Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

IMPROVEMENTS ON SPECTRAL BISECTION*

ISRAEL DE SOUZA ROCHA†

Abstract. In this paper, the third eigenvalue of the Laplacian matrix is used to provide a lower bound on the minimum cutsize. This result has algorithmic implications that are exploited in this paper. Besides, combinatorial properties of certain configurations of a graph partition which are related to the minimality of a cut are investigated. It is shown that such configurations are related to the third eigenvector of the Laplacian matrix. It is well known that the second eigenvector encodes structural information, and that can be used to approximate a minimum bisection. In this paper, it is shown that the third eigenvector carries structural information as well. Then a new spectral bisection algorithm using both eigenvectors is provided. The new algorithm is guaranteed to return a cut that is smaller or equal to the one returned by the classic spectral bisection. Also, a spectral algorithm that can refine a given partition and produce a smaller cut is provided.

Key words. Graph partitioning, Graph bisection, Spectral partitioning.

AMS subject classifications. 05C85, 15A18, 90C10, 90C22, 90C35, 90C27.

1. Introduction. The classic problem of finding a minimum cut of a graph is known to be NP-hard. Nevertheless, the problem has direct applications in VLSI design, data-mining, finite elements and communication in parallel computing, etc. In practice, given the importance of the problem, the solution is generally approximated using heuristic algorithms. The problem is to separate the vertices of a graph in two parts, such that the number of edges connecting vertices in different parts is minimized. Such a partition, also known as a cut, is called a balanced cut or a bisection whenever both parts have the same size.

In many applications, it is desired to obtain the smallest possible cut at a cost of having a partition that is not balanced, but acceptable in the sense both parts have almost the same size. However, even for those cases efficient algorithms that approximates balanced cuts up to a constant factor do not exist. In fact, this approximation problem is NP-hard [2].

Spectral techniques are well-known approaches to this problem and they have its roots in the work of Fiedler [10] and Donath and Hoffman [8, 9]. These spectral methods are known to provide good answers, and they are broadly used in several problems [20, 23, 24]. Spectral partitioning algorithms recover global structural information and connectivity of a graph by means of an eigenvector of the second eigenvalue of the Laplacian matrix of the graph.

In [26], Spielman and Teng provided a recursive spectral bisection algorithm and showed that spectral partitioning methods work well on bounded-degree planar graphs. Guattery and Miller [14] perform an analysis of the quality of the separators produced by such methods. Papers [26] and [14] discuss the difference between guarantees on the size of a balanced cut versus its optimality. Hendrickson and Leland [20] extend the spectral approach to partition a graph into four or eight parts by using multiple eigenvectors. Furthermore, Lee, Gharan, and L. Trevisan [21] provides theoretical justification for the use of multiple eigenvectors in

*Received by the editors on June 12, 2018. Accepted for publication on August 25, 2020. Handling Editor: Bryan L. Shader.
†The Czech Academy of Sciences, Institute of Computer Science, Pod Vodárenskou věží 2, 182 07 Prague, Czech Republic (israelrocha@gmail.com). This work was (partially) done while affiliated with Institute of Computer Science of the Czech Academy of Sciences, with institutional support RVO:67985807.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

multi-way clustering algorithms.

In this paper, instead of using structural information provided by multiple eigenvectors to partition graphs into multiple parts, we develop an approach that uses multiple eigenvectors to create a bisection of the graphs. It is well known that the second eigenvector encodes structural information, and that can be used to approximate a minimum bisection. In this paper, we show that the third eigenvector carries structural information as well, which enables us to apply that information in the bisection problem. We then provide a new spectral bisection algorithm using both eigenvectors.

Beyond the second eigenvector there is a large literature about graph spectra of graphs [5, 6, 7, 13, 18]. Going in a related direction of our investigation, Pati [22] investigated the connection between the third smallest eigenvalue and the graph structure and especially for trees. Grady and Schwartz [12] provide an isoperimetric algorithm derived and motivated by the isoperimetric constant, which share close similarities with spectral partitioning and in particular the method presented in this paper. Also related to our investigation is the work of Lang [17], which uses random hyperplane rounding of multiple eigenvectors. There, the author empirically investigates the behavior of six graph partitioning algorithms on power law graphs.

From a more general perspective, there are several heuristics for the graph partitioning problem, and they can be classified as either:

- Geometric - based solely on the coordinate information of the vertices;
- Combinatorial - which attempts to group together highly connected vertices;
- Spectral - formulates as the optimization of a discrete quadratic function. The relaxed counterpart of the discrete problem becomes a continuous one, which can be solved by computing the second eigenvector of the discrete Laplacian of the graph;
- Multilevel methods - a sequence of smaller graphs is constructed in order to produce a similar coarser graph. The initial bisection is performed on the smallest of these graphs. Finally, the graph is uncoarse and partition refinement is performed on each of the coarse graph.

Each method has its advantages and disadvantages, and many of them are described in [25], where we can find a detailed description of several different methods in each of these classes. Combining those methods is a common strategy to overcome the disadvantages. For instance, spectral schemes can use eigenvectors to produce coordinate information for vertices. Geometric methods can then use these coordinates to partition the graph. Usually, for each application it is unclear which method is better. There are many factors to be considered: degree of parallelism, run time, quality of the cut produced. Karypis and Kumar [16] evaluate different aspects for many combinations of methods. In general, it is agreed that spectral methods are good, especially multilevel spectral bisection.

In this paper, we aggregate more information present in the spectra to improve the traditional spectral bisection algorithm (SB) and produce a new graph bisection algorithm. While SB makes use of one eigenvector only, the new algorithm uses two eigenvectors, which allows us to returns a partition with cut size smaller or equal to the SB cut size. Besides, the additional running time of computing an extra eigenvector is rather small compared to the overall running time of SB.

One one hand, we are especially concerned with the theoretical relations of eigenvectors and cuts on graphs, and also show there is still more to be understood about these relations. Therefore, we do not intend to make an extensive comparison between different classes of algorithms and the new one, since the new algorithm is guaranteed to return a cut that is not worse than the one of SB, at a cost of a rather small

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

running time. Nevertheless, we present some numerical results comparing the quality of the cut between the new algorithm and SB. It is worth it to mention that there is no restriction on using the new algorithm in combination with other methods, and we expect that the new algorithm improves the existing mixed methods that make use of the traditional SB.

To reach our goal, we investigate properties of certain configurations of a graph partition which are related to the minimality of a cut and the structure of the graph, and we prove several results on that. Such configurations, that we call *organized partitions*, are shown in this paper to be related to eigenvectors of the Laplacian matrix. It turns out that organized partitions are related to the maximum cut problem as well, as we will show in Section 2. The ideas behind the results in Section 2 are purely combinatorial; that has algorithmic implications and we are exploiting it in the paper. We make it possible by formulating the problem of finding organized partitions in terms of an optimization problem. This formulation shows the link between organized partitions, minimum bisection, and eigenvectors of graphs.

Finally, we combine the organized partition, the third, and the second eigenvector to construct an algorithm that approximates a minimum graph bisection. For this algorithm, it is proven that the resulting partition has no more edges than the classical spectral bisection algorithm. Besides, we provide a second algorithm that can produce a smaller cut, given a known cut, a procedure known as refining a partition. There are several multilevel algorithms [3, 4, 15, 19] that further refine the partition during the uncoarsening phase. The second algorithm presented in this paper refines the partition by making use of the information about the organized partition present in the third eigenvector.

The rest of the paper is organized as follows: properties of organized partitions are investigated on Section 2 and related to minimum and maximum cuts on graphs. In Section 3, we connect organized partitions with spectral properties of graphs, and we prove bounds on the minimum cut in terms of these properties. In Section 4, we derive both algorithms, the first improving SB, and the second producing a smaller cut based on a given one. In Section 5, we present a few experimental results comparing the quality of partitions returned by SB and the new algorithm. Finally, in Section 6 there is a list of potential lines of investigation that arise from this paper.

