Algebraic Distance on Graphs

Jie Chen* Ilya Safro†

Abstract

Measuring the connection strength between a pair of vertices in a graph is one of the most vital concerns in many graph applications. Simple measures such as edge weights may not be sufficient for capturing the local connectivity. In this paper, we consider a neighborhood of each graph vertex and propagate a certain property value through direct neighbors. We present a measure of the connection strength (called the algebraic distance) defined from an iterative process based on this consideration. The proposed measure is attractive in that the process is simple, linear, and easily parallelized. A rigorous analysis of the convergence property of the process confirms the underlying intuition that vertices are mutually reinforced and that the local neighborhoods play an important role in influencing the vertex connectivity. We demonstrate the practical effectiveness of the proposed measure through several combinatorial optimization problems on graphs and hypergraphs.

1 Introduction

Measuring the connectivity between two vertices in a graph is one of the central questions in many theoretical and applied areas in computer science. Sometimes, the edge weights are artificial quantities assigned to each edge to distinguish how close the vertices are to each other. However, such assignments may not always be possible or sufficient because of practical limitations, including a lack of other than Boolean information and the difficulty in determining all pairwise vertex relationships. In many graph algorithms, we often need to choose a best suitable pair of vertices (e.g., the heaviest edge). In such situations, a greedy strategy might have to arbitrarily break the ties due to equal edge weights (without a better choice). A better performance might be possible if the strategy could recognize a pair of vertices that are not connected by an edge but pose a strong indirect connection (e.g., for node merging).

Because of the practical significance of vertex connectivity, many algorithms have been proposed to model it. Examples include the length of the shortest path, the number of simple paths between a pair of vertices, maximum flows, and minimum edge cuts/vertex separators. In a random-walk approach [10, 24], the average first-passage time/cost and average commute time were used. A similarity measure between nodes of a graph integrating indirect paths, based on the matrix-forest theorem, was proposed in [4]. Effective resistance [13] was used to model a connectivity with

*Department of Computer Science and Engineering, University of Minnesota at Twin Cities, MN 55455. Email: jchen@cs.umn.edu. Work of this author was supported by NSF grant DMS-0810938, a University of Minnesota Doctoral Dissertation Fellowship, and the CSCAPES institute, a DOE project.

†Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439. Email: safro@mcs.anl.gov. This work was funded by the CSCAPES institute, a DOE project, and in part by DOE Contract DE-AC02-06CH11357.

electrical circuit conductance. A convergence of the compatible relaxation [1] was measured in algebraic multigrid (AMG) schemes [2] in order to detect strong connections between fine and coarse points. A similarity method based on a probabilistic interpretation of a diffusion was introduced in [20]. Our goal is to design a family of algorithms (and measures) that are fast and easy to implement and parallelize and that can be applied locally to the data.

The proposed family of measures is called the *algebraic distance*. We will give a formal definition in Section 2; for now we note that the algorithm is a stationary iterative process that propagates some information about every node to its neighborhood. After a few iterations, the propagated values define a distance between all pairs of nodes. Conceptually, a small distance means a strong connection because, by the propagation, closely connected vertices will converge to similar values.

The definition of the algebraic distance is motived by the bootstrap AMG (BAMG) method [3] for solving a symmetric positive definite system $Ax = b$. In this method, a Gauss-Seidel (GS) process is run on the matrix A in order to expose the slow-to-converge variables x_i allow a better interpolation of the low residual errors. In our definition, hence the iterative process is essentially a Jacobi overrelaxation (JOR) process run on the graph Laplacian matrix. Recently, the algebraic distance was used as a component of an AMG-based coarsening scheme for graph linear ordering problems [21]. Despite considerable empirical evidence of success in multilevel linear ordering algorithms, however, the concept of algebraic distance is still not well understood and has not been used widely in combinatorial optimization problems. This paper studies some properties of this relaxation process and interprets the algebraic distances under a mutually reinforcing environment model, where the neighborhood connectivity information governs the connectivity of the vertices. In particular, two vertices are strongly connected if they are surrounded by similar neighborhoods. With this interpretation, the applications of this measure are no longer restricted to multilevel algorithms. Whenever the concept of vertex connectivity is applicable, we can use the algebraic distances to measure the connection strengths between vertices. We show a few such applications in this paper.

An advantage of the proposed measure is its computational efficiency. The other possible heuristics mentioned earlier are, in general, more expensive to compute. For example, the problem of counting the number of simple paths connecting a pair of vertices is #P-complete [23], the Floyd-Warshall algorithm for computing all pairs shortest paths has a cubic time complexity, and the computation of the commute times involves the pseudo-inverse of the graph Laplacian matrix. In contrast, since our method is a JOR process, its time complexity is linear in the number of iterations and the number of graph edges. This is a significant reduction, especially for sparse graphs. Further, the JOR iterations are easy to parallelize because, unlike other iterative processes such as GS, the update of an entry x_i does not require the most recent value of x_{i-1}. Thus, the proposed strategy has a strong potential for large-scale distributed computing.

The rest of the paper is organized as follows. In Section 2 we present a formal definition of algebraic distance and review some important properties of the graph Laplacian matrix, which will be used throughout the paper. In Section 3, we study some general convergence properties of the proposed measure; and in Section 4, we present further results specifically for the JOR process. Based on our theoretical analysis, we revisit the intuition behind the definition of algebraic distance, and in Section 5, discuss a mutually reinforcing environment model. In Section 6, we demonstrate the effectiveness of the proposed measure through a variety of graph applications. We conclude in Section 7.

2 Notation and Preliminaries

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

Algorithm 1 Computing algebraic distances for graphs

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**

We define $s_{ij}^{(k)}$, the *algebraic distance* between vertices i and j, at the kth iteration, to be

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

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

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

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

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

The graph Laplacian matrix, denoted as L, is defined as

$$D - W,$$

where D is the diagonal matrix with diagonal elements d_{ii} equal to $\sum_j w_{ij}$. Then, Algorithm 1 is essentially the JOR method for solving[1] the linear system

$$Lx = 0, \tag{3}$$

using the relaxation parameter ω. The normalized Laplacian, denoted as $\mathcal{L}$, is

$$D^{-1/2} L D^{-1/2}.$$

Then the JOR method converges for any $0 < \omega < 2/\rho(\mathcal{L})$, where $\rho(\cdot)$ means the spectral radius of a matrix. In practice, we usually fix ω at $1/2$, which guarantees convergence and many other nice properties discussed later.

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

2.1 Laplacian and Normalized Laplacian

In this section we review some important properties of L and $\mathcal{L}$ that are necessary for the convergence analysis of system (3) (for details see, e.g., [5]). The Laplacian L is positive semi-definite; therefore, L is diagonalizable and has n nonnegative eigenvalues. Let (λ_i, u_i) denote the eigen-pairs of L ordered in nondecreasing order of λ_i:

$$Lu_i = \lambda_i u_i, \qquad i = 1, 2, \ldots, n \ . \tag{4}$$

The spectrum of L has the following properties:

(i) By the Gershgorin circle theorem, the largest eigenvalue $\lambda_n \leq 2 \max_i d_{ii}$.

(ii) The smallest eigenvalue(s) of L is zero, since $L\mathbf{1} = 0$, where $\mathbf{1}$ is the vector of all ones. The multiplicity of $\lambda_1 = 0$ is c if and only if the graph has c connected components. In particular, if the graph is connected, then λ_1 is simple. In this case, $\text{span}\{u_1\} = \text{span}\{\mathbf{1}\}$.

(iii) When the graph is connected, by the Courant-Fischer minimax theorem, λ_2 has a bound

$$0 < \lambda_2 \leq \frac{x^T L x}{x^T x}, \qquad \forall\ x \neq 0,\ x^T \mathbf{1} = 0.$$

This eigenvalue, λ_2, is called the *algebraic connectivity* of the graph, and the corresponding (normalized) eigenvector u_2 is called the *Fiedler vector*.

The normalized Laplacian $\mathcal{L}$ is also positive semi-definite. Let (μ_i, v_i) denote the eigen-pairs of $\mathcal{L}$ ordered in the nondecreasing order of μ_i:

$$\mathcal{L} v_i = \mu_i v_i, \qquad i = 1, 2, \ldots, n \ . \tag{5}$$

The equivalence of the following three systems

$$\begin{aligned} \mathcal{L} v_i &= \mu_i v_i, & & \\ L\hat{v}_i &= \mu_i D \hat{v}_i, & \hat{v}_i &= D^{-1/2} v_i, & (6) \\ (D^{-1}L)\hat{v}_i &= \mu_i \hat{v}_i, & \hat{v}_i &= D^{-1/2} v_i, & (7) \end{aligned}$$

indicates that the spectral properties of $\mathcal{L}$ can often be derived from those of the matrix $D^{-1}L$ or the pencil (L, D), and vice versa. The spectrum of $\mathcal{L}$ has the following properties.

(i) The spectral radius $\rho(\mathcal{L})$ equals $\mu_n \leq 2$. In particular, $\mu_n = 2$ if and only if there exists a connected component that is bipartite. Indeed, when all the connected components of the graph are bipartite, the eigenvalues μ_i's are distributed symmetrically around 1:

$$\mu_i = 2 - \mu_{n+1-i}, \qquad \text{for } i = 1, \ldots, n.$$

(ii) The multiplicity of the eigenvalue zero of $\mathcal{L}$ is the same as that of L. We have always $\mu_1 = 0$, and $D^{1/2}\mathbf{1}$ is a corresponding eigenvector.

(iii) When the graph is connected, we have a bound for μ_2:

$$0 < \mu_2 \leq \frac{x^T \mathcal{L} x}{x^T x}, \qquad \forall\ x \neq 0,\ x^T D^{1/2} \mathbf{1} = 0.$$

A simple immediate result is that $\mu_2 \leq 1$ if the graph is not complete, which is derived by choosing a suitable x.

We summarize the eigen-systems (4), (5), (6), and (7) in Table 1 for future references. The iteration matrix H will be introduced in the next section. When necessary, the eigenvectors have a "unit" length, that is the vectors u_i, v_i, and ϕ_i have a unit 2-norm, and the vectors $\hat{v}_i$ have a unit D-norm.

Table 1: Matrices and eigen-systems appeared in this paper.

	Matrix	Eig. val.	Eig. vec.	Ordering
Laplacian	L	λ_i	u_i	$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
Normalized Laplacian	$\mathcal{L}$	μ_i	v_i	$0 = \mu_1 \leq \mu_2 \leq \cdots \leq \mu_n \leq 2$
Matrix pencil	(L, D)	μ_i	$\hat{v}_i$	same as above
Matrix	$D^{-1}L$	μ_i	$\hat{v}_i$	same as above
Iteration matrix	H	σ_i	ϕ_i	$1 = \sigma_1 \geq \lvert\sigma_2\rvert \geq \cdots \geq \lvert\sigma_n\rvert$

3 Iterative Methods for Graph Laplacians

Algorithm 1, on which the definition of the algebraic distance $s_{ij}^{(k)}$ is based, is the JOR method for solving the linear system (3). There being rich results for nonsingular systems, however, here the matrix L is singular, and thus we need to first study the convergence properties for this particular system. In this section, we establish some general results for the convergence of several classical iterative methods (including JOR) for the matrix L. The special case for JOR automatically applies.

