graph (MST) yields a good approximation to the solution on the original graph. We provided an alternative explanation for this approximation by considering the limit of minimizers in the Power Watershed framework. We have shown that, in the limiting case, solving the problem on UMST is equivalent to solving the problem on the original graph. We have established bounds on the difference between the solutions of the relaxed seeded isoperimetric partitioning problem on UMST and MST graphs. Empirical experiments were conducted to analyse these techniques in practice.

It is also possible to characterize the limit of minimizers of the solutions of the discrete isoperimetric partitioning problem in the Power Watershed framework. Although, the computation of the exact limit still remains NP-hard, we have shown that these solutions are 'close' to watershed cuts. Further analysis of the limit of minimizers to the discrete isoperimetric partitioning problem is a subject of future research. We mention here two possible directions: (1) MST has been proved a good heuristic in solving the NP-hard travelling salesman problem [17, 16]. Would it be possible to go along the same lines, and prove some theoretical bounds on the solutions to the isoperimetric graph partitioning problem? (2) In the theory of scale-set analysis [22], it is shown that the celebrated Mumford-Shah functional [26] can be solved in linear time on a tree of segmentations, and that the persistence of regions is a good indicator of their relevance with respect to image segmentation. We can envision using the Cheeger constant in a similar way to what is done in [22]. Extensions to such ideas have been proposed in the shaping framework [31], and can be adapted to the case of the Cheeger constant. Would that be possible to estimate bounds on the solutions in such frameworks?

Beyond segmentation, filtering images is another possible direction of research. As the Cheeger cut problem is closely related to total variation (TV) minimization, it would be interesting to explore the utility of PW framework for solving TV minimization problems. More generally, going beyond images, is to explore the application of the UMST-based algorithm as a fast clustering technique for data analysis.

As a final note, as demonstrated in this paper and others [15, 7, 11, 14, 6], Power Watershed framework has proved to be very useful. From a theoretical standpoint, understanding the working principle behind the Power Watershed framework is still an open problem.

7. Acknowledgments. The authors would like to thank Leo Grady for a fruitful discussion that improved the manuscript. The first author would like to thank Vance Faber for providing some useful references. The second author would like to thank CEFIPRA for providing Raman Charpak Fellowship to support the research. This work is partially funded by CEFIPRA, EMR/2015/000853 SERB, ISRO/SSPO/Ch-1/2016-17 ISRO, ANR-15-CE40-0006 CoMeDiC and ANR-14-CE27-0001 GRAPHSIP research grants.

Appendix A. Proof of proposition 2.2 .

Proof. Given a subset S, define $\mathbf{x}(S) = (x_1, x_2, \cdots, x_n)$ by

$$x_i = \begin{cases} 1/2 - \delta & \text{if } i \in S \\ 1/2 + \delta & \text{otherwise} \end{cases} \tag{29}$$

Then,

$$\mathbf{x}^t L\mathbf{x} = \sum_{ij} w_{ij}(x_i - x_j)^2 = \sum_{e_{ij}\in\partial S} w_{ij}(2 * \delta)^2 \tag{30}$$

Without loss of generality, assume that $|S| \leq (1/2)|V|$ (Otherwise, take the complement). Then the denominator is equal to

$$\frac{|V|}{2} - \delta|S| + \delta|V \setminus S| \tag{31}$$

Thus the cost of this vector is given by

$$\frac{\sum_{e_{ij}\in\partial S} w_{ij}(2 * \delta)^2}{\frac{|V|}{2} - \delta|S| + \delta|V \setminus S|} \tag{32}$$

Note that the above cost converges to 0 as $\delta \to 0$. Hence one can find a $\delta > 0$ such that the cost is less than ϵ. ∎

Appendix B. Proof of theorem 3.2. Let $G = (V, E, w)$ denote the graph, and G^{umst} denote the weighted graph induced by the UMST. Let L and L^{umst} denote the Laplacian of the graphs G and G^{umst} respectively. Denote $L^{other} = L - L^{umst}$ to indicate the Laplacian of the graph induced by the edges which do not belong to the UMST. Let L_i denote the Laplacian of the graph induced by edges with weight equal to w_i. Recall that- it is assumed there are k distinct weights $w_1 < w_2 < w_3 < \cdots < w_k$. This notation is also compounded, in the sense that- L_k^{umst} indicates the Laplacian of the subgraph of UMST graph induced by edges with weight equal to w_k.