2. Organized partitions. Let $G = (V, E)$ be a connected graph with $n = 4N$ vertices. Consider a cut $\{A, B\}$ of the vertex set V such that $|A| = |B|$. Such cut is also known as a balanced cut or a bisection. In this paper, we deal only with balanced cuts, thus from now on, we will simply refer to it as a cut. Let $A = A_1 \cup A_2$ and $B = B_1 \cup B_2$. Now create a new partition of vertices $\mathcal{C} = \{A_1, A_2, B_1, B_2\}$. We use $E(A, B)$ to denote the number of edges between the set of vertices A and B. We say that the partition $\mathcal{C}$ is *organized* whenever

$$E(A_1, A_2) + E(B_1, B_2) - E(A_1, B_1) - E(A_2, B_2) \tag{2.1}$$

is minimum among all subsets with $|A_1| = |A_2| = |B_1| = |B_2|$. See Figure 1, which depicts the partition in question.

It is worth mentioning that saying $\mathcal{C}$ is organized is equivalent to say that

$$E(A_1, B_2) + E(A_2, B_1) + E(A_1, A_2) + E(B_1, B_2) \tag{2.2}$$

is minimum among the prescribed sets. To see that, we notice that

$$\begin{aligned} &E(A_1, B_2) + E(A_2, B_1) + E(A_1, A_2) + E(B_1, B_2) \\ &= E(A, B) + E(A_1, A_2) + E(B_1, B_2) - E(A_1, B_1) - E(A_2, B_2). \end{aligned}$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

Since A and B are fixed, we know that $E(A,B)$ is fixed too. Thus, the same subsets that minimize (2.1) also minimize (2.2). We will show later how organized partitions related to minimum and maximum cuts of graphs.

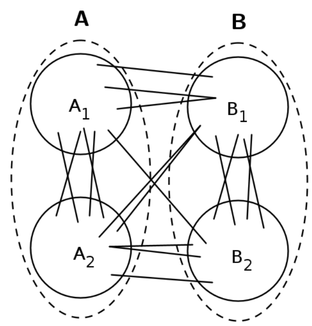


Figure 1: Organized partition.

In this paper, we tacitly assume that any partition $\mathcal{C}$ has $|A|=|B|$ and $|A_1|=|A_2|=|B_1|=|B_2|$. Now, given a cut $\{A,B\}$ we can compute the quantity

$$D_{\mathcal{C}} = \min_{\substack{A=\bar{A}_1\cup\bar{A}_2\\ B=\bar{B}_1\cup\bar{B}_2\\ |\bar{A}_1|=|\bar{A}_2|\\ |\bar{B}_1|=|\bar{B}_2|}} E(\bar{A}_1,\bar{A}_2)+E(\bar{B}_1,\bar{B}_2)-E(\bar{A}_1,\bar{B}_1)-E(\bar{A}_2,\bar{B}_2).$$

In this notation, the solution of the optimization problem $\mathcal{C}=\{A_1,A_2,B_1,B_2\}$ is an organized partition for $\{A,B\}$.

We say that $\mathcal{C}$ is a minimum cut whenever $E(A,B)$ is minimum among all choices of A and B with $|A|=|B|$. In Section 3, we will see that the quantity $D_{\mathcal{C}}$ relates with the eigenvalues of the Laplacian matrix whenever $\mathcal{C}$ is a minimum cut.

The next theorem provides a necessary condition for a cut to be minimum or maximum from the perspective of its organized partition.

THEOREM 2.1. *Let $\{A,B\}$ be any cut with organized partition $\mathcal{C}$. If $D_{\mathcal{C}}<0$, then $\{A,B\}$ is not a minimum cut. If $D_{\mathcal{C}}>0$, then $\{A,B\}$ is not a maximum cut.*

Proof. Let $\mathcal{C}=\{A_1,A_2,B_1,B_2\}$. Notice that

$$E(A,B)=E(A_1,B_1)+E(A_1,B_2)+E(A_2,B_1)+E(A_2,B_2). \tag{2.3}$$

Besides,

$$E(A_1\cup B_1,A_2\cup B_2)=E(A_1,A_2)+E(B_1,B_2)+E(A_2,B_1)+E(A_2,B_2). \tag{2.4}$$

Now, if $D_{\mathcal{C}}<0$, then from the definition, we have

$$E(A_1,A_2)+E(B_1,B_2)<E(A_1,B_1)+E(A_2,B_2).$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

This together with (2.3) and (2.4), gives us

$$\begin{aligned} E(A_1 \cup B_1, A_2 \cup B_2) &< E(A_1, B_1) + E(A_2, B_2) + E(A_2, B_1) + E(A_2, B_2) \\ &= E(A, B). \end{aligned}$$

Therefore, $\{A, B\}$ is not a minimum cut.

If $D_{\mathcal{C}} > 0$, then

$$E(A_1, A_2) + E(B_1, B_2) > E(A_1, B_1) + E(A_2, B_2).$$

Similarly as before, that gives us

$$E(A_1 \cup B_1, A_2 \cup B_2) > E(A, B).$$

Thus, $\{A, B\}$ is not a maximum cut. ∎

In fact, the proof reveals a way to construct a better cut, and this is one of the fundamental ideas behind the algorithm we provide in Section 4. We make this construction explicit in the form of the following corollary.

Corollary 2.2. *If a cut $\{A, B\}$ has $D_{\mathcal{C}} < 0$, then $E(A_1 \cup B_1, A_2 \cup B_2) < E(A, B)$. If $D_{\mathcal{C}} > 0$, then $E(A_1 \cup B_1, A_2 \cup B_2) > E(A, B)$.*

The next result gives some insights on how the organized partition of a minimum/maximum cut looks. We say that a graph has a *trivial partition* if it can be decomposed into four separated components of the same size.

Theorem 2.3. *A graph has no trivial partition if and only if for each minimum cut $\{A, B\}$ the organized partition satisfy $E(A_1, A_2) + E(B_1, B_2) \neq 0$.*

If $\{A, B\}$ is a maximum cut of a graph with no trivial partition, then any of its organized partitions satisfy $E(A_1, B_1) + E(A_2, B_2) \neq 0$.

Proof. First, if the graph has a trivial partition, then there exists an organized partition of a minimum cut such that

$$E(A_1, B_1) + E(A_2, B_2) = E(A_1, A_2) + E(B_1, B_2) = 0.$$

Thus, assume that the graph has no trivial partition. Let $\{A, B\}$ be a minimum cut and assume by contradiction that $E(A_1, A_2) + E(B_1, B_2) = 0$. We can assume that $E(A_1, B_1) + E(A_2, B_2) \neq 0$, otherwise the graph would have a trivial partition. Thus,

$$D_{\mathcal{C}} = E(A_1, A_2) + E(B_1, B_2) - E(A_1, B_1) - E(A_2, B_2) < 0.$$

Therefore, Theorem 2.1 implies that $\{A, B\}$ is not a minimum cut, which is a contradiction. That finishes the proof of the first claim.

Let $\{A, B\}$ be a maximum cut, suppose to the contrary that $E(A_1, B_1) + E(A_2, B_2) = 0$. If $E(A_1, A_2) + E(B_1, B_2) = 0$, then the graph has a trivial partition. Thus, $E(A_1, A_2) + E(B_1, B_2) \neq 0$, and that gives us

$$D_{\mathcal{C}} = E(A_1, A_2) + E(B_1, B_2) - E(A_1, B_1) - E(A_2, B_2) > 0.$$

Finally, Theorem 2.1 implies that $\{A, B\}$ is not a maximum cut, which is a contradiction. That estabilishes the first claim. ∎

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

Organized partitions also indicate conditions for which a graph has more than one minimum or maximum cut and, if that is the case, how to construct them.

