SIAM J. Sci. Comput.
Vol. 38, No. 1, pp. B25–B49

UPDATING AND DOWNDATING TECHNIQUES FOR OPTIMIZING NETWORK COMMUNICABILITY*

FRANCESCA ARRIGO† AND MICHELE BENZI‡

Abstract. The total communicability of a network (or graph) is defined as the sum of the entries in the exponential of the adjacency matrix of the network, possibly normalized by the number of nodes. This quantity offers a good measure of how easily information spreads across the network, and can be useful in the design of networks having certain desirable properties. The total communicability can be computed quickly even for large networks using techniques based on the Lanczos algorithm. In this work we introduce some heuristics that can be used to add, delete, or rewire a limited number of edges in a given sparse network so that the modified network has a large total communicability. To this end, we introduce new edge centrality measures, which can be used as a guide in the selection of edges to be added or removed. Moreover, we show experimentally that the total communicability provides an effective and easily computable measure of how "well-connected" a sparse network is.

Key words. network analysis, eigenvector centrality, subgraph centrality, total communicability, edge centrality, free energy, natural connectivity

AMS subject classifications. 05C82, 15A16, 65F60

DOI. 10.1137/140991923

1. Introduction.

Network models are nowadays ubiquitous in the natural, information, social, and engineering sciences. The last 15 years or so have seen the emergence of the vast, multidisciplinary field of network science, with contributions from a wide array of researchers including physicists, mathematicians, computer scientists, engineers, biologists, and social scientists [3, 18, 34]. Applications of network science can be found in biology, public health, social network analysis, homeland security, economics, the humanities, marketing, and information retrieval. Network analysis is also an essential ingredient in the design of information, communication, and transportation networks, as well as in energy-related disciplines such as power grid maintenance, control, and optimization [35]. Graph theory and linear algebra provide abstractions and quantitative tools that can be employed in the analysis and design of large and complex network models.

Real-world networks are characterized by structural properties that make them very different from both regular graphs and completely random graphs. Real networks frequently exhibit a highly skewed degree distribution (often following a power law), small diameter, high clustering coefficient (the latter two properties together are often referred to as the *small world* property), the presence of motifs, communities, and other signatures of complexity.

Some of the basic questions in network analysis concern node and edge centrality, community detection, communicability, and diffusion [11, 18, 34]. Related to these are the important notions of network robustness (or its opposite, vulnerability) and

*Submitted to the journal's Computational Methods in Science and Engineering section October 17, 2014; accepted for publication (in revised form) October 6, 2015; published electronically January 6, 2016.
http://www.siam.org/journals/sisc/38-1/99192.html
†Department of Science and High Technology, University of Insubria, Como 22100, Italy (francesca.arrigo@uninsubria.it).
‡Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322 (benzi@mathcs.emory.edu). The work of this author was supported by National Science Foundation grants DMS-1115692 and DMS-1418889.

connectivity [13]. These latter properties refer to the degree of resiliency displayed by the network in the face of random accidental failures or deliberate, targeted attacks, which can be modeled in terms of edge or node removal. Generally speaking, it is desirable to design networks that are at the same time highly sparse (in order to reduce costs) and highly connected, meaning that disconnecting or disrupting the network would require the removal of a large number of edges. Such networks should not contain bottlenecks, and they should allow for the rapid exchange of communication between nodes. Expander graphs [17, 28] are an important class of graphs with such properties.

In this paper we describe some techniques that can be brought to bear on the problems described above and related questions. Our approach is based on the notion of *total communicability* of a network, which was introduced in [7] on the basis of earlier work by Estrada and Hatano [20, 21]. Total communicability, defined as the (normalized) sum of the entries in the exponential of the adjacency matrix of the network, provides a global measure of how well the nodes in a graph can exchange information. Communicability is based on the number and length of graph walks connecting pairs of nodes in the network. Pairs of nodes (i, j) with high communicability correspond to large entries $[e^A]_{ij}$ in the matrix exponential of A, the adjacency matrix of the network.

Total network communicability can also be used to measure the connectivity of the network as a whole. For instance, given two alternative network designs (with a similar "budget" in terms of number of candidate edges), one can compare the two designs by computing the respective total communicabilities and pick the network with the highest one, assuming that a well-connected network with high node communicability is the desired goal. It is important to stress that the total communicability of a network can be efficiently computed or estimated even for large networks using Lanczos- or Arnoldi-based algorithms without having to compute any individual entry of e^A (only the ability to perform matrix-vector products with A is required). Of course, total communicability is only one of a number of possible metrics that can be used to quantify network robustness and effectiveness at diffusing information, and we are not claiming that it is necessarily the best one for all types of networks. Indeed, it is fair to say that the comparative study of such metrics is still in its infancy, and much work remains to be done in this area.

In this paper we consider three different problems. Let $G = (V, E)$ be a connected, undirected, and sparse graph. The *downdating problem* consists of selecting an edge (i, j) to be removed from the network so as to minimize the decrease in its total communicability while preserving its connectedness.

The goal when tackling the *updating problem*, on the other hand, is to select a pair of nodes $i \neq j$ such that $(i, j) \notin E$ in such a way that the increase in the total communicability of the network is maximized.

Finally, the *rewiring problem* has the same goal as the updating problem, but it requires the selection of two modifications which constitute the downdate-then-update step to be performed.

The importance of the first two problems for network analysis and design is obvious. We note that an efficient solution to the second problem would also suggest how to proceed if the goal were to identify existing edges whose removal would *maximize* the decrease in communicability, which could be useful, e.g., in planning antiterrorism operations or public health policies (see, e.g., [40, 41]). The third problem is motivated by the observation that for transportation networks (e.g., flight routes) it is sometimes desirable to redirect edges in order to improve the performance (i.e., increase the number of travelers) without increasing the costs by too much. Hence, in such cases, one wants to eliminate a route used only by a few travelers and to add a route that may be used by a lot of people.

The above problems may arise not only in the design of infrastructural networks (such as telecommunication or transportation networks), but also in other contexts. For instance, in social networks the addition of a friendship/collaborative tie may dramatically change the structure of the network, leading to a more cohesive group, and hence preventing the splitting of the community into smaller subgroups.

The work is organized as follows. Section 2 contains some basic facts from linear algebra and graph theory and introduces the modifications of the adjacency matrix we will perform. In this section we also provide further justification for the use of the total network communicability as the objective function. In section 3 we describe bounds for the total communicability via the Gauss–Radau quadrature rule, and we show how these bounds change when a rank-two modification of the adjacency matrix is performed. Section 4 is devoted to the introduction of the methods to controllably modify the graph in order to adjust the value of its total communicability. Numerical studies to assess the effectiveness and performance of the techniques introduced are provided in section 5 for both synthetic and real-world networks. In section 6 we discuss the evolution of a popular measure of network connectivity, known as the *free energy* (or *natural connectivity*), when the same modifications are performed. This section provides further evidence that motivates the use of the total communicability as a measure of connectivity. Finally, in section 7 we draw conclusions and describe future directions.

2. Background and definitions.

In this section we provide some basic definitions, notation, and properties associated with graphs.

A *graph* or *network* $G = (V, E)$ is defined by a set of n nodes (vertices) V and a set of m edges $E = \{(i, j)|i, j \in V\}$ between the nodes. An edge is said to be *incident* to a vertex i if there exists a node $j \neq i$ such that either $(i, j) \in E$ or $(j, i) \in E$. The *degree* of a vertex, denoted by d_i, is the number of edges incident to i in G. The graph is said to be *undirected* if the edges are formed by unordered pairs of vertices. A *walk* of length k in G is a set of nodes $i_1, i_2, \ldots, i_k, i_{k+1}$ such that for all $1 \le l \le k$, $(i_l, i_{l+1}) \in E$. A *closed walk* is a walk for which $i_1 = i_{k+1}$. A *path* is a walk with no repeated nodes. A graph is *connected* if there is a path connecting every pair of nodes. A graph with unweighted edges, no self-loops (edges from a node to itself), and no multiple edges is said to be *simple*. Throughout this work, we will consider undirected, simple, and connected networks.

Every graph can be represented as a matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$, called the *adjacency matrix* of the graph. The entries of the adjacency matrix of an unweighted graph $G = (V, E)$ are

$$a_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases} \qquad \forall i, j \in V.$$

If the network is simple, the diagonal elements of the adjacency matrix are all equal to zero. In the special case of an undirected network, the associated adjacency matrix is symmetric, and thus its eigenvalues are real.

We label the eigenvalues in nonincreasing order: $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n$. Since A is a real-valued, symmetric matrix, we can decompose A into $A = Q\Lambda Q^T$, where Λ is a diagonal matrix containing the eigenvalues of A and $Q = [\mathbf{q}_1, \ldots, \mathbf{q}_n]$ is orthogonal, where $\mathbf{q}_i$ is an eigenvector associated with λ_i. Moreover, if G is connected, A is irreducible, and from the Perron–Frobenius theorem [33, Chapter 8] we deduce that $\lambda_1 > \lambda_2$ and that the leading eigenvector $\mathbf{q}_1$, sometimes referred to as the *Perron vector*, can be chosen such that its components $q_1(i)$ are positive for all $i \in V$.