The idea of the proof is to show that at every level of the algorithm 1, the minimizers obtained with respect to the graph also minimize the corresponding optimization problem with respect to the UMST graph. This implies that the output of the algorithm is same for both the graph and its corresponding UMST, which proves the theorem.

Following the steps algorithm 1, starting at the highest level k and filter the set of minimizers to obtain the limit of minimizers.

At level k solve the following optimization problem.

$$\begin{aligned} \arg\min_{x} \quad & \mathbf{x}_{-j}^t L_{k,(-j,-j)} \mathbf{x}_{-j} \\ \text{subject to} \quad & (\mathbf{x}_{-j})_i \in [0,1] \forall i \\ & \mathbf{x}_{-j}^t \mathbf{1} = \mu \end{aligned} \tag{33}$$

The above optimization problem is stated for the graph. The equivalent optimization problem for the UMST is obtained by replacing the Laplacian of the graph with the Laplacian of the UMST. Note that it is assumed the seed is placed at some arbitrary vertex x_j. From above, the set of solutions to this optimization problem is the set of solutions obtained by solving

$$L_{k,(-j,-j)} \mathbf{x}_{-j} = \lambda_k \mathbf{1} \tag{34}$$

Lemma B.1. *Given a graph G and the corresponding UMST graph G^{umst}, if w_k denotes the highest weight, then*

$$L_{k,(-j,-j)} = L_{k,(-j,-j)}^{umst} \tag{35}$$

The lemma B.1 essentially tells that at the highest level, the Laplacian for the graph and the corresponding UMST are the same. And hence, the set of minimizers obtained after level k is same for both the graph and its UMST.

Let A_k indicate the matrix obtained by stacking the indicator vectors of the components of $G_{\geq w_k}$. Clearly, the following relation holds

$$L_{k,(-j,-j)} A_k = \mathbf{0} \tag{36}$$

This follows from the properties of the Laplacian [10]. Hence the following lemma holds.

Lemma B.2. *Let M_k denote the set of minimizers obtained by solving (34). If $\mathbf{x}_k^*$ denotes some solution to the equation, then the set M_k is characterized by*

$$\mathbf{x}_k^* + A_k \boldsymbol{y} \tag{37}$$

for any y.

At level $k-1$, thanks to the above results, solve the optimization problem

$$\begin{aligned} \arg\min_{\mathbf{x}} \quad & \mathbf{x}_{-j}^t L_{k-1,(-j,-j)} \mathbf{x}_{-j} \\ \text{subject to} \quad & (\mathbf{x}_{-j})_i \in [0,1] \forall i \\ & \mathbf{x}_{-j}^t \mathbf{1} = \mu \\ & \mathbf{x}_{-j}^t \approx \mathbf{x}_k^* + A_k \mathbf{y} \end{aligned} \tag{38}$$

where $\approx$ is to be read as - 'is of the form'. Once again, note that the above optimization problem is stated for the graph. To obtain the equivalent optimization problem for the UMST graph one simply replaces the Laplacian of the graph with the Laplacian of the UMST graph. Simplifying the above optimization problem,

$$\begin{aligned} \underset{\mathbf{y}}{\text{minimize}} \quad & (\mathbf{x}_k^*)^t L_{k-1,(-i,-i)} \mathbf{x}_k^* + 2(\mathbf{x}_k^*)^t L_{k-1,(-i,-i)} A_k \mathbf{y} \quad + \mathbf{y}^t A_k^t L_{k-1,(-i,-i)} A_k \mathbf{y} \\ \text{subject to} \quad & (\mathbf{x}_k^* + A_k \mathbf{y})^t \mathbf{1} = \mu \end{aligned} \tag{39}$$

The following lemma holds.

Lemma B.3. *Recall that* $L_{k-1} = L_{k-1}^{other} + L_{k-1}^{umst}$. *Hence,* $L_{k-1}^{other} A_k = 0$.