THEOREM 2.4. *Let $\{A, B\}$ be any cut with organized partition $\mathcal{C} = \{A_1, A_2, B_1, B_2\}$. If $D_{\mathcal{C}} = 0$, then $E(A, B) = E(A_1 \cup B_1, A_2 \cup B_2)$.*

Proof. From the definition of $D_{\mathcal{C}}$, we have

$$E(A_1, A_2) + E(B_1, B_2) = E(A_1, B_1) + E(A_2, B_2).$$

Thus, we can write

$$\begin{aligned} E(A_1 \cup B_1, A_2 \cup B_2) &= E(A_1, B_1) + E(A_2, B_2) + E(A_2, B_1) + E(A_2, B_2) \\ &= E(A, B). \end{aligned}$$

That finishes the proof. ∎

COROLLARY 2.5. *Let $\{A, B\}$ be a minimum or a maximum cut. If $D_{\mathcal{C}} = 0$, then it is not unique.*

Thus, in some cases finding an organized partition can be useful to construct a different minimum bisection whenever it is not unique. On the other hand, for a graph with a unique minimum bisection, the organized partition can be used to bound the size of the second smallest bisection. As the next theorem shows that any bisection is not too far from the minimum whenever $D_{\mathcal{C}}$ is small.

THEOREM 2.6. *Let $\{A, B\}$ be a minimum bisection and $\mathcal{C} = \{A_1, A_2, B_1, B_2\}$ its organized partition. Let $\{R, S\}$ be any other bisection. Then*

$$E(R, S) - E(A, B) \leq D_{\mathcal{C}}.$$

Proof. Notice that $E(A_1, A_2) + E(B_1, B_2) \leq \frac{n^2}{8}$. Thus, the total number of missing edges between A_1 and B_1 together with the missing edges between A_2 and B_2 is

$$\frac{n^2}{8} - E(A_1, B_1) - E(A_2, B_2) \geq E(A_1, A_2) + E(B_1, B_2) - E(A_1, B_1) - E(A_2, B_2) = D_{\mathcal{C}}.$$

By Theorem 2.1, we have $D_{\mathcal{C}} \geq 0$. That means we can add at least $D_{\mathcal{C}}$ edges between the pairs A_1, B_1 and A_2, B_2.

That fact allows us to create a new graph G^* by adding $D_{\mathcal{C}}$ edges between the pairs A_1, B_1 and A_2, B_2. For this new graph it still holds that $\mathcal{C} = \{A_1, A_2, B_1, B_2\}$ is an organized partition. Similarly, denoting by $D^*_{\mathcal{C}}$ and $E^*(A, B)$ the corresponding quantities in the graph G^*, it holds that $D^*_{\mathcal{C}} = 0$ and

$$E^*(A, B) = E(A, B) + D_{\mathcal{C}}. \tag{2.5}$$

Now, assume by contradiction that there exists a cut $\{R, S\}$ satisfying $E(R, S) > D_{\mathcal{C}} + E(A, B)$. Notice that the graph G^* contains all corresponding edges from the graph G, therefore for any pair of sets of vertices $\{R, S\}$ we have $E^*(R, S) \geq E(R, S)$. That together with equation (2.5), implies

$$E^*(R, S) \geq E(R, S) > E^*(A, B).$$

The last strict inequality tells us that $\{A, B\}$ is a unique minimum cut for G^*. By Theorem 2.1 and Corollary 2.5, this minimum cut satisfies $D^*_{\mathcal{C}} > 0$. This contradicts $D^*_{\mathcal{C}} = 0$. Therefore, $E(R, S) \leq D_{\mathcal{C}} + E(A, B)$, which finishes the proof. ∎

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

3. Integer program formulation. This section is dedicated to relating organized partitions with spectral properties of the graph. We prove bounds on the minimum cut in terms of these properties. In the next theorem, we show how to construct the organized partition of given cut. It turns out it suffices to solve an integer program in terms of the Laplacian matrix of the graph.

THEOREM 3.1. *Let $G=(V,E)$ be a connected graph with $n=4N$ vertices. Let $\{A,B\}$ be any bisection of a graph G and denote by y be the vector with entries*

$$y_i = \begin{cases} 1/\sqrt{n} & \text{if } i \in A \\ -1/\sqrt{n} & \text{if } i \in B. \end{cases}$$

Let L be the Laplacian matrix of the G. Then

$$\frac{4}{n}\left(E(A,B)+D_{\mathcal{C}}\right) = \min_{\substack{x^T\mathbf{1}=0 \\ \|x\|=1 \\ y^Tx=0 \\ x_i\in\left\{1/\sqrt{n},-1/\sqrt{n}\right\}}} x^TLx. \tag{3.6}$$

Furthermore, each solution $\bar{x}$ of (3.6) prescribes an organized partition for $\{A,B\}$ as follows

$$\bar{x}_i = \begin{cases} 1/\sqrt{n} & i \in A_1 \cup B_1 \\ -1/\sqrt{n} & i \in A_2 \cup B_2. \end{cases}$$

Proof. Let A_1, A_2, B_1 and B_2 be disjoint sets such that $A_1\cup A_2=A$ and $B_1\cup B_2=B$, with $|A_1|=|A_2|$ and $|B_1|=|B_2|$. Define the vector x with entries

$$x_i = \begin{cases} 1/\sqrt{n} & i \in A_1 \cup B_1 \\ -1/\sqrt{n} & i \in A_2 \cup B_2 \end{cases}.$$

Clearly, $x^T\mathbf{1}=0$, $\|x\|=1$, and $y^Tx=0$.

Now, we can write x^TLx in terms of the partition $\{A_1,A_2,B_1,B_2\}$ as

$$\begin{aligned} x^TLx &= \sum_{(i,j)\in E} (x_i-x_j)^2 \\ &= \sum_{\substack{(i,j)\in E \\ i\in A_1, j\in B_1}} (x_i-x_j)^2 + \sum_{\substack{(i,j)\in E \\ i\in A_2, j\in B_2}} (x_i-x_j)^2 + \sum_{\substack{(i,j)\in E \\ i\in A_1, j\in B_2}} (x_i-x_j)^2 \\ &\quad + \sum_{\substack{(i,j)\in E \\ i\in A_2, j\in B_2}} (x_i-x_j)^2 + \sum_{\substack{(i,j)\in E \\ i\in A_1, j\in A_2}} (x_i-x_j)^2 + \sum_{\substack{(i,j)\in E \\ i\in B_1, j\in B_2}} (x_i-x_j)^2 \\ &= \frac{4}{n}\left(E(A_1,B_2)+E(A_2,B_1)+E(A_1,A_2)+E(B_1,B_2)\right), \end{aligned}$$

since the first two sums are zero. That gives us

$$\frac{n}{4}x^TLx = E(A,B)+E(A_1,A_2)+E(B_1,B_2)-E(A_1,B_1)-E(A_2,B_2),$$

for each choice of partition $\{A_1,A_2,B_1,B_2\}$. Therefore, in view of the definition of $D_{\mathcal{C}}$, we have

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

$$\min_{\substack{x^T\mathbf{1}=0\\ \|x\|=1\\ y^Tx=0\\ x_i\in\left\{1/\sqrt{n},-1/\sqrt{n}\right\}}} x^TLx = \frac{4}{n}\left(E(A,B)+D_{\mathcal{C}}\right).$$

By the construction of the feasible set of solutions, $\bar{x}$ indicates the organized partition of $\{A,B\}$. That finishes the proof. ∎

Thus, whenever $\{A,B\}$ is a minimum cut, the minimum of (3.6) reduces to $\frac{4}{n}\left(MinCut(G)+D_{\mathcal{C}}\right)$.

In the work of [9] the authors proved the inequality

$$MinCut(G) \geq \frac{n}{4}\lambda_2. \tag{3.7}$$

In light of the concept of organized partitions we can relate minimum cuts and eigenvalues of the Laplacian matrix and prove the next result. For a partition $\{A,B\}$ we call the vector with entries

$$z_i = \begin{cases} 1/\sqrt{n} & \text{if } i\in A\\ -1/\sqrt{n} & \text{if } i\in B\end{cases}$$

the *descriptor* of the partition.