We can now introduce the basic operations which will be performed on the adjacency matrix A associated with the network $G = (V, E)$. We define the *downdating* of the edge $(i, j) \in E$ as the removal of this edge from the network. The resulting graph $\widehat{G} = (V, \widehat{E})$, which may be disconnected, has adjacency matrix

$$\widehat{A} = A - UW^T, \qquad U = [\mathbf{e}_i, \mathbf{e}_j], \quad W = [\mathbf{e}_j, \mathbf{e}_i],$$

where here and in the rest of this work the vectors $\mathbf{e}_i$, $\mathbf{e}_j$ represent the ith and jth vectors of the standard basis of $\mathbb{R}^n$, respectively.

Similarly, let $(i, j) \in \overline{E}$ be an element in the complement of E. We will call this element a *virtual edge* for the graph G. We can construct a new graph $\tilde{G} = (V, \tilde{E})$ obtained from G by adding the virtual edge (i, j) to the graph. This procedure will be referred to as the *updating* of the virtual edge (i, j). The adjacency matrix of the resulting graph is

$$\tilde{A} = A + UW^T, \qquad U = [\mathbf{e}_i, \mathbf{e}_j], \quad W = [\mathbf{e}_j, \mathbf{e}_i].$$

Hence, these two operations can both be described as rank-two modifications of the adjacency matrix of the original graph.

The operation of downdating an edge and successively updating a virtual edge will be referred to as *rewiring*.

Remark 1. These operations are all performed in a symmetric fashion, since in this paper we consider exclusively undirected networks.

2.1. Centrality and total communicability. One of the main goals when analyzing a network is to identify the most influential nodes in the network. Over the years, various measures of the importance, or centrality, of nodes have been developed [11, 18, 34]. In particular the *(exponential) subgraph centrality* of a node i (see [22]) is defined as the ith diagonal element of the matrix exponential [27]:

$$e^A = I + A + \frac{A^2}{2!} + \cdots = \sum_{k=0}^{\infty} \frac{A^k}{k!},$$

where I is the $n \times n$ identity matrix. As is well known in graph theory, given an adjacency matrix A of an unweighted network and $k \in \mathbb{N}$, the element $(A^k)_{ij}$ counts the total number of walks of length k starting from node i and ending at node j. Therefore, the subgraph centrality of node i counts the total number of closed walks centered at node i, weighting walks of length k by a factor $\frac{1}{k!}$, and hence giving more importance to shorter walks. The subgraph centrality then accounts for the returnability of the information to the node which was the source of this same information. Likewise, the off-diagonal entries of the matrix $(e^A)_{ij}$ (*subgraph communicability* of nodes i and j) account for the ability of nodes i and j to exchange information [20, 21].

Starting from these observations and with the aim of reducing the cost of the computation of the rankings, in [7] it was suggested to use as a centrality measure the *total communicability of a node* i, defined as the ith entry of the vector $e^A\mathbf{1}$, where $\mathbf{1}$ denotes the vector of all ones:

$$TC(i) := [e^A\mathbf{1}]_i = \sum_{j=1}^{n} \left[e^A\right]_{ij}. \tag{2.1}$$

This measure of centrality is given by a weighted sum of walks from every node in the network (including node i itself), and thus quantifies both the ability of a node to spread information across the network and the returnability of the information to the node itself.

The value resulting from summing these quantities over all the nodes can be interpreted as a global measure of how effectively the communication takes place across the whole network. This index is called *total (network) communicability* [7] and can be written as

$$TC(A) := \mathbf{1}^T e^A \mathbf{1} = \sum_{i=1}^{n}\sum_{j=1}^{n}(e^A)_{ij} = \sum_{k=1}^{n} e^{\lambda_k}(\mathbf{q}_k^T\mathbf{1})^2. \tag{2.2}$$

This value can be efficiently computed, e.g., by means of a Krylov method as implemented in Güttel's MATLAB toolbox `funm_kryl` (see [1, 26]) or by Lanczos-based techniques as discussed below. In the toolbox [26] an efficient algorithm for evaluating $f(A)\mathbf{v}$ is implemented; with this method the vector $e^A\mathbf{1}$ can be constructed in roughly $O(n)$ operations (note that the prefactor can vary for different types of networks) and the total communicability is easily derived.

As is clear from its definition, the value of $TC(A)$ may be very large. Several normalizations have been proposed; the simplest is the normalization by the number of nodes n (see [7]), which we will use throughout the paper. It is easy to prove that the normalized total communicability satisfies

$$\frac{1}{n}\sum_{i=1}^{n}\left(e^A\right)_{ii} \le \frac{TC(A)}{n} \le e^{\lambda_1}, \tag{2.3}$$

where the lower bound is attained by the graph with n nodes and no edges and the upper bound is attained by the complete graph with n nodes.

Remark 2. The last equality in (2.2) shows that the main contribution to the value of $TC(A)$ is likely to come from the term $e^{\lambda_1}\|\mathbf{q}_1\|_1^2$.

2.2. Rationale for targeting the total communicability. As already mentioned, total communicability provides a good measure of how efficiently information (in the broad sense of the term) is diffused across the network. Typically, very high values of $TC(A)$ are observed for highly optimized infrastructure networks (such as airline routes or computer networks) and for highly cohesive social and information networks (such as certain types of collaboration networks). Conversely, the total network communicability is relatively low for spatially extended, grid-like networks (such as many road networks) or for networks that consist of two or more communities with poor communication between them (such as the Zachary network).[1] As a further example, reduced values of the communicability between different brain regions have been detected in stroke patients compared to healthy individuals [14]. We refer the reader to [21] for an extensive survey on communicability, including applications for which it has been found to be useful.

Another reason in support of the use of total communicability as an objective function is that it is closely related to the *natural connectivity* (or *free energy*) of the network, while being dramatically easier to compute; see section 6. Sparse networks

[1]Numerical values of the normalized total network communicability for a broad collection of networks are reported in the experimental sections of this paper, in the supplementary material to this paper, and in [7].

with high values of $TC(A)$ are very well connected and thus less likely to be disrupted by either random failures or targeted attacks leading to the loss of edges. This justifies trying to design sparse networks with high values of the total communicability.

An important observation is that the total network communicability $TC(A)$ can be interpreted in at least two different ways. Since it is given by the sum of all the pairwise communicabilities $C(i, j) = [e^A]_{ij}$, it is a global measure of the ability of the network to diffuse information. However, recalling the definition (2.1) of total node communicability, the normalized total communicability can also be seen as the "average total communicability" of the nodes in the network:

$$\frac{TC(A)}{n} = \frac{1}{n}\sum_{i=1}^{n} TC(i).$$

Since the total node communicability is a centrality measure [7], our goal can then be rephrased as the problem of constructing sparse networks having high average node centrality, where the node centrality is given by the total node communicability. Since this is merely one of a large number of centrality measures proposed in the literature, a legitimate question to ask is why the total node communicability should be used instead of a different centrality index. In other words, given any node centrality function $f : V \longrightarrow \mathbb{R}_+$, we could consider instead the problem of, say, adding a prescribed number of edges so as to maximize the increase in the global average centrality

$$\bar{f} = \frac{1}{n}\sum_{i=1}^{n} f(i).$$

As it turns out, most other centrality indices either are computationally too expensive to work with (at least for large networks) or lead to objective functions which do not make much sense. The following is a brief discussion of some of the most popular centrality indices used in the field of network science.

1. **Degree:** Consider first the simplest centrality index, the degree. Obviously, adding K edges according to *any* criteria will produce exactly the same variation in the average degree of a network. Hence, one may as well add edges at random. Doing so, however, cannot be expected to be greatly beneficial if the goal is to improve the robustness or efficiency of the network.
2. **Eigenvector centrality:** Let $\mathbf{q}_1$ be the principal eigenvector of A, normalized so that $\|\mathbf{q}_1\|_2 = 1$. The eigenvector centrality of node $i \in V$ is the ith component of $\mathbf{q}_1$, denoted by $q_1(i)$. It is straightforward to see that the problem of maximizing the average eigenvector centrality

$$\frac{q_1(1) + q_1(2) + \cdots + q_1(n)}{n}$$

subject to the constraint $\|\mathbf{q}_1\|_2 = 1$ has as its only solution

$$q_1(1) = q_1(2) = \cdots = q_1(n) = \frac{1}{\sqrt{n}}.$$

This implies that A has constant row sums or, in other words, that the graph is regular—every node in G has the same degree. Hence, any heuristic aimed at maximizing the average eigenvector centrality will result in graphs that are close to being regular. However, regular graph topologies are not, per se, endowed with any especially good properties when it comes to diffusing information or being robust: think of a cycle graph, for example. Regular sparse graphs *can* be very well connected and robust (this is the case of expander graphs), but there is no reason to think that simply making the degree distribution of a given network more regular will improve its expansion properties.

