Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 00 (2011) 1–10
Procedia Computer Science

International Conference on Computational Science, ICCS 2011

A measure of the local connectivity between graph vertices

Jie Chen[a], Ilya Safro[a]

[a]*Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439*

Abstract

Measuring the connectivity strength between a pair of vertices in a graph is one of the most vital concerns in numerous computational graph problems. In this paper we propose a local measure, together with an efficient algorithm that is scalable and parallelizable. In the heart of the algorithm is an iterative process that propagates and refines random values associated with each vertex. The connectivity measure hence defined, named algebraic distance, is the absolute difference between two values after a small number of iterations. This process is inspired by the bootstrap algebraic multigrid (BAMG), where a similar process is employed to expose connections between variables and to determine their role in convergence of a multigrid solver. We show convergence properties of the proposed measure, and provide a mutually reinforcing model to explain why the algebraic distances meaningfully measure the connectivity in a local sense. The practical effectiveness of the proposed measure is demonstrated on several computational (hyper)graph problems.

Keywords: connectivity, algebraic distance, combinatorial scientific computing, computational graph problems

1. Introduction

Measuring the connectivity between two vertices in a graph is one of the central questions in combinatorial scientific computing. In many graph optimization problems where a greedy strategy is employed, a good heuristic often relies on choosing a pair of vertices that are strongly connected. For example, in the case of a multilevel framework [1, 2, 3, 4], the vertex connectivity is used to expose the coupling strength between coarse and fine vertices. Practically, this connectivity measure should represent only a local effect (i.e., locally how to vertices are connected), and it should endow an efficient computation, the cost of which does not constitute an overhead of the whole graph solver. In this paper, we propose a measure with a simple algorithm for this purpose. We associate each vertex with an initial random value and consider an iterative process where a portion of the vertex value is updated by a weighted combination of the values of its neighbors. This process smooths the associated values for nearby vertices. If two vertices share similar neighborhoods, then after a small number of iterations the two values will be similar, indicating a strong connection between them. We call the absolute difference between two values the *algebraic distance*.

The algebraic distance is inspired by the bootstrap algebraic multigrid (BAMG) method [5] for solving linear systems $Ax = b$. In the heart of any multigrid [6] lies a coarsening process that creates a hierarchy of projections of an original problem domain onto the smaller spaces. In contrast to the geometric multigrid that exploits a regular

Email addresses: jiechen@mcs.anl.gov (Jie Chen), safro@mcs.anl.gov (Ilya Safro)
URL: http://www.mcs.anl.gov/~jiechen (Jie Chen), http://www.mcs.anl.gov/~safro (Ilya Safro)

geometric pattern (of the underlying domain) in choosing coarse variables, the algebraic multigrid (AMG) creates a coarse system by automatic exploration of the "geometry" behind the problem using sophisticated rules of "closeness". BAMG defines such a closeness by running several Gauss-Seidel (GS) relaxations with a random initial vector on the corresponding homogeneous system $Ax = 0$. The speed of convergence of the iterate x signifies the closeness between variables, and it is used to determine the rules of aggregation and interpolation. This paper considers a similar process, where GS is replaced by Jacobi overrelaxation (JOR) and A is replaced by the graph Laplacian L. To generalize, we also consider using several initial random vectors as input.

We begin with analyzing the convergence properties of JOR (see Section 3). The associated linear system is not positive definite, a case commonly discussed in numerical linear algebra literature, and, thus, the analysis is nontrivial. With a concurrent scaling of all the vertex values, the algebraic distance between two vertices i and j converges to the absolute difference between the ith and the jth entry of the eigenvector corresponding to the second smallest eigenvalue of some generalized eigen system related to L. Note that the vertex values themselves do not converge to this eigenvector; in fact, they converge to a constant value, and it is the special scaling that makes their absolute differences nontrivial. This convergence is typically slow. However, we show that in practice the convergence is not necessary because the vertex values stabilize after a small number of iterations. In Section 4 we consider a mutually reinforcing model (which occurs in many real-life problems) and explain why these nonconverged values are meaningful.

In essense, the proposed measure, at convergence, resorts to an eigenvector. Eigenvector approaches and spectral analysis have been extensively studied in graph theory for problems such as graph partitioning, connectivity, and random walks. We point out that the eigenvector related to the proposed measure is neither the same as the Fiedler vector used for spectral partitioning nor the same as the stationary distribution of random walks. In fact, in actual computations we do not use the eigenvector at all—the iterations are prematurely terminated. This premature termination both distinguishes our approach from other eigenvector approaches and makes the computations extremely efficient and usefull as a tool in diverse applications. The computational cost of computing the algebraic distances is only linear to the number of edges in the graph, and the process can be easily parallelized by adopting the same well-known parallelization scheme as the matrix JOR solver. (For parallel matrix-vector multiplications, see, e.g., [7, Ch. 11].)

An instant application of the proposed connectivity measure is to replace the role of the edge weights in algorithms of various combinatorial problems. Examples of these problems include graph arrangements and graph/hypergraph partitioning. In particular, we use the algebraic distances to replace the edge weights in some manner ("re-weighting"). The replacement can also be done in conditional statements when some greedy algorithm has to choose the heaviest edge. Our approach significantly improves a large number of graph algorithms that rely on the edge weights in a major way.