Theorem 3.2. *Let $G=(V,E)$ be a connected graph with $n=4N$ vertices. Let $\mathcal{C}$ be an organized partition of a minimum bisection with descriptor z. Then, we have*

$$MinCut(G) \geq \frac{n}{8}\left(\lambda_2 + \min_{\substack{x^T\mathbf{1}=0\\ \|x\|=1\\ z^Tx=0}} x^TLx\right) - \frac{D_{\mathcal{C}}}{2}.$$

Proof. Define the vector y with entries

$$y_i = \begin{cases} 1/\sqrt{n} & i\in A\\ -1/\sqrt{n} & i\in B.\end{cases} \tag{3.8}$$

Clearly, $y^T\mathbf{1}=0$ and $\|y\|=1$. Thus, we can write

$$y^TLy = \sum_{(i,j)\in E}(y_i-y_j)^2 = \sum_{\substack{(i,j)\in E\\ i\in A, j\in B}}(y_i-y_j)^2 + \sum_{\substack{(i,j)\in E\\ i\in A, j\in A}}(y_i-y_j)^2 + \sum_{\substack{(i,j)\in E\\ i\in B, j\in B}}(y_i-y_j)^2.$$

Notice the sum over the edges with both endpoints in the same set vanishes. Thus, we have

$$y^TLy = \sum_{\substack{(i,j)\in E\\ i\in A, j\in B}}\left(1/\sqrt{n}-(-1/\sqrt{n})\right)^2 = \frac{4}{n}E(A,B).$$

An important idea here is that a minimum cut is achieved if we take the minimum over all prescribed vectors, i.e.,

$$MinCut(G) = \frac{n}{4}\min_{\substack{y^T\mathbf{1}=0, \|y\|=1\\ y_i\in\left\{1/\sqrt{n},-1/\sqrt{n}\right\}}} y^TLy. \tag{3.9}$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

Now, we apply Theorem 3.1 for the vector descriptor z that solves the minimization problem (3.9). Thus, we can write the sum of the following minimization problems as

$$\min_{\substack{y^T\mathbf{1}=0,\|y\|=1\\ y_i\in\left\{1/\sqrt{n},-1/\sqrt{n}\right\}}} y^TLy+\min_{\substack{x^T\mathbf{1}=0\\ \|x\|=1\\ z^Tx=0\\ x_i\in\left\{1/\sqrt{n},-1/\sqrt{n}\right\}}} x^TLx=\frac{4}{n}\left(2MinCut(G)+D_{\mathcal{C}}\right).$$

Equivalently, we can write

$$MinCut(G)=\frac{n}{8}\min_{\substack{y^T\mathbf{1}=0,\|y\|=1\\ y_i\in\left\{1/\sqrt{n},-1/\sqrt{n}\right\}}} y^TLy+\min_{\substack{x^T\mathbf{1}=0\\ \|x\|=1\\ z^Tx=0\\ x_i\in\left\{1/\sqrt{n},-1/\sqrt{n}\right\}}} x^TLx-\frac{D_{\mathcal{C}}}{2}.$$

Thus, if we drop the constraint $y_i,x_i\in\{1/\sqrt{n},-1/\sqrt{n}\}$ and consider all unitary vectors $x,y\in\mathbb{R}^n$, we find the inequality

$$\begin{aligned} MinCut(G) &\geq \frac{n}{8}\min_{\substack{y^T\mathbf{1}=0\\ \|y\|=1}} y^TLy+\min_{\substack{x^T\mathbf{1}=0\\ \|x\|=1\\ z^Tx=0}} x^TLx-\frac{D_{\mathcal{C}}}{2}\\ &=\frac{n}{8}\left(\lambda_2+\min_{\substack{x^T\mathbf{1}=0\\ \|x\|=1\\ z^Tx=0}} x^TLx\right)-\frac{D_{\mathcal{C}}}{2}. \end{aligned}$$

That finishes the proof. □

It is worth mentioning that if we drop the constraint $y_i\in\{1/\sqrt{n},-1/\sqrt{n}\}$ in the minimization problem (3.9), then we precisely obtain the lower bound (3.7) as the authors in [9].

Corollary 3.3. *If* $\min_{\substack{x^T\mathbf{1}=0\\ \|x\|=1\\ z^Tx=0}} x^TLx\geq\lambda_3$, *then*

$$MinCut(G)\geq\frac{n}{8}\left(\lambda_2+\lambda_3\right)-\frac{D_{\mathcal{C}}}{2}.$$

Notice that an eigenvector y for λ_2 is an approximation for the descriptor vector z that corresponds to a minimum bisection, since it is the solution of the relaxed problem. Therefore the constraint $z^Tx=0$ in

$$\min_{\substack{x^T\mathbf{1}=0\\ \|x\|=1\\ z^Tx=0}} x^TLx$$

is an approximation to the constraint $y^Tx=0$, i.e,

$$\lambda_3=\min_{\substack{x^T\mathbf{1}=0\\ \|x\|=1\\ y^Tx=0}} x^TLx\approxeq\min_{\substack{x^T\mathbf{1}=0\\ \|x\|=1\\ z^Tx=0}} x^TLx.$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

Besides, whenever

$$D_{\mathcal{C}} < \frac{n}{4}\left(\lambda_3 - \lambda_2\right) \quad \text{and} \quad \min_{\substack{x^T\mathbf{1}=0\\ \|x\|=1\\ z^Tx=0}} x^TLx \geq \lambda_3,$$

Theorem 3.2 provides a tighter lower bound than (3.7). Intuitively, it means that an optimization problem that considers both λ_2 and λ_3 is more likely to reveal a minimum cut than a problem that considers only λ_2.

Next, we relate the Laplacian eigenvalues with a the number of edges between the partitions of interest, which further relates to organized partitions as well.

THEOREM 3.4. *Let $G=(V,E)$ be a connected graph with $n=4N$ vertices. Consider the minimum over all partitions $\{A_1,A_2,B_1,B_2\}$ as follow*

$$D = \min_{\substack{A,B\subset V,|A|=|B|\\ A=A_1\cup A_2,\\ B=B_1\cup B_2\\ |A_1|=|A_2|\\ |B_1|=|B_2|}} E(A,B)+E(A_1,B_2)+E(A_2,B_1)+E(A_1,A_2)+E(B_1,B_2).$$

Then

$$\frac{n}{4}\left(\lambda_3+\lambda_2\right) \leq D.$$

Proof. Define the vector y with entries

$$y_i = \begin{cases} 1/\sqrt{n} & i\in A\\ -1/\sqrt{n} & i\in B. \end{cases} \tag{3.10}$$

Let A_1, A_2, B_1 and B_2 be disjoint sets such that $A_1\cup A_2 = A$ and $B_1\cup B_2 = B$, with $|A_1|=|A_2|$ and $|B_1|=|B_2|$. Define the vector x with entries

$$x_i = \begin{cases} 1/\sqrt{n} & i\in A_1\cup B_1\\ -1/\sqrt{n} & i\in A_2\cup B_2. \end{cases}$$

As in the proofs of Theorems 3.1 and 3.2, we obtain

$$\frac{n}{4}y^TLy = E(A,B) \quad \text{and}$$

$$\frac{n}{4}x^TLx = E(A,B)+E(A_1,A_2)+E(B_1,B_2)-E(A_1,B_1)-E(A_2,B_2).$$

From the identity

$$\begin{aligned} &E(A,B)+E(A_1,A_2)+E(B_1,B_2)-E(A_1,B_1)-E(A_2,B_2)\\ &\qquad = E(A_1,B_2)+E(A_2,B_1)+E(A_1,A_2)+E(B_1,B_2), \end{aligned}$$

we obtain

$$\frac{n}{4}\left(y^TLy+x^TLx\right) = E(A,B)+E(A_1,B_2)+E(A_2,B_1)+E(A_1,A_2)+E(B_1,B_2).$$