3. **Subgraph centrality**: the average subgraph centrality of a network is known in the literature as the normalized *Estrada index:*

$$\frac{1}{n}EE(A) = \frac{1}{n}\mathrm{T}r(e^A) = \frac{1}{n}\sum_{i=1}^{n}[e^A]_{ii} = \frac{1}{n}\sum_{i=1}^{n}e^{\lambda_i}.$$

   It can also be interpreted as the average self-communicability of the nodes. As we mentioned, this is a lower bound for the average total communicability. Evaluation of this quantity requires knowledge of all n diagonal entries of e^A, or of all the eigenvalues of A, and is therefore much more expensive to compute. The heuristics we derive in this paper have a similar effect on $TC(A)$ as on the Estrada index, as we demonstrate in section 6. So, using subgraph centrality instead of total communicability centrality would lead to exactly the same heuristics and results, with the disadvantage that evaluating the objective function, if necessary, would be much more expensive.

4. **Katz centrality**: the Katz centrality of node $i \in V$ is defined as the ith row sum of the matrix resolvent $(I - \alpha A)^{-1}$, where the parameter α is chosen in the interval $(0, \frac{1}{\lambda_1})$, so that the power series expansion

$$(I - \alpha A)^{-1} = I + \alpha A + \alpha^2 A^2 + \cdots$$

   is convergent [30]. Since this centrality measure can be interpreted in terms of walks, using it instead of the total communicability would lead to the same heuristics and very similar results, especially when α is sufficiently close to $\frac{1}{\lambda_1}$ or if the spectral gap $\lambda_1 - \lambda_2$ is large; see [8]. Using Katz centrality, however, requires the careful selection of the parameter α, which leads to some complications. For example, after each update one needs to recompute the dominant eigenvalue of the adjacency matrix in order to check whether the value of α used is still within the range of permissible values or whether it has to be reduced, making this approach computationally very expensive. This problem does not arise if the matrix exponential is used instead of the resolvent.

5. **Other centrality measures**: So far we have only discussed centrality measures that can be expressed in terms of the adjacency matrix A. These centrality measures are all connected to the notion of walk in a graph, and they can often be understood in terms of spectral graph theory. Other popular centrality measures, such as betweenness centrality and closeness centrality (see, e.g., [34]), do not have a simple formulation in terms of matrix properties. They are based on the assumption that all communication in a graph tends to take place along shortest paths, which is not always the case (this was a major motivation for the introduction of walk-based measures, which postulate that communication between nodes can take place along walks of any length, with a preference toward shorter walks). A further disadvantage

is that they are quite expensive to compute, although randomized approximations can bring the cost down to acceptable levels [11]. For these reasons we do not consider them in this paper, where the focus is on linear algebraic techniques. It remains an open question whether heuristics for manipulating graph edges so as to tune some global average of these centrality measures can lead to networks with desirable connectivity and robustness properties.

Finally, in view of the bounds (2.3), the evolution of the total communicability under network modifications is closely tied to the evolution of the dominant eigenvalue λ_1. This quantity plays a crucial role in network analysis, for example, in the definition of the *epidemic threshold*; see, for instance, [34, p. 664] and [41]. In particular, a decrease in the total network communicability can be expected to lead to an increase in the epidemic threshold. Thus, edge modification techniques developed for tuning $TC(A)$ can potentially be used to alter epidemics dynamics.

3. Bounds via quadrature rules. In the previous section we saw the simple bounds (2.3) on the normalized total network communicability. More refined bounds for this index can be obtained by means of quadrature rules as described in [5, 6, 25, 23]. The following theorem contains our result on the bounds for the normalized total communicability.

THEOREM 3.1. *Let A be the adjacency matrix of an unweighted and undirected network. Then*

$$\Phi\left(\beta, \omega_1 + \frac{\gamma_1^2}{\omega_1 - \beta}\right) \le \frac{TC(A)}{n} \le \Phi\left(\alpha, \omega_1 + \frac{\gamma_1^2}{\omega_1 - \alpha}\right),$$

where $[\alpha, \beta]$ is an interval containing the spectrum of $-A$ (i.e., $\alpha \le -\lambda_1$ and $\beta \ge -\lambda_n$), $\omega_1 = -\mu = -\frac{1}{n}\sum_{i=1}^n d_i$ is the negative mean of the degrees, $\gamma_1 = \sigma = \sqrt{\frac{1}{n}\sum_{k=1}^n (d_k - \mu)^2}$ is the standard deviation, and

$$\Phi(x, y) = \frac{c\,(e^{-x} - e^{-y}) + xe^{-y} - ye^{-x}}{x - y}, \qquad c = \omega_1. \tag{3.1}$$

A proof of this result can be found in the supplementary material accompanying the paper.

Analogous bounds can be found for the adjacency matrix of the graph after performing a downdate or an update. These results are summarized in the following corollaries.

COROLLARY 3.2 (downdating). *Let $\widehat{A} = A - UW^T$, where $U = [\mathbf{e}_i, \mathbf{e}_j]$ and $W = [\mathbf{e}_j, \mathbf{e}_i]$, be the adjacency matrix of an unweighted and undirected network obtained after the downdate of the edge (i, j) from the matrix A. Let $\omega_1 = -\mu = -\frac{1}{n}\sum_{i=1}^n d_i$ and $\gamma_1 = \sigma = \sqrt{\frac{1}{n}\sum_{k=1}^n (d_k - \mu)^2}$, where d_i is the degree of node i in the original graph. Then*

$$\Phi\left(\beta_-, \omega_- + \frac{\gamma_-^2}{\omega_- - \beta_-}\right) \le \frac{TC(\widehat{A})}{n} \le \Phi\left(\alpha_-, \omega_- + \frac{\gamma_-^2}{\omega_- - \alpha_-}\right),$$

where

$$\begin{cases} \omega_- = \omega_1 + \frac{2}{n}, \\ \gamma_- = \sqrt{\gamma_1^2 - \frac{2}{n}\left(d_i + d_j - 1 + 2\omega_1 + \frac{2}{n}\right)} \end{cases},$$

α_- and β_- are approximations of the smallest and largest eigenvalues of $-\widehat{A}$, respectively, and Φ is defined as in (3.1) *with $c = \omega_-$.*

Note that if bounds α and β for the extremal eigenvalues of the original matrix are known, we can then use $\alpha_- = \alpha$ and $\beta_- = \beta + 1$. Indeed, if we order the eigenvalues of $\widehat{A}$ in nonincreasing order $\widehat{\lambda}_1 > \widehat{\lambda}_2 \geq \cdots \geq \widehat{\lambda}_n$ we obtain, as a consequence of Weyl's theorem (see [29, section 4.3]), that

$$\alpha - 1 \leq -\lambda_1 - 1 < -\widehat{\lambda}_1 < -\widehat{\lambda}_2 \leq \cdots \leq -\widehat{\lambda}_n < -\lambda_n + 1 \leq \beta + 1.$$

Furthermore, the Perron–Frobenius theorem ensures that, when performing a downdate, the largest eigenvalue of the adjacency matrix cannot increase; hence, we deduce the more stringent bounds $\alpha \leq -\widehat{\lambda}_1 \leq -\widehat{\lambda}_2 \leq \cdots \leq -\widehat{\lambda}_n \leq \beta + 1$.

Similarly, we can derive bounds for the normalized total communicability of the matrix $\tilde{A}$ obtained from the matrix A after performing the update of the virtual edge (i, j).

COROLLARY 3.3 (updating). *Let $\widehat{A} = A + UW^T$, where $U = [\mathbf{e}_i, \mathbf{e}_j]$ and $W = [\mathbf{e}_j, \mathbf{e}_i]$, be the adjacency matrix of an unweighted and undirected network obtained after the update of the virtual edge (i, j) in the matrix A. Let $\omega_1 = -\mu = -\frac{1}{n}\sum_{i=1}^{n} d_i$ and $\gamma_1 = \sigma = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(d_k - \mu)^2}$, where d_i is the degree of node i in the original graph. Then*

$$\Phi\left(\beta_+, \omega_+ + \frac{\gamma_+^2}{\omega_+ - \beta_+}\right) \leq \frac{TC(\tilde{A})}{n} \leq \Phi\left(\alpha_+, \omega_+ + \frac{\gamma_+^2}{\omega_+ - \alpha_+}\right),$$

where

$$\begin{cases} \omega_+ = \omega_1 - \frac{2}{n}, \\ \gamma_+ = \sqrt{\gamma_1^2 + \frac{2}{n}\left(d_i + d_j + 1 + 2\omega_1 - \frac{2}{n}\right)} \end{cases},$$

α_+ and β_+ are bounds for the smallest and largest eigenvalues of $-\tilde{A}$, respectively, and Φ is defined as in (3.1) *with $c = \omega_+$.*

Notice that again, if bounds α and β for the extremal eigenvalues of $-A$ are known, we can then take $\alpha_+ = \alpha - 1$ and $\beta_+ = \beta$. In fact, the spectrum of the rank-two symmetric perturbations UW^T and $-UW^T$ is $\{\pm 1, 0\}$, and hence we can use Weyl's theorem as before and then improve the upper bound using the Perron–Frobenius theorem.

In the next section we will see how the new bounds can be used to guide the updating and downdating process.

4. Modifications of the adjacency matrix. In this section we develop techniques that allow us to tackle the following problems.

(P1) Downdate: select K edges that can be downdated from the network without disconnecting it and that cause the smallest drop in the total communicability of the graph.