2. Algebraic Distances and Graph Laplacians

Denote by $G = (V, E)$ a connected graph, where the set of vertices V is $\{1, 2, ..., n\}$ and E is a set of m edges. Let $W = \{w_{ij}\}$ be the weighted adjacency matrix of G, where w_{ij} is the non-negative weight of the undirected edge ij; if $ij \notin E$, then $w_{ij} = 0$. The following process (Algorithm 1) iteratively updates a vector x from a random initialization $x^{(0)}$. We use superscripts to distinguish successive iterates and subscripts for vector entries.

Algorithm 1 Iterating a vector x

Input: Parameter ω, initial vector $x^{(0)}$

1: **for** $k = 1, 2, \ldots$ **do**
2: $\quad \tilde{x}_i^{(k)} \leftarrow \sum_j w_{ij} x_j^{(k-1)} / \sum_j w_{ij}, \ \ \forall i.$
3: $\quad x^{(k)} \leftarrow (1 - \omega) x^{(k-1)} + \omega \tilde{x}^{(k)}$
4: **end for**

Based on Algorithm 1, the *algebraic distance* between vertices i and j, at the kth iteration, is defined as

$$s_{ij}^{(k)} = \left| x_i^{(k)} - x_j^{(k)} \right|. \tag{1}$$

With R initial vectors $x^{(0,r)}$, $r = 1, \ldots, R$, each vector is independently updated by Algorithm 1, and the *extended p-normed algebraic distance* is defined as

$$\varrho_{ij,p}^{(k)} = \left(\sum_{r=1}^{R} \left| x_i^{(k,r)} - x_j^{(k,r)} \right|^p \right)^{1/p}, \tag{2}$$

where the superscript $^{(k,r)}$ refers to the kth iteration on the rth initial random vector. For $p = \infty$, by convention,

$$\varrho_{ij,\infty}^{(k)} = \max_{r=1,\ldots,R} \left| x_i^{(k,r)} - x_j^{(k,r)} \right|.$$

The graph *Laplacian matrix* $L = D - W$, where D is the diagonal matrix with elements $d_{ii} = \sum_j w_{ij}$, plays a central role in spectral graph analysis. It can be easily verified that Algorithm 1 is nothing but the Jacobi overrelaxation process for solving[1] the linear system $Lx = 0$, using the relaxation parameter ω. The JOR process converges for any $0 < \omega < 2/\rho(\mathcal{L})$, where $\mathcal{L} = D^{-1/2} L D^{-1/2}$ is the *normalized Laplacian* and $\rho(\cdot)$ is the spectral radius. It is not hard to show that $\rho(\mathcal{L}) \leq 2$. In practical cases we fix $\omega = 1/2$, which facilitates later analysis.

3. Analysis

Standard iterative methods for solving a linear system can be written in a general form

$$x^{(k+1)} = H x^{(k)}, \qquad k = 0, 1, 2, \ldots, \tag{3}$$

where H is the iteration matrix. For JOR, we have $H = I - \omega D^{-1} L$. When the graph is connected, its largest eigenvalue (both algebraic and in magnitude) $\sigma_1 = 1$ is simple, with a corresponding eigenvector $\mathbf{1}$ (a column vector of all ones). Therefore, an immediate consequence is that all the entries of x converge to the same constant, and the algebraic distance $s_{ij}^{(k)}$ for all ij pairs converges to zero. This case is not interesting.

However, a deeper investigation allows one to scale the algebraic distance such that it converges to a nontrivial value. We denote σ_2 the second largest eigenvalue in magnitude of H, and consider scaling $s_{ij}^{(k)}$ for all ij pairs by the kth power of σ_2, namely,[2]

$$\hat{s}_{ij}^{(k)} := s_{ij}^{(k)} / \sigma_2^k. \tag{4}$$

The following theorem establishes the convergence of the *scaled algebraic distance* $\hat{s}_{ij}^{(k)}$ as k goes to infinity.

Theorem 1. *Given a connected graph, let* $(\mu_i, \hat{v}_i)$ *be the eigen-pairs of the matrix pencil* (L, D)*, labeled in nondecreasing order of the eigenvalues, namely,*

$$L\hat{v}_i = \mu_i D \hat{v}_i, \qquad i = 1, \ldots, n, \tag{5}$$

and assume that $\mu_2 \neq \mu_3 \neq \mu_{n-1} \neq \mu_n$*. Unless* $\omega = 2/(\mu_2 + \mu_n)$*, the quantity* $\hat{s}_{ij}^{(k)}$ *defined in* (4) *will always converge to a limit* $|\xi_i - \xi_j|$ *in the order* $O(\theta^k)$*, for some* ξ *and* $0 < \theta < 1$*.*

(i) *If* $0 < \omega < \frac{2}{\mu_3 + \mu_n}$*, then* $\xi \in \operatorname{span}\{\hat{v}_2\}$ *and* $\theta = \frac{1 - \omega\mu_3}{1 - \omega\mu_2}$.

