



Clustering Vertex-Weighted Graphs by Spectral Methods

Juan-Luis García-Zapata ^{1,*}  and Clara Grácio ² ¹ Departamento de Matemáticas, Universidad de Extremadura, 10003 Cáceres, Spain² Departamento de Matematica, Universidad de Évora, 7004-516 Évora, Portugal; mgracio@uevora.pt

* Correspondence: jgzapata@unex.es

Abstract: Spectral techniques are often used to partition the set of vertices of a graph, or to form clusters. They are based on the Laplacian matrix. These techniques allow easily to integrate weights on the edges. In this work, we introduce a p -Laplacian, or a generalized Laplacian matrix with potential, which also allows us to take into account weights on the vertices. These vertex weights are independent of the edge weights. In this way, we can cluster with the importance of vertices, assigning more weight to some vertices than to others, not considering only the number of vertices. We also provide some bounds, similar to those of Cheeger, for the value of the minimal cut cost with weights at the vertices, as a function of the first non-zero eigenvalue of the p -Laplacian (an analog of the Fiedler eigenvalue).

Keywords: clustering; partitioning; Laplacian graph; vertex-weighted graph

MSC: 05C22; 05C35; 05C50; 05C70



Citation: García-Zapata, J.-L.; Grácio, C. Clustering Vertex-Weighted Graphs by Spectral Methods. *Mathematics* **2021**, *9*, 2841. <https://doi.org/10.3390/math9222841>

Academic Editors: Shanhe Wu and Frank Werner

Received: 5 September 2021

Accepted: 4 November 2021

Published: 9 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Informally, a cluster in a graph is a subgraph whose vertices are tightly linked between them, and loosely linked with other vertices, out of the subgraph. Such a vague concept is useful in the description of several phenomena: walking, searching, and decomposition of graphs [1]. The concept of a cluster is closely related to that of community [2]. Depending on the intended application, the meaning of the concept can be specified either to better reflect the aspects to be modeled or to ease its calculation. In our case, we were interested in the subdivision of the set of vertices in two parts, of similar size, in such a way that the number of edges between the two parts is kept to a minimum. To solve this problem, there are many references in the literature about spectral tools, based on a certain eigenvector of the Laplacian operator (Fiedler vector) [3]. These references also show how to consider weights at the edges of the graph so that the minimization of the edges takes these weights into account. In this work, we extended this tool in case there are also weights on the vertices independent of those on the edges so that the partition is made in parts of similar total weight, not necessarily a similar number of vertices, minimizing the total edge weight of the cut. Other proposals in the literature deal with this problem using generalized eigenvectors (see [4] and the references therein). In this article, we remain in the context of the usual eigenvectors. Our interest in avoiding generalized eigenvalues is that one of the possible applications is spectral clustering, along the lines of [5]. This method uses several eigenvectors to build a template for each vertex and forms the clusters with these templates. The eigenvectors are orthogonal to each other, but the generalized eigenvectors are not, so the templates formed with these will be less effective. In other words, we needed larger templates with generalized eigenvectors than with usual eigenvectors. An example of our work related to spectral clustering is [6].

A similar problem is studied in [7], where a Laplacian that incorporates both weights at the vertices and at the edges is defined. The vertex weights are multiplicatively integrated directly as matrix factors. The novelty of our approach is that our weights (called ρ) derive

from a potential p that is additively integrated on the diagonal. The relationship between the potential p that must be introduced to obtain the desired weights ρ is specified in Theorem 3. This is significant because it allows us to overcome a technical difficulty of the cited work, obtaining error bounds (Theorem 4) as a function of the potential, not of the weights. A case of application of this method of weights from a potential is our contribution (Chapter 2.4) to the collective work [8]. In that work, we used the Laplacian matrix with potentials to perform spectral graph partitioning for process placement on heterogeneous computing platforms.

In the next section, we review standard notations and concepts about graph and matrix representation. In the following Section 3, we describe known facts about the spectral partition using a Laplacian matrix. Section 4 contains our contribution: we define a Laplacian matrix with potential (p -Laplacian matrices) and show that certain matrix of this type can be used to perform a partition on the set of vertices, in sizes of similar weight, minimizing the edges cut (Theorem 3). It also contains a Cheeger-style bound on the difference between the true value of the minimum partition and the approximated value obtained using the Fiedler eigenvector of the p -Laplacian matrix (Theorem 4). The value considered in this result is the ratio cut of the partition, instead of the total cut. Our purpose was to show that it is possible to give bounds analogous to those that appear in the literature but adapted to this approach of weights at the vertices with independence from the weights at the edges. We emphasize that this bound is an improvement, similar to that of Mohar, for vertex-and-edge-weighted graph spectral analysis.

2. Graphs and Transfer Matrices

A graph $G = (V, E)$ consists of a set V of *vertexes* and a set E of subsets of two vertices (the *edges*). That is, $E \subset \mathcal{P}_2(V)$. For $u, v \in V$, the edge $\{u, v\}$, also noted $u \sim v$, is said to go between u and v . In this study, we used this definition of a graph, which does not model the direction of the edge or loops.

A *weight on edges* is a map

$$w : E \rightarrow \mathbb{R}.$$

The weight of the edge $u \sim v$ is noted $w(u, v)$. If a weight on the edges is not specified, implicitly, the constant unit weight must be considered (that is, $w(u, v) = 1$ for each $\{u, v\} \in E$).

A *weight on vertexes* is a map

$$s : V \rightarrow \mathbb{R}.$$

The set of these maps, that is, the set of all vertex weights, is denoted \mathbb{R}^V . Clearly, it is a vector space.

