SIAM J. MATH. DATA SCI.
Vol. 1, No. 2, pp. 269–292

A Nonlinear Spectral Method for Core–Periphery Detection in Networks*

Francesco Tudisco† and Desmond J. Higham†

Abstract. We derive and analyze a new iterative algorithm for detecting network core–periphery structure. Using techniques from nonlinear Perron–Frobenius theory, we prove global convergence to the unique solution of a relaxed version of a natural discrete optimization problem. On sparse networks, the cost of each iteration scales linearly with the number of nodes, making the algorithm feasible for large-scale problems. We give an alternative interpretation of the algorithm from the perspective of maximum likelihood reordering of a new logistic core–periphery random graph model. This viewpoint also presents a new basis for quantitatively judging a core–periphery detection algorithm. We illustrate the algorithm on a range of synthetic and real networks and show that it offers advantages over the current state of the art.

Key words. core–periphery, mesoscale structure, networks, nonlinear Perron–Frobenius, nonlinear eigenvalues, spectral method

AMS subject classifications. 05C50, 05C70, 68R10, 62H30, 91C20, 91D30, 94C15

DOI. 10.1137/18M1183558

1. Motivation.

Large, complex networks record pairwise interactions between components in a system. In many circumstances, we wish to summarize this wealth of information by extracting high-level information or visualizing key features. Two of the most important and well-studied tasks are

- *clustering*, also known as *community detection*, where we attempt to subdivide a network into smaller modules such that nodes within each module share many connections and nodes in distinct modules share few connections; and
- determination of *centrality* or *rank*, where we assign a nonnegative value to each node such that a larger value indicates a higher level of importance.

A distinct but closely related problem is to assign each node to either the *core* or the *periphery* in such a way that core nodes are strongly connected across the whole network, whereas peripheral nodes are strongly connected only to core nodes; hence there are relatively weak periphery–periphery connections. More generally, we may wish to assign a nonnegative value to each node, with a larger value indicating greater "coreness." The images in the center and right panels of Figure 1 indicate the two-by-two block pattern associated with a core–periphery structure.

*Received by the editors April 26, 2018; accepted for publication (in revised form) January 29, 2019; published electronically April 11, 2019. EPSRC data statement: All data and code related to this work are publicly available, and may be obtained by following the links in the text or by consulting the associated references.
http://www.siam.org/journals/simods/1-2/M118355.html
Funding: The first author is supported by the Marie Curie Individual Fellowship "MAGNET" n. 744014. The second author is supported by grant EP/M00158X/1 from the EPSRC/RCUK Digital Economy Programme.

†Department of Mathematics and Statistics, University of Strathclyde, G11XH Glasgow, UK (f.tudisco@strath.ac.uk, d.j.higham@strath.ac.uk).

The core–periphery concept appeared implicitly in the study of economic, social, and scientific citation networks and was formalized in the seminal paper of Borgatti and Everett [3]. A review of recent work on modeling and analyzing core–periphery structure, and related ideas in degree assortativity, rich-clubs, and nested/bow-tie/onion networks can be found in [9]. We focus here on the issue of *detection*: given a large complex network with nodes appearing in arbitrary order, can we discover, quantify, and visualize any inherent core–periphery organization?

In the next section, we set up our notation and discuss background material. Many detection algorithms can be motivated from an optimization perspective. In section 3, we use such an approach to define and justify the logistic core–periphery detection problem. We also show how it relates to a new random graph model that generates core–periphery networks. In section 4, we prove that a suitably relaxed version of this discrete optimization problem may be solved efficiently using a nonlinear spectral method. The resulting algorithm is described in subsection 4.2. Experiments on real and synthetic networks are performed in section 5, and some conclusions are given in section 6.

2. Background.

2.1. Notation. We use bold letters to denote vectors and uppercase letters to denote matrices. The respective entries are denoted with lowercase, nonbold symbols; for example, $\boldsymbol{x}$ denotes the vector with ith entry x_i, and A denotes the matrix with i,jth entries a_{ij}, $i,j = 1,\dots,n$. We use standard entrywise notation and operations; for instance, $\boldsymbol{x} \geq 0$ denotes a vector with nonnegative entries, $|\boldsymbol{x}|$ the vector with entries $(|\boldsymbol{x}|)_i = |x_i|$, $e^{\boldsymbol{x}}$ the vector with entries $(e^{\boldsymbol{x}})_i = e^{x_i}$, and $\boldsymbol{xy}$ the vector with entries $(\boldsymbol{xy})_i = x_i y_i$. For $p \geq 1$ we denote by $\|\boldsymbol{x}\|_p = (x_1^p + \cdots + x_n^p)^{1/p}$ the p-norm, with $\mathcal{S}_p = \{\boldsymbol{x} : \|\boldsymbol{x}\|_p = 1\}$ the p-unit sphere, and by $\mathbb{R}^n_+ = \{\boldsymbol{x} : x_i \geq 0 \ \forall i\}$ the cone of vectors with nonnegative entries.

We use $A \in \mathbb{R}^{n\times n}$ to represent the adjacency matrix of a network $G = (V, E)$, with vertex set V and edge set E. We consider undirected networks, so A is symmetric. Nonnegative weights are allowed, with a larger value of a_{ij} indicating a stronger connection between nodes i and j. We assume that the network is connected; that is, every pair of nodes may be joined by a path of edges having nonzero weight. For a disconnected network we could simply consider each connected component separately.

2.2. Core–periphery quality functions. Several models for core–periphery detection are based on the definition of various *core–periphery quality functions* f and their optimization over certain discrete or continuous sets of vectors. In this setting, node i is assigned a value $x_i^\star$, where $\boldsymbol{x}^\star$ solves an optimization problem of the form

$$\max_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x}), \quad f(\boldsymbol{x}) = \sum_{i,j=1}^{n} a_{ij}\,\kappa(x_i, x_j), \tag{1}$$

for some choice of kernel function κ and constraint set Ω. A larger value of $x_i^\star$ indicates greater "coreness," and the overall core–periphery structure may be examined by visualizing the adjacency matrix with nodes ordered according to the magnitude of the entries of $\boldsymbol{x}^\star$. We mention below some concrete examples.

The influential work of Borgatti and Everett [3] proposed a discrete notion of core–periphery structure based on comparing the given network with a block model that consists of a fully connected core and a periphery that has no internal edges but is fully connected to the core. Their method aims to find an indicator vector $\boldsymbol{x}$ with binary entries. So $x_i = 1$ assigns nodes to the core and $x_i = 0$ assigns nodes to the periphery. By defining the matrix $C = (c_{ij})$ as $c_{ij} = 1$ if $x_i = 1$ or $x_j = 1$ and $c_{ij} = 0$ otherwise, they look at the quantity $\rho_C = \sum_{ij} a_{ij}c_{ij}$ and aim to compute the binary vector $\boldsymbol{x}$ that maximizes ρ_C among all possible reshufflings of C such that the number of 1 and 0 entries is preserved. Clearly this method corresponds to (1) with $\kappa(x, y) = \text{sign}(x + y)$ and $\Omega = \{\boldsymbol{x} \in \{0, 1\}^n : \sum_i x_i = m\}$ for a fixed positive integer $m \leq n$.

Another popular technique, used, for instance, in UCINET [4], is based on the best rank-one approximation of the off-diagonal entries of A. In other words, this method seeks $\boldsymbol{x} \in \mathbb{R}^n$ that minimizes $\sum_i \sum_{j\neq i}(a_{ij} - x_i x_j)^2$. This is done via the MINRES algorithm, as discussed, for instance, in [5]. Writing

$$A = \lambda_1 \boldsymbol{v}_1 \boldsymbol{v}_1^T + \lambda_2 \boldsymbol{v}_2 \boldsymbol{v}_2^T + \cdots + \lambda_n \boldsymbol{v}_n \boldsymbol{v}_n^T,$$

where $\lambda_1 > 0$ is the largest eigenvalue of A and $\boldsymbol{v}_1$ is the corresponding eigenvector, it follows that the optimal rank-one matrix $\boldsymbol{x}\boldsymbol{x}^T$ we seek is strictly related to $\lambda_1 \boldsymbol{v}_1 \boldsymbol{v}_1^T$. Therefore, the least-squares problem is equivalent to maximizing the Rayleigh quotient of A, that is, the following optimization problem:

$$\max_{\boldsymbol{x}\neq 0} \frac{\boldsymbol{x}^T A \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}. \tag{2}$$

This, in turn, coincides with (1) for $\kappa(x, y) = xy$ and $\Omega = \{\boldsymbol{x} : \boldsymbol{x}^T\boldsymbol{x} = 1\} = \mathcal{S}_2$. Moreover, as the matrix A is symmetric, nonnegative, and irreducible, by the Perron–Frobenius theorem, the maximizer $\boldsymbol{v}_1$ is unique and entrywise positive, and the corresponding eigenvalue λ_1 coincides with the spectral radius of A. Following a different construction, a model for core–periphery detection based on the spectral radius of A and the associated Perron eigenvector is also considered in [24]. Note that, thanks to the Perron–Frobenius theorem, it follows that the constraint set in (1) can be chosen as $\Omega = \mathcal{S}_2^+ = \mathcal{S}_2 \cap \mathbb{R}^n_+$. This observation has practical importance because it constrains the solution space. As we discuss in section 4, this feature is shared by our nonlinear core–periphery model, where existence and uniqueness are proved using a customized nonlinear Perron–Frobenius-type theorem. Moreover, note that having a nonnegative solution $\boldsymbol{x}$ to (1) not only allows for a core–periphery assignment or ranking but also implicitly produces a continuous core–periphery score for the nodes. We note that the Perron–Frobenius eigenvector of A is also a well-known nodal centrality measure [14].

The concept of core–periphery quality measure with general kernel function, as formulated in (1), was introduced by Rombach et al. in [29]. They focus on the choice $\kappa(x, y) = xy$ and introduce a novel continuous constraint set defined in terms of two parameters $0 \leq \alpha, \beta \leq 1$ as follows:

$$\Omega = C_{\alpha,\beta} = \left\{ \boldsymbol{x} \in \mathbb{R}^n : \begin{array}{ll} x_i = \frac{i(1-\alpha)}{2\lfloor \beta n \rfloor} & \text{for } i = 1, \ldots, \lfloor \beta n \rfloor, \\ x_i = \frac{(i-\lfloor \beta n \rfloor)(1-\alpha)}{2(n-\lfloor \beta n \rfloor)} + \frac{1+\alpha}{2} & \text{for } i = \lfloor \beta n \rfloor + 1, \ldots, n \end{array} \right\}. \tag{3}$$

Here α is used to tune the score jump between the peripheral node with highest score and the core node with lowest score, whereas β is used to set the size of the core set. Note that, as $0 \leq \alpha, \beta \leq 1$, we have $C_{\alpha,\beta} \subseteq \mathbb{R}^n_+$, and thus, as for the Perron–Frobenius eigenvector of A, the maximizer of (1) with $\kappa(x, y) = xy$ and $\Omega = C_{\alpha,\beta}$ is a nonnegative vector whose entries define a core–periphery score value, called the *aggregate core score* in [30].

2.3. The optimization problem. The models proposed in [3] and [29] lead to discrete optimization problems whose global solution cannot be computed for large graphs. Both papers propose computational methods that deliver approximate solutions but do not guarantee accuracy. The combinatorial optimization problem of [3] is solved via random reshuffling. For the model proposed in [29] a simulated-annealing algorithm is used. The presence of the two parameters α and β adds a complication, which is addressed there by considering all (α, β) values on a discrete uniform lattice in $[0, 1]^2$. Clearly, refining the discretization level improves the approximation to the solution but raises the computational cost.

For the model used in UCINET based on MINRES [4, 5], recalling (2) we note that an efficient approach is to recast the optimization problem into the computation of a matrix eigenvector, for which well-established algorithms are available.

Since our approach fits into the core–periphery quality function optimization approach of [29, 30], we will use the method developed there, with $\kappa(x, y) = xy$ and $\Omega = C_{\alpha,\beta}$, as a baseline for comparison in our experiments in section 5.