(ii) *If* $\frac{2}{\mu_3 + \mu_n} \leq \omega < \frac{2}{\mu_2 + \mu_n}$*, then* $\xi \in \operatorname{span}\{\hat{v}_2\}$ *and* $\theta = -\frac{1 - \omega\mu_n}{1 - \omega\mu_2}$.

(iii) *If* $\frac{2}{\mu_2 + \mu_n} < \omega < \min\left\{ \frac{2}{\mu_2 + \mu_{n-1}}, \frac{2}{\mu_n} \right\}$*, then* $\xi \in \operatorname{span}\{\hat{v}_n\}$ *and* $\theta = -\frac{1 - \omega\mu_2}{1 - \omega\mu_n}$.

(iv) *If* $\frac{2}{\mu_2 + \mu_{n-1}} \leq \omega < \frac{2}{\mu_n}$*, then* $\xi \in \operatorname{span}\{\hat{v}_n\}$ *and* $\theta = \frac{1 - \omega\mu_{n-1}}{1 - \omega\mu_n}$.

[1] However, we are not interested in actually solving this system, which has infinitely many solutions.

[2] Readers may wonder why the eigenvalue σ_2 does not appear in the rest of the section. Indeed, $(1 - \sigma_2)/\omega$ is an eigenvalue of the matrix pencil (L, D) (see Theorem 1). The eigenvalues μ_i's (replacing σ_2) therefore play a central role in the analysis.

Theorem 1 shows two possible limits ($\hat{v}_2$ or $\hat{v}_n$) for ξ depending on the value of ω. In practice, we may not be able to know which limit ξ achieves, since the μ_i's are not numerically computed, but we can analytically derive some upper/lower bounds for the cutting point $2/(\mu_2 + \mu_n)$ and estimate which of the cases in Theorem 1 is applied. We note that for a graph that is not complete (e.g., a sparse graph), we have $2/(\mu_2 + \mu_n) \geq 2/3$, since $\mu_2 \leq 1$ and $\mu_n \leq 2$. Therefore, when we set $\omega = 1/2$, either case (i) or case (ii) applies. In other words, the scaled algebraic distance $\hat{s}_{ij}^{(k)}$ converges to the difference between the ith and the jth entry of $\hat{v}_2$.

For real-life graphs, the θ corresponding to $\omega = 1/2$ is so close to 1 that the theoretical convergence of $\hat{s}_{ij}^{(k)}$ is of little practical use—it takes an enormous number of iterations before it gets close enough to the limit. (As observed, θ often can be as high as 0.999.) However, an interesting phenomenon is that in practice $x^{(k)}$ soon becomes "stable"; that is, the two iterates $x^{(k+1)}$ and $x^{(k)}$ are almost parallel even when k is small.

Theorem 2. *Given a graph, let $(\mu_i, \hat{v}_i)$ be the eigen-pairs of the matrix pencil (L, D), labeled in nondecreasing order of the eigenvalues. Denote $\hat{V} = [\hat{v}_1, \ldots, \hat{v}_n]$. Let $x^{(0)}$ be the initial vector of the JOR process, and let $a = \hat{V}^{-1}x^{(0)}$ with $a_1 \neq 0$. If the following two conditions are satisfied,*

$$1 - \omega\mu_n \geq 0, \tag{6a}$$

$$f_k := \frac{\alpha {r_k}^{2k}(1 - r_k)^2}{1 + \alpha {r_k}^{2k}(1 + r_k)^2} \leq \frac{1}{\kappa}, \tag{6b}$$

where $\alpha = \left(\sum_{i\neq 1} a_i^2\right) / \left(4a_1^2\right)$, r_k is the unique root of the equation

$$2\alpha r^{2k+2} + 2\alpha r^{2k+1} + (k+1)r - k = 0 \tag{7}$$

on the interval $[0, 1]$, and κ is the condition number of D, then

$$1 - \left\langle \frac{x^{(k)}}{\left\|x^{(k)}\right\|}, \frac{x^{(k+1)}}{\left\|x^{(k+1)}\right\|} \right\rangle^2 \leq \frac{4\kappa f_k}{(1 + \kappa f_k)^2}. \tag{8}$$

We address a few important issues about this result. First, since we use $\omega = 1/2$, condition (6a) is satisfied. Second, f_k is defined as a rational polynomial of r_k, which is the unique root of the polynomial (7) on the interval $[0, 1]$. Therefore, f_k can be easily evaluated and it is typically close to zero. For example, when $\alpha = 100$ and $k = 50$, we have $r_k = 0.9475$, which gives $f_k = 4.6 \times 10^{-4}$. Third, the condition number κ of D is usually not large. For many graphs arising from application areas such as VLSI design and finite-element meshes, if the graph edges have a uniform weight equal to 1, then d_{ii} is the degree of a vertex, and thus for the whole graph the vertex degrees may not vary too much. All this means is that condition (6b) is a mild requirement. The final bound in (8), for $k = 30$ or 50, typically drops to the order of 10^{-4}. Note that $\sin^2(\pi/180) = 3.05 \times 10^{-4}$, which indicates that the angle between $x^{(k)}$ and $x^{(k+1)}$ is around or less than 1°.

The proofs of the two theorems can be found in [8].