To represent graphs using matrices, we chose an ordering of the set of vertexes, $V = \{v_1, \dots, v_n\}$. The *adjacency matrix* of G (for this conventional ordering) is the $n \times n$ matrix $A = (a_{ij})_{i,j=1,\dots,n}$ of values:

$$a_{ij} = \begin{cases} 1 & \text{if } v_i \sim v_j \\ 0 & \text{if not} \end{cases}$$

We represent a vertex weight $s \in \mathbb{R}^V$ as the vector $(s_i)_{i=1,\dots,n}$ of its values $s_i = s(v_i)$. The adjacency matrix operates in \mathbb{R}^V , the set of vertex weights, as a right matrix product (postmultiplication).

$$A : \mathbb{R}^V \longrightarrow \mathbb{R}^V$$

$$(s_i)_{i=1,\dots,n} \longmapsto sA = \left(\sum_{j=0}^n s_j a_{ij} \right)_{i=1,\dots,n}$$

This is the *transference* (or *shift*) of the vertex weight (s_i) by the graph G .

The postmultiplication of s , as a row vector, by A , is usual in matrix analysis of finite Markov chains [9]. We can see the shift as the following action: the shift sA represents that each edge $v_i \sim v_j$ takes the content of v_i , that is, $s(v_i)$, and transports it to the vertex v_j (modifying its amount by the factor of transference a_{ij}). The sum of the values transferred to v_j is $\sum s(v_i)a_{ij}$, with a gain a_{ij} from each adjacent vertex v_i . So, the vertex weight sA is:

$$(sA)(v_j) = \sum_{v_i \sim v_j \in E} s(v_i)a_{ij}$$

In the literature, the interest is focused on symmetrical matrices (because they are modeling undirected graphs), so there is no difference between pre- and postmultiplication by A . Besides, note that the main diagonal is zero because the graphs do not have loops. In Section 4, we introduce potentials that can be viewed as a method to use the diagonal entries to carry vertex weights, independently of the edge weights.

3. Laplacian and Partitions

This section summarizes standard notions about graphs, partitions, and the Laplacian operator, which can be seen in [3] or [10], although the notation is adapted to our particular purposes, and the Lemma 3 is proved in a novel way, without using summatories.

A *partition of a set* V is an array (V_1, V_2) of two subsets of V such that $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \emptyset$. A *partition of a graph* $G = (V, E)$ is a partition of the underlying set V of vertices. An edge $u \sim v$ in G is *cut by a partition* (V_1, V_2) if $u \in V_1$ and $v \in V_2$ or vice versa. If the graph is weighted, the *total cut* is

$$\text{cut}(V_1, V_2) = \sum_{\substack{u \in V_1 \\ v \in V_2}} w(u, v).$$

Any subset $U \subset V$ defines a cut, (U, U^c) , which is called the *cut of* U .

It is usually preferable, between several partitions in a graph, that one has a minimum number of cut edges (or total cut, if weighted). The cut of the graph is:

$$c(G) = \min_{U \subset V} \text{cut}(U, U^c)$$

In this study, we were interested in partitions with minimal cuts but with a balanced number of vertices, that is, $|V_1| = |V_2|$ for an even number of vertices or $|V_1| = \lfloor \frac{|V|}{2} \rfloor$ for any number of vertices (a *bipartition*). The bipartition width [11] is:

$$b(G) = \min_{\substack{U \subset V \\ |U| = \lfloor \frac{|V|}{2} \rfloor}} \text{cut}(U, U^c)$$

We also used the cut ratio of U (the quotient $i(U) = \text{cut}(U, U^c)/|U|$) and the isoperimetric number:

$$i(G) = \min_{\substack{U \subset V \\ |U| \leq \frac{|V|}{2}}} \frac{\text{cut}(U, U^c)}{|U|}$$

We expressed, using linear algebra, the combinatorial problem of finding the partitions that realize these minimums.

Let us suppose we are given an ordering $V = \{v_1, v_2, \dots, v_n\}$ in the set of vertices. A vector $x \in \mathbb{R}^V$ has an entry x_i for each v_i . The *characteristic vector* c_S of a set $S \in V$ is $c_S = (x_i)_{i=1, \dots, n}$ with:

$$x_i = \begin{cases} 1 & \text{if } v_i \in S \\ 0 & \text{if } v_i \notin S \end{cases}$$

Sometimes it is preferable to use other values than 0 or 1 in the vector expression of a combinatorial object like a subset or partition [4]. For two real values b_1, b_2 , the (b_1, b_2) -indicator vector of a partition (V_1, V_2) is the vector $x = (x_i)_{i=1, \dots, n}$ with

$$x_i = \begin{cases} b_1 & \text{if } v_i \in V_1 \\ b_2 & \text{if } v_i \in V_2 \end{cases}$$

For example, the $(0,1)$ -indicator is the characteristic of the second set of the partition. We mainly use $(1,-1)$ -indicators. We denote $x \cdot y$, which is the standard scalar product in \mathbb{R}^n . In this way, a matrix A has associated the bilinear form $x \cdot Ay$. The vector $\mathbf{1}$ has the value 1 in each component. The degree vector is $g = (d_1, d_2, \dots, d_n)$ where $d_i = \text{deg}(v_i)$.

Lemma 1. Being $S, T \subset V$ of characteristic vectors c_S, c_T :

- (i) $\mathbf{1} \cdot c_S = |S|$. Also $c_S \cdot c_S = |S|$.
- (ii) $c_S \cdot c_T = |S \cap T|$.
- (iii) If A is the adjacency matrix of a graph, the vector $A\mathbf{1}$ has, in the i -th entry, the degree $d_i = \text{deg}(v_i)$. That is, $A\mathbf{1} = g$. Also $\mathbf{1} \cdot A\mathbf{1} = \sum_i d_i$.
- (iv) Ac_S has, in entry i -th, the number of edges to v_i from vertices in S .