Although algorithms based on other choices of the kernel function κ have not yet been considered in the literature, both in section 2.2.1 of [29] and section 4.2.1 of [30] it is pointed out that an ideal core–periphery kernel function is

$$\kappa(x, y) = \mu_\alpha(x, y) = (|x|^\alpha + |y|^\alpha)^{1/\alpha} \tag{4}$$

for $\alpha > 0$ large. In fact this function is related to core–periphery structure in a very natural way, as we discuss in the next two sections.

3. Logistic core–periphery detection problem. We propose a new model based on the kernel $\kappa(x, y) = \max\{|x|, |y|\}$. Note that this kernel function arises as the $\alpha \to \infty$ limit of (4). Focusing for now on the ranking problem, our goal is to determine a core–periphery ranking vector that assigns to each node a distinct integer between 1 and n, with a lower rank denoting a more peripheral node. Clearly, any such ranking vector is nothing but a permutation vector $\boldsymbol{\pi}$, where $i \mapsto \pi_i$ is a permutation of the set $\{1, \dots, n\}$. Therefore, if $\mathcal{P}_n$ is the set of permutation vectors of n entries, we formulate our core–periphery detection problem as follows:

$$\max_{\boldsymbol{\pi} \in \mathcal{P}_n} f_\infty(\boldsymbol{\pi}), \quad \text{where} \quad f_\infty(\boldsymbol{\pi}) = \sum_{i,j=1}^{n} a_{ij} \max\{\pi_i, \pi_j\}. \tag{5}$$

We will see in section 4 that, in practice, finite but large enough values of α in (4) provide an accurate approximation of $\max\{x, y\}$. Moreover, relaxing from $\mathcal{P}_n$ to $\mathcal{S}_p^+ = \mathcal{S}_p \cap \mathbb{R}^n_+$ allows for a globally convergent, easily implementable and computationally feasible algorithm.

We will refer to (5) as the *logistic core–periphery detection problem*. In order to motivate this name and the model itself, in the next subsection we discuss a random graph model that provides a natural and flexible model for core–periphery structure.

Figure 1. *Left:* $\sigma_{s,t}(x)$ *for* $t = 1/2$ *and* $s \in \{5, 10, 20, 40\}$. *The piecewise linear plot is the Heaviside function* $H_t(x)$, *which corresponds to* $\lim_{s\to\infty} \sigma_{s,t}(x)$. *Center: heatmap of* A *with entries* $a_{ij} = H_{1/2}(\max\{1 - i/n, 1 - j/n\})$. *Right: heatmap of* A *with entries* $a_{ij} = \sigma_{10,1/2}(\max\{1 - i/n, 1 - j/n\})$. *In these heatmaps, in order to emphasize the overall structure, the diagonal entries have been colored with the probabilities* $\mathbb{P}(i \sim i) = H_{1/2}(1 - i/n)$ *and* $\mathbb{P}(i \sim i) = \sigma_{10,1/2}(1 - i/n)$, *respectively. However, the associated random graphs have no self-loops, and so the actual diagonal probabilities are* $\mathbb{P}(i \sim i) = 0$.

3.1. Logistic core–periphery random graph model.

We now consider random graph models that generate core–periphery structure. For this subsection only, we restrict our attention to the case of unweighted, or binary, networks. We focus on models where the nodes can be placed in a natural ordering, represented by a permutation vector, so $i \mapsto \pi_i$. In this natural ordering, for every pair of nodes i and j the probability of an edge will be a function of π_i and π_j. Moreover, these events will be independent. We note that such models have been studied in other contexts; for example, in an early reference Grindrod [19] used this framework to define a class of range-dependent graphs that captures features of the classic Watts–Strogatz model.

A simple core–periphery model of this type arises when edges are present with probability one within the core and between core and periphery, and with probability zero among peripheral nodes. This model is considered, for instance, in [3, 29]. In this model there exists a permutation of the indices $i \mapsto \pi_i$ such that an edge connecting two different nodes i and j exists with independent probability $\mathbb{P}(i \sim j) = H_t\left(\frac{1}{n}\max\{\pi_i, \pi_j\}\right)$, where, for $t \in (0, 1)$, H_t is the Heaviside function $H_t(x) = 1$ if $x \geq t$ and $H_t(x) = 0$ otherwise. The parameter t allows us to tune the size of the core and of the periphery. Figure 1 (center) shows an example matrix whose ijth entry is the probability $\mathbb{P}(i \sim j)$ from this model for $t = 1/2$ and $\pi_i = n - i$ for any i. The Heaviside function H_t is a discontinuous step function, and it leads to an idealized all-or-nothing structure. Instead, we may consider a family of continuous approximations to H_t based on the logistic sigmoid function. For $s, t \in \mathbb{R}$, $s \geq 0$, we define

$$\sigma_{s,t}(x) = \frac{1}{1 + e^{-s(x-t)}}.$$

Note that for any fixed $x, t \in \mathbb{R}$ we have $\lim_{s\to\infty} \sigma_{s,t}(x) = H_t(x)$. Examples are plotted in Figure 1 (left).

We now introduce the random graph model where an edge connecting two different nodes

i and j exists with independent probability

$$\mathbb{P}(i \sim j) = \sigma_{s,t}\left(\frac{1}{n}\max\{n-i, n-j\}\right).$$

We refer to this as the *logistic core–periphery random graph model*. The rightmost plot in Figure 1 shows a 20×20 example matrix whose ijth entry is the corresponding probability $\mathbb{P}(i \sim j)$ for $s = 10$ and $t = 1/2$. We see that, relative to the Heaviside version, this model gives a smoother transition from core to periphery and has a built-in notion of ranking within each group. The ability of this model to capture core and peripheral nodes was also recently pointed out in [21].

We are interested in the circumstance where a core–periphery structure is present in the graph but must be discovered. In practice, our task is to find a suitable reordering of the nodes that highlights the presence of core and periphery. A natural approach is then to find the permutation of indices $\boldsymbol{\pi} \in \mathcal{P}_n$ that maximizes the likelihood, under the assumption of a logistic core–periphery structure. This likelihood is given by

$$\nu(\boldsymbol{\pi}) = \prod_{i\sim j} \varphi(\pi_i, \pi_j) \prod_{i\not\sim j} (1 - \varphi(\pi_i, \pi_j)), \tag{6}$$

where, for the sake of brevity, we let $\varphi(x, y) = \sigma_{s,t}(\frac{1}{n}\max\{x, y\})$. We now show that solving the proposed logistic core–periphery detection problem (5) is equivalent to solving this maximum likelihood reordering problem.

Theorem 3.1. *$\boldsymbol{\pi}^\star \in \mathcal{P}_n$ is a permutation that maximizes $\nu(\boldsymbol{\pi})$ if and only if $\boldsymbol{\pi}^\star$ is a solution of (5).*

Proof. Our proof exploits a very useful trick that Grindrod [19] used in the case of a range-dependent random graph: the likelihood $\nu(\boldsymbol{\pi})$ can be equivalently written as

$$\nu(\boldsymbol{\pi}) = \prod_{i\sim j} \frac{\varphi(\pi_i, \pi_j)}{1 - \varphi(\pi_i, \pi_j)} \prod_{i,j=1}^{n} 1 - \varphi(\pi_i, \pi_j).$$

As the right-hand factor does not depend on the graph, maximizing $\nu(\boldsymbol{\pi})$ is equivalent to maximizing the left-hand factor. Thus, taking the logarithm on both sides, we observe that $\boldsymbol{\pi}^\star$ maximizes ν if and only if it maximizes

$$\sum_{ij=1}^{n} a_{ij} \log\left(\frac{\varphi(\pi_i, \pi_j)}{1 - \varphi(\pi_i, \pi_j)}\right).$$

Now, using the definition of $\varphi(x, y)$ in terms of the logistic sigmoid function, a short computation shows that $\log(\varphi(x, y)/(1 - \varphi(x, y))) = s(\frac{1}{n}\max\{x, y\} - t)$ for any $x, y, t \in \mathbb{R}$, $s \geq 0$. Therefore, $\boldsymbol{\pi}^\star$ maximizes ν if and only if it maximizes the core–periphery quality function $\sum_{ij} a_{ij} \max\{\pi_i, \pi_j\}$, which concludes the proof. ∎

In words, Theorem 3.1 shows that in the case of unweighted networks, solving the logistic core–periphery detection problem (5) is equivalent to solving the maximum likelihood

reordering problem (6) under the assumption that the network was generated from the logistic core–periphery random graph model. This is somewhat analogous to a known phenomenon in the community detection case [27].

We mention that core–periphery detection via likelihood maximization on a random graph model was also proposed in [35]. There, the authors used a stochastic block model where nodes are independently assigned to the core with probability γ_1 and to the periphery with probability $1-\gamma_1$. Core–core, core–periphery, and periphery–periphery connections then appear with independent probabilities p_{11}, p_{12}, and p_{22}, with $p_{11} > p_{12} > p_{22}$. Inferring model parameters by maximizing the likelihood over all possible node bipartitions leads to a core–periphery assignment. Because solving this discrete optimization problem is not practicable for large networks, the authors develop an approximation technique based on expectation maximization and belief propagation. We emphasize that this random graph reordering/partitioning framework applies to unweighted (binary) networks.

4. Nonlinear spectral method for core–periphery detection. In this section, we introduce an iterative method for the logistic core–periphery detection problem (5) and prove that it converges globally to the solution of a relaxed problem. We refer to this as a *nonlinear spectral method* for two reasons. First, its derivation and analysis are inspired by recent work in nonlinear Perron–Frobenius theory [15, 17, 32, 16]. Second, as shown in Lemma 4.3, there is an equivalence between (5) and a nonlinear eigenvalue problem. Recall that the network is assumed to be (nonnegatively) weighted, connected, and undirected.

The logistic core–periphery model (5) is a combinatorial optimization problem whose exact solution is not feasible for large-scale networks. We therefore introduce two relaxations that lead to a new "smooth" logistic core–periphery problem whose solution may be computed efficiently with a new nonlinear spectral method.

Given $\alpha > 1$, we replace the discontinuous kernel function $\max\{|x|, |y|\}$ with

$$\mu_\alpha(x,y) = (|x|^\alpha + |y|^\alpha)^{1/\alpha}. \tag{7}$$

As mentioned at the end of subsection 2.2, $\max\{|x|, |y|\}$ is the limit of $\mu_\alpha(x,y)$ for $\alpha \to \infty$. More precisely, letting

$$f_\alpha(\boldsymbol{x}) = \sum_{ij} a_{ij}\mu_\alpha(x_i, x_j), \tag{8}$$

a simple computation using the Hölder inequality reveals that

$$f_\infty(\boldsymbol{x}) \le f_\alpha(\boldsymbol{x}) \le 2^{1/\alpha} f_\infty(\boldsymbol{x}) \tag{9}$$

for any $\alpha > 1$. Therefore, when α is large enough, using f_α in place of f_∞ in (5) provides a very accurate approximation.

Second, we relax the discrete constraint set $\mathcal{P}_n$ into a continuous one. In doing this we note that every vector in $\mathcal{P}_n$ is entrywise nonnegative and has fixed length. For instance, $\|\boldsymbol{x}\|_1 = \frac{1}{2}n(n+1)$ for any $\boldsymbol{x} \in \mathcal{P}_n$. Note that the normalization constant $\frac{1}{2}n(n+1)$ can be chosen arbitrarily. In fact, the function f_α we are considering is positively 1-homogeneous; that is, for any $\lambda > 0$ we have $f_\alpha(\lambda\boldsymbol{x}) = \lambda f_\alpha(\boldsymbol{x})$. This implies that if $\boldsymbol{x}$ maximizes f_α among

all the vectors of norm exactly 1, then for any $a > 0$, $a\boldsymbol{x}$ maximizes f_α among all the vectors of norm exactly a. We therefore relax $\mathcal{P}_n$ into a sphere of nonnegative vectors. For convenience, we choose the p-sphere $\mathcal{S}_p = \{\boldsymbol{x} : \|\boldsymbol{x}\|_p = 1\}$ and let $\mathcal{S}_p^+ = \mathcal{S}_p \cap \mathbb{R}^n_+$.