4. Mutually Reinforcing Model

We have defined the algebraic distance $s_{ij}^{(k)}$ based on the JOR process and have established a result that the scaled quantity, $\hat{s}_{ij}^{(k)}$, converges to some value that depends solely on the second eigenvector $\hat{v}_2$ of the matrix pencil (L, D). We have also shown that even though the convergence is slow, the iterate $x^{(k)}$ stabilizes quite early. In this section, we present a model to explain how the (scaled) algebraic distance measures the connectivity between the two involved vertices. Emphasis is placed on the *neighborhood* of each vertex, hence the vertex values computed in this way represent the connectivity *locally*.

Consider a mutually reinforcing environment, where entities are influenced by their neighbors. A graph is such an environment, and the vertices are mutually reinforced such that a portion of a vertex value is a weighted average of the influences from its neighbors. Two vertices are close, or similar, if they are placed in two similar environments.

In other words, the difference of their values is a measure of their connectivity. Specifically, let each vertex i be associated with a real number x_i. Except for a μ portion of itself, x_i is a weighted average of its neighbors:

$$x_i = \mu x_i + \sum_{j\sim i} p_{ij} x_j, \tag{9}$$

where the weights $p_{ij} = w_{ij}/d_{ii}$, and $j \sim i$ means j is a neighbor of i. Here, the edge weights w_{ij} are normalized by d_{ii} such that the weights in the model, p_{ij}, are coefficients of a convex combination. The portion $0 \le \mu \le 1$ is an indicator of how strongly an environment acts on a vertex. When μ tends to zero, the neighborhood plays a major role, whereas when μ tends to one, a vertex is so stubborn that its neighbors cannot have a strong impact on it. The coefficient μ does not need to be explicitly specified; it is an innate property of the graph. For such a mutually reinforcing environment, a small μ is more desired. In the matrix form, (9) becomes

$$x = \mu x + D^{-1} W x, \qquad 0 \le \mu \le 1. \tag{10}$$

It is not surprising to see that the eigen-pair $(\mu_2, \hat{v}_2)$ (as mentioned in Theorem 1) is a solution to the model (10) with $\mu = \mu_2$ and $x = \hat{v}_2$. Furthermore, μ_2 is the smallest μ such that (10) is satisfied with a vector x whose entries are not a constant, since μ has to be an eigenvalue of the matrix pencil (L, D). What is surprising is that Theorem 2 implies that the (normalized) iterate $x^{(k)}$ also approximately satisfies the model:

$$\hat{x}^{(k)} \approx \mu_2 \hat{x}^{(k)} + D^{-1} W \hat{x}^{(k)}, \qquad \text{where } \hat{x}^{(k)} = x^{(k)}/\|x^{(k)}\|,$$

which is a result corresponding to the parallelism of $x^{(k)}$ and $x^{(k+1)}$. Therefore, when k is small, say 50, the x values computed in Algorithm 1 approximately represent the reinforcement between vertices in the graph, even though the vector $x^{(k)}$ has not converged in the standard sense.

The iterating process as presented in Algorithm 1 in effect smoothes the values of nearby vertices. Thus, when k is not large, this smoothing operation cannot be extended over a neighborhood. In other words, the computed algebraic distances represent only a local effect. This "local distance" will be very useful in applications where vertex connectivities are only required to be locally estimated. For example, in graph coarsening, a merging of the vertices is a local operation, and a vertex will have little impact on the decision of merging if it is far away from the pairs being considered. Furthermore, in a general context, a multilevel graph technique also benefits from this local consideration if the coarsening heuristic is based on choosing the smallest algebraic distance, rather than the largest edge weight. In Section 6, concrete numerical examples are presented.

5. Related Work

Distances between vertices in a graph can be measured by using many concepts and algorithms, the initial motivation of which may or may not be for this purpose. Examples include commute times [9], diffusion distances [10], and effective resistances [11]. Chebotarev and Shamis [12] surveyed a number of distance measures and studied their normative properties. Here, we focus on two specific directions—spectral graph theory via graph Laplacians and random walks—and draw connections and distinctions between these works and our approach.

Spectral graph theory is closely related to spectral graph partitioning. Let (λ_i, u_i) be the eigen-pairs of L, labeled in nondecreasing order of the eigenvalues:

$$Lu_i = \lambda_i u_i, \qquad i = 1, \ldots, n. \tag{11}$$

Let us compare (11) with (5). The eigenvector u_2 (*Fiedler vector* [13, 14]) is used as an approximate partition vector in the graph bisection problem, and the corresponding eigenvalue λ_2 (*algebraic connectivity*) plays a central role in the global connectivity of the graph, in the sense that $\lambda_2 \neq 0$ if and only if the graph is connected. Furthermore, the Cheeger's inequality (see, e.g., [15] for a variant) bounds λ_2 on the two sides by using the conductance Φ of the graph, which implies that when λ_2 is small, the graph has a cut of small Φ. An interpretation of this result is that a graph is easy to cut if λ_2 is small. Therefore, this *algebraic connectivity* is a global property, whereas the *algebraic distance* defined in this paper works on a pair of vertices and is a local connectivity measure. Interestingly, the involved

eigenvalue μ_2 does suggest how strongly a vertex is influenced by its neighbors in the mutually reinforcing model, which indicates that a small μ_2 is desired.