Proof. They are straightforward. \square

Lemma 2. Being c_1 and c_2 characteristic vectors of the sets of a partition (V_1, V_2) :

$$\text{cut}(V_1, V_2) = c_1 \cdot Ac_2$$

Proof. By Lemma 1 (iv), Ac_2 contains in i -th entry the total weight of edges between v_i and V_2 . Hence, $c_1 \cdot Ac_2$ is the sum of the weight of the edges in the set $V_1 \sim V_2$, defined as

$$V_1 \sim V_2 = \{u \sim v \mid u \in V_1, v \in V_2\},$$

that is, $\text{cut}(V_1, V_2)$. \square

Calling $D_g = \text{diag}(g)$ the matrix with the degree vector g in the diagonal and zero off-diagonal, we have $\mathbf{1} \cdot D_g \mathbf{1} = \sum_i d_i$. Being x the $(1,-1)$ -indicator of any partition, we also have $x \cdot D_g x = \sum_i d_i$, because the minus signs appear in pairs.

Defining the Laplacian as $L = D_g - A$, we have:

Lemma 3. If x is the $(1,-1)$ -indicator of (V_1, V_2) ,

$$\text{cut}(V_1, V_2) = \frac{x \cdot Lx}{4}$$

Proof. If x is the $(1,-1)$ -indicator of (V_1, V_2) , $x = c_1 - c_2$ with c_1 and c_2 characteristic of V_1 and V_2 , respectively, and:

$$\begin{aligned} x \cdot Ax &= (c_1 - c_2) \cdot A(c_1 - c_2) = c_1 \cdot Ac_1 + c_2 \cdot Ac_2 - (c_1 \cdot Ac_2 + c_2 \cdot Ac_1) = \\ &= c_1 \cdot Ac_1 + c_2 \cdot Ac_2 - 2\text{cut}(V_1, V_2) \end{aligned}$$

Besides, as $c_1 + c_2 = \mathbf{1}$, that is, $c_1 = \mathbf{1} - c_2$, then $c_1 \cdot Ac_1 = c_1 \cdot A(\mathbf{1} - c_2) = c_1 \cdot A\mathbf{1} - c_1 \cdot Ac_2$. Likewise, $c_2 \cdot Ac_2 = c_2 \cdot A\mathbf{1} - c_2 \cdot Ac_1$, hence

$$\begin{aligned} c_1 \cdot Ac_1 + c_2 \cdot Ac_2 &= c_1 \cdot A\mathbf{1} - c_1 \cdot Ac_2 + c_2 \cdot A\mathbf{1} - c_2 \cdot Ac_1 = \\ &= (c_1 + c_2) \cdot A\mathbf{1} - (c_1 \cdot Ac_2 + c_2 \cdot Ac_1) = \mathbf{1} \cdot A\mathbf{1} - 2\text{cut}(V_1, V_2) \end{aligned}$$

So,

$$\begin{aligned} x \cdot Ax &= c_1 \cdot Ac_1 + c_2 \cdot Ac_2 - 2\text{cut}(V_1, V_2) = \\ &= \mathbf{1} \cdot A\mathbf{1} - 2\text{cut}(V_1, V_2) - 2\text{cut}(V_1, V_2) = \\ &= \sum_i d_i - 4\text{cut}(V_1, V_2). \end{aligned}$$

That is, $x \cdot Ax = \sum_i d_i - 4\text{cut}(V_1, V_2)$. Solving for the cut, we have $4\text{cut}(V_1, V_2) = \sum_i d_i - x \cdot Ax$. As $\sum_i d_i = x \cdot D_g x$, and $L = D_g - A$, we can express $\sum_i d_i - x \cdot Ax = x \cdot D_g x - x \cdot Ax = x \cdot (D_g x - Ax) = x \cdot (D_g - A)x = x \cdot Lx$. Therefore,

$$\text{cut}(V_1, V_2) = \frac{x \cdot Lx}{4}.$$

□

We deduced this well-known identity in matrix form, instead of summatory form as usual. So we not only avoided the index chasing but also made explicit the role of the values b_1, b_2 used in indicators. For example, if x is a $(\frac{1}{2}, -\frac{1}{2})$ -indicator of (V_1, V_2) , then $\text{cut}(V_1, V_2) = x \cdot Lx$. In general [4], for (b_1, b_2) -indicators, the cut is $\frac{x \cdot Lx}{(b_2 - b_1)^2}$. This deduction also clarifies the role of the diagonal degree matrix D_g .

In addition to the expression of the cost as a bilinear form L , we expressed the requirement that the partition (V_1, V_2) be balanced as $\mathbf{1} \cdot x = 0$. So, the problem of finding the bipartition of minimal cost is the following problem of combinatorial optimization:

$$\begin{aligned} &\underset{x}{\text{Minimize}} && x \cdot Lx \\ &\text{subject to} && \begin{cases} x_i = \pm 1, i = 1, \dots, n \\ \mathbf{1} \cdot x = 0 \end{cases} \end{aligned}$$

This combinatorial problem is an NP-complete problem [12]. To approximate a solution, with polynomial computational cost, it is customary to relax the constraints $x_i = \pm 1$. This relaxed problem is a numerical one that has several features that ease its resolution: L is symmetrical, hence its eigenvalues are real, and there is an orthonormal basis of eigenvectors [13]. Besides, $\mathbf{1}$ is an eigenvector of eigenvalue 0, because $D_g \mathbf{1} - A\mathbf{1} = 0$. Additionally, L is a weak diagonally dominant of positive diagonal, hence (by the theorem of the Geršgorin discs [14]) its eigenvalues are nonnegative $0 = \lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_k$. If G is connected, $\lambda_0 < \lambda_1$ [3]. These features of L are generally deduced from its expression as summatory of squares, which we have avoided. The result about diagonal dominance that we used instead is also easy to see.