Overall, for $\mu_\alpha(x, y)$ in (7) and $f_\alpha(\boldsymbol{x})$ in (8), we modify the original logistic core–periphery problem (5) into

$$\max_{\boldsymbol{x}\in\mathcal{S}_p^+} f_\alpha(\boldsymbol{x}), \quad \text{where} \quad f_\alpha(\boldsymbol{x}) = \sum_{i,j=1}^{n} a_{ij}\,\mu_\alpha(x_i, x_j)\,. \tag{10}$$

We devote the remainder of this section to proving that, for any $\alpha > 0$ and any $p > \alpha$, the relaxed logistic core–periphery model (10) has a unique, entrywise positive solution that can be efficiently computed via a globally convergent iterative method.

4.1. Existence and uniqueness of a solution to the relaxed problem. We begin by observing that the function f_α attains its maximum on a positive vector.

Lemma 4.1. *The problem* (10) *is solved by a vector* $\boldsymbol{x}^\star$ *such that* $\boldsymbol{x}^\star > 0$.

Proof. As $f_\alpha(\boldsymbol{x}) = f_\alpha(|\boldsymbol{x}|)$ for any $\boldsymbol{x} \in \mathbb{R}^n$, we easily deduce that the maximum is attained on a vector $\boldsymbol{x}^\star \geq 0$. Now suppose that there exists $1 \leq k \leq n$ such that $x_k^\star = 0$. As the graph is connected, there exists ℓ such that $a_{k\ell} > 0$. Then the vector $\boldsymbol{y}$ defined by $y_i = x_i^\star$ for $i \neq k$ and $y_k = \varepsilon > 0$ would be such that

$$\begin{aligned} f_\alpha(\boldsymbol{y}) &= \sum_{i,j\neq k} a_{ij}\mu_\alpha(x_i^\star, x_j^\star) + 2\sum_{j} a_{kj}\mu_\alpha(x_j^\star, \varepsilon) \\ &\geq \sum_{i,j\neq k} a_{ij}\mu_\alpha(x_i^\star, x_j^\star) + 2a_{k\ell}\mu_\alpha(x_\ell^\star, \varepsilon) > f_\alpha(\boldsymbol{x}^\star), \end{aligned}$$

which contradicts the maximality of $\boldsymbol{x}^\star$. We conclude that the solution of (10) is attained on an entrywise positive vector. ∎

Now, by using the positive 1-homogeneity of f_α, we show that the constrained optimization problem (10) is equivalent to an unconstrained problem for the normalized function $f_\alpha(\boldsymbol{x})/\|\boldsymbol{x}\|_p$.

Lemma 4.2. *For any* $p > 1$ *and any* $\alpha > 1$ *we have*

$$\max_{\boldsymbol{x}\in\mathcal{S}_p^+} f_\alpha(\boldsymbol{x}) = \max_{\boldsymbol{x}\in\mathbb{R}^n} \frac{f_\alpha(\boldsymbol{x})}{\|\boldsymbol{x}\|_p}.$$

Proof. By the 1-homogeneity of f_α we have the following chain of inequalities:

$$\begin{aligned} \max_{\|\boldsymbol{x}\|_p\leq 1} f_\alpha(\boldsymbol{x}) \geq \max_{\|\boldsymbol{x}\|_p=1} f_\alpha(\boldsymbol{x}) &= \max_{\boldsymbol{x}\in\mathbb{R}^n} f_\alpha\left(\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_p}\right) = \max_{\boldsymbol{x}\in\mathbb{R}^n} \frac{f_\alpha(\boldsymbol{x})}{\|\boldsymbol{x}\|_p} \\ &\geq \max_{\|\boldsymbol{x}\|_p\leq 1} \frac{f_\alpha(\boldsymbol{x})}{\|\boldsymbol{x}\|_p} \geq \max_{\|\boldsymbol{x}\|_p\leq 1} f_\alpha(\boldsymbol{x})\,. \end{aligned}$$

This implies that the inequalities above are all identities. Together with $f_\alpha(\boldsymbol{x}) = f_\alpha(|\boldsymbol{x}|)$, this shows the claim. ∎

We have the following consequence.

Lemma 4.3. *Let $F_\alpha = \nabla f_\alpha : \mathbb{R}^n \to \mathbb{R}^n$ be the gradient of f_α, that is,*

$$F_\alpha(\boldsymbol{x})_i = 2\sum_{j=1}^{n} a_{ij}\,|x_i|^{\alpha-2}x_i(|x_i|^\alpha + |x_j|^\alpha)^{1/\alpha-1}\,, \quad i = 1,\dots,n\,.$$

Then, for any $p > 1$, the following statements are equivalent:

1. *$\boldsymbol{x}$ is a solution of (10).*
2. *$\boldsymbol{x}$ satisfies the eigenvalue equation $F_\alpha(\boldsymbol{x}) = \lambda\,|\boldsymbol{x}|^{p-2}\boldsymbol{x}$ with $\lambda > 0$.*
3. *$\boldsymbol{x}$ is a fixed point of the map $G_\alpha(\boldsymbol{x}) = |F_\alpha(\boldsymbol{x})|^{q-2}F_\alpha(\boldsymbol{x})/\|F_\alpha(\boldsymbol{x})\|_q^{q-1}$, where q is the Hölder conjugate of p, i.e., $1/p + 1/q = 1$.*

Proof. For convenience, let us write $r_\alpha(\boldsymbol{x}) = f_\alpha(\boldsymbol{x})/\|\boldsymbol{x}\|_p$. By differentiating $r_\alpha(\boldsymbol{x})$ we see that

$$\nabla r_\alpha(\boldsymbol{x}) = 0 \iff \frac{1}{\|\boldsymbol{x}\|_p}\Big\{F_\alpha(\boldsymbol{x}) - \lambda\,|\boldsymbol{x}|^{p-2}\boldsymbol{x}\Big\} = 0,$$

with $\lambda = f_\alpha(\boldsymbol{x})/\|\boldsymbol{x}\|_p^p > 0$. Together with Lemma 4.2, this proves (1) $\iff$ (2). Now note that the map $\boldsymbol{x} \mapsto \psi(\boldsymbol{x}) = |\boldsymbol{x}|^{q-2}\boldsymbol{x}$ is such that $\psi(|\boldsymbol{x}|^{p-2}\boldsymbol{x}) = \boldsymbol{x}$. In fact,

$$\psi(|\boldsymbol{x}|^{p-2}\boldsymbol{x}) = |\,|\boldsymbol{x}|^{p-2}\boldsymbol{x}|^{q-2}\,|\boldsymbol{x}|^{p-2}\boldsymbol{x} = |\boldsymbol{x}|^{(p-1)(q-2)+p-2}\boldsymbol{x} = \boldsymbol{x}\,.$$

As ψ is bijective we have $F_\alpha(\boldsymbol{x}) = \lambda\,|\boldsymbol{x}|^{p-2}\boldsymbol{x} \iff |F_\alpha(\boldsymbol{x})|^{q-2}F_\alpha(\boldsymbol{x}) = \psi(\lambda)\boldsymbol{x}$. Therefore, recalling that $\|\boldsymbol{x}\|_p = 1$ and $\lambda > 0$ we have

$$F_\alpha(\boldsymbol{x}) = \lambda\,|\boldsymbol{x}|^{p-2}\boldsymbol{x} \iff \frac{|F_\alpha(\boldsymbol{x})|^{q-2}F_\alpha(\boldsymbol{x})}{\|F_\alpha(\boldsymbol{x})\|_q^{q-1}} = \frac{|F_\alpha(\boldsymbol{x})|^{q-2}F_\alpha(\boldsymbol{x})}{\||F_\alpha(\boldsymbol{x})|^{q-2}F_\alpha(\boldsymbol{x})\|_p} = \frac{\psi(\lambda)\boldsymbol{x}}{\|\psi(\lambda)\boldsymbol{x}\|_p} = \boldsymbol{x},$$

where the first identity follows by $\||\boldsymbol{y}|^{q-2}\boldsymbol{y}\|_p = \|\boldsymbol{y}^{q-1}\|_p = \|\boldsymbol{y}\|_q^{q-1}$. This shows (2) $\iff$ (3) and concludes the proof. ∎

We need one final, rather technical, lemma. For the sake of completeness, we state it for the case where α may attain both positive and negative values.

Lemma 4.4. *For $\alpha \in \mathbb{R}$ let G_α be defined as in Lemma 4.3 above, and let $g_i : \mathbb{R}^n \to \mathbb{R}$ be the scalar functions such that $G_\alpha(\boldsymbol{x}) = (g_1(\boldsymbol{x}),\dots,g_n(\boldsymbol{x}))$. Then,*

$$\left\|\frac{\nabla g_k(\boldsymbol{x})\boldsymbol{x}}{g_k(\boldsymbol{x})}\right\|_1 \le \frac{|1-\alpha|}{p-1}$$

for any vector $\boldsymbol{x} \ge 0$ and any $k = 1,\dots,n$.

Proof. First, note that $\boldsymbol{y} \ge 0$ implies that both $F_\alpha(\boldsymbol{y})$ and $G_\alpha(\boldsymbol{y})$ are nonnegative. Now,

let $i,k \in \{1,\dots,n\}$. By the definition of g_k and using the chain rule, we obtain

$$\begin{aligned}\left|\frac{\partial_i\{g_k(\boldsymbol{y})\}}{g_k(\boldsymbol{y})}\right| &= \left|\frac{\partial_i\left\{\left(F_\alpha(\boldsymbol{y})_k\right)^{q-1}\right\}}{\left(F_\alpha(\boldsymbol{y})_k\right)^{q-1}} - \frac{\partial_i\left\{\|F_\alpha(\boldsymbol{y})\|_q^{q-1}\right\}}{\|F_\alpha(\boldsymbol{y})\|_q^{q-1}}\right| \\ &= (q-1)\left|\frac{\partial_i F_\alpha(\boldsymbol{y})_k}{F_\alpha(\boldsymbol{y})_k} - \frac{\sum_m \left(F_\alpha(\boldsymbol{y})_m\right)^{q-1}\partial_i F_\alpha(\boldsymbol{y})_m}{\|F_\alpha(\boldsymbol{y})\|_q^q}\right| \\ &= (q-1)\left|\frac{\partial_i F_\alpha(\boldsymbol{y})_k}{F_\alpha(\boldsymbol{y})_k} - \sum_m \left(\frac{F_\alpha(\boldsymbol{y})_m}{\|F_\alpha(\boldsymbol{y})\|_q}\right)^q \frac{\partial_i F_\alpha(\boldsymbol{y})_m}{F_\alpha(\boldsymbol{y})_m}\right|.\end{aligned} \tag{11}$$

Now note that for any nonnegative $\boldsymbol{y} \in \mathbb{R}^n$ we have

$$\frac{\partial_i F_\alpha(\boldsymbol{y})_m y_i}{F_\alpha(\boldsymbol{y})_m} = (1-\alpha)\,\frac{y_i^\alpha}{y_i^\alpha + y_m^\alpha}\,\frac{a_{mi}(y_m^\alpha + y_i^\alpha)^{1/\alpha-1}}{\sum_j a_{mj}(y_m^\alpha + y_j^\alpha)^{1/\alpha-1}} \geq 0\,.$$

Let $\widetilde{m} \in \{1,\dots,n\}$ be the index for which the quantity above is maximal. From (11) we deduce that $|\partial_i\{g_k(\boldsymbol{y})\}y_i/g_k(\boldsymbol{y})|$ is of the form $|\beta_k - \sum_m \gamma_m\beta_m|$, where $\beta_m,\gamma_m \geq 0$ for all $m = 1,\dots,n$ and $\sum_m \gamma_m = 1$. This implies that $|\beta_k - \sum_m \gamma_m\beta_m| \leq \beta_{\widetilde{m}}$, and as $y_i^\alpha/(y_i^\alpha + y_{\widetilde{m}}^\alpha) \leq 1$, we find

$$\left|\frac{\partial_i\{g_k(\boldsymbol{y})\}y_i}{g_k(\boldsymbol{y})}\right| \leq (q-1)|1-\alpha|\left(\frac{a_{\widetilde{m},i}(y_{\widetilde{m}}^\alpha + y_i^\alpha)^{1/\alpha-1}}{\sum_j a_{\widetilde{m},j}(y_{\widetilde{m}}^\alpha + y_j^\alpha)^{1/\alpha-1}}\right).$$

Summing this formula over i and recalling that $q-1 = 1/(p-1)$ concludes the proof. ∎

This leads to our main result.