A distance defined as the difference of two entries of $\hat{v}_2$ can be interpreted from graph embedding (or dimensionality reduction in the data mining literature). As such, graph vertices are embedded on the real axis; that is, the coordinate of i is the ith entry of $\hat{v}_2$. Laplacian eigenmaps [16] explains that this embedding maps "close-by" vertices to "close-by" values, in a manner that it minimizes the difference between i and j scaled by the edge weight w_{ij}. The graph embedding computed in this way induces an Euclidean distance measure for graph vertices. This distance happens to be the limit of the scaled algebraic distance $\hat{s}_{ij}^{(k)}$ as we analyzed. Nevertheless, in practice we do not use the eigenvector $\hat{v}_2$. Apart from unexplained observations that premature termination of the JOR process yields a better performance in graph applications, a major consideration for defining a connectivity measure is that we can afford only a lightweight computational procedure, preferably with easy parallelization, since it is used within an external application such as graph partitioning. Computing eigenvectors, on the other hand, is an extensively studied subject in numerical linear algebra and parallel computing [17, 18], and it is also considered in other special situations, such as in a decentralized distributed computing environment [19] or where only an approximate vector with high successful probability is needed [20]. For the current setting, the Lanczos algorithm is the preferable choice to compute $\hat{v}_2$; the time cost is generally linear to m with additional small overheads, depending on the eigen gap. However, it can still not beat Algorithm 1 both in time and in memory, considering that the latter runs in km time and requires only $2n + m$ memory space.

The iteration matrix (see equation (3)) of the JOR process,

$$H = I - \omega D^{-1} L = (1 - \omega) I + \omega D^{-1} W,$$

suggests that H happens to be the matrix of a lazy random walk, where the walker does not change states with probability $1 - \omega$. Despite this close relation, the stationary distribution of the lazy random walk is the *principal left eigenvector* of H, whereas the involved eigenvector $\hat{v}_2$ in the analysis of algebraic distances is the *second right eigenvector* of H. It would be interesting to consider an interpretation of the algebraic distances from the perspective of random walks. Indeed, the idea of iteratively applying a vector to the left-hand side of the lazy random walk matrix has been exploited to measure a local connectivity and to perform a local partitioning of the graph [21, 22]. In our case, the right eigenvector of H does not correspond to any stationary distributions, and this feature distinguishes our approach with those in [21, 22].

6. Applications

In this section, we demonstrate how the algebraic distances can be used as a practical tool for improving existing algorithms for graph applications. For this purpose, we have chosen two strategies. In the first strategy we substituted the original edge weights with the algebraic distances while keeping the original algorithms unmodified. In the second strategy we substituted the decision criteria based on chosing the heaviest edge with one based on algebraic distances. We considered the following baseline algorithms: spectral graph arrangements, spectral graph bisection, multilevel bisection of hypergraphs, and multilevel compression-firendly ordering of a network. All these problems are of great practical significance. Thus, fast and qualitative heuristics are much appreciated in areas such as combinatorial scientific computing, VLSI placement, graph visualization, and biological applications. The experimental graphs were of different sizes ($|E|$ was between 10^3 and 10^7). Most of them were selected from the UFL database [23] and the SNAP project [24].

Spectral graph arrangements and bisection. Graph arrangements consist of a well-known family of (usually) NP-complete problems, such as minimum p-sum, cutwidth, profile of a graph, and sum cut. The common goal of these problems is to arrange the graph vertices along the line with integer coordinates from 1 to $|V|$ such that some functional will be minimized. In this paper we consider the minimum p-sum problem with $p = 1$ (also known as the minimum linear arrangement) and $p = 2$. Both problems are known to be NP-complete [25]. The objective is to minimize the functional

$$\sigma_p(G, \pi) = \left(\sum_{ij} w_{ij} |\pi(i) - \pi(j)|^p \right)^{1/p}, \tag{12}$$

where π is a bijection from V to $\{1, 2, ..., n\}$.

A graph bisection is also a well-known NP-hard problem [25]. The goal of the problem is to find a partitioning of V into two disjoint nonempty subsets Π_1 and Π_2, while enforcing the following:

$$\text{minimize} \sum_{i\in\Pi_p \Rightarrow j\notin\Pi_p} w_{ij} \quad \text{such that } \forall p \in [1,2], |\pi_p| \leq (1+\alpha)\cdot\frac{|V|}{2}\,, \tag{13}$$

where α is a given imbalance factor. This problem can be easily generalized to the k-partitioning case, and many heuristics for such a case are based on the bisection.

A well-known and successful heuristic, called the *spectral approach*, uses the Fiedler vector (the eigenvector u_2 in (11)) as a relaxed/approximate solution to the linear arrangement [26] and partitioning [27, 28, 29] problems. It is also widely used for solving many other problems, such as envelope reduction of sparse matrices [30]. The spectral approach consists of the following steps: (a) calculate u_2 (see (11)); (b) get ϕ, an ordering of the graph nodes according to the corresponding coordinates in u_2; and (c) calculate either an edge cut between $\phi(\lfloor n/2\rfloor)$ and $\phi(\lfloor n/2\rfloor+1)$ for (13) or a sum of stretched edge lengths for (12). This approach can be viewed as a converged averaging process, which leads to the solution of a quadratic optimization problem if the restriction on the solution coordinates is relaxed; that is, the coordinates need not all be integers, as in the case where all vertices are equal (that is not true in many scientific computing applications and models). It is well known that u_2 provides the best nontrivial solution for this problem. Often in the process of global averaging, the role of dominating edges can be diminished. This is a reason we use the algebraic distance to improve the spectral approach by introducing a dominancy for the edges.