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

$$x^{(k+1)} = H x^{(k)}, \qquad k = 0, 1, 2, \ldots, \tag{8}$$

where H is the iteration matrix. Let the Laplacian $L = D - W_L - W_U$, where W_L and W_U are the strict lower and upper triangular parts of W, respectively. Then the iteration matrices for Gauss Seidel, Jacobi, SOR, and JOR are, respectively,

$$H_{GS} = (D - W_L)^{-1} W_U, \qquad H_{SOR} = (D/\omega - W_L)^{-1} \left((1/\omega - 1)D + W_U\right),$$
$$H_{JAC} = D^{-1}(W_L + W_U), \qquad H_{JOR} = (D/\omega)^{-1} \left((1/\omega - 1)D + W_L + W_U\right).$$

When $\omega = 1$, GS is equivalent to SOR, and Jacobi is equivalent to JOR. We will use the notation H when the discussions are general or apply to all the iterative methods; we will add subscripts when an individual method is emphasized. In the next subsection, we first study the eigen-structure of H.

3.1 The Iteration Matrix H

A matrix $A \in \mathbb{R}^{n\times n}$ is said to be *convergent* if $\lim_{k\to\infty} A^k$ exists.[2] Let α_i denote the eigenvalues of A, where $i = 1, \ldots, n$. Then, A is convergent if and only if (i) $|\alpha_i| \leq 1$, (ii) $|\alpha_i| = 1$ implies $\alpha_i = 1$, and (iii) the algebraic multiplicity of the eigenvalue 1 equals its geometric multiplicity. In other words, the Jordan canonical form of a convergent matrix looks like the following:

$$\begin{bmatrix} I_{t\times t} & 0 \\ 0 & J \end{bmatrix},$$

where $t \geq 0$ is the algebraic/geometric multiplicity of the eigenvalue 1, and J consists of Jordan blocks for all the other eigenvalues $|\alpha_i| < 1$. The following theorem implies that unless the graph has a bipartite connected component and Jacobi iterations are used (or equivalently JOR with $\omega = 1$), the iteration matrix H for the system (3) is always convergent.

Theorem 1. *The iteration matrix H for the linear system (3) has the following properties.*

(*i*) *The matrices H_{GS}, H_{SOR} (with $0 < \omega < 2$), and H_{JOR} (with $0 < \omega < 2/\rho(\mathcal{L})$) are convergent.*

(*ii*) *The matrix H_{JAC} is convergent if and only if none of the connected components of the graph is bipartite. When the graph has a bipartite connected component, H_{JAC} has an eigenvalue -1.*

(*iii*) *The multiplicity of the eigenvalue 1 of H is the same as that of the eigenvalue 0 of L.*

(*iv*) *If the graph is connected, then the eigenvalue 1 of H is simple.*

(*v*) *The spectral radii of H_{GS}, H_{JAC}, H_{SOR}, and H_{JOR} are all 1.*

Proof. For the proof of the first property, see [8]. Property (ii) is easy to verify, by noting that

$$L\hat{v} = \mu D\hat{v} \iff H_{JAC}\ \hat{v} = (1-\mu)\hat{v}$$

and that the largest eigenvalue of (L, D) is strictly less than 2 if and only if none of the connected components of the graph is bipartite. Property (iii) follows from

$$Lu = 0 \iff u = Hu.$$

Property (iv) immediately follows from property (iii). For GS, SOR, and JOR, the fact that the spectral radius of H is 1 follows from properties (i) and (iii). For Jacobi, by using the Gershgorin circle theorem, it is clear that the eigenvalues of H_{JAC} all have a modulus less than or equal to 1. Thus, by property (iii), $\rho(H_{JAC}) = 1$. □

[2]Some authors (e.g., [17]) use the term *convergent* for a matrix A where the limit A^k is zero. However, the interesting case in this paper is that the limit is nonzero. Thus, we make a broader inclusion in the definition here.

3.2 Convergence of the Iterative Process (8)

When H is convergent, we denote its similarity transform to the Jordan canonical form as

$$PHP^{-1} = \begin{bmatrix} I_{t\times t} & 0 \\ 0 & J \end{bmatrix},$$

where P is nonsingular. The initial vector $x^{(0)}$ can be uniquely decomposed as the sum of a vector in range$(I-H)$ and a vector z in null$(I-H)$:

$$x^{(0)} = (I-H)y + z.$$

Since $z \in \text{null}(I-H)$, we have $Hz = z$. Therefore,

$$\begin{aligned} x^{(k)} &= H^k x^{(0)} \\ &= H^k(I-H)y + H^k z \\ &= P^{-1}\begin{bmatrix} I_{t\times t} & 0 \\ 0 & J^k \end{bmatrix}\begin{bmatrix} 0 & 0 \\ 0 & I-J \end{bmatrix} Py + z. \end{aligned}$$

Note that the limit of J^k is zero. Thus,

$$\lim_{k\to\infty} x^{(k)} = z.$$

This result is summarized in the following theorem.

Theorem 2. *When the iteration matrix H for the linear system* (3) *is convergent, the iterative process* (8) *converges. In such a case, the iterate $x^{(k)}$ converges to zero if the initial vector $x^{(0)} \in$* range$(I-H)$*, otherwise $x^{(k)}$ converges to a vector in* null$(I-H)$*.*

In particular, if the eigenvalue 1 of H is simple, the subspace null$(I-H)$ is spanned by the vector $\mathbf{1}$. Hence we have the following corollary.

Corollary 3. *If the graph is connected and the iteration matrix H for the linear system* (3) *is convergent, then the iterate $x^{(k)}$ converges to zero if the initial vector $x^{(0)} \in$* range$(I-H)$*; otherwise $x^{(k)}$ converges to a nonzero scalar multiple of $\mathbf{1}$.*

An immediate consequence of the above corollary is that the algebraic distance vanishes as k goes to infinity.

Corollary 4. *Under the conditions of Corollary 3, the quantity $s_{ij}^{(k)}$ defined in* (1) *converges to zero for all i and j.*

Corollary 4 seems to suggest that the definition of the algebraic distance as a measure of the strength of connection is inappropriate. However, we are actually interested in comparing the relative magnitudes of $s_{ij}^{(k)}$ for different (i,j) pairs. In other words, a concurrent scaling of the quantity $s_{ij}^{(k)}$ for all i and j will not compromise the measure. In the following, we will discuss such a quantity, denoted as $\hat{s}_{ij}^{(k)}$.

To this end, we consider a connected graph and make an additional mild assumption that H is diagonalizable. Let (σ_i, ϕ_i) denote the eigen-pairs of H:

$$H\phi_i = \sigma_i \phi_i, \qquad i = 1, \ldots, n, \tag{9}$$

where the eigenvalues are labeled in the order

$$1 = \sigma_1 > |\sigma_2| \geq |\sigma_3| \geq \cdots \geq |\sigma_n|$$

according to Theorem 1. The eigenvectors ϕ_i's are linearly independent, and

$$\begin{aligned} \text{span}\{\phi_1\} &= \text{span}\{\mathbf{1}\} = \text{null}(I - H), \\ \text{span}\{\phi_\ell \mid \ell = 2, \ldots, n\} &= \text{range}(I - H). \end{aligned}$$

Let the initial vector $x^{(0)}$ be expressed as a linear combination of the eigenvectors:

$$x^{(0)} = a_1\phi_1 + a_2\phi_2 + \cdots + a_n\phi_n. \tag{10}$$

Then, the kth iterate $x^{(k)}$ is

$$x^{(k)} = H^k x^{(0)} = a_1\phi_1 + a_2\sigma_2^k\phi_2 + \cdots + a_n\sigma_n^k\phi_n. \tag{11}$$

We first have the following simple result on the convergence rate of $x^{(k)}$.

Theorem 5. *Under the conditions of Corollary 3, assume that H is diagonalizable with eigen-pairs (σ_i, ϕ_i) labeled in nonincreasing order of the magnitudes of the eigenvalues. Then the iterate $x^{(k)}$ approaches the limit in the order $O(|\sigma_2|^k)$, and the quantity $s_{ij}^{(k)}$ defined in* (1) *approaches zero in the same order.*

The algebraic distance is now expressed as

$$s_{ij}^{(k)} = \left|(e_i - e_j)^T x^{(k)}\right| = \left|(e_i - e_j)^T \sum_{\ell=2}^{n} a_\ell \sigma_\ell^k \phi_\ell\right|,$$

where e_i is the ith column of the identity matrix. Note that the summation starts from $\ell = 2$. Since σ_2^k is a common factor for all the (i, j) pairs, we define the quantity $\hat{s}_{ij}^{(k)}$ as

$$s_{ij}^{(k)} / \sigma_2^k \tag{12}$$

and are interested in its behavior as k increases. It turns out that, in contrast to $s_{ij}^{(k)}$, the scaled quantity $\hat{s}_{ij}^{(k)}$ does not always converge. When it does, however, it converges to some interesting value other than zero. We consider two cases:

(i) $\sigma_2 = \sigma_3 = \cdots = \sigma_t$, and $|\sigma_t| > |\sigma_{t+1}|$, for some $t \geq 2$. This case implies that $\sigma_2, \ldots, \sigma_t$ are all real, since otherwise the complex conjugate of σ_2 is also an eigenvalue of H. In practice, this case occurs for most real-life graphs, with $t = 2$; in other words, σ_2 is real and $|\sigma_2| > |\sigma_3|$.

(ii) $|\sigma_2| = |\sigma_3| = \cdots = |\sigma_t| > |\sigma_{t+1}|$ for some $t \geq 3$, and $\sigma_2, \ldots, \sigma_t$ are not all the same. In this case, $\sigma_2, \ldots, \sigma_t$ are not necessarily all reals. The iteration matrices H_{GS} and H_{SOR} for random graphs frequently yield $|\sigma_2| = |\sigma_3| > |\sigma_4|$, where σ_2 and σ_3 are conjugate complex numbers.

Theorem 6. *Under the conditions of Theorem 5, let the initial vector $x^{(0)}$ be expanded in the eigenbasis of H as in* (10).