Theorem 4.5. *Assume $\alpha > 1$ and $p > \alpha$. Then (10) has a unique solution $\boldsymbol{x}^\star$ that is entrywise positive. Moreover, for any $\boldsymbol{x}_0 > 0$, if G_α is defined as in Lemma 4.3, then the sequence defined by $\boldsymbol{x}_{k+1} = G_\alpha(\boldsymbol{x}_k)$ belongs to $\mathcal{S}_p^+$ and converges to $\boldsymbol{x}^\star$.*

Proof. The fact that $\boldsymbol{x}_k \in \mathcal{S}_p^+$ for any k is an obvious consequence of the identities $\||\boldsymbol{z}|^{q-2}\boldsymbol{z}\|_p = \|\boldsymbol{z}^{q-1}\|_p = \|\boldsymbol{z}\|_q^{q-1}$. Now we show that the map G_α defined in Lemma 4.3 is Lipschitz contractive which, due to the Banach fixed point theorem, gives convergence of the sequence and uniqueness of the solution. To this end we use the Thomson metric d_T defined for $\boldsymbol{x},\boldsymbol{y} \in \mathcal{S}_p^+$ as $d_T(\boldsymbol{x},\boldsymbol{y}) = \|\log(\boldsymbol{x}) - \log(\boldsymbol{y})\|_\infty$.

As before, for $i = 1,\dots,n$, let $g_i : \mathbb{R}^n \to \mathbb{R}$ be the scalar functions such that $G_\alpha(\boldsymbol{x}) = (g_1(\boldsymbol{x}),\dots,g_n(\boldsymbol{x}))$. By the mean value theorem we have

$$\phi(\boldsymbol{x}) - \phi(\boldsymbol{y}) = \nabla\phi(\boldsymbol{\xi})^T(\boldsymbol{x}-\boldsymbol{y})$$

for any differentiable function $\phi : \mathbb{R}^n \to \mathbb{R}$ and with $\boldsymbol{\xi}$ being a point in the segment joining $\boldsymbol{x}$ and $\boldsymbol{y}$. Consider the function $\phi(\boldsymbol{x}) = \log(g_i(e^{\boldsymbol{x}}))$. Then $\nabla\phi(\boldsymbol{\xi}) = \nabla g_i(e^{\boldsymbol{\xi}})e^{\boldsymbol{\xi}}/g_i(e^{\boldsymbol{\xi}})$, and we obtain

$$|\phi(\boldsymbol{x}) - \phi(\boldsymbol{y})| = |\log(g_i(e^{\boldsymbol{x}})) - \log(g_i(e^{\boldsymbol{y}}))| = \left|\frac{(\nabla g_i(e^{\boldsymbol{\xi}})e^{\boldsymbol{\xi}})^T(\boldsymbol{x}-\boldsymbol{y})}{g_i(e^{\boldsymbol{\xi}})}\right|.$$

As the exponential function maps positive vectors into positive vectors bijectively, the previous equation implies that for any two positive vectors $\overline{\boldsymbol{x}} = e^{\boldsymbol{x}}$ and $\overline{\boldsymbol{y}} = e^{\boldsymbol{y}}$ we have

$$|\log(g_i(\overline{\boldsymbol{x}})) - \log(g_i(\overline{\boldsymbol{y}}))| \le \left\| \frac{\nabla g_i(e^{\boldsymbol{\xi}}) e^{\boldsymbol{\xi}}}{g_i(e^{\boldsymbol{\xi}})} \right\|_1 \|\log(\overline{\boldsymbol{x}}) - \log(\overline{\boldsymbol{y}})\|_\infty = \left\| \frac{\nabla g_i(e^{\boldsymbol{\xi}}) e^{\boldsymbol{\xi}}}{g_i(e^{\boldsymbol{\xi}})} \right\|_1 d_T(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}).$$

Together with Lemma 4.4, this gives us $d_T(G_\alpha(\boldsymbol{x}), G_\alpha(\boldsymbol{y})) \le C d_T(\boldsymbol{x}, \boldsymbol{y})$ with $C = |1-\alpha|/(p-1) < 1$. Thus, G_α is a contraction and $\boldsymbol{x}_k \to \boldsymbol{x}^\star \in \mathcal{S}_p^+$ as $k \to \infty$. Finally, Lemmas 4.1 and 4.3 imply that $\boldsymbol{x}^\star$ is entrywise positive and solves (10), concluding the proof. ∎

4.2. Algorithm. Theorem 4.5 leads naturally to the following algorithm.

Algorithm 1: Nonlinear spectral method for core–periphery detection.

Input: Adjacency matrix A, initial guess $\boldsymbol{x}_0 > 0$
Fix $\alpha \gg 1$, $p > \alpha$, $q = p/(p-1)$
Let $F_\alpha(\boldsymbol{x})_i = 2 \sum_{j=1}^n a_{ij} |x_i|^{\alpha-2} x_i (|x_i|^\alpha + |x_j|^\alpha)^{1/\alpha - 1}$, for $i = 1, \dots, n$

1 **For** $k = 0, 1, 2, 3, \dots$ **repeat**
2 $\quad \boldsymbol{y}_{k+1} = F_\alpha(\boldsymbol{x}_k)$
3 $\quad \boldsymbol{x}_{k+1} = \|\boldsymbol{y}_{k+1}\|_q^{1-q} |\boldsymbol{y}_{k+1}|^{q-2} \boldsymbol{y}_{k+1}$
4 **until** $\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\| / \|\boldsymbol{x}_{k+1}\| <$ tolerance
5 $\boldsymbol{c} = \boldsymbol{x}_{k+1} / \max(\boldsymbol{x}_{k+1})$
6 Reorder the network nodes according to the magnitude of the entries of $\boldsymbol{c}$

Output: Core–periphery score $\boldsymbol{c}$ and approximate maximizer $\boldsymbol{x}_{k+1}$ of (10)

Recall that, since $\boldsymbol{x}_0 > 0$, each element of the sequence $\boldsymbol{x}_k$ generated by the algorithm is a positive vector, due to Theorem 4.5. Each iteration requires the computation of a vector norm, at step 3, and the computation of the action of the nonlinear map F_α on a nonnegative vector $\boldsymbol{x}$, at step 2. Thus, if $m \ge n$ is the number of edges in the network (or, equivalently, half the number of nonzero entries of A), the order of complexity per iteration of Algorithm 1 is $O(m) + O(n)$. For large-scale, sparse, real-world networks, m is typically linearly proportional to n or $n \log n$. In this setting the method is scalable to high dimensions, as confirmed by Figure 2.

Further comments on the algorithm are in order. First, recall that the convergence is independent of the starting point, $\boldsymbol{x}_0$, provided that $\boldsymbol{x}_0$ is entrywise positive. In practice, we use a uniform vector. Concerning the choice of α, recall that we want α large enough to give a good approximation to the original kernel $\max\{|x|, |y|\}$. As quantified in (9), the approximation error is bounded by a factor $2^{1/\alpha}$. Thus, in practice, moderate values of the parameter are sufficient. In order to avoid numerical issues, in the experiments presented in section 5 we use $\alpha = 10$. As for the choice of p, from the proof of Theorem 4.5 it follows that the larger $p > \alpha$, the smaller the contraction ratio $C = (\alpha - 1)/(p - 1)$ and thus the faster the convergence of Algorithm 1. This is made more precise in Corollary 4.6 below, where we explicitly bound $\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|$ and $\|\boldsymbol{x}_k - \boldsymbol{x}^\star\|$ in terms of C. Finally, the choice of the norm in the stopping criterion is not critical. We typically use the p-norm because the sequence $\boldsymbol{x}_k$ is designed so that $\|\boldsymbol{x}_k\|_p = 1$ for any k. Hence, in the stopping criterion we require one fewer norm computation at each step. However, this reduction in cost is likely to be negligible,

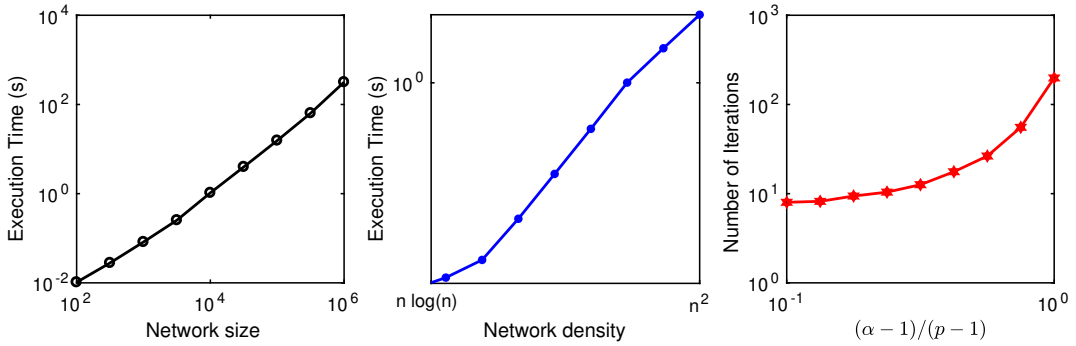


Figure 2. *All plots show mean values over* 10 *runs. Left and center: Time required by Algorithm* 1 *to converge to a tolerance of* 10^{-8}*, with* $\alpha = p/2 = 10$*, for random Erdős–Renyi graphs. Left:* n *nodes and* $m = O(n \log n)$ *edges, with* $n \in [10^2, 10^6]$*. Center:* $n = 1000$ *nodes and* $m \in [n \log n, n^2]$ *edges. Right: Number of iterations required by Algorithm* 1 *when the ratio* $(\alpha - 1)/(p - 1)$ *varies.*

and thus we expect other distance functions to work equally well. Moreover, we point out that, due to Corollary 4.6, a computationally cheaper stopping criterion is available from the contraction ratio $(\alpha - 1)/(p - 1)$ and its integer powers. However, in our experience, this upper bound on the iteration error can be far from sharp.