The proof of the above lemma is seen by noting that - 'other' edges apart from UMST are intra-component edges in the components of $\mathcal{G}_{\geq w_k}$. So, components of the graph whose Laplacian would be L_{k-1}^{other} are subsets of the components of $\mathcal{G}_{\geq w_k}$. Thus, from the properties of the Laplacian, the above lemma holds true.

Thanks to the above lemma, the optimization problem reduces to

$$\begin{aligned} \underset{y}{\text{minimize}} \quad & (\mathbf{x}_k^*)^t L_{k-1,(-i,-i)}^{umst} \mathbf{x}_k^* + 2(\mathbf{x}_k^*)^t L_{k-1,(-i,-i)}^{umst} A_k \mathbf{y} + \mathbf{y}^t A_k^t L_{k-1,(-i,-i)}^{umst} A_k \mathbf{y} \\ \text{subject to} \quad & (\mathbf{x}_k^* + A_k \mathbf{y})^t \mathbf{1} = \mu \end{aligned} \tag{40}$$

Observe that the above optimization problem is nothing but the optimization problem at level $k-1$ of the UMST graph. Hence, at level $k-1$, the solutions to the optimization problem of the graph and the solutions to the optimization problem of the UMST graph are equal.

Continuing this argument, one can easily see that the output of the algorithm 1 for both the graph and its UMST is the same. Hence the theorem is proved.

Appendix C. Proof of theorem 3.3.

Observe that from above, the limit of minimizers can be written as

$$\mathbf{x}^* = \mathbf{x}_k^* + A_k \mathbf{x}_{k-1}^* + A_k A_{k-1} x_{k-2}^* + \cdots + A_k A_{k-1} A_{k-2} \cdots \mathbf{x}_1^* \tag{41}$$

Note that $L^{umst} = L_1^{umst} + L_2^{umst} + L_3^{umst} + \cdots + L_k^{umst}$. So,

$$L^{umst}\mathbf{x}^* = (L_k^{umst} + L_{k-1}^{umst} + \cdots + L_1^{umst})(\mathbf{x}_k^* + A_k \mathbf{x}_{k-1}^* + \cdots + A_k A_{k-1}..\mathbf{x}_1^*) \tag{42}$$

$$= L_k^{umst}\mathbf{x}_k^* + L_{k-1}^{umst}(\mathbf{x}_k^* + A_k \mathbf{x}_{k-1}^*) + \cdots \tag{43}$$

The second step follows from noting that $L_k A_k = 0$, $L_{k-1} A_k A_{k-1} = 0$, and so on. This, inturn, follows from the following lemma.

Lemma C.1. *The matrix $A_1 A_2$ consists of the indicator vectors of the connected components of $\mathcal{G}_{\leq w_2}$.*

Note that $L_k^{umst}\mathbf{x}_k^* = \lambda_k \mathbf{1}$. Now, the solutions at level 2 , optimization problem in (40) satisfy

$$A_k L_{k-1}^{umst}(\mathbf{x}_k^* + A_k \mathbf{y}) = A_k \lambda_{k-1}\mathbf{1} \tag{44}$$

Lemma C.2. *If given $AB\,\mathbf{x} = A\boldsymbol{y}$, then $B\,\mathbf{x} = \boldsymbol{y}$ when A is non singular (has an inverse) or if the intersection of column space of B and the null space of A is the zero vector.*

The proof of the above lemma is standard. From lemma C.2, it can be deduced that $L_{k-1}^{umst}(\mathbf{x}_k^* + A_k\mathbf{y}) = \lambda_{k-1}\mathbf{1}$. Thus, continuing from (43)

$$L^{umst}\mathbf{x}^* = L_k^{umst}\mathbf{x}_k^* + L_{k-1}^{umst}(\mathbf{x}_k^* + A_1\mathbf{x}_{k-1}^*) + \cdots \tag{45}$$

$$= \lambda_k\mathbf{1} + \lambda_{k-1}\mathbf{1}\cdots \tag{46}$$

$$= \Lambda\mathbf{1} \tag{47}$$

Appendix D. Proof of Proposition 3.4. For this proof, we need the cut-property of MST.