(*i*) *If $\sigma_2 = \sigma_3 = \cdots = \sigma_t$ and $|\sigma_t| > |\sigma_{t+1}|$ for some $t \geq 2$, and if $a_2, \ldots, a_t$ are not all zero, then the quantity $\hat{s}_{ij}^{(k)}$ defined in* (12) *approaches the limit $\left|(e_i - e_j)^T \xi\right|$ in the order $O\left(|\sigma_{t+1}/\sigma_t|^k\right)$, where ξ is an eigenvector corresponding to the eigenvalue σ_2 (with multiplicity $t-1$).*

(*ii*) *If $|\sigma_2| = |\sigma_3| = \cdots = |\sigma_t| > |\sigma_{t+1}|$ for some $t \geq 3$, where $\sigma_2, \ldots, \sigma_t$ are not all the same, $a_2, \ldots, a_t$ are not all zero, and if there exists an integer m such that $(\sigma_\ell/\sigma_2)^m = 1$ for $\ell = 3, \ldots, t$, then the pth subsequence $\{\hat{s}_{ij}^{(mk+p)}\}_{k=0,1,2,\ldots}$ approaches the limit $\left|(e_i - e_j)^T \eta_p\right|$ in the order $O\left(|\sigma_{t+1}/\sigma_t|^{mk}\right)$, where $\eta_p = a_2\phi_2 + a_3(\sigma_3/\sigma_2)^p\phi_3 + \cdots + a_t(\sigma_t/\sigma_2)^p\phi_t$ for $p = 0, 1, \ldots, (m-1)$.*

Proof. Case (i): Equation (12) becomes

$$\hat{s}_{ij}^{(k)} = \left|(e_i - e_j)^T \xi^{(k)}\right|,$$

where

$$\xi^{(k)} = a_2\phi_2 + \cdots + a_t\phi_t + \sum_{\ell=t+1}^{n} a_\ell \left(\frac{\sigma_\ell}{\sigma_2}\right)^k \phi_\ell.$$

When k tends to infinity, the summation term in $\xi^{(k)}$ vanishes. Let $\xi = a_2\phi_2 + \cdots + a_t\phi_t$, then $\lim_{k\to\infty} \hat{s}_{ij}^{(k)} = \left|(e_i - e_j)^T \xi\right|$, and $\hat{s}_{ij}^{(k)}$ converges in the order $O(|\sigma_{t+1}/\sigma_2|^k) = O(|\sigma_{t+1}/\sigma_t|^k)$.

Case (ii): Equation (12) becomes

$$\hat{s}_{ij}^{(k)} = \left|(e_i - e_j)^T \eta^{(k)}\right|,$$

where

$$\eta^{(k)} = a_2\phi_2 + a_3\tau_3^k\phi_3 + \cdots + a_t\tau_t^k\phi_t + \sum_{\ell=t+1}^{n} a_\ell \left(\frac{\sigma_\ell}{\sigma_2}\right)^k \phi_\ell,$$

and $\tau_\ell = \sigma_\ell/\sigma_2$ for $\ell = 3, \ldots, t$. If there exists a positive integer m such that $\tau_\ell^m = 1$ for all ℓ, then $\eta^{(k)}$ has m limit points $\eta_0, \ldots, \eta_{m-1}$. □

Remark 1. Case (ii) of the above theorem indicates that the quantity $\hat{s}_{ij}^{(k)}$ oscillates at the "limit". However, we may not often have a positive integer m such that $\tau_\ell^m = 1$ for $\ell = 3, \ldots, t$. Thus, we consider a case where $\tau_\ell^m = e^{i\epsilon_\ell}$ for a small $|\epsilon_\ell|$ for each ℓ (here e is the base of the natural logarithm and i is the imaginary unit). In such a case, we still can derive some similar results indicating oscillations. Note that when k is large enough, the summation term in $\eta^{(k)}$ is negligible. Therefore,

$$\begin{aligned}
\left|\hat{s}_{ij}^{(k)} - \hat{s}_{ij}^{(k+m)}\right| &= \left|\left|(e_i - e_j)^T \eta^{(k)}\right| - \left|(e_i - e_j)^T \eta^{(k+m)}\right|\right| \\
&\leq \left|(e_i - e_j)^T (\eta^{(k)} - \eta^{(k+m)})\right| \\
&\approx \left|(e_i - e_j)^T [a_3\tau_3^k(1 - e^{i\epsilon_3})\phi_3 + \cdots + a_t\tau_t^k(1 - e^{i\epsilon_t})\phi_t]\right| \\
&\leq \sum_{\ell=3}^{t} \left|1 - e^{i\epsilon_\ell}\right| \cdot \left|(e_i - e_j)^T (a_\ell\phi_\ell)\right|
\end{aligned}$$

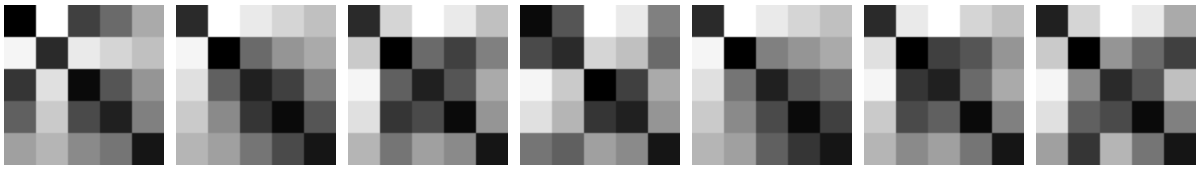
$$\leq \sum_{\ell=3}^{t} |\epsilon_\ell| \cdot \left|(e_i - e_j)^T (a_\ell \phi_\ell)\right| .$$

The above bound means that when k is large enough, the value of $\hat{s}_{ij}^{(k)}$ will be close to the one in m iterations later. In other words, when k tends to infinity, $\hat{s}_{ij}^{(k)}$ approximately oscillates with a period m, similar to the case where we have an m such that $\tau_\ell^m = 1$ for all ℓ (case (ii) of the theorem).

The following example shows such an oscillation. The weighted adjacency matrix is

$$W = \begin{bmatrix} 0 & 0.8235 & 0.6948 & 0.3171 & 0.9502 \\ 0.8235 & 0 & 0.3816 & 0.7655 & 0.7952 \\ 0.6948 & 0.3816 & 0 & 0.6463 & 0.7094 \\ 0.3171 & 0.7655 & 0.6463 & 0 & 0.1626 \\ 0.9502 & 0.7952 & 0.7094 & 0.1626 & 0 \end{bmatrix} .$$

The second and third eigenvalues of H_{GS} are $\sigma_2 = -0.0501 + 0.0637i$ and $\sigma_3 = -0.0501 - 0.0637i$. Thus, $\tau_3 = \sigma_3/\sigma_2 = -0.2348 + 0.9721i$, and $\tau_3^7 = e^{0.0881i}$. Hence, $\epsilon_3 = 0.0881$ which can be considered small enough. Therefore, it is expected that the quantity $\hat{s}_{ij}^{(k)}$ approximately oscillates with a period 7. Figure 1 shows a visualization of the matrix $[\hat{s}_{ij}^{(k)}]$ as k increases. Note the repetition of the appearance of the matrix every 7 iterations.



(a) $k = 10$ (b) $k = 11$ (c) $k = 12$ (d) $k = 13$ (e) $k = 14$ (f) $k = 15$ (g) $k = 16$

(h) $k = 17$ (i) $k = 18$ (j) $k = 19$ (k) $k = 20$ (l) $k = 21$ (m) $k = 22$ (n) $k = 23$

Figure 1: Visualization of the matrix $[\hat{s}_{ij}^{(k)}]$. Dark color means small value and bright color means large value.

4 Algebraic Distances and Jacobi Overrelaxations

In this section, we discuss the implications of the theorems in the preceding section for the JOR iterations. Specifically, the eigenvalues and vectors of the iteration matrix H_{JOR} are closely related to those of the matrix pencil (L, D). Further, because of the distributions of the eigenvalues, the convergence of Algorithm 1 often is slow. Hence, we also study the behavior of the iterations at an early stage.

4.1 Algebraic Distances at the Limit

An immediate result is that H_{JOR} is diagonalizable and all the eigenvalues of H_{JOR} are real, since

$$H_{JOR}\ \phi_i = \sigma_i \phi_i \iff L\phi_i = \frac{1-\sigma_i}{\omega} D\phi_i.$$

This equivalence implies that if μ_j is an eigenvalue of (L, D), then $\mu_j = (1-\sigma_i)/\omega$ for some i. In general, we may not have $\mu_i = (1-\sigma_i)/\omega$ for all i, since the eigenvalues of H_{JOR} are sorted in the order of their absolute values, whereas the eigenvalues of (L, D) are sorted in their natural order. Figure 2 pictorially shows the relative positions of the eigenvalues σ_i of H_{JOR}. It is clear that $\sigma_1 = 1 - \omega\mu_1 = 1$ regardless of the value ω, since $\mu_1 = 0$. However, σ_2 can be either $1 - \omega\mu_2$ or $1 - \omega\mu_n$, depending on the value of ω. A special case is that $1 - \omega\mu_2 = -(1 - \omega\mu_n)$. In this case, $\sigma_2 = -\sigma_3$, and case (ii) of Theorem 6 indicates that the scaled algebraic distance $\hat{s}_{ij}^{(k)}$ will not converge; rather, it oscillates when k is large. Otherwise, we enumerate all the other possible cases for ω, and we have the following theorem as a corollary of case (i) of Theorem 6.

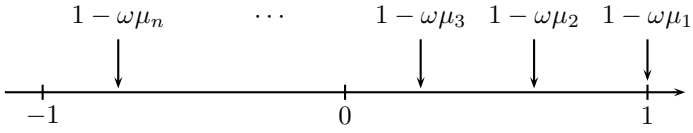


Figure 2: Eigenvalues of H_{JOR}.