In our heuristics for minimizing (12) and (13) we use the spectral approach as a black-box algorithm. We substitute the original graph edge weights with the corresponding algebraic distances and use the same sprectral approach on the modified graph. This extension is presented in Algorithm 2. The experimental results for graph arrangement and partitioning problems are presented in Figures 1(a) and 1(b), respectively, which show a favorable improvement.

Algorithm 2 Improved spectral approach

Input: Graph G, problem type: minimum p-sum (MPS) or partitioning (PART)

1: For all edges $ij \in E$ calculate $\varrho^{(k)}_{ij,\infty}$ for $k = 20$ and $R = 5$
2: $G' \leftarrow G$ with modified edge weights $w_{ij} = 1/\varrho^{(k)}_{ij,\infty}$
3: $\phi \leftarrow$ order obtained from Fiedler vector of G'
4: If MPS then return $\sigma(G, \phi)$
5: If PART then return cost of edge cut between $\phi(\lfloor n/2\rfloor)$ and $\phi(\lfloor n/2\rfloor+1)$

Hypergraph partitioning. A hypergraph $\mathcal{H}$ is a pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of nets (hyperedges). Each $h \in \mathcal{E}$ is a subset of $\mathcal{V}$. Since a hypergraph is a generalization of graphs, a hypergraph bisection is NP-hard. Similar to the goal of the graph bisection, the goal of the hypergraph bisection is to find a partitioning of $\mathcal{V}$ into two disjoint nonempty subsets Π_1 and Π_2, while enforcing the following:

$$\text{minimize} \sum_{\substack{h\in\mathcal{E} \text{ s.t. } \exists i,j\in h \text{ and} \\ i\in\Pi_p \Rightarrow j\notin\pi_p}} w_h \quad \text{such that } \forall p \in [1,2], |\Pi_p| \leq (1+\alpha)\cdot\frac{|V|}{2}\,, \tag{14}$$

where α is a given imbalance factor.

HMetis2 [31] is one of the fastest and most successful modern multilevel solvers for partitioning problems. We use it as a black-box solver but modify the hyperedge weights before invoking it (see Algorithm 3). First, we construst a bipartite graph model for the hypergraph. In particular, we create $G = (V, E)$ with $V = \mathcal{V} \bigcup \mathcal{E}$ and $ij \in E$ if $i \in \mathcal{V}$ appears in $j \in \mathcal{E}$. The edge weights are preserved with no changes. Second, we compute algebraic distances on G, from which we assign new weights to the hyperedges of $\mathcal{H}$. The numerical results of comparing the two algorithms are presented in Figure 1(c). Clearly, a significant improvement can be obtained by using the algebraic distance substitutes.

Compression-friendly ordering. Qualitative solutions of the minimum logarithmic arrangement problem (MLogA, also NP-hard) can be used to achieve a better network compression [32]. We demonstrate the role of the algebraic

Algorithm 3 Hypergraph multilevel bisection with algebraic distance based preprocessing

Input: Hypergraph $\mathcal{H}$, $k = 20$, $R = 10$

$G = (V, E) \leftarrow$ bipartite graph model
Create R initial vectors $x^{(0,r)}$
for $r = 1, 2, \ldots, R$ **do**
  for $m = 1, 2, \ldots, k$ **do**
    $x_i^{(m,r)} \leftarrow \sum_j w_{ij} x_j^{(m-1,r)} / \sum_j w_{ij}, \ \forall i.$
  end for
end for
return algebraic distances $s_h^{(k)} \leftarrow \sum_r \max_{i,j \in h} \left| x_i^{(k,r)} - x_j^{(k,r)} \right|, \ \forall h \in \mathcal{E}.$

Input: Hypergraph $\mathcal{H}$ with net weights $1/s_h^{(k)}$

$C \leftarrow$ net cut obtained by HMetis2 on $\mathcal{H}$ with the modified net weights
return cost of C with original net weights

distance in a recently developed fast multilevel method for MLogA [33]. Similar to (12), the goal of MLogA is to find a permutation of nodes that minimizes $\sum_{ij} w_{ij} \lg |\pi(i) - \pi(j)|$. In contrast to the previous approach of substituting edge weights by algebraic distances, we substitute all decision statements that chose the heaviest edge (during a coarsening) with decisions based on the smallest algebraic distance. The numerical results are shown in Figure 1(d), which also demonstrate favorable improvement by using our approach.

7. Conclusion

We have presented an iterative process for smoothing values associated with graph vertices and defined a notion of algebraic distance between vertices when the process stabilizes (not converges). The distances thus defined represent the local connectivity of a pair of vertices; that is, two vertices are strongly connected if their algebraic distance is small. We show several applications to demonstrate how the algebraic distance can be used to define quantities that replace the graph edge weights in algorithms for combinatorial optimization problems. The experiments show that with an algebraic distance preprocessing, the quality of several baseline algorithms can be greatly improved. Furthermore, its easy parallelization makes it particularly attractive for dealing with large-scale graphs and problems.