The *Rayleigh quotient* of a $n \times n$ symmetric matrix M is

$$R_M(x) = \frac{x \cdot Mx}{x \cdot x}$$

defined for $x \neq 0$ in \mathbb{R}^n . It plays a role in the following min-max theorem of Courant-Weyl [13], which we use without proof:

Theorem 1. Let S_k be the set of subspaces of \mathbb{R}^n of dimension lesser or equal than k , for $k = 1, 2, \dots, n$, M symmetric with eigenvalues $\lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_{n-1}$ and corresponding eigenvectors f_0, f_1, \dots, f_{n-1} . Then

$$\lambda_{k-1} = \min_{E \in S_k} \max_{\substack{x \in E \\ \|x\|=1}} R_M(x).$$

Besides, the argument E giving the minimum is $E = \text{span}(f_0, f_1, \dots, f_{k-1})$, and an argument x giving the maximum is f_{k-1} .

In particular, $\lambda_0 = \min_{\substack{x \in \mathbb{R}^n \\ \|x\|=1}} R_M(x)$, because each $x \neq 0$ span an $E \in \mathcal{S}_1$. In the case that $M = L$, the eigenvector x_0 is (a scalar multiple of) $\mathbf{1}$, as commented above, and the others are orthogonal to it: $\mathbf{1} \cdot f_i = 0, i = 1, \dots, k - 1$. In the case $i = 1$,

$$\lambda_1 = \min_{E \in \mathcal{S}_2} \max_{\substack{x \in E \\ \|x\|=1}} R_L(x) = \min_{\substack{x \in \mathbb{R}^n \\ \mathbf{1} \cdot x=0, \|x\|=1}} R_L(x)$$

because each x with $\mathbf{1} \cdot x = 0$ span a $E = \langle \mathbf{1}, x \rangle \in \mathcal{S}_2$. The minimum is reached in f_1 .

To relate this result with the cut value of partitions note that if x an indicator vector of a partition ($x_i = \pm 1$), the cut $\frac{x \cdot Lx}{4}$ is proportional to the Rayleigh quotient. The minimum $\lambda_1 = \min_{\mathbf{1} \cdot x=0} \frac{x \cdot Lx}{x \cdot x}$ is reached in a vector f_1 that is an eigenvector for λ_1 . That is, f_1 is a solution to the relaxed problem, although it may not be an indicator vector. The first non-null eigenvalue λ_1 is termed the *Fiedler value*, and its eigenvector f_1 is the *Fiedler vector*.

There are several rounding or truncation methods to obtain an indicator vector $x = (x_i)_i$ (i.e., with integer values $-1, 1$ for x_i) from f_1 (whose entries are not necessarily integers). The most direct rounding is the partition by sign: $x_i = 1$ if $(f_1)_i > 0, x_i = -1$ otherwise. This rounding can give a partition that is not a bipartition. Another rounding method uses the median, achieving precisely bipartitions: if m is the median value of the entries of $f_i, x_i = 1$ if $f_i > m$ and $x_i = -1$ otherwise.

For these rounding methods and others that appear in the literature [15], there is an error bound. The error is the difference between the partition obtained by the rounding and the partition that actually minimizes the cut, which is obtainable by combinatorial methods. These bounds are known as discrete Cheeger bounds since they involve the first nonzero eigenvalue. In particular, we use the following Cheeger bound developed by Mohar [16]. It compares the cut ratio $i(U_s) = \frac{\text{cut}(U_s, U_s^c)}{|U_s|}$ of the partition induced by the sign rounding, $U_s = \{x_i \mid (f_1)_i > 0\}$, and the isoperimetric number $i(G)$.

Theorem 2. *If G has more than three vertexes and its maximal degree is Δ , then*

$$\frac{\lambda_1}{2} \leq i(G) \leq i(U_s) \leq \sqrt{\lambda_1(2\Delta - \lambda_1)}$$

This bound uses the sign rounding and the cut ratio, $i(U_s)$, instead of the median rounding and the edge cut, $\text{cut}(U, U^c)$, which we use to describe our framework. However, we chose it because it is easier to present the generalization that we make for vertex-weighted graphs in the next section. The cut ratio has also been studied in the stochastic setting [17]. In principle, it is also possible to generalize for vertex-weighted graphs the similar bounds that there are in the literature for median rounding and edge cut [18].

4. Laplacians with Potential

To motivate our contribution, we recall here the usual interpretation, for instance in [19], of the values of a weight s on vertices using flows on graphs. The vertex weight $s(v_i)$ corresponds to the amount or magnitude of some physical substance placed at v_i . The weights at the edges $w_{ij}, i \neq j$ between different vertices correspond, in this interpretation, to a transmission factor or gain that affects the substance when it flows from one vertex v_i to another v_j , increasing or decreasing its amount. Following this interpretation of the weight matrix as a transference or shift, the weights in the loop edges w_{ii} are the gain that suffers the substance that stays in the same vertex v_i .

In this diffusion process interpretation, the eigenvectors of a shift matrix are the stationary substance distributions. In particular, the Laplacian matrix has a stationary distribution that is uniform (corresponding to the null eigenvalue) because the degree values on the diagonal make the total gain of substance equal to zero. In this sense, the Laplacian process is conservative. Another example is the shift by the adjacency matrix of a connected graph, which has a positive stationary distribution (the random walk limit

distribution, corresponding to the Perron eigenvector [3]), with a substance gain given by the Perron eigenvalue.

Following this vein, we show how to control the weights in the diagonal w_{ii} to obtain any positive $\rho : V \rightarrow \mathbb{R}$ as a stationary distribution, the eigenvector of a matrix similar to the Laplacian.