(P2) Update: select K edges to be added to the network (without creating self-loops or multiple edges) so as to increase as much as possible the total communicability of the graph.

(P3) Rewire: select K edges to be rewired in the network so as to increase as much as possible the value of $TC(A)$. The rewiring process must not disconnect the network or create self-loops or multiple edges in the graph.

As we will show below, (P3) can be solved using combinations of methods developed to solve (P1) and (P2). Hence, we first focus on the downdate and the update separately. Note that to decrease as little as possible the total communicability when removing an edge, it would suffice to select $(i^*, j^*) \in E$ so as to minimize the quantities

$$\mathbf{1}^T A^k \mathbf{1} - \mathbf{1}^T (A - UW^T)^k \mathbf{1} \quad \forall k = 1, 2, \ldots,$$

since $TC(A) = \sum_{k=0}^{\infty} \frac{\mathbf{1}^T A^k \mathbf{1}}{k!}$. Similarly, to increase as much as possible $TC(A)$ by addition of a virtual edge, it would suffice to select $(i^*, j^*) \in \overline{E}$ that maximizes the differences

$$\mathbf{1}^T (A + UW^T)^k \mathbf{1} - \mathbf{1}^T A^k \mathbf{1} \quad \forall k = 1, 2, \ldots.$$

However, it is easy to show that in general, one cannot find a choice for (i^*, j^*) that works for all such k. Indeed, numerical experiments on small synthetic graphs (not shown here) show that in general, the optimal edge selection for $k = 2$ is different from that for $k = 3$. Because of this, it is unlikely that one can find a simple "closed form solution" to the problem, and we need to develop approximation techniques.

The majority of the heuristics we will develop are based on new edge centrality measures. The idea underlying these is that it seems reasonable to assume that an edge is more likely used as a communication channel if its adjacent nodes are given a lot of information to spread. Thus, in the following definitions we introduce three new centrality measures for edges based on the principle that edges connecting important nodes are themselves important.

Definition 4.1. *For any $i, j \in V$ we define the* edge subgraph centrality *of an existing/virtual edge (i, j) as*

$$^{e}SC(i,j) = \left(e^A\right)_{ii} \left(e^A\right)_{jj}. \tag{4.1}$$

This definition, based on the subgraph centrality of nodes, exploits the fact that the matrix exponential is symmetric positive definite, and hence $(e^A)_{ii}(e^A)_{jj} > (e^A)_{ij}^2$. Therefore, the diagonal elements of e^A somehow control its off-diagonal entries, and hence they may contain enough information to infer the "payload" of the edges or of the virtual edges of interest.

Definition 4.2. *For any $i, j \in V$ we define the* edge total communicability centrality *of an existing/virtual edge (i, j) as*

$$^{e}TC(i,j) = [e^A \mathbf{1}]_i [e^A \mathbf{1}]_j. \tag{4.2}$$

It is important to observe that when the spectral gap $\lambda_1 - \lambda_2$ is "large enough," the subgraph centrality $\left(e^A\right)_{ii}$ and the total communicability centrality $[e^A \mathbf{1}]_i$ are essentially determined by $e^{\lambda_1} q_1(i)^2$ and $e^{\lambda_1} q_1(i) \|\mathbf{q}_1\|_1$, respectively (see, e.g., [7, 8, 17]); it follows that in this case the two centrality measures introduced and a centrality measure based on the eigenvector centrality for nodes can be expected to provide similar rankings. This is especially true when attention is restricted to the top edges (or nodes). This observation motivates the introduction of the following edge centrality measure.

Definition 4.3. *For any $i, j \in V$ we define the* edge eigenvector centrality *of an existing/virtual edge (i, j) as*

$$^{e}EC(i,j) = q_1(i) q_1(j). \tag{4.3}$$

As a further justification for this definition, note that

$$\lambda_1 - 2\left({}^e EC(i,j)\right) \leq \widehat{\lambda}_1 \leq \lambda_1, \qquad \tilde{\lambda}_1 \geq \lambda_1 + 2\left({}^e EC(i,j)\right),$$

where $\widehat{\lambda}_1$ is the leading eigenvalue of the matrix $\widehat{A}$ and $\tilde{\lambda}_1$ is the leading eigenvalue of the matrix $\tilde{A}$, as defined in section 2. These inequalities show that the edge eigenvector centrality of an existing/virtual edge (i, j) is strictly connected to the change in the value of the leading eigenvalue of the adjacency matrix, which influences the evolution of the total communicability when we modify A (see Remark 2).

Remark 3. The edge eigenvector centrality has been used in [40, 41] to devise edge removal techniques aimed at significantly reducing λ_1, so as to increase the *epidemic threshold* of networks.

Note that we defined these measures of centrality for both existing and virtual edges (as in [9]). The reason for this, as well as the justification for these definitions, will become clear in the next subsections.

We now discuss how to use these definitions to tackle the problems previously described. The computational aspects concerning the implementation of the heuristics we are about to introduce and the derivation of their computational costs are described in the supplementary material to this paper.

(P1) Downdate. The downdate of any edge in the network will result in a reduction of its total communicability. Note that since we are focusing on the case of connected networks, we will only perform downdates that keep the resulting graph connected. In practice, it is desirable to further restrict the choice of downdates to a subset of all existing edges, on the basis of criteria to be discussed shortly.

An "optimal" approach would select at each step of the downdating process a candidate edge corresponding to the minimum decrease of communicability.[2] Note that for large networks, this method is too costly to be practical. For this reason we aim to develop inexpensive techniques that will hopefully give close-to-optimal results. Nevertheless, for small networks we will use the "optimal" approach (where we systematically try all feasible edges and delete the one causing the least drop in total communicability) as a baseline method against which we compare the various algorithms discussed below. This method will be henceforth referred to as `optimal`.

The next methods we introduce perform the downdate of the lowest ranked existing edge according to the edge centrality measures previously introduced, whose removal does not disconnect the network. We will refer to these methods as `subgraph`, `nodeTC`, and `eigenvector`, which are based on Definitions 4.1, 4.2, and 4.3, respectively. From the point of view of the communicability, these methods downdate an edge connecting two nodes which are peripheral (i.e., have low centrality) and therefore are not expected to give a large contribution to the spread of information along the network. Hence, the selected edge is connecting two nodes whose ability to exchange information is already very low, and we do not expect the total communicability to suffer too much from this edge removal. This observation also suggests that such downdates can be repeatedly applied without the need to recompute the ranking of the edges after each downdate. As long as the number of downdates performed remains small compared to the total number of edges, we expect good results at a greatly reduced total cost. Note also that such downdates can be performed simultaneously

[2]Strictly speaking, this would correspond to a greedy algorithm, which is only locally optimal. In general, this is unlikely to result in "globally optimal" network communicability. In this paper, the term "optimal" will be understood in this limited sense only.

rather than sequentially. We will refer to these variants as `subgraph.no`, `nodeTC.no`, and `eigenvector.no`.

Finally, we consider a technique motivated by the bounds obtained via quadrature rules derived in section 3. From the expression for the function Φ in the special case of the downdate (cf. Corollary 3.2), we infer that a potentially good choice may be to remove the edge having incident nodes i, j, for which the sum $d_i + d_j$ is minimal, if its removal does not disconnect the network. Indeed, this choice reduces the upper bound only slightly, and the total communicability may mirror this behavior. Another way to justify this strategy is to observe that it is indeed the optimal strategy if we approximate e^A with its second-order approximation $I + A + \frac{1}{2}A^2$ in the definition of total communicability. This technique will be henceforth referred to as `degree`. We note that a related measure, namely, the average of the out-degrees $\frac{d_i+d_j}{2}$, was proposed in [9] as a measure for the centrality of an edge (i, j) in directed graphs.

(P2) Update. Most real-world networks are characterized by low average degree. As a consequence, the adjacency matrices of such networks are sparse ($m = O(n)$). For the purpose of selecting a virtual edge to be updated, this implies that we have approximately $\frac{1}{2}(n^2 - cn)$ possible choices if we want to avoid the formation of multiple edges or self-loops (here c is a moderate constant). Each one of these possible updates will result in an increase of the total communicability of the network, but not every one of these will result in a significant increment.

One natural updating technique is to connect two nodes having high centralities, i.e., add the virtual edge having the highest ranking according to the corresponding edge centrality. Its incident nodes, being quite central, can be expected to have an important role in the spreading of information along the network; on the other hand, the communication between them may be relatively poor (for example, think of the case where the two nodes sit in two distinct communities). Hence, giving them a preferential communication channel, such as an edge between them, should result in a better spread of information along the whole network. Again, we will use the labels `subgraph`, `nodeTC`, and `eigenvector` to describe these updating strategies. As before, in order to reduce the computational cost, we also test the effectiveness of these techniques without the recomputation of the ranking of the virtual edges after each update. These variants (referred to as `subgraph.no`, `nodeTC.no`, and `eigenvector.no`) are expected to return good results as well, since the selected update should not radically change the ranking of the edges. Indeed, they make central nodes even more central, and consequently the ranking of the edges remains almost unchanged. Note again that these updates can be performed simultaneously rather than sequentially.