Theorem 7. *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, 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 (12) will always converge to a limit $|(e_i - e_j)^T \xi|$ in the order $O(\theta^k)$, for some ξ and $0 < \theta < 1$.*

(i) If $0 < \omega < 2/(\mu_3 + \mu_n)$, then

$$\xi \in \text{span}\{\hat{v}_2\} \qquad \textit{and} \qquad \theta = \frac{1-\omega\mu_3}{1-\omega\mu_2};$$

(ii) If $2/(\mu_3 + \mu_n) \leq \omega < 2/(\mu_2 + \mu_n)$, then

$$\xi \in \text{span}\{\hat{v}_2\} \qquad \textit{and} \qquad \theta = -\frac{1-\omega\mu_n}{1-\omega\mu_2};$$

(iii) If $2/(\mu_2 + \mu_n) < \omega < \min\{2/(\mu_2 + \mu_{n-1}), 2/\mu_n\}$, then

$$\xi \in \text{span}\{\hat{v}_n\} \qquad \textit{and} \qquad \theta = -\frac{1-\omega\mu_2}{1-\omega\mu_n};$$

(iv) If $2/(\mu_2 + \mu_{n-1}) \leq \omega < 2/\mu_n$, then

$$\xi \in \text{span}\{\hat{v}_n\} \qquad \textit{and} \qquad \theta = \frac{1-\omega\mu_{n-1}}{1-\omega\mu_n}.$$

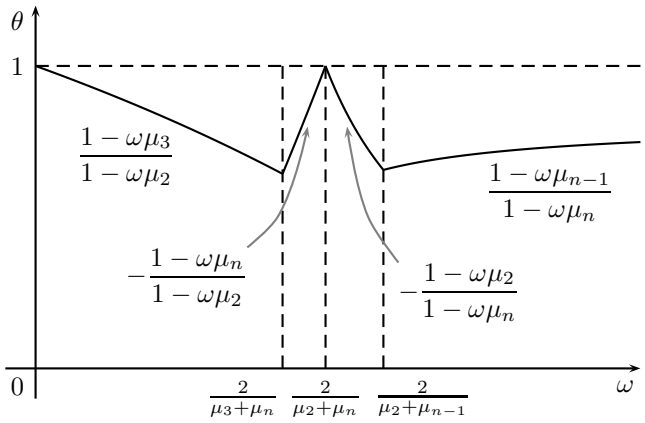


Figure 3: The θ as a function of ω. Note that the value $2/\mu_n$ can be less than, equal to, or greater than $2/(\mu_2 + \mu_{n-1})$.

A graphical illustration of the dependence of θ on ω is shown in Figure 3.

Proof. In case (i), $-(1-\omega\mu_n) < (1-\omega\mu_3) < (1-\omega\mu_2)$; therefore

$$\sigma_2 = 1-\omega\mu_2 > 0, \qquad \sigma_3 = 1-\omega\mu_3 > 0.$$

In case (ii), $(1-\omega\mu_3) < -(1-\omega\mu_n) < (1-\omega\mu_2)$; therefore

$$\sigma_2 = 1-\omega\mu_2 > 0, \qquad \sigma_3 = 1-\omega\mu_n < 0.$$

In case (iii), $-(1-\omega\mu_{n-1}) < (1-\omega\mu_2) < -(1-\omega\mu_n)$; therefore

$$\sigma_2 = 1-\omega\mu_n < 0, \qquad \sigma_3 = 1-\omega\mu_2 > 0.$$

In case (iv), $(1-\omega\mu_2) < -(1-\omega\mu_{n-1}) < -(1-\omega\mu_n)$; therefore

$$\sigma_2 = 1-\omega\mu_n < 0, \qquad \sigma_3 = 1-\omega\mu_{n-1} < 0.$$

The theorem is established by following case (i) of Theorem 6. □

Remark 2. Sometimes $\mu_2 = \mu_3$ or $\mu_{n-1} = \mu_n$ for graphs from real-life problems. Thus, Theorem 7 does not apply. However, we can use the same technique as in the proof to analyze the convergence of $\hat{s}_{ij}^{(k)}$, by checking the possible values of σ_2 and σ_3.

Theorem 7 shows two possible limits for $\hat{s}_{ij}^{(k)}$ depending on the value of ω, which in turn depends on the relative sizes of the eigenvalues μ_i. In practice, the eigenvalues 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 7 is applied. A simple bound exploits the fact that $\mu_2 \leq \mu_n \leq 2$; thus

$$2/(\mu_2 + \mu_n) \geq 1/2,$$

which indicates that for any $\omega < 1/2$, the quantity $\hat{s}_{ij}^{(k)}$ will converge to some value proportional to $|(e_i - e_j)^T \hat{v}_2|$. A slightly better bound is

$$2/(\mu_2 + \mu_n) \geq 2/3,$$

for a graph that is not complete, since in such a case $\mu_2 \leq 1$. For more sharper bounds of μ_2 and μ_n, see, for example, [5]. Since in practice we deal with sparse graphs and set $\omega = 1/2$, the quantity $\hat{s}_{ij}^{(k)}$ always converges to $|(e_i - e_j)^T \xi|$ with $\xi \in \text{span}\{\hat{v}_2\}$.

4.2 Algebraic Distances at Early Iterations

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 steps 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.

To make the above statement precise, we want to measure the angle between two consecutive iterates. Specifically, we are interested in how close

$$1 - \left\langle \frac{x^{(k)}}{\left\|x^{(k)}\right\|}, \frac{x^{(k+1)}}{\left\|x^{(k+1)}\right\|} \right\rangle^2$$

is to zero (this is the squared sine of the angle between $x^{(k)}$ and $x^{(k+1)}$). Note that even though $x^{(k)}$ and $x^{(k+1)}$ become close to parallel at early iterations, it does not necessarily mean that $x^{(k)}$ has converged, nor has the quantity $s_{ij}^{(k)}$ or $\hat{s}_{ij}^{(k)}$.

Theorem 8. *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:*

$$1 - \omega\mu_n \geq 0, \tag{13a}$$

$$f_k := \frac{\alpha {r_k}^{2k} (1 - r_k)^2}{1 + \alpha {r_k}^{2k} (1 + r_k)^2} \leq \frac{1}{\kappa}, \tag{13b}$$

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

$$2\alpha r^{2k+2} + 2\alpha r^{2k+1} + (k+1)r - k = 0 \tag{14}$$

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

$$1 - \left\langle \frac{x^{(k)}}{\left\|x^{(k)}\right\|}, \frac{x^{(k+1)}}{\left\|x^{(k+1)}\right\|} \right\rangle^2 \leq \frac{4\kappa f_k}{(1 + \kappa f_k)^2}. \tag{15}$$

Since the proof of the above theorem is long and technical, it is deferred to the Appendix. We note several important issues in this theorem. First, since we use $\omega = 1/2$, condition (13a) is satisfied. Second, f_k is defined as a rational polynomial of r_k, which is the unique root of the polynomial (14) 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.[3] All this means is that condition (13b) is not a strong requirement. The final bound in (15), for $k = 30$ or 50, typically drops to the order of 10^{-4}. Note that $\sin^2(\pi/180) = 3.05 \times 10^{-4}$.

Of course, not every graph with an arbitrary initialization will yield $x^{(k)}$ close to parallel to $x^{(k+1)}$ for a small k. Hence, some assumptions, such as these in Theorem 8, are needed. On closing this subsection, we present a "bad" example. In this example, if the relaxation parameter ω is not small, then for any k we can construct a corresponding initialization such that $x^{(k)}$ and $x^{(k+1)}$ are far from parallel.

Example 1. Consider a graph with only two vertices and an edge between them. The JOR iteration matrix for this graph is

$$H_{JOR} = \begin{bmatrix} 1-\omega & \omega \\ \omega & 1-\omega \end{bmatrix}.$$

Its two eigenvalues are $\sigma_1 = 1$ and $\sigma_2 = 1 - 2\omega$. Given an initialization

$$x^{(0)} = \begin{bmatrix} 1+\delta \\ -1+\delta \end{bmatrix},$$

the iterations generate the iterates

$$x^{(k)} = \begin{bmatrix} +(1-2\omega)^k + \delta \\ -(1-2\omega)^k + \delta \end{bmatrix} = \begin{bmatrix} +\sigma_2^k + \delta \\ -\sigma_2^k + \delta \end{bmatrix}.$$

Then,

$$1 - \left\langle \frac{x^{(k)}}{\left\|x^{(k)}\right\|}, \frac{x^{(k+1)}}{\left\|x^{(k+1)}\right\|} \right\rangle^2 = \frac{(1-\sigma_2)^2}{1 + \sigma_2^2 + \left(\dfrac{\sigma_2^{2k+2}}{\delta^2} + \dfrac{\delta^2}{\sigma_2^{2k}} \right)}.$$

If we choose

$$\delta = \sigma_2^{k+0.5} = (1-2\omega)^{k+0.5},$$

then the above equation becomes

$$1 - \left\langle \frac{x^{(k)}}{\left\|x^{(k)}\right\|}, \frac{x^{(k+1)}}{\left\|x^{(k+1)}\right\|} \right\rangle^2 = \frac{(1-\sigma_2)^2}{(1+\sigma_2)^2} = \left(\frac{\omega}{1-\omega} \right)^2.$$

When ω approaches 1/2, the angle between $x^{(k)}$ and $x^{(k+1)}$ can be arbitrarily close to 90°. In other words, when the relaxation parameter ω is not small, for any k, we can choose an appropriate δ such that $x^{(k)}$ and $x^{(k+1)}$ are far from parallel. An example set of such parameters is

$$\omega = 1/8, \qquad k = 20, \qquad \delta = 0.75^{20.5} \approx 0.0027.$$

Hence, the angle between $x^{(20)}$ and $x^{(21)}$ is 8.21°.

[3] This may not be true for power-law graphs.

5 Mutually Reinforcing Model

We have defined the algebraic distance $s_{ij}^{(k)}$ based on an iterative process and have proved 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 that incorporates the local structure of the graph for quantitatively evaluating the vertex connectivity. We use the vectors $\hat{v}_2$ and $x^{(k)}$, as an (approximate) solution to the model, to explain that the algebraic distance defined in this way well agrees with the model.

Consider a mutually reinforcing environment, where entities are influenced by their neighbors. Intuitively, for an abstract property that is characteristic in such an environment, a part of the property value for an entity should be a weighted average of the influences from its neighbors in some way. Two entities are said to be close, or similar, if they are placed in two similar environments, or consequently, their property values are close. If we consider the graph itself as an integral environment and vertices as individual entities each of which is surrounded by a neighborhood (the neighboring vertices), then two vertices have a strong connection if they have similar values for an afore mentioned abstract property. Let each vertex i be associated with a real number x_i. Except for a μ portion of itself, i is influenced by its neighbors, which is quantitatively a weighted average:

$$x_i = \mu x_i + \sum_{j \sim i} p_{ij} x_j, \tag{16}$$

where $j \sim i$ means j is a neighbor of i. Here, the portion $0 \leq \mu \leq 1$ 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 internal property of the entire environment (i.e., the graph). For such a mutually reinforcing environment, a small μ is more desired. The weight of the influence by a neighbor j, p_{ij}, should be nonnegative, and all the p_{ij} related to the same i should sum to 1. The weight p_{ij} reflects how strongly a neighbor can influence i, and therefore a natural choice is $p_{ij} = w_{ij} / \sum_j w_{ij}$. Thus, (16) is formally written in the following form

$$x_i = \mu x_i + \sum_j \frac{w_{ij}}{d_{ii}} x_j \qquad 0 \leq \mu \leq 1. \tag{17}$$

The coupling of two vertices i and j is measured by $|x_i - x_j|$. A small value means a strong connection, which equivalently means that their neighborhoods have a similar influence on the two vertices.