As commented, a weight on vertices is a function $p : V \rightarrow \mathbb{R}$. Its diagonal form is the matrix $D_p = \text{diag}(p(x_i))$. The *generalized Laplacian with potential p* (or *p-Laplacian*) is:

$$L_p = L + D_p$$

That is, $L_p = D_g - A + D_p$. In particular, with $p = 0$, the 0-Laplacian is the ordinary Laplacian. Some properties of p -Laplacians are similar to those of ordinary Laplacians, as can be viewed in [20] under the name of generalized Laplacian. For our purposes, we highlight the following ones, whose proof require some technicalities:

Lemma 4. *If the graph G is connected and the potential p verifies*

$$p(v_i) \geq -\text{deg}(v_i),$$

then:

- (a) *The eigenvalues of L_p are real, and the minimum eigenvalue has multiplicity 1. That is, $\lambda_0 < \lambda_1 \leq \dots \leq \lambda_{n-1}$.*
- (b) *There is a positive eigenvector corresponding to λ_0 , unique up to a scalar multiple.*

Proof. If $p(v_i) \geq -\text{deg}(v_i)$, then $L_p = D_g - A + D_p$ is a symmetrical Z-matrix [21]. As G is connected, L_p is irreducible. By Observation 1.4.3 of [21], the claims follow. \square

The minimum eigenvalue is the *Perron eigenvalue* λ_0 . It has multiplicity 1, and there is an eigenvector of the Perron eigenvalue with all positive entries. To fix one such eigenvector, we define the *Perron eigenvector* ρ as that with $\|\rho\| = 1$.

The min-max theorem for the operator L_p gives us that $\lambda_1 = \min_{x \neq 0} \frac{x \cdot L_p x}{x \cdot x}$, and the minimum is reached in an eigenvector of λ_1 of norm 1 (the *Fiedler vector* ϕ).

With these properties, we can replicate the spectral partition methodology because the spectral decomposition of L_p assures that $\phi \cdot \rho = 0$. This can be understood, as in the ordinary Laplacian L above, that the positive and negative values of the Fiedler vector give us an indicator of two sets of vertexes. This indicator cuts V in two parts of equal absolute sum of Perron values. That is, $\sum_{x \in V} \phi(x)\rho(x) = 0$ and if $V_1 = \{x_i \mid \phi(x_i) > 0\}$, $V_2 = V_1^c$, then

$$\sum_{x \in V_1} \phi(x)\rho(x) = \sum_{x \in V_2} |\phi(x)|\rho(x)$$

However, in this case, the Perron vector is not the constant distribution, $\mathbf{1}$, but a positive distribution ρ . We use the distribution ρ as a measure of the relative importance of the vertexes in a partition or clustering.

Note that the Perron eigenvector ρ is defined up to a constant factor, in the sense that $\rho' = k\rho$, for $k > 0$, is also a positive eigenvector of the same eigenvalue. Our choice of $\|\rho\| = 1$ is conventional.

Note also that, as in the ordinary case commented in Section 3, the Fiedler vector ϕ does not necessarily have ± 1 components and should be considered as an approximation to the optimal partition.

To build a potential p such that the Perron distribution of L_p will be a predefined given ρ , we apply the formula of the following theorem. Note that it is scale-invariant, in the sense that ρ and $k\rho$, for $k > 0$, will produce the same potential p .

Remember that for a vector x , we denote its i -th component as x_i , and a function $x : V \rightarrow \mathbb{R}$ is identified with the vector $x_i = x(v_i)$. We can build a potential p such that the Perron vector ρ of L_p have predetermined positive values ρ_i :

Theorem 3. For any vector ρ such that $\rho_i > 0$ for $i = 1, \dots, n$, if

$$p(v_i) = \frac{(A\rho)_i}{\rho_i} - \text{deg}(v_i),$$

being $A = (a_{ij})$ the adjacency matrix, then the Perron vector of L_p is ρ , and the Perron value is 0.

Proof. As p verifies the hypothesis of Lemma 4, L_p has a Perron eigenvalue. Note that for each $i = 1, \dots, n$, $(D_g\rho)_i = \text{deg}(v_i)\rho_i$. Also

$$(D_p\rho)_i = p(v_i)\rho_i = (A\rho)_i - \text{deg}(v_i)\rho_i.$$

Therefore

$$\begin{aligned} (L_p\rho)_i &= ((D_g - A + D_p)\rho)_i = (D_g\rho)_i - (A\rho)_i + (D_p\rho)_i = \\ &= \text{deg}(v_i)\rho_i - (A\rho)_i + ((A\rho)_i - \text{deg}(v_i)\rho_i) = 0. \end{aligned}$$

So, $L_p\rho = 0$. To conclude that 0 and ρ are the Perron eigenvalue and eigenvector, we use the fact that in a symmetrical matrix the eigenvectors of different eigenvalues are orthogonal. Consequently, only one eigenvalue can have associated a positive eigenvector as ρ , and this is 0. \square

With this result, we can do spectral partition with preassigned weights ρ on the vertexes. By the above discussion, the p -Laplacian for the potential p corresponding to the given ρ has a Fiedler vector orthogonal to the Perron vector ρ (that is, it produces a partition in parts of equal total weight at the vertices). The value of the cut is also well expressed by the p -Laplacian, in the following lemma. For a vertex set $U \subset V$, we denote $p(U) = \sum_{v \in U} p(v)$.

Lemma 5. If x is the $(1, -1)$ -indicator of (V_1, V_2) ,

$$\text{cut}(V_1, V_2) = \frac{x \cdot L_p x}{4} - \frac{p(V)}{4}$$

Proof. As $L_p = L + D_p$,

$$x \cdot L_p x = x \cdot (L + D_p)x = x \cdot (Lx + D_p x) = x \cdot Lx + x \cdot D_p x$$

By Lemma 3, $x \cdot Lx = 4\text{cut}(V_1, V_2)$, and $x \cdot D_p x = \sum_i p(v_i) = p(V)$, hence the claim. \square

Therefore, to find a partition (i.e., an indicator x) that minimizes $x \cdot L_p x$ is equivalent to minimizing $\text{cut}(V_1, V_2)$, because $p(V)/4$ is a constant, given the potential p .