As for the case of downdating, the bounds via quadrature rules derived in section 3 suggest an updating technique, i.e., adding the virtual edge (i, j) for which $d_i + d_j$ is maximal. Indeed, such a choice would maximize the lower bound on the total communicability; see Corollary 3.3. Again, this choice can also be justified by noting that it is optimal if e^A is replaced by its quadratic Maclaurin approximant. We will again use the label `degree` to refer to this updating strategy.

All these techniques will be compared with the `optimal` one, based on systematically trying all feasible virtual edges and selecting at each step the one resulting in the largest increase of the total communicability. Due to the very high cost of this brute force approach, we will use it only on small networks.

The heuristics introduced to tackle (P1) and (P2) are summarized in Table 1.

TABLE 1
Brief description of the techniques introduced in the paper.

Method	Downdate: $(i,j) \in E$	Update: $(i,j) \notin E$
`optimal`	$\arg\min\{TC(A) - TC(\widehat{A})\}$	$\arg\max\{TC(\tilde{A}) - TC(A)\}$
`subgraph(.no)`	$\arg\min\{^eSC(i,j)\}$	$\arg\max\{^eSC(i,j)\}$
`eigenvector(.no)`	$\arg\min\{^eEC(i,j)\}$	$\arg\max\{^eEC(i,j)\}$
`nodeTC(.no)`	$\arg\min\{^eTC(i,j)\}$	$\arg\max\{^eTC(i,j)\}$
`degree`	$\arg\min\{d_i + d_j\}$	$\arg\max\{d_i + d_j\}$

(P3) Rewire. As we have already noted, there are situations in which the rewire of an edge may be preferable to the addition of a new one. There are various possible choices for the rewiring strategy to follow. The greatest part of those found in the literature are variants of random rewiring (see, for example, [10, 31]). In this paper, on the other hand, we are interested in devising mathematically informed rewiring strategies. For comparison purposes, however, we will compare our rewiring methods to the random rewire method, `random`, which downdates an edge (chosen uniformly at random from among all edges whose removal does not disconnect the network) and then updates a virtual edge, also chosen uniformly at random.

Combining the various downdating and updating methods previously introduced, we obtain different rewiring strategies based on the centralities of edges and on the bounds for the total communicability. Concerning the methods based on the edge subgraph, eigenvector, and total communicability centralities, we note that since a single downdate does not dramatically change the communication capability of the network, we do not need to recompute the centralities and the ranking of the edges after each downdating step, at least as long as the number of rewired edges remains relatively small (numerical experiments not shown here support this claim). On the other hand, after each update we may or may not recalculate the edge centralities. As before, we use `subgraph/subgraph.no`, `eigenvector/eigenvector.no`, and `nodeTC/nodeTC.no` to refer to these three variants of rewiring. Additionally, we introduce another rewiring strategy, henceforth referred to as `node`, based on the subgraph centrality of the nodes. In this method we disconnect the most central node from the least central node among its immediate neighbors; then we connect it to the most central node among those it is not linked to. It is worth emphasizing that this strategy is philosophically different from the previous ones based on the edge subgraph centrality in the downdating phase (the updating step is the same). In fact, in those methods we use information on the nodes in order to deduce some information on the edges connecting them; on the other hand, the `node` algorithm does not take into account the potentially high "payload" of the edges involved, whose removal may result in a dramatic drop in the total communicability.

5. Numerical studies.

In this section we discuss the results of numerical studies performed in order to assess the effectiveness and efficiency of the proposed techniques. The tests have been performed on both synthetic and real-world networks, as described below. We refer the reader to the supplementary material for the results of computations performed on four small social networks, aimed at comparing our heuristics with `optimal`. These results show that for these small networks, the resulting total communicabilities are essentially identical to those obtained with the `optimal` strategy.

Table 2
Description of the data set.

NAME	n	m	λ_1	λ_2	$\lambda_1 - \lambda_2$
Minnesota	2640	3302	3.2324	3.2319	0.0005
USAir97	332	2126	41.233	17.308	23.925
as-735	6474	12572	46.893	27.823	19.070
Erdös02	5534	8472	25.842	12.330	13.512
ca-HepTh	8638	24806	31.034	23.004	8.031
as-22july06	22963	48436	71.613	53.166	18.447
usroad-48	126146	161950	3.911	3.840	0.071

5.1. Real-world networks. All the networks used in the tests can be found in the University of Florida Sparse Matrix Collection [15] under different "groups." The USAir97 and Erdös02 networks are from the Pajek group. The USAir97 network describes the US Air flight routes in 1997, while the Erdös02 network represents the Erdös collaboration network, with Erdös included. The network as-735, from the SNAP group, is the communication network of a group of autonomous systems (ASs) measured over 735 days between November 8, 1997 and January 2, 2000. Communication occurs when routers from two ASs exchange information. The Minnesota network from the Gleich group represents the Minnesota road network. These latter three networks are not connected, and therefore the tests were performed on their largest connected component. We point out that the original largest connected component of the network as-735 has 1323 ones on the main diagonal, which were retained in our tests. The network ca-HepTh is from the SNAP group and represents the collaboration network of arXiv High Energy Physics Theory; the network as-22july06 is from the Newman group and represents the (symmetrized) structure of Internet routers as of July 22, 2006. Finally, the network usroad-48, which is from the Gleich group, represents the continental US road network. For each network, Table 2 reports the number of nodes (n), the number of edges (m), the two largest eigenvalues, and the spectral gap. We use the first four networks to test all methods described in the previous section (except for `optimal`, which is only applied to the four smallest networks; see the supplementary material) and use the last three to illustrate the performance of the most efficient among the methods tested.

We first consider the networks Minnesota, as-735, USAir97, and Erdös02, for which we perform $K = 50$ modifications. For these networks the set $\overline{E}$ (the complement of the set E of edges) is large enough that performing an extensive search for the edge to be updated is expensive. Hence, we form the set S containing the top 10% of the nodes ordered according to the eigenvector centrality, and we restrict our search to virtual edges incident to these nodes only. An exception is the network USAir97, where we have used the set S corresponding to the top 20% of the nodes, since in the case of 10% this set contained only 52 virtual edges. In Figures 1 and 2 we show results for the methods `eigenvector`, `eigenvector.no`, `subgraph`, `subgraph.no,` and `degree`. Before commenting on these, we want to stress the poor performance of `node` when tackling (P3); this shows that the use of edge centrality measures (as opposed to node centralities alone) is indispensable in this framework. The results for these networks clearly show the effectiveness of the `eigenvector` and `subgraph` algorithms and of their less expensive variants `eigenvector.no` and `subgraph.no` in nearly all cases; similar results were obtained with `nodeTC` and `nodeTC.no` (not shown). The only exception is in the downdating of the Minnesota network, where the eigenvector-based techniques give slightly worse results. This fact is easily explained in view of

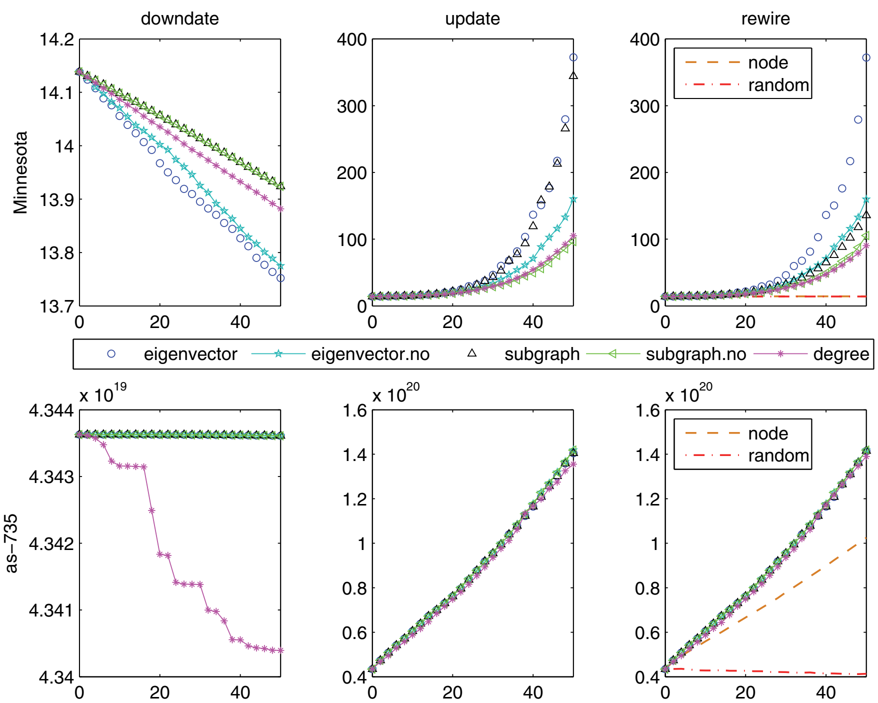

FIG. 1. *Evolution of the normalized total communicability versus number of downdates, updates, and rewires for networks Minnesota and* as-735*.*

the tiny spectral gap characterizing this and similar networks[3] (see Table 2). Because of this property, eigenvector centrality is a poor approximation of subgraph centrality and cannot be expected to give results similar to those obtained with `subgraph` and `subgraph.no`.