This implies

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

$$\frac{n}{4} \min_{\substack{y^T\mathbf{1}=0, x^T\mathbf{1}=0, y^Tx=0\\ \|x\|=\|y\|=1 \\ y_i\in\left\{1/\sqrt{n},-1/\sqrt{n}\right\} \\ x_i\in\left\{1/\sqrt{n},-1/\sqrt{n}\right\}}} x^TLx + y^TLy = D.$$

Therefore, when we remove the constraints $y_i, x_i \in \{1/\sqrt{n}, -1/\sqrt{n}\}$ and consider all unitary vectors $x, y \in \mathbb{R}^n$, we find the inequality

$$\frac{n}{4} \min_{\substack{y^T\mathbf{1}=0, x^T\mathbf{1}=0, y^Tx=0\\ \|x\|=\|y\|=1}} x^TLx + y^TLy \leq D,$$

which implies the result. ∎

Corollary 3.5. *If $D \leq 2MinCut(G) + D_{\mathcal{C}}$, then*

$$MinCut(G) \geq \frac{n}{8}\left(\lambda_2 + \lambda_3\right) - \frac{D_{\mathcal{C}}}{2}.$$

Proof. From Theorem 3.4 we obtain

$$\frac{n}{4}\left(\lambda_3 + \lambda_2\right) \leq D \leq 2MinCut(G) + D_{\mathcal{C}},$$

which implies the stated bound. ∎

We notice that as an integer program, the problem of finding an organized partition is NP-hard. That gives rise to the heuristic developed in the next section. We finish this section with a result that summarizes all its underlying ideas.

Theorem 3.6. *Let $G = (V, E)$ be a connected graph with $n = 4N$ vertices and with Laplacian matrix L. Then, the solution $(\bar{x}, \bar{y})$ of the problem*

$$\min_{\substack{y^T\mathbf{1}=0, \|y\|=1 \\ y_i\in\left\{1/\sqrt{n},-1/\sqrt{n}\right\}}} y^TLy + \min_{\substack{x^T\mathbf{1}=0 \\ \|x\|=1 \\ \bar{y}^Tx=0 \\ x_i\in\left\{1/\sqrt{n},-1/\sqrt{n}\right\}}} x^TLx$$

constructs a minimum cut $\{A, B\}$ together with its organized partition
$\mathcal{C} = \{A_1, A_2, B_1, B_2\}$, as follows:

$$\bar{y}_i = \begin{cases} 1/\sqrt{n} & i \in A \\ -1/\sqrt{n} & i \in B \end{cases} \quad \text{and} \quad \bar{x}_i = \begin{cases} 1/\sqrt{n} & \text{if } i \in A_1 \cup B_1 \\ -1/\sqrt{n} & \text{if } i \in A_2 \cup B_2. \end{cases}$$

Proof. Follows from equation (3.9) and Theorem 3.1. ∎

4. Derivation of the algorithms. In this section, we provide an intuitive description of the main ideas behind our new algorithms, which arise from the theoretical background developed in the previous sections. We do that by showing how to improve the bisection provided by the traditional SB algorithm by means of properties of organized partitions. We will prove that there are infinitely many solutions for the minimization problem that finds the organized partition of a cut, if we apply relaxation. Thus, these solutions construct better candidates for a minimum cut. First, we consider some examples where SB fails to approximate a good bisection.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

As an approximation algorithm, SB sometimes provides a cut that is too far from optimal. There are investigations about this phenomenon, and the best known example where SB fails is given by the roach graph, due to Guattery and Miller [14]. The roach graph consists of two path graphs with the same even size connected by a few edges, as illustrated in Figure 2.

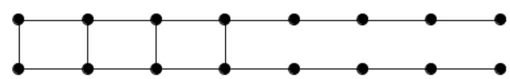


Figure 2: Roach graph on 16 vertices.

This is a very good example which seems to be tailor made to defeat SB. The roach graph is an important example not only because SB provides a cut that is far from optimal, in fact it is the prototype of many cases where this algorithm gives a very bad result. Let us look closer to what is happening with the algorithm on this kind of graph.

For a roach graph the minimum bisection consists of two edges separating the antennae - the pendant paths on the right side of Figure 2. But that is not what SB returns. Taking a roach graph on 16 vertices, we label the upper and lower path from 1 to 8 and 9 to 16, respectively. For this ordering, its eigenvector associated with λ_2 is approximately given by

$$\begin{aligned} y = [&-0.0028 - 0.0083 - 0.0295 - 0.1068 - 0.3869796 - 0.6270 - 0.8024 - 0.8948 \\ &0.0028\ 0.0083\ 0.0295\ 0.1068\ 0.3869\ 0.6270\ 0.8024\ 0.8948]^T. \end{aligned}$$

Now, we can plot the entries of y displayed in Figure 3. The upper path corresponds to the points above the origin and the lower path bellow it. SB will split the graph in two paths, which provides a cut with 4 edges, which is not a minimum cut. In [14] the authors showed this is true for the whole class of roach graphs, therefore showing a class of graphs where the resulting bisection from SB is far from optimal, i.e., with a bisection of order $\mathcal{O}(n)$.

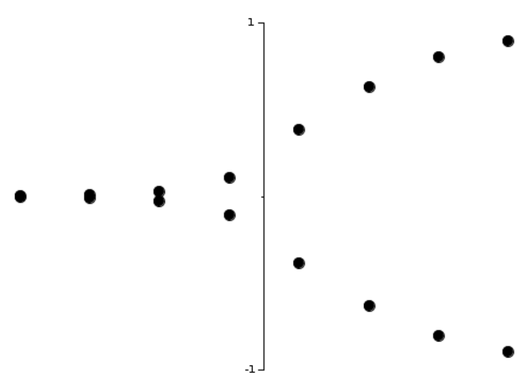


Figure 3: The y eigenvector for the roach graph.

This is the prototype of what happens with SB when it returns a wrong bisection. In view of this problem, it is natural to ask how to overcome this pathology for the SB algorithm. Here we show how that

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

can be done using the concept of organized partitions. In light of Corollary 2.2 if a cut has $D_{\mathcal{C}} < 0$, then its organized partition can be used to construct a smaller cut. Thus, it would be useful to have an algorithm that approximates a minimum cut and which computes its organized partition as well. That means if we could solve both problems simultaneously, then we can obtain a better cut than the original algorithm, whenever this cut has $D_{\mathcal{C}} < 0$.

That is the case of the roach graph and many examples of this nature. Notice that the cut provided by SB for the roach graph on 16 vertices is $\mathcal{C} = \{A_1, A_2, B_1, B_2\}$, where
$A_1 = \{v_1, \dots, v_4\}$, $A_2 = \{v_5, \dots, v_8\}$, $B_1 = \{v_9, \dots, v_{12}\}$, and $B_2 = \{v_{13}, \dots, v_{16}\}$. Therefore, we have

$$D_{\mathcal{C}} = E(A_1, A_2) + E(B_1, B_2) - E(A_1, B_1) - E(A_2, B_2) = 1 + 1 - 4 - 0,$$

which gives us the desired property $D_{\mathcal{C}} < 0$. For this reason, the organized partition of this cut will provide a smaller bisection.

Now, let us see what the eigenvector of λ_3 tells about the organized partition. Theorem 3.6 constructs the organized partition based on the solution of an integer program. Theorem 3.2 and its proof suggest that the eigenvectors of λ_2 and λ_3 can be used to approximate the solution. Thus, if we drop the constraints on x and y putting $x, y \in \mathbb{R}^n$, it is expected that the solution of the new program

$$\min_{\substack{y^T\mathbf{1}=x^T\mathbf{1}=0\\ \|y\|=\|x\|=1\\ y^Tx=0}} y^TLy + x^TLx \tag{4.11}$$

approximates the minimum cut and its organized partition by the eigenvector x associated with λ_3.