References

[1] G. Karypis, R. Aggarwal, V. Kumar, S. Shekhar, Multilevel hypergraph partitioning: applications in VLSI domain, IEEE Trans. Very Large Scale Integr. Syst. 7 (1) (1999) 69–79. doi:http://dx.doi.org/10.1109/92.748202.

[2] D.-Z. Du, P. Pardalos, Handbook of Combinatorial Optimization, Kluwer Academic Publishers, 1999.
URL `citeseer.ist.psu.edu/213350.html`

[3] E. Sharon, A. Brandt, R. Basri, Fast multiscale image segmentation, in: Proceedings IEEE Conference on Computer Vision and Pattern Recognition, 2000, pp. 70–77.
URL `citeseer.nj.nec.com/sharon99fast.html`

[4] I. Safro, D. Ron, A. Brandt, Multilevel algorithms for linear ordering problems, Journal of Experimental Algorithmics 13 (2008) 1.4–1.20.

[5] A. Brandt, Multiscale scientific computation: Review 2001, in: Multiscale and Multiresolution Methods, Vol. 20, Springer Verlag, 2002, pp. 3–95.

[6] U. Trottenberg, A. Schuller, Multigrid, Academic Press, Inc., Orlando, FL, USA, 2001.

[7] Y. Saad, Iterative methods for sparse linear systems, 2nd Edition, SIAM, 2003.

[8] J. Chen, I. Safro, Algebriac distance on graphs, Tech. Rep. ANL/MCS-P1696-1009, Mathematics and Computer Science Division, Argonne National Laboratory (2009).

[9] F. Fouss, A. Pirotte, J.-M. Renders, M. Saerens, Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation, IEEE Transactions on Knowledge and Data Engineering 19 (3) (2007) 355–369.

[10] B. Nadler, S. Lafon, R. R. Coifman, I. G. Kevrekidis, Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators, in: Advances in Neural Information Processing Systems, Vol. 18, MIT Press, 2005, pp. 955–962.

[11] A. Ghosh, S. Boyd, A. Saberi, Minimizing effective resistance of a graph, SIAM Rev. 50 (1) (2008) 37–66. doi:http://dx.doi.org/10.1137/050645452.

[12] P. Chebotarev, E. Shamis, On proximity measures for graph vertices, Automation and Remote Control 59 (10) (1998) 1443–1459.

[13] M. Fiedler, Algebraic connectivity of graphs, Czechoslovak Math. J. 23 (1973) 298–305.