Corollary 4.6. *For* $\boldsymbol{x}_0 > 0$*, let* $\boldsymbol{x}_k$ *be the sequence defined by Algorithm* 1*, and let* $\gamma = \|\log(\boldsymbol{x}_1) - \log(\boldsymbol{x}_0)\|_\infty$*. For any* $k = 0, 1, 2, \ldots$ *we have*

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_\infty \le \gamma \left(\frac{\alpha - 1}{p - 1}\right)^k \quad \text{and} \quad \|\boldsymbol{x}_k - \boldsymbol{x}^\star\|_\infty \le \gamma \left(\frac{p - 1}{p - \alpha}\right)\left(\frac{\alpha - 1}{p - 1}\right)^k,$$

where $\boldsymbol{x}^\star = \lim_k \boldsymbol{x}_k$ *is the unique positive solution of* (10)*.*

Proof. From the mean value theorem we have $|e^a - e^b| \le |a - b| \max\{e^a, e^b\}$. Thus, for any $\boldsymbol{x}, \boldsymbol{y} > 0$ with $\|\boldsymbol{x}\|_p = \|\boldsymbol{y}\|_p = 1$ we have

$$\|\log(\boldsymbol{x}) - \log(\boldsymbol{y})\|_\infty \ge \|\boldsymbol{x} - \boldsymbol{y}\|_\infty \Big(\max_i(\max\{x_i, y_i\})\Big)^{-1} \ge \|\boldsymbol{x} - \boldsymbol{y}\|_\infty,$$

as both x_i and y_i are not larger than one for any $i = 1, \ldots, n$. With the notation of the proof of Theorem 4.5, this implies that $d_T(\boldsymbol{x}, \boldsymbol{y}) \ge \|\boldsymbol{x} - \boldsymbol{y}\|_\infty$ for any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{S}_p^+$. Moreover, from the proof of that theorem we have that $d_T(G_\alpha(\boldsymbol{x}), G_\alpha(\boldsymbol{y})) \le C d_T(\boldsymbol{x}, \boldsymbol{y})$, with $C = (\alpha - 1)/(p - 1)$. Therefore, as $\boldsymbol{x}_k \in \mathcal{S}_p^+$ for any k we have

$$\begin{aligned}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_\infty &\le d_T(\boldsymbol{x}_{k+1}, \boldsymbol{x}_k) = d_T(G_\alpha(\boldsymbol{x}_k), G_\alpha(\boldsymbol{x}_{k-1})) \\ &\le C d_T(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}) \le C^k d_T(\boldsymbol{x}_1, \boldsymbol{x}_0)\,.\end{aligned}$$

This proves the first inequality. As for the second one, first note that it is enough to show that $d_T(\boldsymbol{x}_k, \boldsymbol{x}^\star) \le (1 - C)^{-1} d_T(\boldsymbol{x}_{k+1}, \boldsymbol{x}_k)$ as we then can argue as before to obtain $\|\boldsymbol{x}^\star - \boldsymbol{x}_k\|_\infty \le (1 - C^{-1}) C^k d_T(\boldsymbol{x}_1, \boldsymbol{x}_0)$, which is the rightmost inequality in the statement. Now, observe that adding $d_T(\boldsymbol{x}_{i+1}, \boldsymbol{x}_i)$ to both sides of the inequality $d_T(\boldsymbol{x}_{i+2}, \boldsymbol{x}_{i+1}) \le C d_T(\boldsymbol{x}_{i+1}, \boldsymbol{x}_i)$ and

rearranging terms leads to

$$d_T(\boldsymbol{x}_{i+1}, \boldsymbol{x}_i) \leq \frac{1}{1-C}\Big(d_T(\boldsymbol{x}_{i+1}, \boldsymbol{x}_i) - d_T(\boldsymbol{x}_{i+2}, \boldsymbol{x}_{i+1})\Big)$$

for any $i = 0, 1, 2, \ldots$. Therefore, using the triangle inequality for d_T, for any k, h with $h > k$ we obtain

$$d_T(\boldsymbol{x}_{h+1}, \boldsymbol{x}_k) \leq \sum_{i=k}^{h} d_T(\boldsymbol{x}_{i+1}, \boldsymbol{x}_i) \leq \frac{1}{1-C}\Big(d_T(\boldsymbol{x}_{k+1}, \boldsymbol{x}_k) - d_T(\boldsymbol{x}_{h+2}, \boldsymbol{x}_{h+1})\Big).$$

Finally, letting h grow to infinity in the previous inequality gives the desired bound and concludes the proof. ∎

5. Experiments. In this section we describe results obtained when the logistic core–periphery score computed via Algorithm 1 is used to rank nodes in some example networks. All experiments were performed using MATLAB version 9.1.0.441655 (R2016b) on a laptop running Ubuntu 16.04 LTS with a 3.2 GHz Intel Core i5 processor and 8 GB of RAM. The experiments can be reproduced using the code available at https://github.com/ftudisco/nonlinear-core-periphery.

We compare results with those obtained from the following other core-quality function optimization approaches: the degree vector, the Perron eigenvector of the adjacency matrix (eigenvector centrality), and the simulated-annealing algorithm proposed in [29]. Furthermore, we compare with the k-core decomposition coreness score [22], computed as the limit of the H-index operator sequence discussed in [23].

The use of the degree vector $\boldsymbol{d}$ and of the eigenvector centrality $\boldsymbol{v}$ may be regarded as linear counterparts of our method. If $\alpha = 1$, then for any $\boldsymbol{x} \geq 0$ the functional $f_\alpha(\boldsymbol{x}) = \sum_{ij} a_{ij}\mu_\alpha(x_i, x_j)$ is linear and has the form $f_1(\boldsymbol{x}) = \sum_{ij} a_{ij}(x_i + x_j) = 2\,\boldsymbol{x}^T\boldsymbol{d}$. Thus, the maximum is attained when $\boldsymbol{x}$ is the degree vector $\boldsymbol{d}$. The eigenvector centrality $A\boldsymbol{v} = \rho(A)\boldsymbol{v}$, $\|\boldsymbol{v}\|_2 = 1$, instead, somewhat corresponds to the case where α goes to 0. To obtain $\boldsymbol{v}$, however, we need to slightly modify the approximate kernel μ_α from (8) to

$$\widetilde{\mu}_\alpha(x, y) = \left(\frac{|x|^\alpha + |y|^\alpha}{2}\right)^{1/\alpha}.$$

This is because μ_α diverges when $\alpha \to 0$, whereas $\lim_{\alpha\to 0} \widetilde{\mu}_\alpha(x, y) = \sqrt{|xy|}$. On the other hand, notice that both μ_α and $\widetilde{\mu}_\alpha$ coincide with the maximum operator when $\alpha \to \infty$, and for any fixed $\alpha > 0$ a vector that maximizes $\sum_{ij} A_{ij}\widetilde{\mu}_\alpha(x_i, x_j)$ maximizes f_α as well. Replacing μ_α with $\widetilde{\mu}_\alpha$ we have $f_0(\boldsymbol{x}) = \sqrt{\boldsymbol{x}}^T A\sqrt{\boldsymbol{x}}$. Thus, if we choose $p = 1$, the maximum is attained when $\boldsymbol{x} = \boldsymbol{v}^2$, the square of the entrywise positive eigenvector centrality. Note that with this choice of p, the solution $\boldsymbol{v}$ is constrained on the Euclidean sphere $\|\boldsymbol{v}\|_2 = 1$. Notice, moreover, that this is confirmed by Theorem 3.1 because, for $\alpha \to 0$ and $p = 1$, the nonlinear operator F_α boils down to the matrix A, and Algorithm 1 is equivalent to the standard linear power method.

As for the simulated-annealing method, recall that it aims to maximize the core-quality function $\sum_{ij} a_{ij}x_ix_j$ over $C_{\alpha,\beta}$, defined as in (3). To this end the method requires a uniform

discretization of the square $[0,1]^2$. In all our experiments below we choose the discretization $\{1/h, 2/h, \dots, 1\}^2$ with $h = 50$.

Algorithm 1 requires the selection of two positive scalars, α and $p > \alpha$, and the norm in the stopping criterion. In all our experiments we set $\alpha = 10$ and $p = 2\alpha$ and terminate when

$$\frac{\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|_p}{\|\boldsymbol{x}_{k+1}\|_p} = \|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|_p < 10^{-8}.$$

For the sake of brevity, we refer to the nonlinear spectral method, simulated-annealing method, degree-based method, eigenvector centrality method, and H-index k-core decomposition method as NSM, Sim-Ann, Degree, Eig, and Coreness, respectively. We point out that, in order to reduce computing time we implement Sim-Ann in parallel on four cores, whereas all other methods are run on a single computing core.

5.1. Synthetic networks. In practice, of course, it is typically not known ahead of time whether a given network contains any inherent core–periphery structure. However, in order to conduct a set of controlled tests, we begin with two classes of random networks that have a built-in core–periphery structure. The first takes the form of a stochastic block model, a widespread benchmark where community structure is imposed in block form. We then consider the new logistic core–periphery random model discussed in section 3.1. For the sake of brevity, we only compare NSM, Sim-Ann, and Degree in these synthetic tests, noting that Eig and Coreness were comparable to or less effective than Sim-Ann.

5.1.1. Stochastic block model. We consider synthetic networks that have a planted core–periphery structure arising from a stochastic block model. For the sake of consistency with previous works, we denote this ensemble of unweighted networks by $\mathrm{CP}(n, \delta, p, k)$. Each network drawn has δn core nodes and $(1-\delta)n$ periphery nodes, with $\delta \in [0,1]$. We consider two parameter settings. The first reproduces the following case analyzed in [30, sect. 5.1]: for $p \in [0,1]$ and $k \in [1, 1/\sqrt{p}]$, each edge between nodes i and j is assigned independently at random with probability kp if either i or j (or both) is/are in the periphery, and with probability $k^2 p$ if both i and j are in the core. In the second setting, edges between nodes i and j have probability kp only if both i and j are in the periphery; otherwise, they have probability $k^2 p$.

In our experiment we fix $n = 100$, $\delta = 1/2$, $p = 1/4$, and for each $k = \{1, 1.05, 1.1, \dots, 2\}$ we compute the core–periphery assignment for a network drawn from $\mathrm{CP}(n, \delta, p, k)$. Figure 3 shows the percentage of nodes correctly assigned to the ground-truth core–periphery structure in the two settings described above. In the left and center panels of the figure, NSM is indicated by red circles, Sim-Ann by blue crosses, and Degree by black plus symbols. Each plot shows the mean over 20 random instances of $\mathrm{CP}(n, \delta, p, k)$, for each fixed value of k. The right plot of the figure shows the median execution time of the three methods over 20 runs. We see that all three approaches give similar results in the first parameter regime, whereas Degree and NSM outperform Sim-Ann in the second. Degree is the cheapest method, and Sim-Ann is around two orders of magnitude more expensive than NSM.

5.1.2. Logistic core–periphery random model. We now consider the unweighted logistic core–periphery random model described in subsection 3.1. More precisely, given n, s, and t,

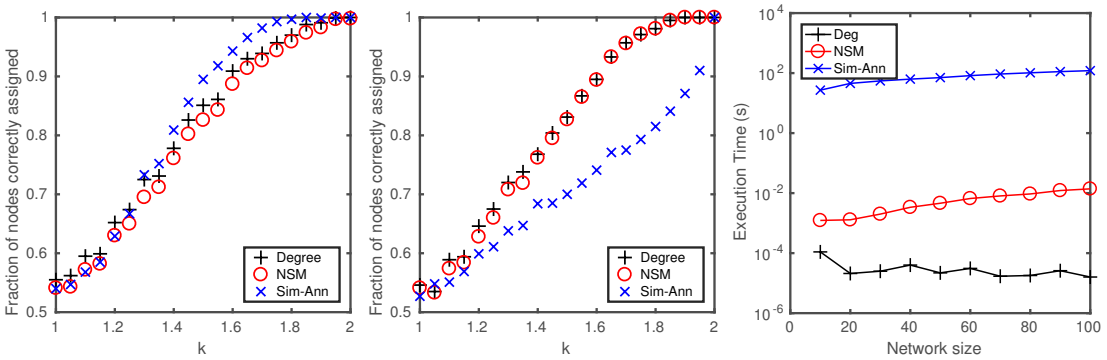


Figure 3. *Experiments on stochastic block model graphs. Left and center: fraction of nodes correctly assigned to the core–periphery ground-truth versus model parameter k for three methods: our nonlinear spectral method (red circles), the Degree vector (black plus symbols), and the simulated-annealing technique of Rombach et al. [30] (blue crosses). Each value is the mean over 20 random instances. Each network has 100 nodes and k ranges in $\{1, 1.05, 1.1, \ldots, 2\}$. Left: nodes within the periphery and between core and periphery connected with probability $k/4$, and nodes within the core connected with probability $k^2/4$. Center: nodes within the core and between core and periphery connected with probability $k^2/4$, and nodes within the periphery connected with probability $k/4$. Right: median execution time of the three methods over 20 instances, when the number of nodes varies within $\{10, 20, \ldots, 100\}$.*

we sample from the family $\mathrm{LCP}(n, s, t)$ of random graphs with n nodes such that an edge between any pair of nodes i and j is assigned independently at random with probability

$$\mathbb{P}(i \sim j) = \sigma_{s,t}\left(\frac{1}{n}\max\{n-i, n-j\}\right), \quad \text{where} \quad \sigma_{s,t}(x) = \frac{1}{1+e^{-s(x-t)}}.$$

Unlike the stochastic block model discussed in section 5.1.1, if s is not too large, the logistic core–periphery model does not give rise to a binary core–periphery structure. Instead, it uses a sliding scale for the nodes where node n is at the center of the core and node 1 is the most peripheral. We therefore look at the ability of the algorithms to recover a suitable ordering.