For a predefined weight on vertices $\rho = (\rho_1, \rho_2, \dots, \rho_n)$, we build the potential p by Theorem 3. The p -Laplacian L_p specifies the vertex ρ -weighted bipartition problem:

$$\begin{aligned} &\text{Minimize}_x && x \cdot L_p x \\ &\text{subject to} && \begin{cases} x_i = \pm 1, i = 1, \dots, n \\ \rho \cdot x = 0 \end{cases} \end{aligned}$$

This combinatorial problem is at least NP-complete as the conventional bipartition problem, which includes it as a subproblem. In any case, we consider the relaxed problem (without the restriction $x_i = \pm 1$) that has as solution, by the min-max Theorem 1, the Fiedler vector ϕ of L_p .

By taking this Fiedler vector as an approximation to the combinatorial solution, that is, the unrelaxed problem, the error can be bounded with a Cheeger expression similar to

that of Mohar (Theorem 2). To express this bound, in the following Theorem 4, we define the cut ratio of U with respect to p as

$$i_p(U) = \frac{\text{cut}(U, U^c)}{p(U)},$$

and the isoperimetric number with respect to p as

$$i_p(G) = \min_{\substack{U \subset V \\ p(U) \leq \frac{p(V)}{2}}} \frac{\text{cut}(U, U^c)}{p(U)}.$$

Given a vector $x \in R^V$, $x_i = x(v_i)$, we order the set of its values as $t_1 > t_2 > \dots > t_m$. That is, for each i in $1, \dots, n$ there is one j in $1, \dots, m$ and only one with $t_j = x(v_i)$. The level sets of x are, for each $k = 1, \dots, m$, $V_k = \{v \in V \mid x(v) \geq t_k\}$. The sweep cut $h_p(x)$ is the minimum cut following level sets of x , that is:

$$h_p(x) = \min_{k=1, \dots, m} \frac{\text{cut}(V_k, V_k^c)}{p(V_k)}$$

It is clear that, for any x , $i_p(G) \leq h_p(x)$. We define also $S(x)$, the increment of square values along x , as

$$S(x) = \sum_{v_i \sim v_j} |x_i^2 - x_j^2|.$$

Lemma 6. Let L be the ordinary Laplacian, for any x :

- (a) $h_p(x) x \cdot D_p x \leq S(x)$.
- (b) $S(x) \leq \sqrt{x \cdot Lx} \sqrt{2x \cdot D_g x}$.

Proof. For (a), consider the level sets of x , that is, vertices sorted such that $x_1 \geq x_2 \geq \dots \geq x_n$, being $t_1 > t_2 > \dots > t_m$ the different values of x . For each level set V_k , by the definition of sweep cut we have:

$$h_p(x)p(V_k) \leq \text{cut}(V_k, V_k^c) \tag{1}$$

Besides

$$\begin{aligned} x \cdot D_p x &= \sum_{i=1}^n p(v_i)x_i^2 = \\ &= \sum_{v_i \in V_1} p(v_i)t_1^2 + \sum_{v_i \in V_2 \setminus V_1} p(v_i)t_2^2 + \sum_{v_i \in V_3 \setminus V_2} p(v_i)t_3^2 + \dots + \sum_{v_i \in V_m \setminus V_{m-1}} p(v_i)t_m^2 = \\ &= \sum_{v_i \in V_1} p(v_i)t_1^2 + \sum_{v_i \in V_2} p(v_i)t_2^2 - \sum_{v_i \in V_1} p(v_i)t_2^2 + \sum_{v_i \in V_3} p(v_i)t_3^2 - \sum_{v_i \in V_2} p(v_i)t_3^2 + \dots \\ &\quad \dots + \sum_{v_i \in V_m} p(v_i)t_m^2 - \sum_{v_i \in V_{m-1}} p(v_i)t_m^2 = \\ &= p(V_1)(t_1^2 - t_2^2) + p(V_2)(t_2^2 - t_3^2) + \dots + p(V_{m-1})(t_{m-1}^2 - t_m^2) + p(V_m)t_m^2 = \\ &= \sum_{k=1}^m p(V_k)(t_k^2 - t_{k+1}^2) \tag{2} \end{aligned}$$

extending with a value $t_{m+1} = 0$ to ease the notation.

Combining (1) and (2):

$$\begin{aligned} h_p(x)x \cdot D_p x &= \sum_{k=1}^m h_p(x)p(V_k)(t_k^2 - t_{k+1}^2) \leq \sum_{k=1}^m \text{cut}(V_k, V_k^c)(t_k^2 - t_{k+1}^2) = \\ &= \sum_{k=1}^m \sum_{\substack{v_i \sim v_j \\ x_i \in V_k \\ x_j \in V_k^c}} (t_k^2 - t_{k+1}^2) = \sum_{k=1}^m \sum_{\substack{v_i \sim v_j \\ x_i \geq t_k \\ x_j < t_k}} (t_k^2 - t_{k+1}^2) = \sum_{v_i \sim v_j} |x_i^2 - x_j^2| = S(x). \end{aligned}$$

The last equality is because, for each pair of vertices $v_i \sim v_j$, if $x_i = t_{k_0}$ and $x_j = t_{k_1}$ with $1 \leq k_0 < k_1 \leq m$, the double summation includes a chain of terms $(t_{k_0}^2 - t_{k_0+1}^2) + (t_{k_0+1}^2 - t_{k_0+2}^2) + \dots + (t_{k_1-1}^2 - t_{k_1}^2)$ that collapses to $t_{k_0}^2 - t_{k_1}^2 = |x_i^2 - x_j^2|$.