For the same roach graph, that eigenvector is approximately

$$\begin{aligned} x = [&-0.6935\ \ -0.5879\ \ -0.3928\ \ -0.1379\ 0.1379\ 0.3928\ 0.5879\ 0.6935\\ &-0.6935\ \ -0.5879 - 0.3928\ \ -0.1379\ \ 0.1379497\ \ 0.3928475\ \ 0.5879\ \ 0.6935]^T. \end{aligned}$$

Notice that if we use x as an approximation for the integer solution of the program in Theorem 3.6, then x induces the correct organized partition
$\mathcal{C} = \{A_1, A_2, B_1, B_2\}$ as described above. Here we simply used the entries of x as an approximation for the integer solution

$$\bar{x}_i = \begin{cases} 1/\sqrt{n} & \text{if } i \in A_1 \cup B_1 \\ -1/\sqrt{n} & \text{if } i \in A_2 \cup B_2. \end{cases}$$

Since $D_{\mathcal{C}} < 0$, this implies that we can construct a smaller bisection than the one provided by SB by using the eigenvector x. More precise, by Corollary 2.2 the partition $\{A_1 \cup B_1, A_2 \cup B_2\}$ gives a smaller bisection. In fact, this is the minimum bisection for the roach graph.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

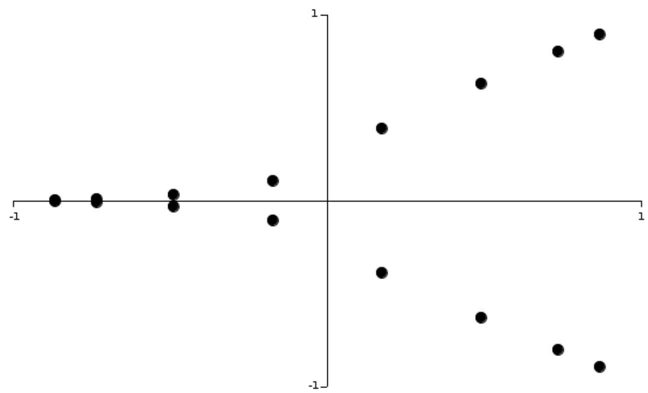


Figure 4: Roach graph with x and y as embedding.

Figure 4 depicts the underlying idea behind the proof of Theorem 3.2. We plotted points using the entries of both eigenvectors of the roach graph x and y as coordinates. There, each point corresponds to a vertex. It is clear to see that if we separate the vertices by the signs of the coordinates in x, then we would get the minimum cut.

The previous discussion suggests to consider both eigenvectors in a new algorithm, in the sense either x or y will approximate a minimum cut. Essentially, when y gives a cut with $D_{\mathcal{C}} < 0$, we can appeal to the cut provided by x. Thus, it would suffice to check which one gives a better cut. Actually, this neat idea can be taken further when we look from the perspective of integer programming.

As we will see in the next Lemma, certain specific linear combinations of x and y are solutions for (4.11) as well. Thus, those new solutions can be used to approximate a minimum bisection. The next Lemma can be proved in different ways using results from matrix theory.

LEMMA 4.1. *Let x and y be a solution of (4.11). Let $\theta \in [0, 2\pi)$ and let $u = cos\theta x + sin\theta y$ and $v = sin\theta x - cos\theta y$. Then u and v is a solution of (4.11).*

Proof. We proceed by showing that $x^T Lx + y^T Ly = u^T Lu + v^T Lv$. Hence, we write

$$\begin{aligned} u^T Lu &= (\cos\theta x + \sin\theta y)^T L(\cos\theta x + \sin\theta y) \\ &= \cos\theta x^T L\cos\theta x + \sin\theta y^T L\sin\theta y + 2\sin\theta y^T L\cos\theta x. \end{aligned}$$

Also, we can write

$$\begin{aligned} v^T Lv &= (\sin\theta x - \cos\theta y)^T L(\sin\theta x - \cos\theta y) \\ &= \sin\theta x^T L\sin\theta x + \cos\theta y^T L\cos\theta y - 2\sin\theta y^T L\cos\theta x. \end{aligned}$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

Therefore, we obtain

$$\begin{aligned}u^T Lu + v^T Lv &= \cos\theta x^T L\cos\theta x + \sin\theta y^T L\sin\theta y + 2\sin\theta y^T L\cos\theta x\\&\quad + \sin\theta x^T L\sin\theta x + \cos\theta y^T L\cos\theta y - 2\sin\theta y^T L\cos\theta x\\&= \cos^2\theta x^T Lx + \sin^2\theta y^T Ly + \sin^2\theta x^T Lx + \cos^2\theta y^T Ly\\&= (\cos^2\theta + \sin^2\theta)(x^T Lx + y^T Ly)\\&= x^T Lx + y^T Ly.\end{aligned}$$

It follows that u and v is also a minimizer of (4.11).

It remains to verify that u and v satisfy the constraints $u^T\mathbf{1} = 0$, $v^T\mathbf{1} = 0$, $\|u\| = \|v\| = 1$ and $u^T v = 0$.

To see that $u^T\mathbf{1} = 0$, we notice that $u^T\mathbf{1} = \cos\theta x^T\mathbf{1} + \sin\theta y^T\mathbf{1} = 0$. Now, using the fact that $x^T x = y^T y = 1$ and $y^T x = 0$, we can write

$$\begin{aligned}u^T u &= (\cos\theta x + \sin\theta y)^T(\cos\theta x + \sin\theta y)\\&= \cos^2\theta x^T x + \sin^2\theta y^T y + 2\sin\theta\cos\theta y^T x\\&= \cos^2\theta + \sin^2\theta = 1,\end{aligned}$$

which implies $\|u\| = 1$. Similarly, we obtain $v^T\mathbf{1} = 0$ and $\|v\| = 1$.

To show $u^T v = 0$, again we use the fact that $x^T x = y^T y = 1$ and $y^T x = 0$

$$\begin{aligned}u^T v &= (\cos\theta x + \sin\theta y)^T(\sin\theta x - \cos\theta y)\\&= \cos\theta\sin\theta x^T x - \cos^2\theta x^T y + \sin^2\theta x^T y - \sin\theta\cos\theta y^T y\\&= \cos\theta\sin\theta x^T x - \sin\theta\cos\theta y^T y = 0.\end{aligned}$$

This concludes the proof. ∎

The solutions described by Lemma 4.1 are related to the work with random hyperplane rounding of the eigenvectors [17]. This relates back to the vector partitioning problem as well.

By constructing a infinite set of solutions for the problem (4.11), the last Theorem introduces a degree of freedom in the solutions of (4.11). We can explore this degree of freedom in order to create different bisections. As discussed before, solutions of (4.11) can be used to approximate a minimum bisection and its organized partition. However, there are infinitely many u and v described in the last theorem. Naturally, all of them can be used to approximate a minimum bisection. That is a key idea in the next algorithm. The next theorem shows how to construct n different bisections based on the solutions of (4.11).

Theorem 4.2. *Let x and y be solutions of (4.11). For each pair x_i and y_i, $i = 1, \dots, n = 4N$, define the vector $u = \frac{x_i}{\sqrt{x_i^2+y_i^2}}x + \frac{y_i}{\sqrt{x_i^2+y_i^2}}y$. Then u induces a bisection that approximates the vector $\bar{u_i}$ with entries*

$$\bar{u_i} = \begin{cases}1/\sqrt{n} & i \in A\\ -1/\sqrt{n} & i \in B.\end{cases}$$

Proof. In order to construct different bisections using Lemma 4.1, we need to choose $\theta \in [0, 2\pi)$, then define u and v, and finally define a new partition $\{A, B\}$ based on u and v. To this end, consider the set of euclidean points (x_i, y_i) given by the corresponding entries of the eigenvectors x and y. Choose a point

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

(x_i, y_i), and let θ_i be the angle between the point (x_i, y_i) and the abscissa. Now define u and v as in Theorem 4.1, and let (u_i, v_i) be points defined by the corresponding entries of u and v. The point (u_i, v_i) is simply a rotation of angle θ_i for the point (x_i, y_i).