Lemma D.1 (Cut Property [4]). *A spanning tree T of a connected graph G is a MST if and only if for every edge e in T, any edge e' in the cut of $T \setminus \{e\}$ satisfies $w(e') \geq w(e)$*

It is enough to show that one can transform an MST T to any other MST T' via a sequence of operations that keep the edge-weight distribution invariant.

Suppose T is different from T' then let $e \in T \setminus T'$. Now consider the cut edges of $T \setminus \{e\}$. It is evident from the cut property that $w(e) \geq w(f)$ for any cut-edge f of $T \setminus \{e\}$. Firstly, T' being a spanning tree has to contain a cut-edge of $T \setminus \{e\}$. Secondly total weight of T' and T are same as both are MST's and $e \notin T'$ implies existence of a cut-edge $e' \neq e$ of $T \setminus \{e\}$ with $w(e) = w(e')$ and $e' \in T'$. Now observe that $T \setminus \{e\} \cup \{e'\}$ is a MST with same edge-weight distribution as that of T. Since, we are working on finite graphs, after a finite sequence of steps, we would end up with T' starting from T. We remark that at every step, the edge-weight distribution of the MST remains invariant and hence the proof.

Appendix E. Proof of theorem 3.5. Recall that,

$$T(\mathbf{x}) = D^{-1}W\,\mathbf{x} + f \tag{48}$$

Assuming that f in column space of $D^{-1}W$, which is the case if the graph is connected, one can rewrite the operator as,

$$T(\mathbf{x}) = D^{-1}W(\mathbf{x} + \overline{f}) \tag{49}$$

for some $\overline{f}$. Hence we have that

$$\|T_{umst} - T_{mst}\| = \|(D^{-1}W)_{umst} - (D^{-1}W)_{mst}\|$$
$$= \sum_{i,j}\left(\frac{w_{ij,umst}}{d_{i,umst}} - \frac{w_{ij,mst}}{d_{i,mst}}\right)^2$$

Let,

$$K_1 = \frac{1}{(\max_i\{d_{i,umst}\})^2}$$

$$K_2 = \frac{1}{(\min_i\{d_{i,mst}\})^2}$$

Note that,

$$\sum_{i,j}\left(\frac{w_{ij,umst}}{d_{i,umst}} - \frac{w_{ij,mst}}{d_{i,mst}}\right)^2 \leq \sum_{i,j}\frac{1}{d^2_{i,umst}}\left(w_{ij,umst} - w_{ij,mst}\right)^2 \tag{50}$$

$$\leq \frac{1}{(\min_i\{d_{i,umst}\})^2}\sum_{i,j}\left(w_{ij,umst} - w_{ij,mst}\right)^2 \tag{51}$$

$$= \frac{1}{(\min_i\{d_{i,umst}\})^2}\sum_{i}\left(n_i - m_i\right)^2 w_i^2 \tag{52}$$

(53)

Similarly, one can obtain the other relation as well.

Appendix F. Proof (sketch) of theorem 4.1. Recall that the edge weights are assumed to take one of k distinct values $w_1 < w_2 < \cdots < w_k$.

The proof of this theorem follows the similar lines as that of theorem 3.2. That is, we trace the working of the algorithm 1 to obtain the proof. The main difference is that, at level m we solve the problem

$$\begin{aligned} &\text{Find } \arg\min_{\mathbf{x}} && \frac{\mathbf{x}^t L_m \mathbf{x}}{\min\{\mathbf{x}^t\mathbf{1}, (\mathbf{1}-\mathbf{x})^t\mathbf{1}\}} \\ &\text{subject to} && x_i \in \{0,1\} \quad \text{for all } i \\ & && \mathbf{x} \text{ is a minimizer at all levels } n > m \end{aligned} \tag{54}$$

If $G_{\geq w_m}$ has components $\{C_1, C_2, \ldots C_l\}$, with $l > 1$ consider the following vector

$$\mathbf{x}^*(i) = \begin{cases} 1 & \text{if } i \in C_1 \\ 0 & \text{otherwise} \end{cases} \tag{55}$$

Then it is easy to see that for all $n > m$, we have

$$(\mathbf{x}^*)^t L_n \mathbf{x}^* = 0 \tag{56}$$

and hence $\mathbf{x}^*$ is a solution to (54).