The results for the downdate show that the inexpensive `degree` method does not perform as well on these networks, except perhaps on Minnesota. The relatively poor performance of this method is due to the fact that the information used by this method to select an edge for downdating is too local.

Note, however, the scale on the vertical axis in Figures 1–2, suggesting that for these networks (excluding perhaps Minnesota) all the edge centrality-based methods perform well with only very small relative differences between the resulting total communicabilities.

Overall, these results indicate that the edge centrality-based methods, especially the inexpensive `eigenvector.no` and `nodeTC.no` variants, are an excellent choice in almost all cases and for tackling all the problems. In the case of downdating networks with small spectral gaps, `subgraph.no` may be preferable but at a higher cost.

The behavior of the `degree` method depends strongly on the network on which it is used. Our tests indicate that it behaves well in some cases (for example, P2 for Erdös02) but poorly in others (P2 for Minnesota). We speculate that this method may perform adequately when tackling (P2) on scale-free networks (such as Erdös02), where a high degree is an indication of centrality in spreading information across the network.

[3]Small spectral gaps are typical of large, grid-like networks such as the road networks or the graphs corresponding to uniform triangulations or discretizations of physical domains.

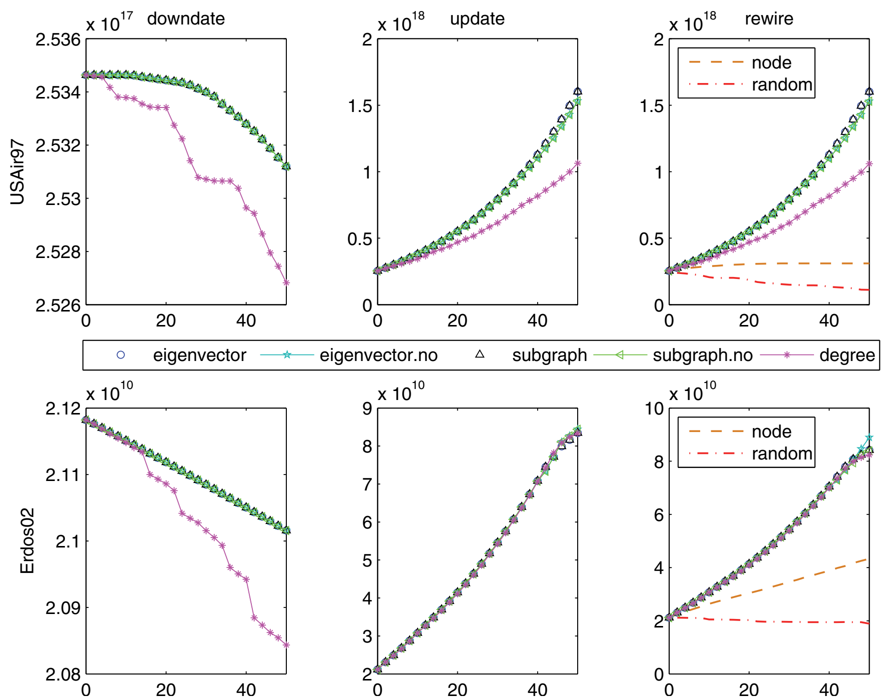

FIG. 2. *Evolution of the normalized total communicability versus number of downdates, updates, and rewires for networks* USAir97 *and* Erdös02.

Some comments on the difference in the results for updating compared to those for rewiring (downdating followed by updating) are in order. Recall that our downdating strategies aim to reduce as little as possible the decrease in the value of the total communicability, whereas the updating techniques aim to increase this index as much as possible. With this in mind, it is not surprising to see that the trends of the evolution of the total communicability after rewiring reflect those obtained with the updating strategies. The values obtained using the updates are, in general, higher than those obtained using the rewiring strategies, since updating implies the addition of edges, whereas in rewiring the number of edges remains the same. Experiments not reported here indicate that the methods based on the edge eigenvector and total communicability centrality are more stable than the others under rewiring and for dampening the effect of the downdates.

In Figures 3–4 we show results for the three largest networks in our data set (ca-HepTh, as-22july06 and usroad-48). In the case of the updating, we have selected the virtual edges from among those in the subgraph containing the top 1% of nodes ranked according to the eigenvector centrality. We compare the methods `eigenvector`, `eigenvector.no`, `nodeTC`, `nodeTC.no`, `subgraph.no`, and `degree`; random downdating was also tested and found to give poor results. Note that network usroad-48 behaves similarly to Minnesota; this is not surprising in view of the fact that these are both road networks with a tiny spectral gap. Looking at the scale on the vertical axis, however, it is clear that the decrease in total communicability is negligible with all the methods tested here. The results on these networks confirm the general trend observed so far; in particular, we note the excellent behavior of `nodeTC` and `nodeTC.no`.

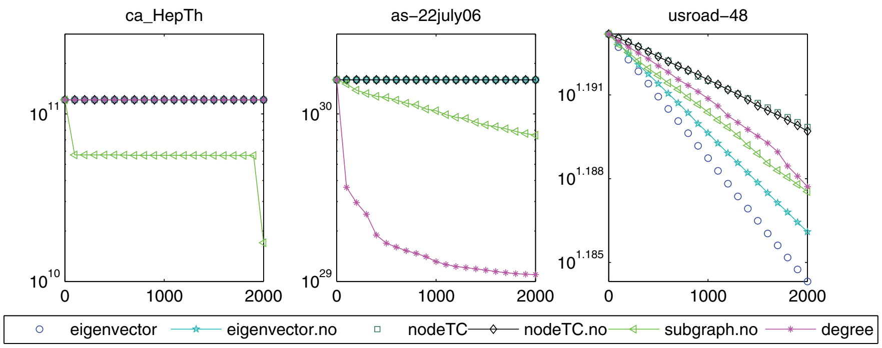

FIG. 3. *Downdates for large networks: normalized total communicability versus number of modifications.*

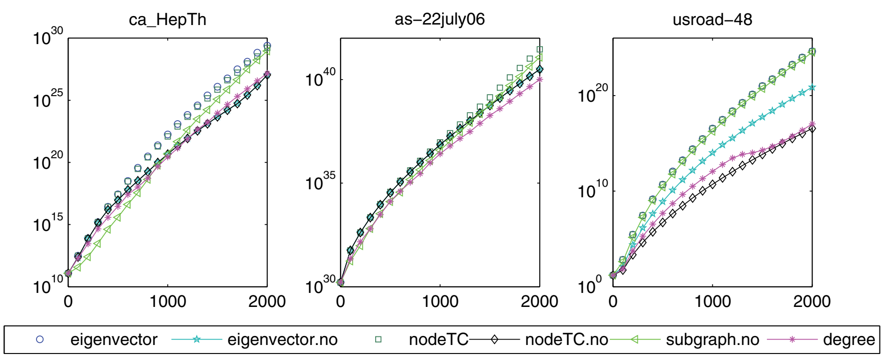

FIG. 4. *Updates for large networks: normalized total communicability versus number of modifications.*

5.2. Synthetic networks. The synthetic examples used in the tests were produced using the CONTEST toolbox for MATLAB (see [38, 39]). We tested two types of graphs: the preferential attachment (Barabási–Albert) model and the small world (Watts–Strogatz) model.

The preferential attachment model [4] was designed to produce networks with scale-free degree distributions as well as the small world property, characterized by short average path length and relatively high clustering coefficient. In CONTEST, preferential attachment networks are constructed using the command `pref(n,d)`, where n is the number of nodes and $d \geq 1$ is the number of edges each new node is given when it is first introduced to the network. The network is created by adding nodes one by one (each new node with d edges). The edges of the new node connect to nodes already in the network with a probability proportional to the degree of the already existing nodes. This results in a scale-free degree distribution.

The second class of synthetic test matrices used in our experiments corresponds to Watts–Strogatz small world networks. The small world model was developed as a way to impose a high clustering coefficient onto classical random graphs [42]. The function used to build these matrices takes the form `smallw(n,k,p)`. Here n is the number of nodes in the network, originally arranged in a ring and connected to their k nearest neighbors. Then each node is considered independently and, with probability p, an edge is added between the node and one of the other nodes in the graph, chosen