Now using the solution of (4.11), we can approximate the solution of the integer program in Theorem 3.6. By Theorem 3.6, its solution defines a minimum cut, and we can define the cut $\{A, B\}$ using the entries of u as an approximation for

$$\bar{u_i} = \begin{cases} 1/\sqrt{n} & i \in A \\ -1/\sqrt{n} & i \in B. \end{cases}$$

Finally, to simplify the computation of u we can calculate $\cos\theta$ and $\sin\theta$ instead of θ. That follows straightforward from

$$\cos\theta = \frac{x_i}{\sqrt{x_i^2 + y_i^2}} \quad \text{and} \quad \sin\theta = \frac{y_i}{\sqrt{x_i^2 + y_i^2}}.$$

That finishes the proof. ∎

4.0.1. Spectral bisection with three eigenvectors. Now we are ready to give the complete algorithm that approximates a minimum bisection of a graph.

Algorithm 1 Graph bisection.

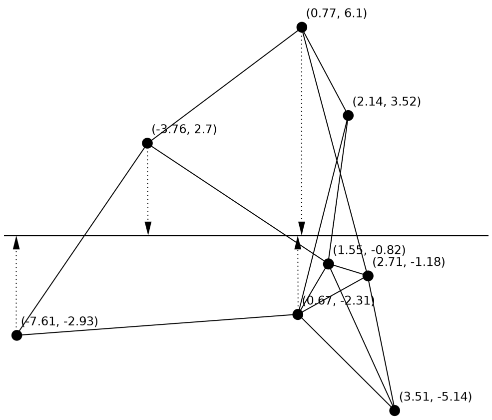
Require: G=(V,E)

Compute eigenvectors y and x corresponding to the second and third smallest eigenvalues of L.
Set A with the $n/2$ vertices with largest entries of the vector y and B with the remaining vertices.
for $i = 1, \dots, n$ **do**
  $u = \frac{x_i}{\sqrt{x_i^2+y_i^2}} x + \frac{y_i}{\sqrt{x_i^2+y_i^2}} y$
  Set R with the $n/2$ vertices with largest u_j and S with the remaining vertices.
  if $E(R, S) < E(A, B)$ **then**
    $A = R$
    $B = S$
  end if
end for
return $\{A, B\}$

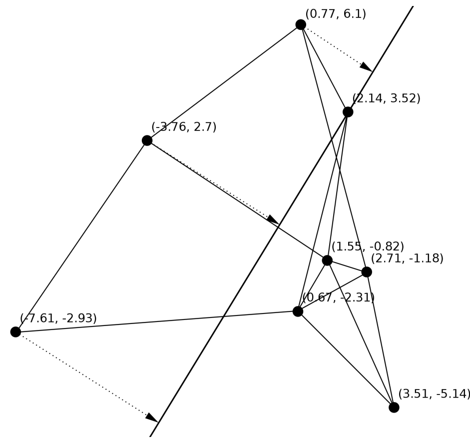
As an illustration of Algorithm 1, Figures 5a and 5 show the same graph embedded on the coordinates given by the second and the third eigenvalue. Figure 5a depicts the SB algorithm choosing a set of vertices based on a Fiedler vector only. The straight line has the same direction of the Fiedler vector. Since SB sorts the vertices based on the this vector and chooses the top largest to construct the bisection, it is clear that it is simply a projection of points along the straight line. As more linear combinations of the Fiedler vector and the third eigenvector are considered, different cuts are created. Figure 5 depicts the optimal choice of vertices induced by one of those linear combinations.

Notice that the cut induced by x, the standard spectral bisection solution, is among the possible cuts $\{R, S\}$ constructed by Algorithm 1. Therefore, the number of edges in the partition provided by Algorithm 1 is not larger than the one in the partition returned by SB, which leads us to the next theorem.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.



(a) SB chooses vertices from a Fiedler vector.



(b) Algorithm 1 considers both, the Fiedler vector and the third eigenvector to choose vertices.

Figure 5: Algorithm 1 considers both, the Fiedler vector and the third eigenvector to choose vertices.

Theorem 4.3. *The cut returned by Algorithm 1 is not larger than that in the SB partition.*

For any roach graph, its eigenvectors have the same shape of the previous example with 16 vertices. That leads us to the next theorem.

Theorem 4.4. *For any roach graph, Algorithm 1 returns a minimum cut.*

Proof. By Lemma 5.1 of [14], the third eigenvector of a roach graph induces a cut separating the pending paths of the roach graph, which is a minimum cut. This cut is among the possible cuts constructed by Algorithm 1. That finishes the proof. ∎

4.0.2. Spectral refinement. Now we will turn our attention to the derivation of an algorithm that refines a given bisection. Since an organized partition can be used to construct a better bisection, the next algorithm constructs an approximation for an organized partition of a given bisection. In the same fashion as in Algorithm 1, these approximations are candidates for a smaller cut.

Theorem 3.1, provide us with a way to construct the organized partition of a given cut. If $\{A, B\}$ is the cut in question, we can denote by y be the vector with entries

$$y_i = \begin{cases} 1/\sqrt{n} & \text{if } i \in A \\ -1/\sqrt{n} & \text{if } i \in B. \end{cases}$$

Now, if we use relaxation on the set of solutions of the integer program (3.6) and drop the constraint $x_i \in \{1/\sqrt{n}, -1/\sqrt{n}\}$, we obtain the following program

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

$$\min_{\substack{x^T\mathbf{1}=0\\ \|x\|=1\\ y^Tx=0}} x^T L x. \tag{4.12}$$

The minimization problem (4.12) is not an eigenvalue problem anymore, because the vector y is not necessarily an eigenvector of the matrix L. However, it is easy to transform problem (4.12) into a standard eigenvalue problem, as shown in [11] by Gene and Golub. Therefore, the solution of the program (4.12) can be used as an approximation for the organized partition: the half largest entries of x indicate the vertices in the set $A_1 \cup B_1$ of the organized partition, and the other half indicates the remaining vertices in the organized partition. Again, we can use linear combinations of x and y to construct different approximations for the organized partition. The algorithm can be described as follows.

Algorithm 2 Spectral bisection refinement.

Require: $G = (V, E), y$
  Set A with the $n/2$ vertices with largest entries in y and B with the remaining vertices.
  Compute x, the solution of $\min_{\substack{x^T\mathbf{1}=0\\ \|x\|=1\\ y^Tx=0}} x^T L x$
  for $i = 1, \ldots, n$ **do**
    $u = \frac{x_i}{\sqrt{x_i^2+y_i^2}}x + \frac{y_i}{\sqrt{x_i^2+y_i^2}}y$
    Set R with the $n/2$ vertices with largest u_j and S with the remaining vertices.
    if $E(R, S) < E(A, B)$ **then**
      $A = R$
      $B = S$
    end if
  end for
  return $\{A, B\}$

5. Experimental results. We would like to emphasize that the purpose of this paper is theoretical and not to create an extensive empirical investigation relating different algorithms to the new one. This is reasonable since the new algorithm is guaranteed to return a cut that is no worse than the one of SB, at a cost of a rather small running time, and there exists a vast literature comparing SB with other methods. Therefore, we restrict ourselves to compare Algorithm 1 only with the classic SB method.

Here we examine the quality of partitions returned by SB and Algorithm 1 on a wide range of graph matrices. The matrices represents graphs arising in different application domains found in Matrix Market. Table 1 describes the characteristics of these matrices and the comparison between cut sizes of both algorithms.