In our experiment we fix $s = 7$, $t = 2/3$ and let the dimension n vary within $\{30, 60, 90, 120\}$. For each n we draw an instance from the ensemble $\mathrm{LCP}(n, s, t)$ and compute the core–periphery score from each of the three methods. We sort each score vector in descending order and consider the associated permutations $\boldsymbol{\pi}_1$, $\boldsymbol{\pi}_2$, and $\boldsymbol{\pi}_3$ for NSM, Degree, and Sim-Ann, respectively. We then evaluate the likelihoods $\nu(\boldsymbol{\pi}_i)$, as defined in (6). In Figure 4 we show medians and quartiles of the three likelihood ratios $\nu(\boldsymbol{\pi}_1)/\nu(\boldsymbol{\pi}_2)$ (in red), $\nu(\boldsymbol{\pi}_1)/\nu(\boldsymbol{\pi}_3)$ (in black), and $\nu(\boldsymbol{\pi}_3)/\nu(\boldsymbol{\pi}_2)$ (in blue). We see that in this test NSM outperforms Degree, which itself outperforms Sim-Ann.

5.2. Real-world datasets. In this subsection we show results for several real-world networks taken from the following different fields: social interaction, academic collaboration, transportation, Internet structure, neural connections, and protein-protein interaction. These networks are freely available online; below we describe their key features and give references for further details.

Cardiff tweets. This is an unweighted network of reciprocated Twitter mentions among

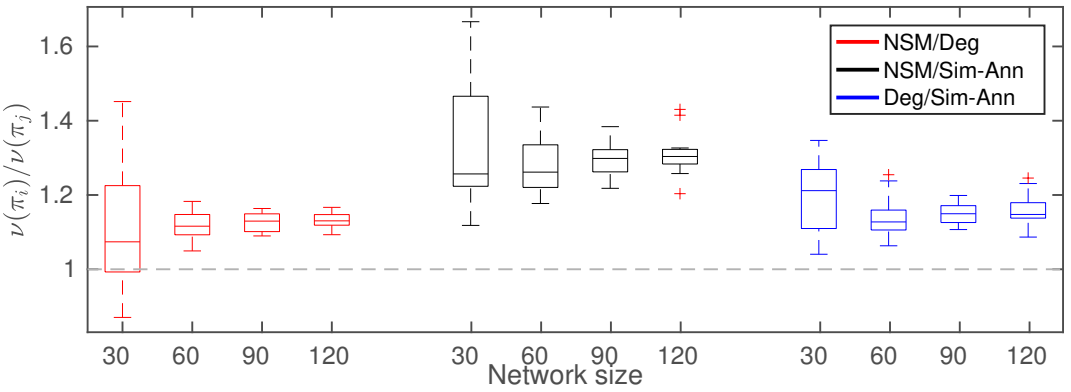


Figure 4. *Boxplots of the ratio of the likelihood $\nu(\boldsymbol{\pi})$ over 20 trials for different sizes n of the random network ranging within $\{30, 60, 90, 120\}$. Three permutation vectors $\boldsymbol{\pi}_1$, $\boldsymbol{\pi}_2$, $\boldsymbol{\pi}_3$ are obtained by sorting the entries of the score vectors obtained with* (1) *the proposed Algorithm 1 ("NSM" in the legend),* (2) *the degree vector of the graph ("Deg" in the legend), and* (3) *the simulated–annealing method of [29] ("Sim-Ann" in the legend), respectively. The three boxplot groups, with different colors, show (from left to right) the ratios $\nu(\boldsymbol{\pi}_1)/\nu(\boldsymbol{\pi}_2)$ (in red), $\nu(\boldsymbol{\pi}_1)/\nu(\boldsymbol{\pi}_3)$ (in black), and $\nu(\boldsymbol{\pi}_3)/\nu(\boldsymbol{\pi}_2)$ (in blue).*

users whose bibliographical information indicates they are associated with the city of Cardiff in the UK. Data refers to the period of October 1–28, 2014. There is a single connected component of 2685 nodes and 4444 edges. The mean degree of the network is 3.31, with a variance of 21.24 and diameter 29. This dataset is part of a larger collection from [20] of geolocated reciprocated Twitter mentions within UK cities.

Network scientists. This is a weighted co-authorship network among scholars who study network science. This network, compiled in 2006, involves 1589 authors. We use the largest connected component, which has 379 nodes and coincides with the network considered in [26]. This component contains 914 edges. Its mean degree is 4.82, with variance 15.46 and diameter 17.

Erdős. This is an instance of the Erdős collaboration unweighted network with 472 nodes representing authors. We use the largest connected component, which contains 429 nodes and 1312 edges. Its mean degree is 6.12, with variance 45.98 and diameter 11. This dataset is one of seven Erdős collaboration networks made available by Batagelj and Mrvar in the Pajek datasets collection [1] and therein referred to as "Erdos971."

Yeast. This is an unweighted protein-protein interaction network described and analyzed in [6]. Like the Erdős dataset, this network is available through the datasets collection in [1]. The whole network consists of 2361 nodes. We use the largest connected component, consisting of 2224 nodes and 6829 edges. Its mean degree is 6.14, with variance 65.76 and diameter 11.

Internet 2006. This is a symmetrized snapshot of the structure of the Internet at the level of autonomous systems, reconstructed from Border Gateway Protocol tables posted by the University of Oregon Route Views Project. This snapshot was created by Mark Newman from data for July 22, 2006 and is available from [28, 10]. The network is connected, with 22963 nodes and 48436 edges. Its mean degree is 4.2186, with variance 108.5 and diameter 11.

Jazz. This is a network of Jazz bands that performed between 1912 and 1940 obtained from "The Red Hot Jazz Archive" digital database [18]. It consists of 198 nodes (jazz bands) and 2742 edges (musicians). Mean degree is 27.7, with variance 304.6 and diameter 12.

Drugs. This is a social network of injecting drug users (IDUs) that have shared a needle within a six-month time period. This is a connected network consisting of 616 nodes and 2012 edges. The average degree is 6.5, with variance 59.17 and diameter 13; see, e.g., [25, 34].

C. elegans. This is a neural network of neurons and synapses in Caenorhabditis elegans, a type of worm. It contains 277 nodes and 2105 edges. Mean degree is 7.6, with variance 48 and diameter 6. The network was created in [8]. The data we used is collected from [2].

London trains. This is a transportation network representing connections between train stations in the city of London. The undirected weighted network that we consider here is the aggregated version of the original multilayer network. It consists of a single connected component with 369 nodes, each corresponding to a train station. Direct connections between stations form a set of 430 edges with nonzero weights. Each such weight takes an integer value of 1, 2, or 3 according to the number of different types of connection, from the three possibilities of underground, overground, and Docklands Light Railway (DLR). The average degree is 2.33, with variance 1.04 and diameter 39. This network is studied in [11] and the data we used was collected from [13].

Analysis. In Figure 5 we use adjacency matrix sparsity plots to show how the three algorithms Degree, Sim-Ann, and NSM compare on five networks of varying size. In each case, the nodes are reordered in descending magnitude of core–periphery score. We see that the three methods give very different visual representations of the data, with NSM generally finding a more convincing core–periphery structure. On the Cardiff, Erdős, and Yeast networks, NSM gives a well-defined "antidiagonal contour" that essentially separates the reordered matrix into two regions. This type of behavior has been observed for other spectral reordering methods [31] but does not seem to be fully understood.

We note that the reciprocated Twitter mentions for the city of Cardiff show a strong core–periphery structure in all three orderings. Very similar results were observed for all 10 city-based networks of reciprocated Twitter mentions collected in [20]. (For the sake of brevity, we do not show those results here.)

To quantify the quality of the core–periphery assignments and to compare different methods on all of the datasets, we perform two further tests.

In Figure 6 we show the *core–periphery profile* of five networks obtained with different methods. This analysis is inspired by the core–periphery profiling approach proposed in [12] and evaluates the core–periphery profile function $\gamma(\boldsymbol{x})$ associated with a given core–periphery quality vector $\boldsymbol{x} > 0$, defined as

$$\gamma(\boldsymbol{x})_k = \frac{\sum_{i,j=1}^{k} A_{\pi_i,\pi_j}}{\sum_{i=1}^{k}\sum_{j=1}^{n} A_{\pi_i,j}}, \tag{12}$$

where $\boldsymbol{\pi}$ is a permutation such that $x_{\pi_1} \leq \cdots \leq x_{\pi_n}$. In words, if for each k we regard $\pi_1, \pi_2, \ldots, \pi_k$ as peripheral nodes and $\pi_{k+1}, \pi_{k+2}, \ldots, \pi_n$ as core nodes, then $\gamma(\boldsymbol{x})_k$ in (12) measures the ratio of periphery-periphery links to periphery-all links. Hence, $\boldsymbol{x}$ reveals a strong core–periphery structure if $\gamma(\boldsymbol{x})_k$ remains small for large k.

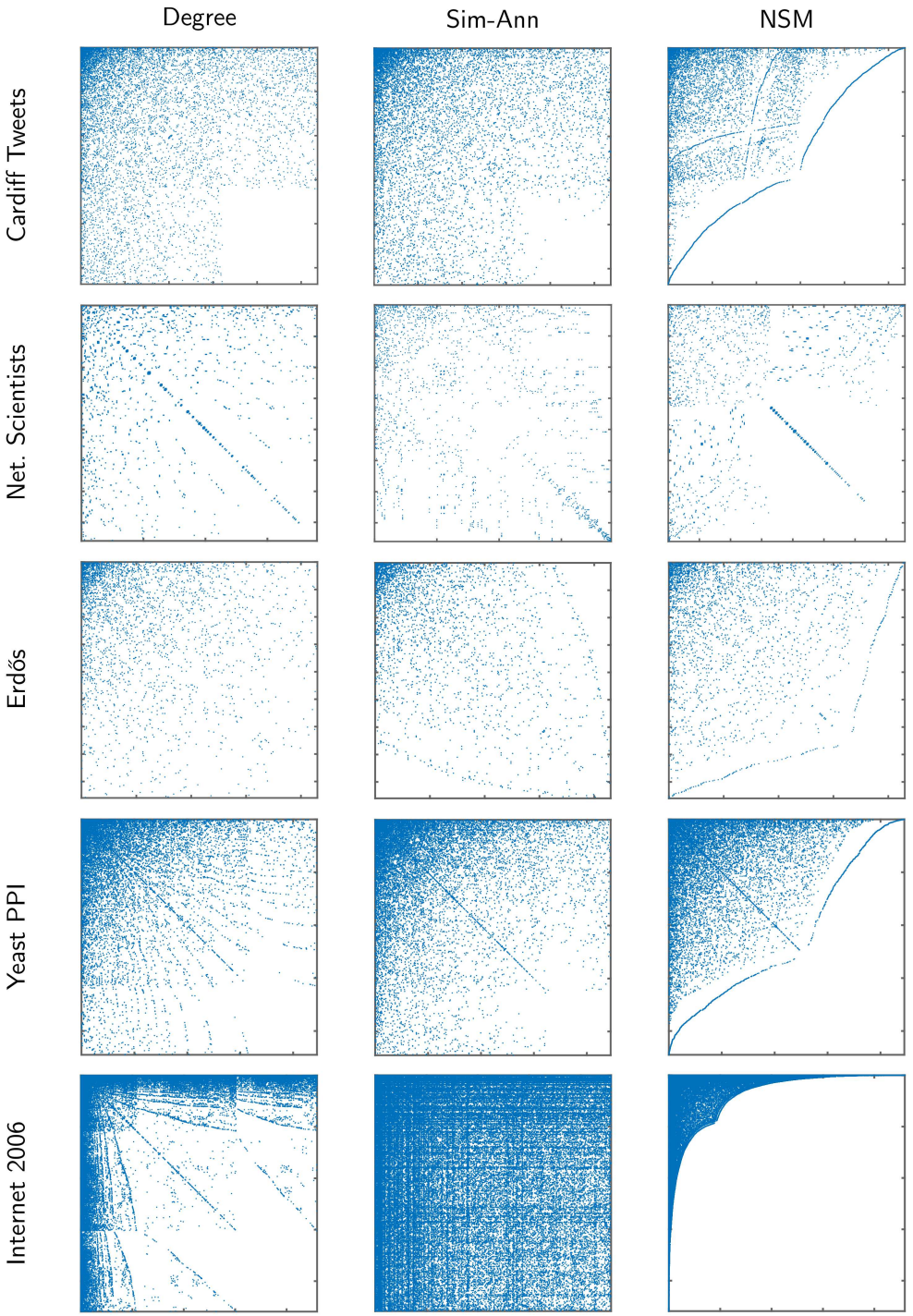


Figure 5. *Sparsity plots of adjacency matrices for various real-world networks. Each row of three plots corresponds to a different dataset. Each column corresponds to a different ordering of the network nodes. Left column: nodes ordered by decreasing degree. Middle column: nodes ordered by aggregate core score of* [30]. *Right column: nodes ordered by the nonlinear spectral method in Algorithm* 1.