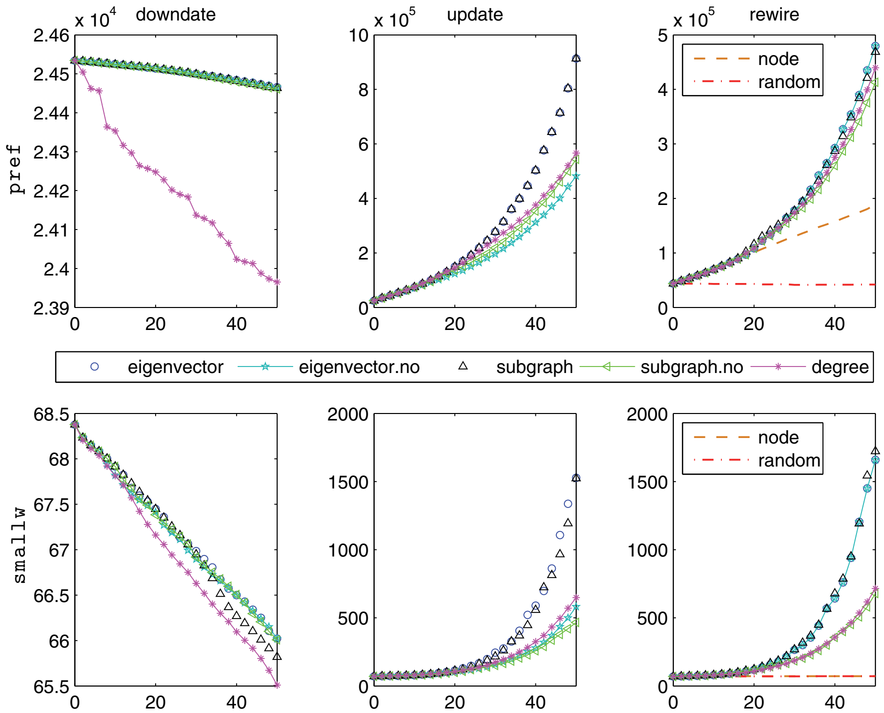

FIG. 5. *Evolution of the total communicability when* 50 *downdates, updates, or rewires are performed on two synthetic networks with* $n = 1000$ *nodes.*

uniformly at random (self-loops and multiple edges are not allowed). In our tests, we have used matrices with $n = 1000$ nodes, which were built using the default values for the functions previously described. We used $d = 2$ in the Barabási–Albert model and $k = 2$, $p = 0.1$ in the Watts–Strogatz model.

The results for our tests are presented in Figure 5. These results agree with what we have seen previously on real-world networks. Interestingly, `degree` does not perform well for the downdate when working on the preferential attachment model; this behavior reflects what we have seen for the networks USAir97, as-735, and Erdös02, which are indeed scale-free networks.

5.3. Timings for synthetic networks. We have performed some experiments with synthetic networks of increasing size in order to assess the scalability of the various methods introduced in this paper. A sequence of seven adjacency matrices corresponding to Barabási–Albert scale-free graphs was generated using the CONTEST toolbox. The order of the matrices ranges from 1000 to 7000; the average degree is kept constant at 5. A fixed number of modifications ($K = 500$) was carried out on each network. All experiments were performed using MATLAB Version 7.12.0.635 (R2011a) on an IBM ThinkPad running Ubuntu 12.04.5 LTS, a 2.5 GHZ Intel Core i5 processor, and 3.7 GiB of RAM. We used the built-in MATLAB function `eigs` (with the default settings) to approximate the dominant eigenvector of the adjacency matrix A, the MATLAB toolbox `mmq` [32] to estimate the diagonal entries of e^A (with a fixed number of five nodes in the Gauss–Radau quadrature rule, hence five Lanczos steps per estimate), and the toolbox `funm_kryl` to compute the vector $e^A\mathbf{1}$ of total communicabilities, also with the default parameter settings.

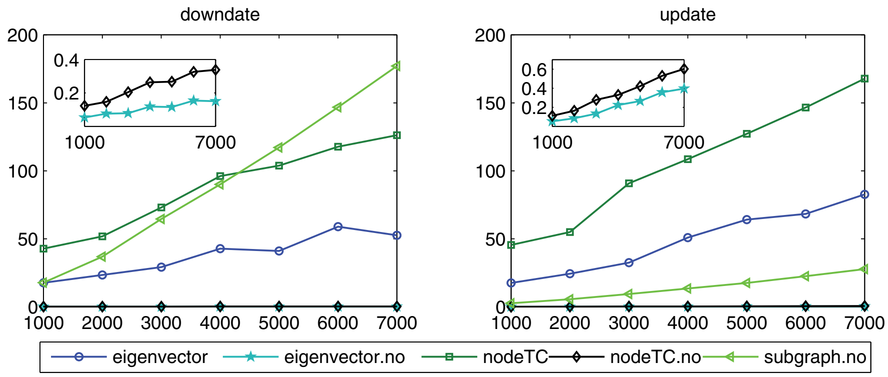

Fig. 6. *Timings in seconds for scale-free graphs of increasing size (*500 *modifications).*

The results are shown in Figure 6. The approximate (asymptotic) linear scaling behavior of the various methods (in particular of `nodeTC.no` and `eigenvector.no`, which are by far the fastest; see the insets) is clearly displayed in these plots.

5.4. Timings for larger networks. In Tables 3–4 we report the timings for various methods when $K = 2000$ downdates and updates are selected for the three largest networks listed in Table 2.

The timings presented refer to the selection of the edges to be downdated or updated, which dominates the computational effort. For the method `subgraph.no` in the case of downdates, we restricted the search of candidate edges to a subset of E in order to reduce costs. For the three test networks we used 40%, 45%, and 15% of the nodes, respectively, chosen by taking those with lowest eigenvector centrality, and the corresponding edges. We found the results to be very close to those obtained working with the complete set E, but at a significantly lower cost (especially for the largest network).

These results clearly show that algorithms `nodeTC.no` and `eigenvector.no` are orders of magnitude faster than the other methods; method `subgraph.no`, while significantly more expensive, is still reasonably efficient[4] and can be expected to give better results in some cases (e.g., on networks with a very small spectral gap). The `degree` algorithm, on the other hand, cannot be recommended in general since it gives somewhat inferior results. The remaining methods `eigenvector`, `nodeTC`, and `subgraph` (not shown here) are prohibitively expensive for large networks, at least when the number K of modifications is high (as it is here).

We also observe that downdating is generally a more expensive process than updating, since in the latter case the edges are to be chosen from among a fairly small subset of all virtual edges, whereas in the downdating process we work on the whole set E of existing edges (or on a large subset of E). For some methods the difference in cost becomes significant when the networks are sufficiently large and the number of modifications to be performed is high.

Summarizing, the method labeled `nodeTC.no` is the fastest and gives excellent results, quite close to those of the more expensive methods, and therefore we can recommend its use for the type of problems considered here. The methods labeled

[4]It is worth mentioning that in principle it is possible to greatly reduce the cost of this method using parallel processing, since each subgraph centrality can be computed independently of the others.

TABLE 3
Timings in seconds for $K = 2000$ downdates performed on the three largest networks in Table 2.

	ca-HepTh	as-22july06	usroad-48
`eigenvector`	278.13	599.83	11207.39
`eigenvector.no`	0.07	1.79	4.08
`nodeTC`	553.04	1234.49	2634.27
`nodeTC.no`	0.34	0.83	1.34
`subgraph.no`	107.36	383.34	1774.07
`degree`	29.67	53.42	153.52

TABLE 4
Timing in seconds for $K = 2000$ updates performed on the three largest networks in Table 2.

	ca-HepTh	as-22july06	usroad-48
`eigenvector`	192.8	436.9	1599.5
`eigenvector.no`	0.19	0.33	5.85
`nodeTC`	561.9	1218.8	2932.
`nodeTC.no`	0.30	0.55	1.59
`subgraph.no`	3.13	7.20	121.4
`degree`	11.1	12.4	175.8

`eigenvector.no` and `subgraph.no` are also effective and may prove useful in some settings, especially for updating.

6. Evolution of other connectivity measures. In this section we want to highlight another facet of the methods we have introduced for (approximately) optimizing the total communicability. In particular, we look at the evolution of other network properties under our updating strategies. When building or modifying a network, there are various features that one may want to achieve. Typically, there are two main desirable properties: first, the network should do a good job at spreading information, i.e., have a high total communicability; second, the network should be robust under targeted attacks or random failure, which is equivalent to the requirement that it should be difficult to "isolate" parts of the network, i.e., the network should be "well-connected." This latter property can be measured by means of various indices. One such measure is the spectral gap $\lambda_1 - \lambda_2$. As a consequence of the Perron–Frobenius theorem, adding an edge to a connected network causes the dominant eigenvalue λ_1 of A to increase. Test results (not shown here) show that when a network is updated using one of our techniques, the first eigenvalue increases rapidly with the number of updates. On the other hand, the second eigenvalue λ_2 tends to change little with each update, and it may even decrease (recall that the matrix $UW^T = \mathbf{e}_i\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_i^T$ being added to A in an update is indefinite). Therefore, the spectral gap $\lambda_1 - \lambda_2$ widens rapidly with the number of updates.[5] It has been pointed out by various authors (see, e.g., [17, 36]) that a large spectral gap is typical of complex networks with good expansion properties.

Here we focus on a related measure, the so-called *free energy* (also known in the literature as *natural connectivity*) of the network. In particular, we investigate the effect of our proposed methods of network updating on the evolution of this index.