The last column of Table 1 indicates percentage of improvement of Algorithm 1 over SB. We highlight the best results

Next, we compared the quality of partitions for several random graphs by computing the average gain

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

Matrix	Description	Order	SB	Algorithm 1	
			Cut	Cut	Improv
cegb3306	Structural engineering	3306	18281	2421	**86%**
cegb3024	Structural engineering	3024	19660	19534	0.6%
dwt_1242	Structural engineering	1242	101	72	**28%**
dwt_2680	Structural engineering	2680	85	85	0%
dwt_918	Structural engineering	918	71	61	14%
eris1176	Electrical network	1176	313	202	**35%**
bcspwr10	Power network	5300	44	31	**29%**
jagmesh1	Finite element model	936s	50	50	0%
jagmesh7	Finite element model	1138	29	28	3.4%
lock2232	Structural engineering	2232	1008	977	3%
lshp1270	Finite element model	1270	73	73	0%
lshp1882	Finite element model	1882	89	89	0%
commanche_dual	Structural engineering	7920	46	42	8.6%
lshp2614	Finite element model	2614	105	105	0%
lshp3466	Finite element model	3466	121	121	0%
man_5976	Structural engineering	5976	55682	55391	0.5%

Table 1: Comparative analysis between SB and Algorithm 1.

of Algorithm 1 over SB. Here, random graphs on n vertices are constructed via Erdős–Rnyi model, where an edge is present between two vertices uniformly with probability p. For different combinations of probabilities and number of vertices, we sampled 1000 random graphs and calculated the average gain. The experiments discarded graphs that are disconnected. Table 2 shows the resulting ratio of improvement, where each column corresponds to a given number of vertices n and each row to a given probability p.

$p\backslash n$	100	500	1000
0.1	**7.68%**	**1.87%**	**1.01%**
0.2	**4.67%**	0.90%	0.46%
0.3	**3.29%**	0.61%	0.30%
0.4	2.73%	0.64%	0.32%
0.5	2.62%	**1.10%**	**0.82%**
0.6	2.91%	**1.20%**	**0.68%**
0.7	1.98%	0.39%	0.19%
0.8	1.00%	0.19%	0.09%
0.9	0.57%	0.15%	0.07%

Table 2: Average gain for 1000 random graphs.

The expected number of edges of these random graphs is $pn(n-1)/2$. Thus, Table 2 indicates that

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

Algorithm 1 performs better for sparse graphs than for dense graphs. We highlighted three best results for each column of Table 2. We notice that in multilevel algorithms, the coarsest graph is usually small, with 100 vertices or less. Putting $p = 0.1$ we obtain on the average 495 edges for random graphs with 100 vertices. Table 2 indicates a good improvement ratio for those graphs, with average of 7.6%. That suggests that very often the new algorithm provides better cuts for the initial partition in multilevel algorithms.

6. Open questions. Naturally, one would expect that more eigenvectors can lead to an improvement on the bisection problem. The what extent can this idea be used? Is the improvement related to the largest eigengap, as in the multi-way partitioning problem?

Let $\mathcal{C}$ be an organized partition of a minimum bisection with descriptor vector z. Theorem 3.2 provides a tighter lower bound than (3.7) whenever

$$D_{\mathcal{C}} < \frac{n}{4}(\lambda_3 - \lambda_2) \quad \text{and} \quad \min_{\substack{x^T\mathbf{1}=0\\ \|x\|=1\\ z^Tx=0}} x^T L x \geq \lambda_3. \tag{6.13}$$

This is related to finding conditions such that the third eigenvector improves SB. It is easy to find graphs with this property. However, it remains an open question to characterize graphs such that (6.13) holds.

Is there a better way to construct an organized partition without the use of an integer program? The solution to this question can provide new methods to construct or refine a bisection.

Barnard and Simon [1] introduced a multilevel recursive spectral bisection method. It basically constructs a sequence of smaller and smaller graphs that retains the structure of the original one. Then given a Fiedler vector of a coarse graph it interpolates it to provide an approximation to next Fiedler vector. Is it possible to do the same for the third eigenvector and provide a multilevel recursive spectral bisection with two eigenvectors?

Acknowledgments. Israel Rocha was supported by the Czech Science Foundation, grant no. GJ16-07822Y. The author thanks S. Atan for inspiring the writing of this work. The Institute of Computer Science of the Czech Academy of Sciences is supported by RVO:67985807.

REFERENCES

[1] S.T. Barnard, and H.D. Simon. A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. *Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing*, 711–718, 1993.

[2] T. Bui and C. Jones. Finding good approximate vertex and edge partitions is NP-hard. *Information Processing Letters*, 42(3):153–159, 1992.

[3] T. Bui and C. Jones. A heuristic for reducing fill in sparse matrix factorization. *Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing*, 445–452, 1993.

[4] C.-K. Cheng and Y.-C. A. Wei. An improved two-way partitioning algorithm with stable performance. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 10:1502–1511, 1991.

[5] F. Chung. *Spectral Graph Theory.* CBMS Regional Conference Series in Mathematics, Vol. 92, American Mathematical Society, Providence, RI, 1997.

[6] K.C. Das. The Laplacian spectrum of a graph. *Computers and Mathematics with Applications*, 48:715–724, 2004.

[7] D.M. Cvetković, M. Doob, and H. Sachs. *Spectra of Graphs - Theory and Applications.* VEB Deutscher Verlag d. Wiss., Berlin, 1979; Academic Press, New York, 1979.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 36, pp. 857-877, December 2020.

[8] W.E. Donath and A.J. Hoffman. Algorithms for partitioning of graphs and computer logic based on eigenvectors of connection matrices. *IBM Technical Disclosure Bulletin*, 15(3):938–944, 1972.

[9] W. E. Donath and A.J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973.

[10] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(4):619–633, 1975.

[11] G.H. Golub. Some modified eigenvalue problems. *SIAM Review*, 15(2):318–334, 1973.

[12] L. Grady and E.L. Schwartz. Isoperimetric partitioning: a new algorithm for graph partitioning. *SIAM Journal on Scientific Computing*, 27(6):1844–1866, 2006.

[13] R. Grone and R. Merris. The Laplacian spectrum of a graph II. *SIAM Journal on Discrete Mathematics*, 7(2):221–229, 1994.

[14] S. Guattery and G. L.Miller. On the quality of spectral separators. *SIAM Journal on Matrix Analysis and Applications*, 19(3):701–719, 1998.

[15] G. Karypis and V. Kumar. A parallel algorithm for multilevel graph partitioning and sparse matrix ordering. *Journal of Parallel and Distributed Computing*, 48:71–95, 1998.

[16] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1999.

[17] K. Lang. Finding good nearly balanced cuts in power law graphs. Yahoo Research Labs, technical report, 2004.

[18] B. Mohar. *The Laplacian Spectrum of Graphs*. Wiley, New York, 1991.

[19] B. Hendrickson and R. Leland. A multilevel algorithm for partitioning graphs. Sandia National Laboratories, technical report SAND93-1301, Albuquerque, NM, 1993.

[20] B. Hendrickson and R. Leland. An improved spectral graph partitioning algorithm for mapping parallel computations. *SIAM Journal on Scientific Computing*, 16(2):452–469, 1995.

[21] J.R. Lee, S. Gharan, and L. Trevisan. Multi-way spectral partitioning and higher-order Cheeger inequalities. *Proceedings of the forty-fourth annual ACM Symposium on Theory of Computing (STOC'12)*, 1117–1130, 2012.

[22] S. Pati. The third smallest eigenvalue of the Laplacian matrix. *Electronic Journal of Linear Algebra*, 8:128–139, 2001.

[23] A. Pothen, H.D. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal on Matrix Analysis and Applications*, 11:430–452, 1990.

[24] A. Pothen, H.D. Simon, L. Wang, and S.T. Bernard. Towards a fast implementation of spectral nested dissection. *Supercomputing '92 Proceedings*, IEEE Computer Society Press, Washington, DC, 42–51, 1992.

[25] K. Schloegel, G. Karypis, and V. Kumar. Graph partitioning for high-performance scientific simulations. In: J. Dongara, I. Foster, G. Fox, W. Gropp, K. Kennedy, L. Torczon, and A. White (editors), *CRPC PArallel Computing Handbook*, Chapter 18, Morgan Kaufmann, San Francisco, CA, 491–541, 2002.

[26] D.A. Spielman and S.-H. Teng. Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications*, 421(2-3):284–305, 2007.