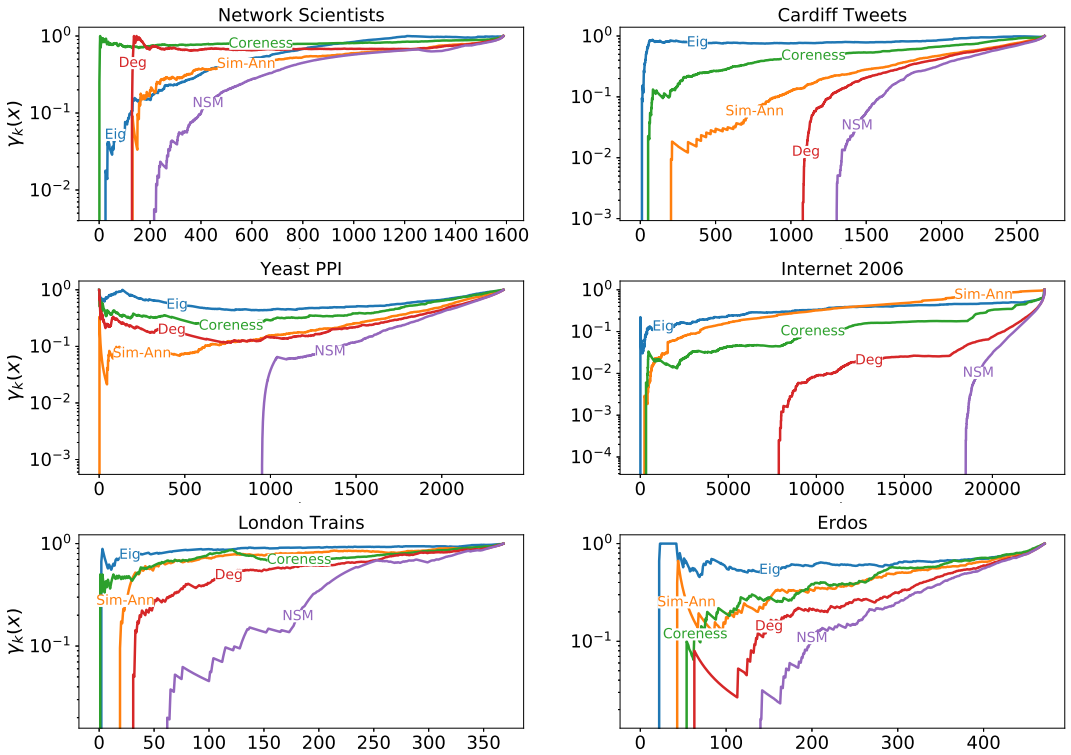


Figure 6. *Core–periphery profile $\gamma(\boldsymbol{x})$ of three networks, where $\boldsymbol{x}$ is the core-score vector obtained with five different methods.*

The quantity $\gamma(\boldsymbol{x})_k$ also has an interesting random walk interpretation. Given $\boldsymbol{x} > 0$, let $S_k = \{\pi_1, \ldots, \pi_k\}$ and consider the standard random walk on G with transition matrix $T = (t_{ij})$ defined by $t_{ij} = a_{ij}/\sum_k a_{ik}$. As the graph is undirected, the stationary distribution of the chain $\boldsymbol{y} > 0$ is the (normalized) degree vector $\boldsymbol{y} = \boldsymbol{d}/\sum_i d_i$. Therefore,

$$\gamma(\boldsymbol{x})_k = \frac{\sum_{i,j\in S_k} y_i t_{ij}}{\sum_{i\in S_k} y_i},$$

which corresponds to the persistence probability of S_k, i.e., the probability that a random walker currently in any of the nodes of S_k remains in S_k at the next time step. Clearly, $\gamma(\boldsymbol{x})_k \le \gamma(\boldsymbol{x})_h$ if $k \le h$ and $\gamma(\boldsymbol{x})_n = 1$ for any $\boldsymbol{x}$. This further justifies why having small values of $\gamma(\boldsymbol{x})_k$ for large values of k is a good indication of the presence of a core and periphery [12]. Figure 6 shows that the smallest core–periphery profile $\gamma(\boldsymbol{x})$ is obtained when $\boldsymbol{x}$ is the output of Algorithm 1. This confirms the behavior shown in Figure 5—Algorithm 1 is the most effective at transforming each network into core–periphery form.

Finally, in Figure 7 we compare the value of the core–periphery quality function f_∞ on all of the datasets and all of the methods. To cover networks of different sizes we plot the normalized value

$$\widetilde{f}_\infty(\boldsymbol{x}) = \frac{f_\infty(\boldsymbol{x})}{(\max_i x_i)\sum_{ij} a_{ij}}.$$

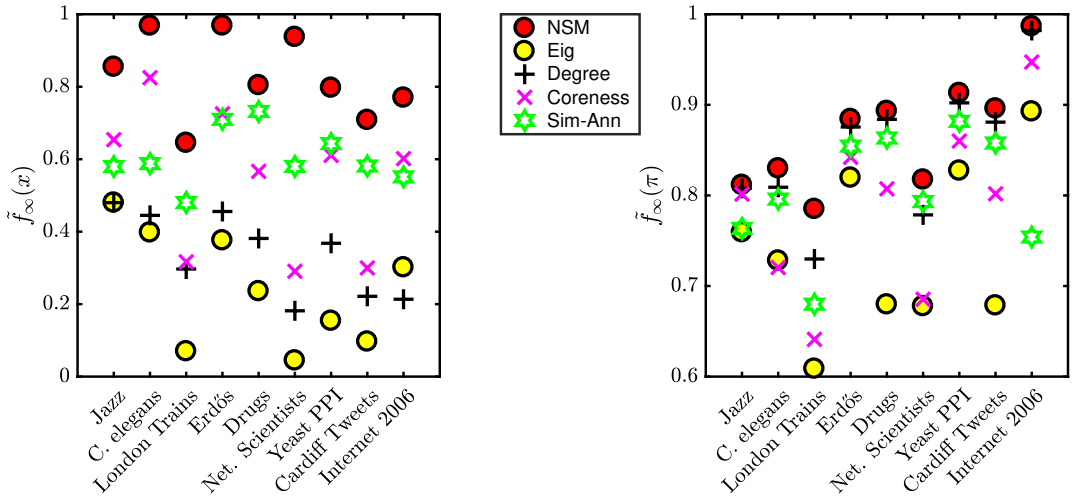


Figure 7. *Normalized core–periphery quality measure $\widetilde{f}_\infty$ for different methods. Left: value of $\widetilde{f}_\infty(\boldsymbol{x})$ for different core-score vectors $\boldsymbol{x}$ normalized so that* $\max(\boldsymbol{x}) = 1$ *and* $\min(\boldsymbol{x}) = 0$. *Right: value of $\widetilde{f}_\infty(\boldsymbol{\pi})$ where $\boldsymbol{\pi}$ is the permutation that sorts the entries of $\boldsymbol{x}$ in increasing order.*

Precisely, the figure shows two plots. On the left we evaluate $\widetilde{f}_\infty(\boldsymbol{x})$ on core-score vectors $\boldsymbol{x}$ obtained by the methods, rescaled so that $\max(\boldsymbol{x}) = 1$ and $\min(\boldsymbol{x}) = 0$, whereas on the right we evaluate $\widetilde{f}_\infty$ on the corresponding permutation vector $\boldsymbol{\pi}$ such that $x_{\pi_1} \leq \cdots \leq x_{\pi_n}$. The NSM is designed to optimize f_α (recall $\alpha = 10$ in our experiments), so the value of $\widetilde{f}_\infty$ is significantly larger than the value obtained with other methods. We see that NSM continues to give the best results when $\widetilde{f}_\infty$ is evaluated on the associated permutation.

London transportation network. In a final experiment we examine in more detail the London transportation network, where further nodal information is available, using the Perron–Frobenius eigenvector of the adjacency matrix as a baseline for comparison. As discussed in section 2.2, this vector can be viewed as both a centrality measure and a core–periphery score, and it corresponds to a linear counterpart of our approach, retrieved when $\alpha \to 0$. We compare central nodes in the London train network obtained from Eig, NSM, and Sim-Ann. Note that important nodes for both Eig and Sim-Ann are somewhat related to the concept of centrality, as both methods aim to maximize the same core-quality function $\sum_{ij} A_{ij}x_i x_j$ but force different constraints sets, $\Omega = \{\boldsymbol{x} : \|\boldsymbol{x}\|_2 = 1\}$ for Eig and $\Omega = C_{\alpha,\beta}$ for Sim-Ann. On the London train network we find that the core assignments of these two techniques highly correlate. The important nodes captured by the NSM are, instead, more directly related to core and periphery features and significantly differ from those captured by Eig and Sim-Ann. For the sake of brevity we do not make comparisons with other methods here.

In Figure 8 we display the edges (underground, overground, and DLR connections) in physical space, with darker lines indicating larger weight. The top 10 stations are highlighted for the three measures, with node size proportional to the value. Although four stations are highlighted in all three plots, there are clear differences in the results. Eigenvector centrality and Sim-Ann produce similar results, focusing on a set of stations that are geographically close, whereas NSM assigns higher core scores to some stations at key intersections that are further from the city center.

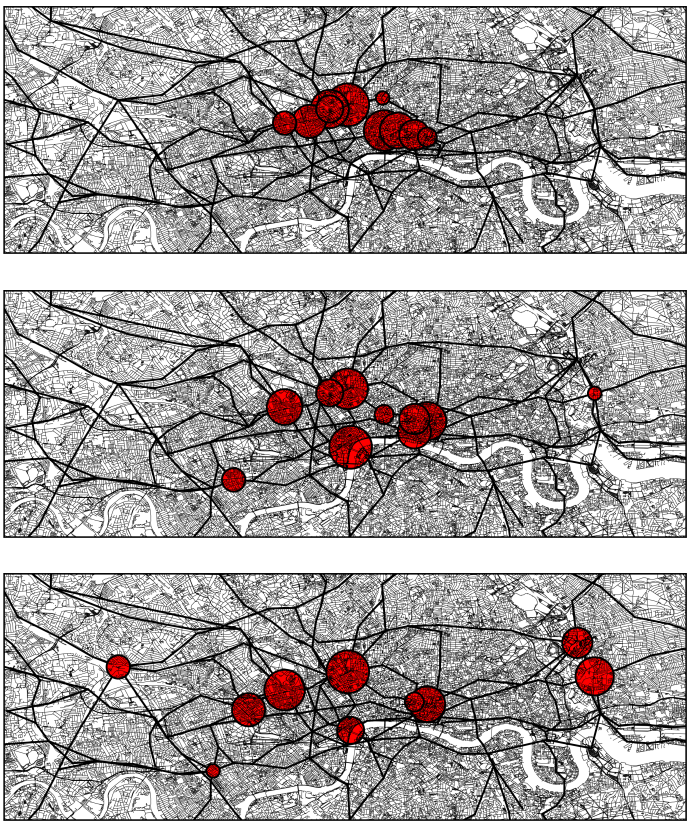


Figure 8. *Physical layout of the aggregate London transportation network. Red circles indicate the* 10 *nodes with highest core–periphery score for the three algorithms, with node size proportional to score. From top to bottom: eigenvector centrality, Sim-Ann, and NSM.*