For (b), we consider vectors $u, v \in \mathbb{R}^E$, that is, indexed by the edges

$$u = (u_1, u_2, \dots, u_{|E|}) = (u_{i,j})_{v_i \sim v_j}.$$

If we apply the Cauchy–Schwarz inequality $u \cdot v \leq \|u\| \|v\|$ to the particular vectors of \mathbb{R}^E $u = (|x_i - x_j|)_{v_i \sim v_j}$ and $v = (x_i + x_j)_{v_i \sim v_j}$, as $u \cdot v = (|x_i - x_j|(x_i + x_j))_{v_i \sim v_j} = (|x_i^2 - x_j^2|)_{v_i \sim v_j}$, we have:

$$S(x) = \sum_{v_i \sim v_j} |x_i^2 - x_j^2| = \sum_{v_i \sim v_j} |x_i - x_j|(x_i + x_j) \leq \sqrt{\sum_{v_i \sim v_j} |x_i - x_j|^2} \sqrt{\sum_{v_i \sim v_j} (x_i + x_j)^2}$$

For the first root factor, it is known that $x \cdot Lx = \sum_{v_i \sim v_j} (x_i - x_j)^2$ being L the ordinary Laplacian (see for example [15]). For the second root factor, by the trivial fact that $(a + b)^2 \leq 2(a^2 + b^2)$ we have:

$$\sum_{v_i \sim v_j} (x_i + x_j)^2 \leq 2 \sum_{v_i \sim v_j} (x_i^2 + x_j^2) = 2 \sum_i \text{deg}(v_i)x_i^2 = 2x \cdot D_g x.$$

□

To prove the following claim, we apply the above lemma to ϕ , the Fiedler eigenvalue of L_p .

Theorem 4. Being $m_p = \min_i(p(v_i))$, $\Delta = \max_i(\text{deg}(v_i))$, λ_1 and ϕ the Fiedler eigenvalue and eigenvector of L_p , then:

$$\frac{n\lambda_1 - p(V)}{4} \leq i_p(G) \leq h_p(\phi) \leq \sqrt{\frac{2\Delta(\lambda_1 - m_p)}{m_p^2}}$$

Proof. For the first inequality, calling x_b the $(-1,1)$ -indicator of the minimal bipartition (V_1, V_2) , that is, such that $i_p(G) = \text{cut}(V_1, V_2)$, by Lemma 5 we have:

$$i_p(G) = \text{cut}(V_1, V_2) = \frac{x_b \cdot L_p x_b - p(V)}{4},$$

that is, $x_b \cdot L_p x_b = 4i_p(G) + p(V)$. Besides, being $\lambda_1 = \min_{x \neq 0} \frac{x \cdot L_p x}{x \cdot x}$ for each x is $\lambda_1 x \cdot x \leq x \cdot L_p x$. In particular, $\lambda_1 n = \lambda_1 x_b \cdot x_b \leq x_b \cdot L_p x_b$, hence $\lambda_1 n \leq 4i_p(G) + p(V)$ and:

$$\frac{n\lambda_1 - p(V)}{4} \leq i_p(G)$$

For the second inequality, being $h_p(\phi)$, a particular cut (the sweep cut of ϕ) the minimal cut $i_p(G)$ is lesser or equal than it.

Finally, for the third inequality, we use Lemma 6 inequalities for the particular case $x = \phi$, that is

- (a) $h_p(\phi) \phi \cdot D_p \phi \leq S(\phi)$
- (b) $S(\phi) \leq \sqrt{\phi \cdot L \phi} \sqrt{2 \phi \cdot D_g \phi}$.

From (a) we have $h_p(\phi) \leq \frac{S(\phi)}{\phi \cdot D_p \phi}$, that from (b) is lesser or equal than $\frac{\sqrt{\phi \cdot L \phi} \sqrt{2 \phi \cdot D_g \phi}}{\phi \cdot D_p \phi}$. The value of these terms are $\phi \cdot D_g \phi = \sum_i \deg(v_i) \phi_i^2$ and $\phi \cdot D_p \phi = \sum_i p(v_i) \phi_i^2$. Additionally, as $L_p = L + D_p$ and $L_p \phi = \lambda_1 \phi$, we have

$$\phi \cdot L \phi = \phi \cdot L_p \phi - \phi \cdot D_p \phi = \lambda_1 \sum_i \phi_i^2 - \sum_i p(v_i) \phi_i^2 = \sum_i (\lambda_1 - p(v_i)) \phi_i^2.$$

Substituting those values in the inequalities:

$$\begin{aligned} h_p(\phi) &\leq \frac{S(\phi)}{\phi \cdot D_p \phi} \leq \frac{\sqrt{2 \phi \cdot L \phi} \sqrt{\phi \cdot D_g \phi}}{\phi \cdot D_p \phi} \leq \frac{\sqrt{2 \sum_i (\lambda_1 - p(v_i)) \phi_i^2} \sqrt{\sum_i \deg(v_i) \phi_i^2}}{\sum_i p(v_i) \phi_i^2} \leq \\ &\leq \sqrt{2 \frac{\sum_i (\lambda_1 - p(v_i)) \phi_i^2}{\sum_i p(v_i) \phi_i^2} \frac{\sum_i \deg(v_i) \phi_i^2}{\sum_i p(v_i) \phi_i^2}} \leq \sqrt{2 \frac{(\lambda_1 - m_p) \sum_i \phi_i^2}{m_p \sum_i \phi_i^2} \frac{\Delta \sum_i x_i^2}{m_p \sum_i \phi_i^2}} = \\ &= \sqrt{\frac{2 \Delta (\lambda_1 - m_p)}{m_p^2}} \end{aligned}$$

The last inequality is because, by definition of m_p and Δ , for each $i = 1, \dots, n$, we have $\deg(v_i) \leq \Delta$, $p(v_i) \geq m_p$ and also $(\lambda - p(v_i)) \leq (\lambda - m_p)$. \square

This result is similar to the above Theorem 2, in this case bounding the cost of the minimal cut $i_p(G)$ and the sweep cut $h_p(\phi)$ of the Fiedler eigenvector of L_p .