The theorem follows from extending this to all levels.

REFERENCES

[1] V. E. Alekseev, R. Boliac, D. V. Korobitsyn, and V. V. Lozin, *NP-hard graph problems and boundary classes of graphs*, Theoretical Computer Science, 389 (2007), pp. 219–236.
[2] S. Alpert, M. Galun, R. Basri, and A. Brandt", *Image segmentation by probabilistic bottom-up aggregation and cue integration.*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2007.
[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, *Contour detection and hierarchical image segmentation*, IEEE transactions on pattern analysis and machine intelligence, 33 (2011), pp. 898–916.
[4] K. Bernhard and J. Vygen, *Combinatorial optimization: Theory and algorithms*, Springer, Third Edition, 2005., (2008).
[5] X. Bresson, X.-C. Tai, T. F. Chan, and A. Szlam, *Multi-class transductive learning based on 1 relaxations of Cheeger cut and Mumford-Shah-Potts model*, Journal of mathematical imaging and vision, 49 (2014), pp. 191–201.
[6] A. Challa, S. Danda, B. S. D. Sagar, and L. Najman, *An introduction to Gamma-convergence for spectral clustering*, in Discrete Geometry for Computer Imagery - 20th IAPR International Conference, DGCI 2017, Vienna, Austria, September 19-21, 2017, Proceedings, W. G. Kropatsch, N. M. Artner, and I. Janusch, eds., vol. 10502 of Lecture Notes in Computer Science, Springer, 2017, pp. 185–196, https://doi.org/10.1007/978-3-319-66272-5_16, https://doi.org/10.1007/978-3-319-66272-5_16.
[7] A. S. Challa, S. Danda, B. S. S. Daya Sagar, and L. Najman, *Power Spectral Clustering*. working paper or preprint, Jan. 2018, https://hal.archives-ouvertes.fr/hal-01516649.
[8] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock, *An introduction to Total Variation for image analysis*, Theoretical foundations and numerical methods for sparse recovery, 9 (2010), p. 227.
[9] K. Chang, S. Shao, and D. Zhang, *The* 1*-Laplacian Cheeger cut: Theory and algorithms*, arXiv preprint arXiv:1603.01687, (2016).
[10] F. R. K. Chung, *Spectral Graph Theory*, American Mathematical Society, 1997.
[11] C. Couprie, L. J. Grady, L. Najman, and H. Talbot, *Power watershed: A unifying graph-based optimization framework*, IEEE Trans. Pattern Anal. Mach. Intell., 33 (2011), pp. 1384–1399, https://doi.org/10.1109/TPAMI.2010.200, https://doi.org/10.1109/TPAMI.2010.200.
[12] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, *Watershed cuts: Minimum spanning forests and the drop of water principle*, IEEE Trans. Pattern Anal. Mach. Intell., 31 (2009), pp. 1362–1374, https://doi.org/10.1109/TPAMI.2008.173, https://doi.org/10.1109/TPAMI.2008.173.
[13] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, *Watershed cuts: Thinnings, shortest path forests, and topological watersheds*, IEEE Trans. Pattern Anal. Mach. Intell., 32 (2010), pp. 925–939, https://doi.org/10.1109/TPAMI.2009.71, https://doi.org/10.1109/TPAMI.2009.71.
[14] S. Danda, A. Challa, B. S. D. Sagar, and L. Najman, *Power tree filter: A theoretical framework linking shortest path filters and minimum spanning tree filters*, in Mathematical Morphology and Its Applications to Signal and Image Processing - 13th International Symposium, ISMM 2017, Fontainebleau, France, May 15-17, 2017, Proceedings, J. Angulo, S. Velasco-Forero, and F. Meyer, eds., vol. 10225 of Lecture Notes in Computer Science, 2017, pp. 199–210, https://doi.org/10.1007/978-3-319-57240-6_16, https://doi.org/10.1007/978-3-319-57240-6_16.
[15] S. Danda, A. S. Challa, B. S. Daya Sagar, and L. Najman, *Some Theoretical Links Between Shortest Path Filters and Minimum Spanning Tree Filters*. working paper or preprint, May 2018, https://hal.archives-ouvertes.fr/hal-01617799.
[16] D.-Z. Du and F. K. Hwang, *A proof of the gilbert-pollak conjecture on the steiner ratio*, Algorithmica, 7 (1992), pp. 121–135.
[17] E. N. Gilbert and H. O. Pollak, *Steiner minimal trees*, SIAM Journal on Applied Mathematics, 16 (1968), pp. 1–29.
[18] G. Gilboa, *Graph and nonlocal framework*, in Nonlinear Eigenproblems in Image Processing and Computer Vision, Springer, 2018, pp. 123–132.
[19] L. Grady, *Fast, quality, segmentation of large volumes - isoperimetric distance trees*, in Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part III, A. Leonardis, H. Bischof, and A. Pinz, eds., vol. 3953 of Lecture Notes in Computer Science, Springer, 2006, pp. 449–462, https://doi.org/10.1007/11744078_35, https://doi.org/10.1007/11744078_35.