In the matrix form, (17) becomes

$$x = \mu x + D^{-1} W x,$$

which is equivalent to $Lx = \mu D x$. Clearly, x is an eigenvector of the matrix pencil (L, D), and μ is its corresponding eigenvalue. Unless the graph is complete, we have at least two sets of x and μ that satisfy this system. In the first set, μ is zero, and x is a scalar multiple of $\mathbf{1}$. In this case, since μ is 0, the value of each vertex is entirely determined by its neighbors. This would have been the most desirable situation for a mutually reinforcing environment because it means that every entity is influenced only by its neighborhood. However, this situation leads to the result that every

entity is the same (x_i constant for all i), and therefore no discriminating power is presented. In the second set, μ is equal to μ_2, the second smallest eigenvalue of (L, D), and $x = \hat{v}_2$. When the graph is not complete, $\mu_2 \leq 1$. Indeed, frequently μ_2 is close to zero in practice. This is naturally a desirable solution for our problem. The neighborhood has a strong impact on a vertex, and vertices have different values such that the strengths of the connectivity for different vertex pairs can be distinguished.

Algorithm 1 computes the above solution. According to Theorem 7, in the limit, the scaled algebraic distance $\hat{s}_{ij}^{(k)}$ converges to a value proportional to $\left|(e_i - e_j)^T \hat{v}_2\right|$. Regardless of the fact that the iterate $x^{(k)}$ converges to a scalar multiple of $\mathbf{1}$, we are actually interested in the entry differences of $x^{(k)}$, and so it is legitimate for us to say informally that $x^{(k)}$ "converges" to $\hat{v}_2$. On the other hand, even when far from convergence, Theorem 8 indicates that the (normalized) iterate $x^{(k)}$ approximately satisfies the model:

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

since consecutive iterates are close to parallel. In a word, from either point of view, Algorithm 1 computes an approximate solution to the model (17).

We remark that for small k, the iterate $x^{(k)}$ can be quite different from its "limit" $\hat{v}_2$, and for different initializations, $x^{(k)}$ will be different. However, they all satisfy (or approximately satisfy) the mutually reinforcing model (17). This feature gives us the flexibility, yet not the arbitrariness, to estimate the connectivity for different vertex pairs. Readers may question why an iterate $x^{(k)}$ is preferred over the eigenvector $\hat{v}_2$ as the measure. A major reason is that the JOR method with a few iterations is computationally much less expensive than computing an eigenvector, even when the matrix L is sparse. Solving a large-scale sparse eigenvalue problem for a symmetric matrix, say, by using the Lanczos method [19, 22, 14], involves frequent convergence tests, each of which needs to solve an eigenvalue subproblem for a tridiagonal matrix. On the other hand, inside each JOR iteration is nothing but weighted averages. Therefore, the JOR process is particularly inexpensive compared with computing an eigenvector. Besides, the simplicity of Algorithm 1 makes it particularly attractive, and thus it is advocated as the preferred algorithm for the proposed measure in this paper.

6 Applications

Before demonstrating the use of algebraic distances in applications, we first show a simple experiment to help one understand how the algebraic distances represent the local connectivity and neighborhoods. In this experiment we used two unweighted graphs generated from finite-element instances (airfoil and 3elt [7]). For every edge ij we define the quantity l_{ij}, which is the length of a shortest path between i and j if ij is deleted. Intuitively, an edge is "critical" if l_{ij} is large, since by the removal of this edge, i and j have to be connected by a long path. In the experiment, we extended the graph by creating 10% of additional random unweighted edges, where the new edges all satisfied $2 < l_{ij} < 11$. The goal of the experiment was to check whether the algebraic distance would give some indications on noisy random edges on well structured instances.

The results of the extended 2-normed algebraic distance with 15 iterations ($\varrho_{ij}^{(15)}$) calculated for all edges are presented in Figure 4(a) and 4(b). In both figures, each tick on the x-axis corresponds to one edge, and the edges are ordered from the longest algebraic distance to the shortest. The

respective y-coordinate of every point (i.e., edge ij) is l_{ij}. Clearly, almost all edges with $l_{ij} \neq 2$ are concentrated at the very left of the figures. This result shows that random/noise edges in general have a connection strength much weaker than those of the original edges. The example shows that the algebraic distance can serve as a separator between strongly and weakly connected pairs of vertices in well structured instances that have good local connectivity.

To demonstrate the practical potential of the algebraic distance measure, we show four applications: maximum weighted matching, maximum independent set, and minimum τ-partitioning of graphs and of hypergraphs. In all these cases, existing fast and widely used in practice baseline algorithms were modified by taking into account quantities based on the algebraic distances instead of the original graph edge weights. The experimental graphs had different sizes ($10^3 < |E| < 10^7$) and structures and were selected from the UF sparse matrix collection [7] based on their nontriviality for several state-of-the-art solvers for NP-hard optimization problems. The algebraic distances for graph partitioning, maximum matching, and maximum independent set problems were calculated by using the standard JOR relaxation with $\omega = 1/2$. For the hypergraph partitioning problem, we extended the definition of algebraic distance for hypergraphs and demonstrate the results for both GS and JOR relaxations.

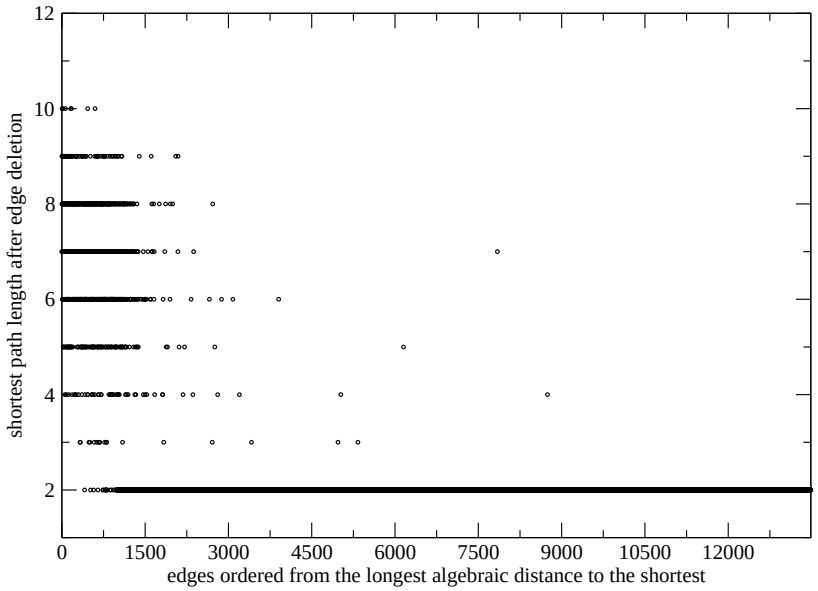
The implementation of stationary iterative processes and their running time are well studied issues. These topics are out of scope of this paper and we refer the reader to the following books in which one can find the discussions about sequential and parallel matrix-vector multiplications and general relaxations [15, 16]

6.1 Maximum Weighted Matching

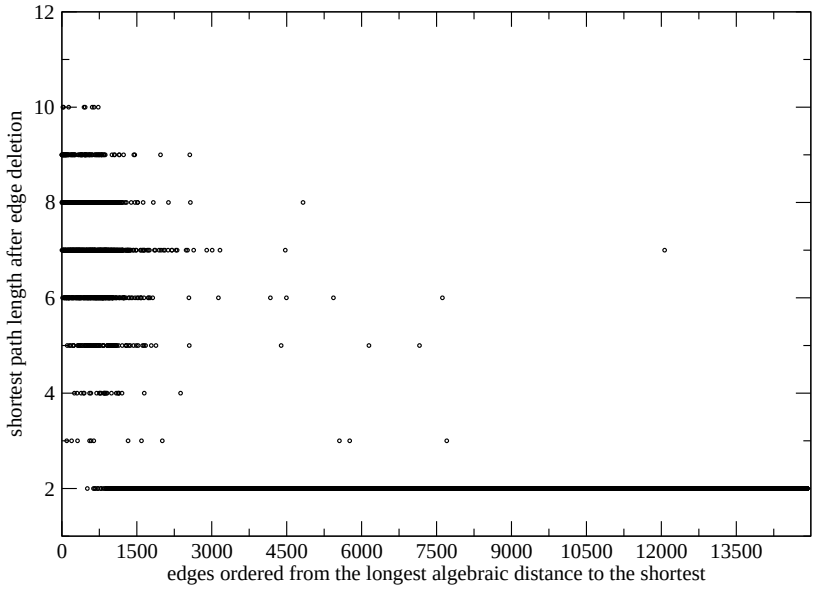
A matching M of a graph is a subset of the graph edges such that no vertex is incident to more than one edge in M. A matching M is said to be *maximum* if $|M| \geq |M'|$ for any other matching M'. Similarly, a matching M is said to be *maximum weighted* if $w(M) \geq w(M')$ for any other matching M', where $w(M) = \sum_{ij \in M} w_{ij}$.

Although the maximum weighted matching problem admits a polynomial time solution, often two well-known 2-approximation methods, which have a linear time complexity (without sorting), are used in practical applications. One is a textbook greedy algorithm, which successively adds a next legal heaviest edge to the existing matching; the other is an improved version of the greedy algorithm, based on a path-growing principle. Both algorithms are presented in [9].

In both algorithms, there exists a greedy step in which the next heaviest edge (one that has the largest weight) has to be chosen. The criterion for choosing an edge was changed according to the following heuristic observation: a better matching can be found in a dense graph with less effort than in a sparse graph. Hence, we give preference to matching two nodes that are not well connected with other nodes from their neighborhood, to give a chance to the less connected nodes to participate in the matching. We define a value s'_{ij} for each graph edge ij based on the extended algebraic distance $\varrho_{ij}^{(k)}$ and use it to replace the edge weight w_{ij} in greedy choice steps (and sorting if applicable). The quantity s'_{ij} has a meaning opposite that of the edge weight: A small value means weak connection. Hence, in the greedy steps, instead of choosing an edge with the largest weight, we choose one that has the smallest s'_{ij}. A heuristic for defining such a quantity s'_{ij} is first to define for each vertex a quantity $a_i = \sum_{ij \in E} 1/\varrho_{ij}^{(k)}$ that captures the connectivity between this vertex and its neighborhood. Then, s'_{ij} is the weighted average between a_i and a_j. The first three lines of Algorithm 2 clearly show the computation of this quantity.



(a) airfoil



(b) 3elt

Figure 4: Algebraic distances of the noisy edges.