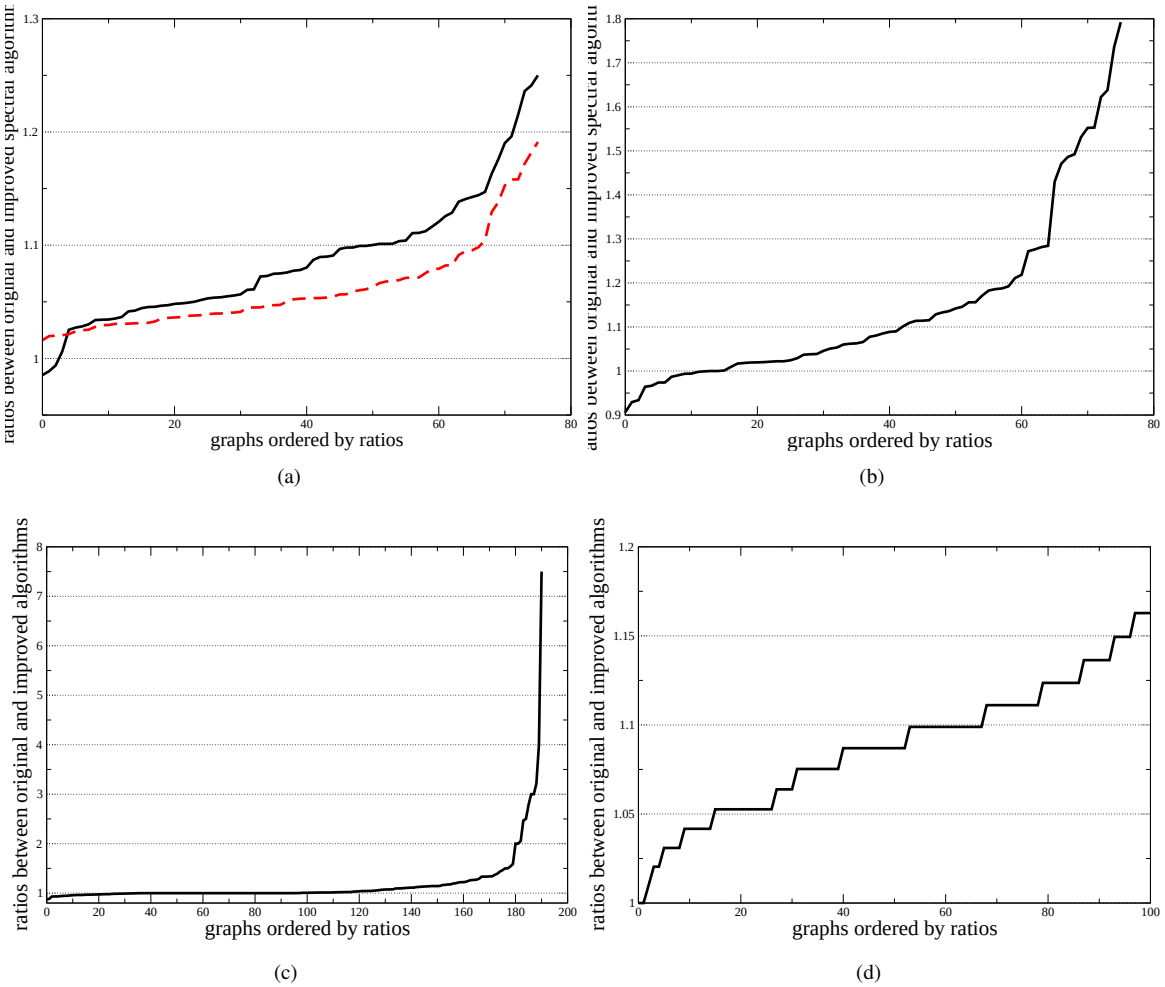

Figure 1: Improvement of several baseline algorithms with algebraic distance preprocessing. Each point corresponds to the average of ratios between minimization results produced by the original and our algorithms. A value larger than one implies improvement. In all experiments the averages were calculated over 50 repeated executions with different random initializations. **(a)** Spectral ordering. The solid and dashed curves correspond to the minimum linear arrangement and the minimum 2-sum problems, respectively. **(b)** Spectral bisection. **(c)** Hypergraph bisection. **(d)** Minimum logarithmic arragement. In the improved algorithm all conditions involving the heaviest edge have been replaced by ones with the smallest algebraic distance.

[14] M. Fiedler, A property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory, Czechoslovak Math. J. 25 (1975) 619–633.
[15] F. R. K. Chung, Spectral Graph Theory, American Mathematical Society, 1997.
[16] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Computation 16 (6) (2003) 1373–1396.
[17] Y. Saad, Numerical Methods for Large Eigenvalue Problems, Halstead Press, 1992.
[18] G. H. Golub, C. F. Van Loan, Matrix Computations, Johns Hopkins University Press, 1996.
[19] D. Kempe, F. McSherry, A decentralized algorithm for spectral analysis, in: Proceedings of the 36th annual ACM Symposium on Theory of Computing, 2004.
[20] D. A. Spielman, S.-H. Teng, Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems, arXiv cs.NA/0607105.
[21] D. A. Spielman, S.-H. Teng, A local clustering algorithm for massive graphs and its application to nearly-linear time graph partitioning, arXiv cs.DS/0809.3232.
[22] R. Andersen, F. Chung, K. Lang, Local graph partitioning using pagerank vectors, in: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, 2006.
[23] T. Davis, University of Florida Sparse Matrix Collection, NA Digest 97 (23).
URL `http://www.cise.ufl.edu/research/sparse/matrices`
[24] J. Leskovec, Stanford Network Analysis Package (SNAP), http://snap.stanford.edu/index.html.

[25] M. R. Garey, D. S. Johnson, Computers and Intractability. A Guide to the Theory of NP-Completeness, Freemann and Company, 1979.
[26] M. Juvan, B. Mohar, Optimal linear labelings and eigenvalues of graphs, Discrete Appl. Math. 36 (2) (1992) 153–168. doi:http://dx.doi.org/10.1016/0166-218X(92)90229-4.
[27] B. Hendrickson, R. Leland, An improved spectral graph partitioning algorithm for mapping parallel computations, SIAM Journal on Scientific Computing 16 (2) (1995) 452–469.
URL `citeseer.ist.psu.edu/hendrickson95improved.html`
[28] A. Pothen, H. D. Simon, K.-P. Liou, Partitioning sparse matrices with eigenvectors of graphs, SIAM J. Matrix Anal. Appl. 11 (3) (1990) 430–452. doi:http://dx.doi.org/10.1137/0611030.
[29] D. A. Spielman, S.-H. Teng, Spectral partitioning works: Planar graphs and finite element meshes, in: IEEE Symposium on Foundations of Computer Science, 1996, pp. 96–105.
URL `citeseer.ist.psu.edu/spielman96spectral.html`
[30] S. Barnard, A. Pothen, H. Simon, A spectral algorithm for envelope reduction of sparse matrices, Numerical Linear Algebra with Applications 2 (4) (1995) 317–334.
[31] G. Karypis, V. Kumar, Metis A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center, Minneapolis, MN (Sep. 1998).
[32] F. Chierichetti, R. Kumar, S. Lattanzi, M. Mitzenmacher, A. Panconesi, P. Raghavan, On compressing social networks, in: KDD '09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 2009, pp. 219–228. doi:http://doi.acm.org/10.1145/1557019.1557049.
[33] I. Safro, B. Temkin, Multiscale approach for the network compression-friendly ordering, submitted, Preprint ANL/MCS-P1751-0410.