[20] L. Grady and E. L. Schwartz, *Isoperimetric graph partitioning for image segmentation*, IEEE Trans. Pattern Anal. Mach. Intell., 28 (2006), pp. 469–475, https://doi.org/10.1109/TPAMI.2006.57, https://doi.org/10.1109/TPAMI.2006.57.

[21] L. Grady and E. L. Schwartz, *Isoperimetric partitioning: A new algorithm for graph partitioning*, SIAM J. Scientific Computing, 27 (2006), pp. 1844–1866, https://doi.org/10.1137/040609008, https://doi.org/10.1137/040609008.

[22] L. Guigues, J. P. Cocquerez, and H. Le Men, *Scale-sets image analysis*, International Journal of Computer Vision, 68 (2006), pp. 289–317.

[23] L. Hubert and P. Arabie, *Comparing partitions*, Journal of Classification, 2 (1985), pp. 193–218, https://doi.org/10.1007/BF01908075, https://doi.org/10.1007/BF01908075.

[24] E. Merkurjev, A. Bertozzi, X. Yan, and K. Lerman, *Modified Cheeger and ratio cut methods using the Ginzburg–Landau functional for classification of high-dimensional data*, Inverse Problems, 33 (2017), p. 074003.

[25] B. Mohar, *Isoperimetric numbers of graphs*, J. Comb. Theory Ser. B, 47 (1989), pp. 274–291, https://doi.org/10.1016/0095-8956(89)90029-4, http://dx.doi.org/10.1016/0095-8956(89)90029-4.

[26] J.-M. Morel and S. Solimini, *Variational methods in image segmentation: with seven image processing experiments*, vol. 14, Springer Science & Business Media, 2012.

[27] L. Najman, *Extending the power watershed framework thanks to* Γ*-convergence*, SIAM J. Imaging Sciences, 10 (2017), pp. 2275–2292, https://doi.org/10.1137/17M1118580, https://doi.org/10.1137/17M1118580.

[28] J. Pont-Tuset and F. Marqués, *Supervised evaluation of image segmentation and object proposal techniques*, IEEE Trans. Pattern Anal. Mach. Intell., 38 (2016), pp. 1465–1478, https://doi.org/10.1109/TPAMI.2015.2481406, https://doi.org/10.1109/TPAMI.2015.2481406.

[29] N. X. Vinh, J. Epps, and J. Bailey, *Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance*, J. Mach. Learn. Res., 11 (2010), pp. 2837–2854, http://dl.acm.org/citation.cfm?id=1756006.1953024.

[30] U. Von Luxburg, *A tutorial on spectral clustering*, Statistics and computing, 17 (2007), pp. 395–416.

[31] Y. Xu, E. Carlinet, T. Géraud, and L. Najman, *Hierarchical segmentation using tree-based shape spaces*, IEEE transactions on pattern analysis and machine intelligence, 39 (2017), pp. 457–469.