Algorithm 2 Greedy maximum weighted matching with algebraic distance preprocessing

For all edges $ij \in E$ calculate $\varrho_{ij}^{(k)}$ for some k, R and p. ▷ Preprocessing
For all nodes $i \in V$ compute $a_i = \sum_{ij \in E} 1/\varrho_{ij}^{(k)}$.
For all edges $ij \in E$ compute $s'_{ij} = a_i/\delta_i + a_j/\delta_j$, where δ_i is the degree of i.
$M \leftarrow \emptyset$. ▷ Start of greedy algorithm
while $E \neq \emptyset$ **do**
    $e \leftarrow$ lightest edge in E (edge with smallest s'_{ij}).
    Add e to M.
    Remove e and all its incident edges from E.
end while
return M

The experimental results are presented in Figure 5, which shows for each graph the ratio of the sizes of the matchings between our algorithm (a greedy matching with algebraic distance preprocessing) and the textbook greedy matching (without preprocessing). All ratios were higher than 1, which indicated that our algorithm yielded a better matching than does the baseline greedy algorithm. Almost identical results were obtained by improving a greedy path growing algorithm from [9]. These particular results were obtained with $k = 20$, $R = 10$, and $p = 1$. However, results of almost the same quality have been obtained with many different combinations of $R \geq 5$, $10 \leq k \leq 100$, and $p = 2, \infty$.

6.2 Maximum Independent Set

An independent set I is a subset of V in which no two vertices are incident. An independent set I is said to be *maximum* if $|I| \geq |I'|$ for any other independent set I'. Finding the maximum independent set (MIS) in a graph is an NP-complete problem [12], and fast and qualitative approximations are of great interest for many applications.

Although many existing approximation algorithms for MIS have been proposed, a popular one in practice is still a textbook greedy algorithm [6]. In this algorithm, the vertices are examined in increasing order of their degrees, and the greedy step consists of choosing the next available vertex that does not contradict the current independent set. We exploit a similar heuristic to the one in maximum weighted matching. That is, for each vertex i, we define a quantity b_i that indicates the connection strength between the vertex itself and its neighborhood. To be precise, a small b_i means a weak connection, and we sort the vertices in the increasing order of b_i. Hence, we give preference to including a vertex that is weakly connected to its neighbors. The first four lines of Algorithm 3 show the computation of this quantity.

The experimental results are presented in Figure 6, which shows for each graph the ratio of the sizes of the independent sets between our algorithm (sorting the vertices according to connection strengths) and the textbook greedy algorithm (sorting the vertices according to vertex degrees). A ratio higher than 1 means that our algorithm computes a larger independent set. As can be seen from the figure, for almost all the graphs our algorithm yielded a better result than did the textbook algorithm.

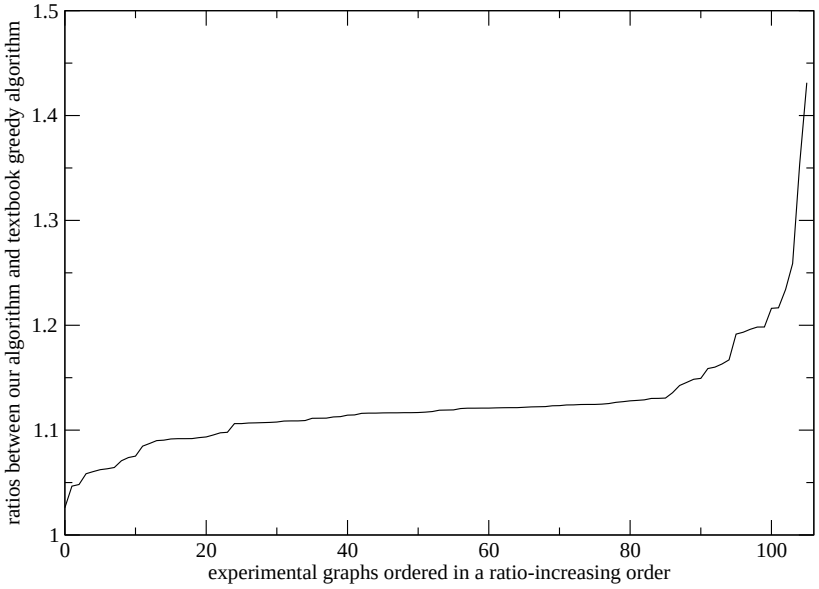


Figure 5: Comparison of greedy algorithms for matching with (Algorithm 2) and without algebraic distance preprocessing. Each point corresponds to the average of ratios between matching weights produced by Algorithm 2 and the textbook greedy algorithm for one graph. The average was calculated over 20 different executions with different random initial vectors. The total number of experimental graphs is 105.

6.3 Graph Partitioning

Graph partitioning is a well known NP-hard problem [11]. The goal is to find a partitioning of V into a family of τ disjoint nonempty subsets $(\pi_p)_{1\le p\le\tau}$, restricted to the following:

$$\begin{aligned} &\text{minimize} \sum_{i\in\pi_p \Rightarrow j\notin\pi_p} w_{ij} \\ &\text{such that } \forall p,\ |\pi_p| \le (1+\alpha)\cdot\frac{|V|}{\tau}\ , \end{aligned} \tag{18}$$

where α is a given *imbalance factor*. In this paper, we consider the 2-partitioning, $\tau = 2$.

HMetis2 [18] is one of the fastest and most successful modern solvers for the partitioning problem. We define its simple extension "HMetis2+ for graphs" in Algorithm 4. In this algorithm we substitute the edge weights with the inverses of algebraic distance $s_{ij}^{(k)}$ and use HMetis2 to produce the 2-partitioning. Figure 7 shows for each graph the ratio of the cut costs (the objective value in (18)) between our extension and the original HMetis2. For most of the graphs our extension yielded a smaller cut, and for the best case the cut cost was reduced by more than 30%.

Indeed, HMetis2 is a multilevel algorithm. A better way to improve HMetis2 and other similar multilevel heuristics is to use the algebraic distances at all levels of the hierarchy, rather than

Algorithm 3 Greedy MIS with algebraic distance preprocessing

For all edges $ij \in E$ calculate $s_{ij}^{(k)}$ for some k. ▷ Preprocessing
For all nodes $i \in V$ compute $a_i = \sum_{ij \in E} 1/s_{ij}^{(k)}$.
For all edges $ij \in E$ compute $s'_{ij} = s_{ij}^{-1}/(a_i + a_j)$.
For all nodes $i \in V$ compute $b_i = \sum_{ij \in E} s'_{ij}$.
(Sorting) Relabel the vertices i in the increasing order of b_i. ▷ Start of greedy algorithm
$I \leftarrow \emptyset$.
for $i = 1, 2, \ldots, n$ **do**
    If $\{i\} \cup I$ is an independent set, add i to I.
end for
return I

substituting the edge weights only at the top level. However, even when we use it as a black-box only, the obtained improvement is quite systematic and non-negligible.

Algorithm 4 HMetis2+ for graphs

For all edges $ij \in E$ calculate $s_{ij}^{(k)}$ for some k (typically 50).
For all edges $ij \in E$ modify the weight $w_{ij} = 1/s_{ij}^{(k)}$.
Produce the graph cut using HMetis2 with modified edge weights.
Return the cut weight with original edge weights

6.4 Hypergraph Partitioning

With encouraging results for the graph problems, the next natural step is to generalize the notion of algebraic distances for hypergraphs. We will give a similar definition for the hyperedges and demonstrate its application for the hypergraph partitioning problem.

Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be a hypergraph, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of hyperedges. Each $h \in \mathcal{E}$ is a subset of $\mathcal{V}$. The hypergraph partitioning problem is a well-known NP-hard problem (see [11] for its graph version). Its goal is to find a partitioning of $\mathcal{V}$ into a family of τ disjoint nonempty subsets $(\pi_p)_{1 \le p \le \tau}$, restricted to the following:

$$\begin{aligned} &\text{minimize} \sum_{\substack{h \in \mathcal{E} \text{ s.t. } \exists i,j \in h \text{ and} \\ i \in \pi_p \Rightarrow j \notin \pi_p}} w_h \\ &\text{such that } \forall p,\ |\pi_p| \le (1+\alpha) \cdot \frac{|V|}{\tau}\ , \end{aligned} \tag{19}$$

where α is a given *imbalance factor*. In this paper, we deal with $\tau = 2$.

While for graphs we use the algebraic distance in place of the weight of an edge, which involves only a pair of vertices, for hypergraphs we have to define similar concepts that can act on a hyperedge, which consists of a subset of vertices. To this end, we consider a bipartite graph model for hypergraphs. We create a bipartite graph $G = (V, E)$ with the vertex set $V = \mathcal{V} \bigcup \mathcal{E}$ and $ih \in E$ if $i \in \mathcal{V}$ appears in hyperedge $h \in \mathcal{E}$. After running k iterations with R random initial vectors on

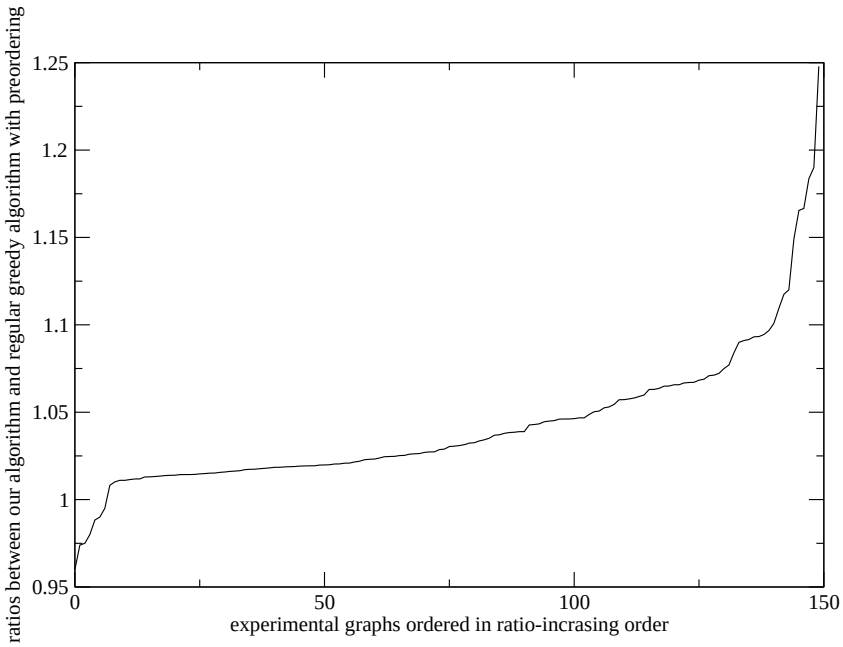


Figure 6: Comparison of greedy algorithms for maximum independent set with (Algorithm 3) and without algebraic distance preprocessing. Each point corresponds to the average of ratios between set sizes produced by Algorithm 3 and the simple textbook greedy algorithm for one graph. The average was calculated over 20 different executions with different random initial vectors. The total number of experimental graphs is 150.

this bipartite model of $\mathcal{H}$, we define $\varrho_S^{(k)}$, the *extended p-normed algebraic distance for a subset of nodes S in $\mathcal{H}$*, as

$$\left(\sum_{r=1}^{R} \max_{i,j \in S} \left| x_i^{(k,r)} - x_j^{(k,r)} \right|^p \right)^{1/p} . \tag{20}$$

We can use any reasonable relaxation process to compute the iterates $x^{(k,r)}$. Here we consider GS and JOR with $\omega = 1/2$.