To underscore the differences that are apparent in Figure 8, in Table 1 we list the names of the top 10 stations drawn in Figure 8, for each of the three rankings. Whereas four major stations, namely Baker Street, King's Cross, Liverpool Street, and Moorgate, are shared by all three methods, four stations appearing in the NSM top 10 do not appear in the top 10 of the other two methods. Table 1 also gives the overall number of passengers entering or exiting each station. A station may play more than one role (underground, overground, or DLR), and we list the most recently reported total annual usage. More precisely, we sum the records for

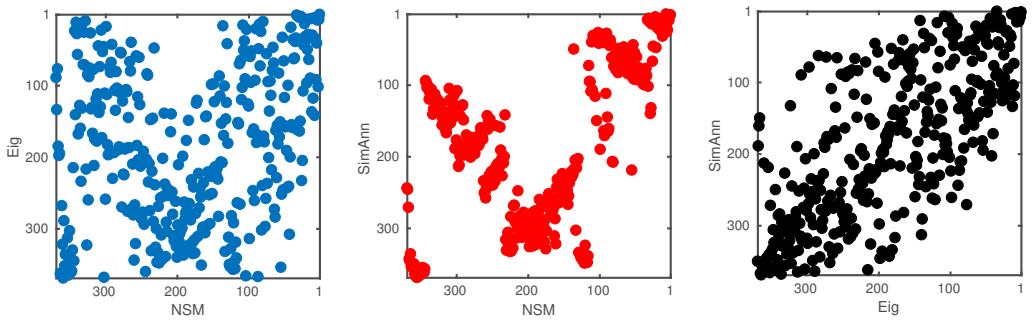
- London Underground annual entries and exits for 2016,
- National Rail annual entries and exits for 2016–2017, and
- DLR annual boardings and alightings for 2016

as reported by Wikipedia in April of 2018. Numbers indicate millions of passengers. The last row shows the overall total number of passengers using the top 10 stations identified by each method. We note that none of the rankings orders the stations strictly by passenger usage. However, while the top 10 stations selected by both Eigenvector and Sim-Ann involve around 600 million passengers per year, the top 10 NSM stations involve almost 800 million passengers.

Table 1

Ten London train stations with highest core value, according to eigenvector centrality (left column), Sim-Ann (middle column), and NSM (right column), applied to the weighted London trains network. The numbers beside each station show overall (underground, overground, DLR) annual usage in millions of passengers. Totals listed in the bottom row show the sum of annual usage across the top 10 stations selected by each method. King's Cross refers to a combination of the two King's Cross and St. Pancras main-line stations and the single King's Cross St. Pancras underground station.

Eigenvector		**Sim-Ann**		**NSM**	
King's Cross	128.85	Embankment	26.84	King's Cross	128.85
Farringdon	29.75	King's Cross	128.85	Baker St.	29.75
Euston Square	14.40	Liverpool St.	138.95	West Ham	77.10
Barbican	11.97	Baker St.	29.75	Liverpool St.	138.95
Great Portland St.	86.60	Bank	96.52	Paddington	85.32
Moorgate	38.40	Moorgate	38.40	Stratford	129.01
Euston	87.16	Euston Square	14.40	Embankment	26.84
Baker St.	29.75	Gloucester Road	13.98	Willesden Junction	109.27
Liverpool St.	138.95	Farringdon	27.92	Moorgate	38.40
Angel	22.10	West Ham	77.10	Earl's Court	20.00
Total	586.09		592.70		783.48



Kendal τ	0.1442	0.2930	0.5455

Figure 9. *Top: Scatter plots comparing the ranking associated with the three core-score functions of Eigenvector centrality, Sim-Ann, and NSM. Bottom: Kendal* τ *correlation coefficient between the three pairs of rankings shown in the corresponding scatter plot.*

For a comparison across all 369 stations, Figure 9 scatter plots the rankings for the three methods in a pairwise manner. We see that the left and middle plots, NSM versus Eig and NSM versus Sim-Ann, show much less agreement than the right plot, Eig versus Sim-Ann. This is confirmed by the Kendal τ correlation coefficients between the different rankings, shown at the bottom of Figure 9.

6. Discussion. The approach in [3, 29, 35] sets up a discrete optimization problem and then applies heuristic algorithms that are not guaranteed to find a global minimum. Our work differs by relaxing the problem before addressing the computational task. We showed that a relaxed analogue of a natural discrete optimization problem allows for a globally convergent iteration that is feasible for large, sparse networks. This philosophy is in line with classical and widely used reordering and clustering methods that make use of the Fiedler or Perron–Frobenius eigenvectors [14]. However, in the core–periphery setting considered here, the

resulting relaxed problem is equivalent to an eigenvalue problem that is inherently nonlinear and is reminiscent of more recent clustering and reordering techniques that exploit nonlinear eigenvectors [7, 32, 33, 34]. Hence, we developed new results in nonlinear Perron–Frobenius theory in order to derive and analyze the algorithm.

As with all clustering, partitioning, and reordering methods in network science, there is no absolute gold standard against which to judge results—the underlying problems may be defined in many different ways. In this work we introduced a new random graph model that (a) gives further justification for our algorithm, and (b) provides one basis for systematic comparison of methods. Maximum likelihood results on synthetic networks with planted structure showed the effectiveness of the new method, as did qualitative visualizations and quantitative tests across a range of application areas.

Acknowledgment. We thank Mason A. Porter for supplying code that implements the algorithm in [29, 30].

REFERENCES

[1] V. Batagelj and A. Mrvar, *Pajek Datasets Collection*, http://vlado.fmf.uni-lj.si/pub/networks/data/, 2006.
[2] A. Benson, *Data*, https://www.cs.cornell.edu/~arb/data/index.html.
[3] S. P. Borgatti and M. G. Everett, *Models of core/periphery structures*, Social Networks, 21 (2000), pp. 375–395.
[4] S. P. Borgatti, M. G. Everett, and L. C. Freeman, *UCINET for Windows: Software for Social Network Analysis*, Analytic Technologies, Harvard, MA, 2002.
[5] J. P. Boyd, W. J. Fitzgerald, M. C. Mahutga, and D. A. Smith, *Computing continuous core/periphery structures for social relations data with MINRES/SVD*, Social Networks, 32 (2010), pp. 125–137.
[6] D. Bu, Y. Zhao, L. Cai, H. Xue, X. Zhu, H. Lu, J. Zhang, S. Sun, L. Ling, N. Zhang, and R. Chen, *Topological structure analysis of the protein–protein interaction network in budding yeast*, Nucleic Acids Research, 31 (2003), pp. 2443–2450.
[7] T. Bühler and M. Hein, *Spectral clustering based on the graph p-Laplacian*, in Proceedings of the 26th Annual International Conference on Machine Learning (ICML), ACM, New York, 2009, pp. 81–88.
[8] Y. Choe, B. McCormick, and W. Koh, *Network connectivity analysis on the temporally augmented C. elegans web: A pilot study*, Society for Neuroscience Abstracts, 30 (2004).
[9] P. Csermely, A. London, L.-Y. Wu, and B. Uzzi, *Structure and dynamics of core/periphery networks*, J. Complex Networks, 1 (2013), pp. 93–123.
[10] T. A. Davis and Y. Hu, *The university of Florida sparse matrix collection*, ACM Trans. Math. Softw. (TOMS), 38 (2011), 1.
[11] M. De Domenico, A. Solé-Ribalta, S. Gómez, and A. Arenas, *Navigability of interconnected networks under random failures*, Proc. Natl. Acad. Sci. USA, 111 (2014), pp. 8351–8356.
[12] F. Della Rossa, F. Dercole, and C. Piccardi, *Profiling core-periphery network structure by random walkers*, Scientific Reports, 3 (2013), p. 1467.
[13] M. De Domenico, *Multilayer Network Dataset*, https://comunelab.fbk.eu/data.php, 2017.
[14] E. Estrada and D. J. Higham, *Network properties revealed through matrix functions*, SIAM Rev., 52 (2010), pp. 696–714, https://doi.org/10.1137/090761070.
[15] A. Gautier, Q. Nguyen, and M. Hein, *Globally optimal training of generalized polynomial neural networks with nonlinear spectral methods*, in Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS '16), Barcelona, 2016, pp. 1695–1703.
[16] A. Gautier and F. Tudisco, *The Contractivity of Cone-Preserving Multilinear Mappings*, preprint, https://arxiv.org/abs/1808.04180, 2018.

[17] A. Gautier, F. Tudisco, and M. Hein, *The Perron-Frobenius Theorem for Multi-homogeneous Mappings*, https://arxiv.org/abs/1801.05034, 2018.
[18] P. M. Gleiser and L. Danon, *Community structure in jazz*, Adv. Complex Systems, 6 (2003), pp. 565–573.
[19] P. Grindrod, *Range-dependent random graphs and their application to modeling large small-world proteome datasets*, Phys. Rev. E, 66 (2002), 066702.
[20] P. Grindrod and T. E. Lee, *Comparison of social structures within cities of very different sizes*, R. Soc. Open Sci., 3 (2016), 150526.
[21] J. Jia and A. R. Benson, *Random spatial network models for core-periphery structure*, in Proceedings of the 12th ACM International Conference on Web Search and Data Mining (WSDM '19), ACM, New York, 2019, pp. 366–374.
[22] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, *Identification of influential spreaders in complex networks*, Nature Physics, 6 (2010), pp. 888–893.
[23] L. Lü, T. Zhou, Q.-M. Zhang, and H. E. Stanley, *The H-index of a network node and its relation to degree and coreness*, Nature Commun., 7 (2016), 10168.
[24] R. J. Mondragón, *Network partition via a bound of the spectral radius*, J. Complex Networks, 5 (2017), pp. 513–526.
[25] A. Neaigus, *The network approach and interventions to prevent HIV among injection drug users*, Public Health Reports, 113 (Suppl. 1) (1998), pp. 140–150.
[26] M. E. J. Newman, *Finding community structure in networks using the eigenvectors of matrices*, Phys. Rev. E, 74 (2006), 036104.
[27] M. E. J. Newman, *Equivalence between modularity optimization and maximum likelihood methods for community detection*, Phys. Rev. E, 94 (2016), 052315.
[28] M. J. Newman, *Network Data*, http://www-personal.umich.edu/~mejn/netdata/.
[29] M. P. Rombach, M. A. Porter, J. H. Fowler, and P. J. Mucha, *Core-periphery structure in networks*, SIAM J. Applied Math., 74 (2014), pp. 167–190, https://doi.org/10.1137/120881683.
[30] P. Rombach, M. A. Porter, J. H. Fowler, and P. J. Mucha, *Core-periphery structure in networks (revisited)*, SIAM Rev., 59 (2017), pp. 619–646, https://doi.org/10.1137/17M1130046.
[31] A. Taylor and D. J. Higham, *NESSIE: Network example source supporting innovative experimentation*, in Network Science: Complexity in Nature and Technology, E. Estrada, M. Fox, D. J. Higham, and G.-L. Oppo, eds., Springer, Berlin, 2010, Chapter 5, pp. 85–106.
[32] F. Tudisco, F. Arrigo, and A. Gautier, *Node and layer eigenvector centralities for multiplex networks*, SIAM J. Appl. Math., 78 (2018), pp. 853–876, https://doi.org/10.1137/17M1137668.
[33] F. Tudisco and M. Hein, *A nodal domain theorem and a higher-order Cheeger inequality for the graph p-Laplacian*, EMS J. Spectral Theory, 8 (2018), pp. 883–908.
[34] F. Tudisco, P. Mercado, and M. Hein, *Community detection in networks via nonlinear modularity eigenvectors*, SIAM J. Appl. Math., 78 (2018), pp. 2393–2419, https://doi.org/10.1137/17M1144143.
[35] X. Zhang, T. Martin, and M. E. J. Newman, *Identification of core-periphery structure in networks*, Phys. Rev. E, 91 (2015), 032803.