5. Conclusions

We introduced a spectral partitioning method for arbitrary (positive) vertex weights not related to edges' weights or vertex degrees. We also provided a bound of the Cheeger type for the error in taking the eigenvector numerical solution as an approximation for the combinatorial problem. This bound is similar to others that appear in the literature for cuts with uniform vertex weights.

In this work, we did not take into account a distribution of weights on the edges. The usual methods of considering ordinary Laplacians with edge weights (for example, [3]) can be extended to generalized Laplacians as we exposed (that is, including a vertex potential that induces vertex weighting in the spectral partitioning). The two types of weights, on vertices and on edges, are independent. This was not considered for simplicity, but it will be the content of future work.

Author Contributions: Conceptualization, J.-L.G.-Z. and C.G.; Writing—original draft, J.-L.G.-Z.; Writing—review and editing, C.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was jointly supported by the European Regional Development Fund "A way to achieve Europe" (ERDF) and the Extremadura Local Government (Ref. IB20040) and by the Spanish Ministerio de Ciencia e Innovación through project PID2019-110315RB-I00 (APRISA).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Schaeffer, S.E. Survey: Graph Clustering. *Comput. Sci. Rev.* **2007**, *1*, 27–64. [[CrossRef](#)]
2. Shang, Y. Generalized K-Core Percolation in Networks with Community Structure. *SIAM J. Appl. Math.* **2020**, *80*, 1272–1289. [[CrossRef](#)]
3. Chung, F.R. *Spectral Graph Theory*; CBMS; American Mathematical Soc.: Providence, RI, USA, 1997; Volume 92.
4. Shewchuk, J.R. *Allow Me to Introduce Spectral and Isoperimetric Graph Partitioning*; Technical Report; University of California Berkeley: Berkeley, CA, USA, 2016.
5. Ng, A.Y.; Jordan, M.I.; Weiss, Y. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*; (NIPS 2001); MIT press: Cambridge, MA, USA, 2002; pp. 849–856.
6. Martins, A.; Grácio, C.; Teixeira, C.; Rodrigues, I.P.; Zapata, J.L.; Ferreira, L. Historia Augusta Authorship: An Approach based on Measurements of Complex Networks. *Appl. Netw. Sci.* **2021**, *6*, 50. [[CrossRef](#)]
7. Xu, S.; Fang, J.; Li, X. Weighted Laplacian Method and Its Theoretical Applications. In *IOP Conference Series: Materials Science and Engineering 2020*; IOP Publishing: Bristol, UK, 2020; Volume 768, p. 072032.
8. Da Costa, G.; Lastovetsky, A.; Barbosa, J.; Díaz-Martín, J.C. Chapter 2: “Programming models and runtimes”. In *Ultrascale Computing Systems*; Carretero, J., Jeannot, E., Zomaya, A., Eds.; Institution of Engineering and Technology: London, UK, 2019.
9. Berman, A.; Plemmons, R.J. *Nonnegative Matrices in the Mathematical Sciences*; SIAM: Philadelphia, PA, USA, 1994.
10. Cvetkovic, D.M.; Rowlinson, P.; Simic, S. *An Introduction to the Theory of Graph Spectra*; Cambridge University Press: Cambridge, UK, 2010.
11. Beineke, L.W.; Wilson, R.J.; Cameron, P.J. (Eds.) *Topics in Algebraic Graph Theory*; Encyclopedia of Mathematics and Its Applications; Cambridge University Press: Cambridge, UK, 2004; Volume 102.
12. Papadimitriou, C.H.; Steiglitz, K. *Combinatorial Optimization: Algorithms and Complexity*; Dover Publications: Mineola, NY, USA, 1998.
13. Lancaster, P.; Tismenetsky, M. *The Theory of Matrices: With Applications*; Computer Science and Scientific Computing Series; Academic Press: Cambridge, MA, USA, 1985.
14. Horn, R.A.; Johnson, C.R. *Matrix Analysis*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2012.
15. Spielman, D.A.; Teng, S.H. Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra Appl.* **2007**, *421*, 284–305. [[CrossRef](#)]
16. Mohar, B. Isoperimetric numbers of graphs. *J. Comb. Theory Ser. B* **1989**, *47*, 274–291. [[CrossRef](#)]
17. Shang, Y. Isoperimetric Numbers of Randomly Perturbed Intersection Graphs. *Symmetry* **2019**, *11*, 452. [[CrossRef](#)]
18. Boppana, R.B. Eigenvalues and graph bisection: An average-case analysis. In *Proceedings of the 28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, Los Angeles, CA, USA, 12–14 October 1987; IEEE: Piscataway, NJ, USA, 1987; pp. 280–285.
19. Chung, F.R. Laplacians of graphs and Cheeger’s inequalities. *Comb. Paul Erdos Eighty* **1996**, *2*, 13–20.
20. Bıyıkoglu, T.; Leydold, J.; Stadler, P.F. *Laplacian Eigenvectors of Graphs*; Lecture Notes in Mathematics; Springer: Berlin/Heidelberg, Germany, 2007; Volume 1915.
21. Molierno, J.J. *Applications of Combinatorial Matrix Theory to Laplacian Matrices of Graphs*; CRC Press: Boca Raton, FL, USA, 2012.