Using this definition and similarly to the graph partitioning problem, we can design a heuristic for the hypergraph partitioning (see Algorithm 5). First, the algebraic distances for all hyperedges are computed, and, second, the original hyperedge weights are substituted by the inverse of the algebraic distances. In other words, we use a small algebraic distance to replace a heavy hyperedge weight. Finally, we use HMetis2 as a black-box solver and partition the hypergraph with the modified hyperedge weights. We call Algorithm 5 "HMetis2+ for hypergraphs".

Figure 8 shows for each hypergraph the ratio of the cut costs (the objective value in (19)) between our extension and the original HMetis2. For most of the hypergraphs our extension yielded a smaller cut, and for the best case the cut cost was reduced by around 90%. We also see that a GS relaxation process in general yields slightly better results than does the standard

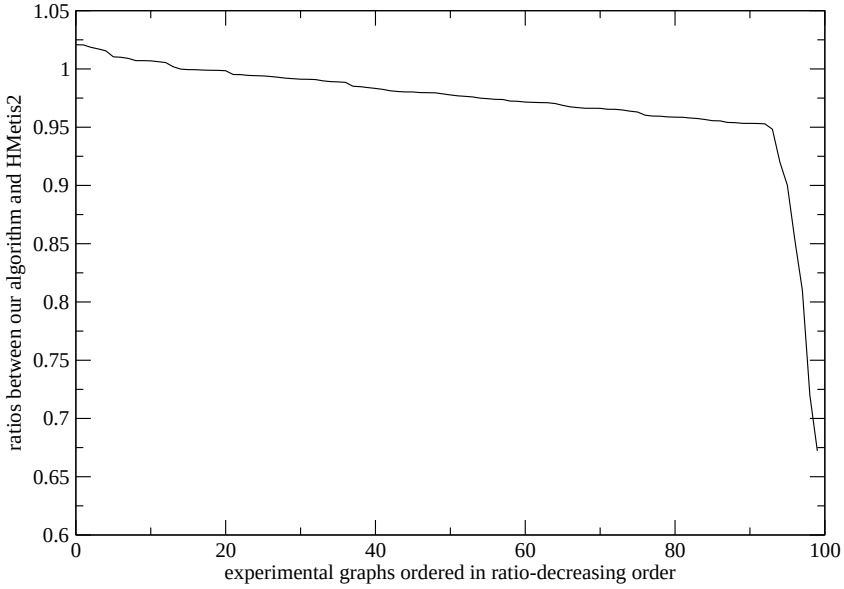


Figure 7: Comparison of Algorithm 4 and HMetis2. Each point corresponds to the average of ratios between cut costs produced by Algorithm 4 and one V-cycle of HMetis2 for one graph. The average was calculated over 20 executions with different random seeds. The total number of experimental graphs is 100.

JOR process. Note that similar to the case of graph partitioning, a more correct way to apply the algebraic distances is to use them at all levels, as was demonstrated in [21]. However, even though we apply them to only the top level, we see here a significant improvement. To the best of our knowledge, this is the first evidence that the multilevel hypergraph partitioning scheme admits such a great improvement.

7 Conclusion

We have presented an iterative process for propagating random initial values on graph nodes through direct neighbors, and defined a notion of algebraic distances between vertices when the process stabilizes. The distances thus defined represent the local connectivity of a pair of vertices; that is,

Algorithm 5 HMetis2+ for hypergraphs

1: For all hyperedges $h \in \mathcal{E}$ calculate $\varrho_h^{(k)}$ for some k.
2: For all hyperedges $h \in \mathcal{E}$ modify the weight $w_h = 1/\varrho_h^{(k)}$.
3: Produce the hypergraph cut using HMetis2 with modified hyperedge weights.
4: Return the cut weight with original hyperedge weights

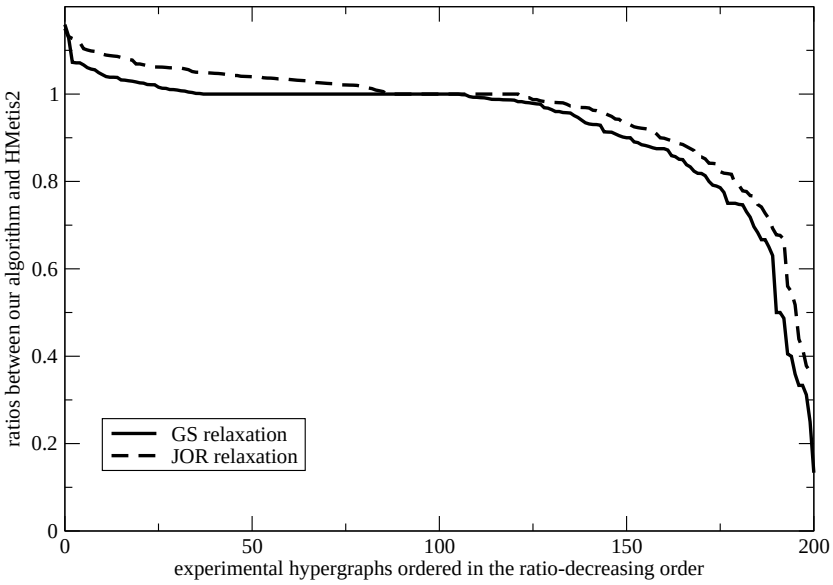


Figure 8: Comparison of Algorithm 5 and HMetis2. Each point corresponds to the average of ratios between cut costs produced by Algorithm 5 and one V-cycle of HMetis2 for one hypergraph. The average was calculated over 20 executions with different random seeds. The total number of experimental hypergraphs is 200 and $k = 50$.

two vertices are strongly connected if their algebraic distance is small. This measure remedies the insufficiency of using solely the edge weights to determine the vertex connection strengths, since neighborhood information is considered.

The proposed iterative process is motivated by the bootstrap AMG method for solving a linear system, where a GS relaxation is used on a positive definite matrix to expose slow-to-converge variables. On the other hand, the process studied in this paper is a JOR relaxation run on the graph Laplacian matrix L, which is only positive semidefinite (i.e., singular). An analysis shows that the algebraic distance between two vertices i and j converges to a value proportional to the difference between the ith and the jth entry of the second smallest eigenvector of the pencil (L, D). Also, it reveals that the convergence is usually slow. However, the changes between consecutive iterations soon become very small, and thus we can run the proposed process using only a few iterations and obtain a good connectivity measure.

By noting that the algebraic distance well represents the local connectivity, we show several applications to demonstrate how it 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, the computation of the algebraic distances occupies only a small fraction of the overall solution time. Thus, its easy parallelization makes it particularly attractive for dealing with large-scale graphs and problems.

A Proof of Theorem 8

The proof requires two lemmas.

Lemma 9. *Define a series of column vectors (indexed by k)*

$$\zeta^{(k)} := [a_1\sigma_1^k,\ a_2\sigma_2^k,\ \ldots,\ a_n\sigma_n^k]^T \in \mathbb{R}^n,$$

where $\sigma_1 = 1$, $0 \le \sigma_i \le 1$ for $i = 2, \ldots, n$, and $a_1 \neq 0$. If

$$f_k := \frac{\alpha {r_k}^{2k}(1-r_k)^2}{1+\alpha {r_k}^{2k}(1+r_k)^2} \le 1,$$

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

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

on the interval $[0,1]$, then

$$1 - \left\langle \frac{\zeta^{(k)}}{\left\|\zeta^{(k)}\right\|}, \frac{\zeta^{(k+1)}}{\left\|\zeta^{(k+1)}\right\|} \right\rangle^2 \le \frac{4f_k}{(1+f_k)^2}. \tag{21}$$

Proof. Let

$$b_i = \frac{1+\sigma_i}{2} \qquad \text{and} \qquad c_i = \frac{1-\sigma_i}{2}.$$

Then,

$$\begin{aligned}
1 - \left\langle \frac{\zeta^{(k)}}{\left\|\zeta^{(k)}\right\|}, \frac{\zeta^{(k+1)}}{\left\|\zeta^{(k+1)}\right\|} \right\rangle^2 &= 1 - \frac{\left(\sum a_i^2\sigma_i^{2k+1}\right)^2}{\left(\sum a_i^2\sigma_i^{2k}\right)\left(\sum a_i^2\sigma_i^{2k+2}\right)} \\
&= 1 - \frac{\left(\sum a_i^2\sigma_i^{2k}(b_i^2 - c_i^2)\right)^2}{\left(\sum a_i^2\sigma_i^{2k}(b_i + c_i)^2\right)\left(\sum a_i^2\sigma_i^{2k}(b_i - c_i)^2\right)} \\
&= \frac{4\left(\sum a_i^2\sigma_i^{2k}b_i^2\right)\left(\sum a_i^2\sigma_i^{2k}c_i^2\right) - 4\left(\sum a_i^2\sigma_i^{2k}b_ic_i\right)^2}{\left(\sum a_i^2\sigma_i^{2k}(b_i^2 + c_i^2)\right)^2 - 4\left(\sum a_i^2\sigma_i^{2k}b_ic_i\right)^2} \\
&\le \frac{4\left(\sum a_i^2\sigma_i^{2k}b_i^2\right)\left(\sum a_i^2\sigma_i^{2k}c_i^2\right)}{\left(\sum a_i^2\sigma_i^{2k}(b_i^2 + c_i^2)\right)^2}. \qquad (*)
\end{aligned}$$

Denote $t = \left(\sum a_i^2\sigma_i^{2k}c_i^2\right) / \left(\sum a_i^2\sigma_i^{2k}b_i^2\right)$, then,

$$(*) = \frac{4}{(1+t)(1+1/t)}.$$

We would like to find an upper bound for t.

Note that $\sigma_1 = 1$; it is not hard to see that t achieves maximum only when $\sigma_2 = \sigma_3 = \cdots = \sigma_n$. Let them all be equal to r. Then t becomes

$$\frac{\left(\sum_{i\neq 1} a_i^2\right) r^{2k}(1-r)^2}{4a_1^2 + \left(\sum_{i\neq 1} a_i^2\right) r^{2k}(1+r)^2} =: f_k(r).$$

The stationary points of $f_k(r)$ satisfy the first-order condition:

$$2\alpha r^{2k+1}(1+r) = k - (k+1)r.$$

In this expression, the left-hand side is a monotonically increasing function of r, and the right-hand side is a monotonically decreasing function for any $k \geq 1$. Therefore, it's not hard to see that the above expression has a unique root on the interval $[0, 1]$. Compared with the boundaries, we conclude that this root is the global maximum of $f_k(r)$.

Thus, $t \leq f_k \leq 1$, and therefore

$$(*) \leq \frac{4}{(1+f_k)\,(1+1/f_k)} = \frac{4f_k}{(1+f_k)^2}. \qquad \square$$

Lemma 10. *Let the squared sine of the angle between two unit vectors $x, y \in \mathbb{R}^n$ be*

$$1 - \langle x, y \rangle^2 = \epsilon,$$

and let a diagonal matrix $D \in \mathbb{R}^{n \times n}$ have condition number κ. If ϵ and κ satisfy

$$\kappa^2 \left(\frac{1 - \sqrt{1-\epsilon}}{1 + \sqrt{1-\epsilon}} \right) \leq 1, \tag{22}$$

then

$$1 - \left\langle \frac{Dx}{\|Dx\|}, \frac{Dy}{\|Dy\|} \right\rangle^2 \leq \frac{4}{\left[1 + \kappa^2 \left(\frac{1 - \sqrt{1-\epsilon}}{1 + \sqrt{1-\epsilon}} \right)\right] \left[1 + \frac{1}{\kappa^2} \left(\frac{1 + \sqrt{1-\epsilon}}{1 - \sqrt{1-\epsilon}} \right)\right]}. \tag{23}$$

Proof. Let

$$z = \frac{x+y}{2} \qquad \text{and} \qquad \delta = \frac{x-y}{2}.$$

Then,

$$\begin{aligned}
1 - \left\langle \frac{Dx}{\|Dx\|}, \frac{Dy}{\|Dy\|} \right\rangle^2 &= 1 - \frac{(y^T D^2 x)^2}{(y^T D^2 y)(x^T D^2 x)} \\
&= 1 - \frac{[(z-\delta)^T D^2 (z+\delta)]^2}{[(z-\delta)^T D^2 (z-\delta)][(z+\delta)^T D^2 (z+\delta)]} \\
&= \frac{4(z^T D^2 z)(\delta^T D^2 \delta) - 4(\delta^T D^2 z)^2}{(z^T D^2 z + \delta^T D^2 \delta)^2 - 4(\delta^T D^2 z)^2} \\
&\leq \frac{4(z^T D^2 z)(\delta^T D^2 \delta)}{(z^T D^2 z + \delta^T D^2 \delta)^2}. \qquad (**)
\end{aligned}$$

Denote $t = (\delta^T D^2 \delta)/(z^T D^2 z)$. Then,

$$(**) = \frac{4}{(1+t)(1+1/t)}.$$

If $x^T y \geq 0$, then we have

$$t = \frac{\delta^T D^2 \delta}{z^T D^2 z} \leq \frac{d_{\max}^2 \|\delta\|^2}{d_{\min}^2 \|z\|^2} = \kappa^2 \left(\frac{1 - x^T y}{1 + x^T y} \right) = \kappa^2 \left(\frac{1 - \sqrt{1-\epsilon}}{1 + \sqrt{1-\epsilon}} \right) \leq 1,$$

otherwise

$$t = \frac{\delta^T D^2 \delta}{z^T D^2 z} \geq \frac{d_{\min}^2 \|\delta\|^2}{d_{\max}^2 \|z\|^2} = \frac{1}{\kappa^2}\left(\frac{1-x^T y}{1+x^T y}\right) = \frac{1}{\kappa^2}\left(\frac{1+\sqrt{1-\epsilon}}{1-\sqrt{1-\epsilon}}\right) \geq 1,$$

where $d_{\max}$ and $d_{\min}$ are the maximum and the minimum of the absolute values of the diagonal elements of D, respectively. For both cases, we have

$$(**) = \frac{4}{(1+t)(1+1/t)} \leq \frac{4}{\left[1+\kappa^2\left(\frac{1-\sqrt{1-\epsilon}}{1+\sqrt{1-\epsilon}}\right)\right]\left[1+\frac{1}{\kappa^2}\left(\frac{1+\sqrt{1-\epsilon}}{1-\sqrt{1-\epsilon}}\right)\right]}. \qquad \square$$

Proof of Theorem 8. Since for JOR the eigenvectors of the iteration matrix are the eigenvectors $\hat{v}_i$ of (L, D), we write

$$x^{(k)} = \hat{V}\zeta^{(k)},$$

where $\zeta^{(k)} = [a_1\sigma_1^k, a_2\sigma_2^k, \ldots, a_n\sigma_n^k]^T$. Note that condition (13a) implies that $\sigma_i = 1 - \omega\mu_i$, and all the σ_i's satisfy $0 \leq \sigma_i \leq 1$. Then by Lemma 9,

$$1 - \left\langle \frac{\zeta^{(k)}}{\left\|\zeta^{(k)}\right\|}, \frac{\zeta^{(k+1)}}{\left\|\zeta^{(k+1)}\right\|} \right\rangle^2 \leq \frac{4f_k}{(1+f_k)^2}. \tag{24}$$

To simplify the notations, let $x' = \zeta^{(k)}/\left\|\zeta^{(k)}\right\|$ and $y' = \zeta^{(k+1)}/\left\|\zeta^{(k+1)}\right\|$. Then,

$$1 - \left\langle \frac{x^{(k)}}{\left\|x^{(k)}\right\|}, \frac{x^{(k+1)}}{\left\|x^{(k+1)}\right\|} \right\rangle^2 = 1 - \frac{\left\langle \hat{V}x', \hat{V}y' \right\rangle^2}{\left\|\hat{V}x'\right\|^2 \left\|\hat{V}y'\right\|^2} = 1 - \frac{\left(y'^T \hat{V}^T \hat{V} x'\right)^2}{\left(y'^T \hat{V}^T \hat{V} y'\right)\left(x'^T \hat{V}^T \hat{V} x'\right)}.$$

Let $\hat{V}^T\hat{V}$ have the eigen-decomposition $U^T\Sigma U$, where U is orthogonal and Σ is positive definite diagonal, then

$$1 - \left\langle \frac{x^{(k)}}{\left\|x^{(k)}\right\|}, \frac{x^{(k+1)}}{\left\|x^{(k+1)}\right\|} \right\rangle^2 = 1 - \frac{\left(y'^T U^T \Sigma U x'\right)^2}{\left(y'^T U^T \Sigma U y'\right)\left(x'^T U^T \Sigma U x'\right)} = 1 - \frac{\left\langle \Sigma^{1/2}Ux', \Sigma^{1/2}Uy' \right\rangle^2}{\left\|\Sigma^{1/2}Ux'\right\|^2 \left\|\Sigma^{1/2}Uy'\right\|^2}.$$

Let $x = Ux'$ and $y = Uy'$. Then

$$1 - \left\langle \frac{x^{(k)}}{\left\|x^{(k)}\right\|}, \frac{x^{(k+1)}}{\left\|x^{(k+1)}\right\|} \right\rangle^2 = 1 - \left\langle \frac{\Sigma^{1/2}x}{\left\|\Sigma^{1/2}x\right\|}, \frac{\Sigma^{1/2}y}{\left\|\Sigma^{1/2}y\right\|} \right\rangle^2.$$

Denote the right-hand side of the inequality (24) ϵ_k. We have four facts. First, $\|x\| = \|y\| = 1$. Second,

$$1 - \langle x, y \rangle^2 = 1 - \left\langle Ux', Uy' \right\rangle^2 = 1 - \left\langle x', y' \right\rangle^2 \leq \epsilon_k.$$

Third, by noting that $\hat{V}$ is D-orthogonal, that is, $\hat{V}^T D \hat{V} = I$, we have

$$\kappa^2(\Sigma^{1/2}) = \kappa(\Sigma) = \kappa(\hat{V}^T\hat{V}) = \kappa(\hat{V}\hat{V}^T) = \kappa(D^{-1}) = \kappa.$$

Fourth, since ϵ_k and f_k satisfy the relation

$$\frac{1-\sqrt{1-\epsilon_k}}{1+\sqrt{1-\epsilon_k}} = f_k,$$

we have

$$\kappa^2(\Sigma^{1/2})\left(\frac{1-\sqrt{1-\epsilon_k}}{1+\sqrt{1-\epsilon_k}}\right) = \kappa f_k \leq 1.$$

Therefore, by Lemma 10, we conclude that

$$1 - \left\langle \frac{\Sigma^{1/2}x}{\|\Sigma^{1/2}x\|}, \frac{\Sigma^{1/2}y}{\|\Sigma^{1/2}y\|} \right\rangle^2 \leq \frac{4}{(1+\kappa f_k)(1+1/(\kappa f_k))}. \qquad \square$$

References

[1] A. Brandt. General highly accurate algebraic coarsening. *Electronic Trans. Num. Anal.*, 10:1–20, 2000.

[2] A. Brandt and D. Ron. Multigrid solvers and multilevel optimization strategies. In J. Cong and J. R. Shinnerl, editors, *Multilevel Optimization and VLSICAD*. Kluwer, 2003.

[3] Achi Brandt. Multiscale scientific computation: review 2001. In *Multiscale and multiresolution methods*, volume 20, pages 3–95. Springer Verlag, 2002.

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

[5] Fan R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[6] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. McGraw-Hill, 2nd edition, 2001.

[7] T. Davis. University of Florida sparse matrix collection. *NA Digest*, 97(23), 1997.

[8] Achiya Dax. The convergence of linear stationary iterative processes for solving singular unstructured systems of linear equations. *SIAM Rev.*, 32(4):611–635, 1990.

[9] Doratha E. Drake and Stefan Hougardy. A simple approximation algorithm for the weighted matching problem. *Inf. Process. Lett.*, 85(4):211–213, 2003.

[10] Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco 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):355–369, 2007.

[11] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.

[12] Michael R. Garey and David S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. Freemann And Company, 1979.

[13] Arpita Ghosh, Stephen Boyd, and Amin Saberi. Minimizing effective resistance of a graph. *SIAM Rev.*, 50(1):37–66, 2008.

[14] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

[15] Ananth Grama, George Karypis, Anshul Gupta, and Vipin Kumar. *Introduction to Parallel Computing: Design and Analysis of Algorithms*. Addison-Wesley, 2003.

[16] Michael T. Heath. *Scientific Computing*. McGraw-Hill, Inc., New York, NY, USA, 2002.

[17] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[18] G. Karypis and 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, September 1998.

[19] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Research of the National Bureau of Standards*, 45:255–282, 1950.

[20] Boaz Nadler, Stéphane Lafon, Ronald R. Coifman, and Ioannis G. Kevrekidis. Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators. In *Advances in Neural Information Processing Systems*, volume 18, pages 955–962. MIT Press, 2005.

[21] Dorit Ron, Ilya Safro, and Achi Brandt. Relaxation based coarsening for combinatorial optimization problems. *submitted*, 2009.

[22] Y. Saad. *Numerical Methods for Large Eigenvalue Problems*. Halstead Press, 1992.

[23] Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[24] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press, 1994.