6.1. Tracking the free energy (or natural connectivity). The authors of [43] discuss a measure of network connectivity which is based on an intuitive notion

[5] This fact, incidentally, may serve as further justification for the effectiveness of algorithms such as `nodeTC.no` and `eigenvector.no`.

of robustness and whose analytical expression has a clear meaning and can be derived from the eigenvalues of A; they refer to it as the *natural connectivity* of the network (see also [44]). The idea underlying this index is that a network is more robust if there exists more than one route to get from one node to another; this property ensures that if a route becomes unusable, there is an alternative way to get from the source of information to the target. Therefore, intuitively a network is more robust if it has a lot of (apparently) redundant routes connecting its vertices or, equivalently, if each of its nodes is involved in a lot of closed walks. The natural connectivity aims at quantifying this property by using an existing measure for the total number of closed walks in a graph, namely, the *Estrada index* [16]. This index, denoted by $EE(G)$, is defined as the trace of the matrix exponential. Normalizing this value and taking the natural logarithm, one obtains the *natural connectivity* of the graph:

$$\overline{\lambda}(A) = \ln\left(\frac{1}{n}\sum_{i=1}^{n} e^{\lambda_i}\right) = \ln(EE(G)) - \ln(n).$$

It turns out, however, that essentially the same index was already present in the literature. Indeed, the natural connectivity is only one of the possible interpretations that can be given to the logarithm of the (normalized) Estrada index. Another, earlier interpretation was given in [19], where the authors related this quantity to the Helmholtz free energy of the network $F = -\ln(EE(G))$. Therefore, since $\overline{\lambda} = -F - \ln(n)$, the behavior of F completely describes that of $\overline{\lambda}$ (and conversely) as the graph is modified by adding or removing links.

The natural connectivity has been recently used (see [12]) to derive manipulation algorithms that directly optimize this robustness measure. In particular, the updating algorithm introduced in [12] appears to be superior to existing heuristics, such as those proposed in [10, 24, 37]. This algorithm, which costs $O(mt + Kd_{max}^2 t + Knt^2)$, where $d_{max} = \max_{i\in V} d_i$ and t is the (user-defined) number of leading eigenpairs, selects K edges to be added to the network by maximizing a quantity that involves the elements of the leading t eigenpairs of A.[6]

We have compared our updating techniques with that described in [12]. Results for four representative networks are shown in Figure 7. In our tests, we use the value $t = 50$ (as in [12]), and we select $K = 500$ edges. Note that, when K is large, the authors recommend recomputing the set of t leading eigenpairs every l iterations. This operation requires an additional effort that our faster methods do not need. Since the authors of [12] show numerical experiments in which the methods with and without the recomputation return almost exactly the same results, we did not recompute the eigenpairs after any of the updates.

Figure 7 displays the results for the evolution of both the natural connectivity and the normalized total communicability, where the latter is plotted in a semilogarithmic scale. A total of 500 updates have been performed. The method labeled `Chan` selects the edges according to the algorithm described in [12] choosing from all the virtual edges of the graph. For our methods we used, as before, the virtual edges in the subgraph obtained selecting the top 10% or 20% of nodes ranked according to the eigenvector centrality. As one can easily see, our methods generally outperform the algorithm proposed in [12]. In particular, `nodeTC.no` and `eigenvector.no` give generally better results than `Chan` and are much faster in practice. For instance, the execution time with `Chan` on the network ca-HepTh was over 531 seconds and much

[6]A description of the algorithm can be found in the supplementary material.

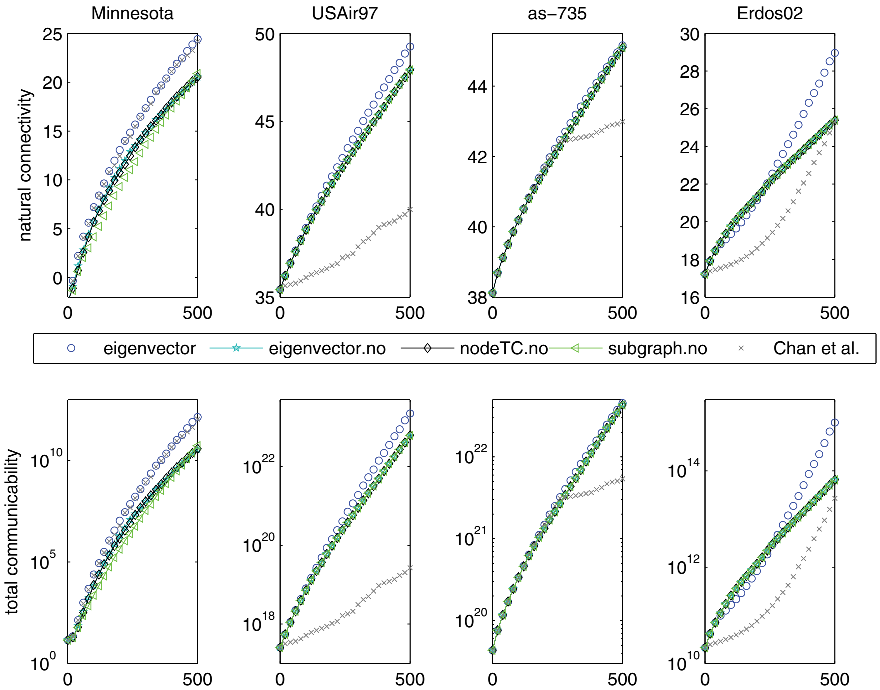

Fig. 7. *Evolution of the natural connectivity and of the normalized total communicability (in a semilogarithmic scale plot) when up to* 500 *updates are performed on four real-world networks.*

higher for the two larger networks. We recall (see Table 4) that the execution times for `nodeTC.no` and `eigenvector.no` are about three orders of magnitude smaller.

It is striking to see how closely the evolution of the natural connectivity mirrors the behavior of the normalized total communicability. This is likely due to the fact that both indices depend on the eigenvalues of A (with a large contribution coming from the terms containing λ_1; see (2.2) and Remark 2), and all the updating strategies used here tend to make λ_1 appreciably larger.

Returning to the interpretation in terms of statistical physics, from Figure 7 we deduce that the free energy of the graph decreases as we add edges to the network. In particular, this means that the network is evolving toward a more stable configuration and, in the limit, toward equilibrium, which is the configuration with maximum entropy.[7]

These findings indicate that the normalized total communicability is equally effective an index as the natural connectivity (equivalently, the free energy) for the purpose of characterizing network connectivity. Since the network total communicability can be computed very fast (in $O(n)$ time), we believe that the normalized total communicability should be used instead of the natural connectivity, especially for large networks. Indeed, computing the natural connectivity requires evaluating the trace of e^A; even when stochastic trace estimation is used [2], this is several times more expensive for large networks than the total communicability.

[7]The relation between the free energy and the Gibbs entropy is described in more detail in the supplementary material.

7. Conclusions and future work. In this paper we have introduced several techniques that can be used to modify an existing network so as to obtain networks that are highly sparse and yet have a large total communicability.

These heuristics make use of various measures of edge centrality, a few of which have been introduced in this work. Far from being ad hoc, these heuristics are widely applicable and mathematically justified. All our techniques can be implemented using well-established tools from numerical linear algebra: algorithms for eigenvector computation, Gauss-based quadrature rules for estimating quadratic forms, and Krylov subspace methods for computing the action of a matrix function on a vector. Ultimately, the Lanczos algorithm is the main player. High quality, public domain software exists to perform these modifications efficiently.

Among all the methods introduced here, the best results are obtained by the `nodeTC.no` and `eigenvector.no` algorithms, which are based on the edge total communicability and eigenvector centrality, respectively. These methods are extremely fast and returned excellent results in virtually all the tests we performed. For updating networks characterized by a small spectral gap, a viable alternative is the algorithm `subgraph.no`. While more expensive than `nodeTC.no` and `eigenvector.no`, this method scales linearly with the number of nodes and yields consistently good results.

Finally, we have shown that the total communicability can be effectively used as a measure of network connectivity, which plays an important role in designing robust networks. Indeed, the total communicability does a very good job at quantifying two related properties of networks: the ease of spreading information, and the extent to which the network is "well-connected." Our results show that the total communicability behaves in a manner very similar to the natural connectivity (or free energy) under network modifications, while it can be computed much more quickly.

Future work should include the extension of these techniques to other types of networks, including directed and weighted ones.

Acknowledgments. We are grateful to Ernesto Estrada for providing some of the networks used in the numerical experiments and for pointing out some useful references. The first author would like to thank Emory University for the hospitality offered in 2014, when this work was completed. We also thank two anonymous referees for helpful suggestions.

REFERENCES

[1] M. Afanasjew, M. Eiermann, O. G. Ernst, and S. Güttel, *Implementation of a restarted Krylov subspace method for the evaluation of matrix functions*, Linear Algebra Appl., 429 (2008), pp. 2293–2314.

[2] H. Avron and S. Toledo, *Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix*, J. ACM, 58 (2011), Art. 8.

[3] A.-L. Barabási, *Linked: The New Science of Networks*, Perseus, Cambridge, MA, 2002.

[4] A. L. Barabási and R. Albert, *Emergence of scaling in random networks*, Science, 286 (1999), pp. 509–512.

[5] M. Benzi and P. Boito, *Quadrature rule-based bounds for functions of adjacency matrices*, Linear Algebra Appl., 433 (2010), pp. 637–652.

[6] M. Benzi and G. H. Golub, *Bounds for the entries of matrix functions with application to preconditioning*, BIT, 39 (1999), pp. 417–438.

[7] M. Benzi and C. Klymko, *Total communicability as a centrality measure*, J. Complex Networks, 1 (2013), pp. 124–149.

[8] M. Benzi and C. Klymko, *On the limiting behavior of parameter-dependent network centrality measures*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 686–706.

[9] M. W. Berry, T. P. Chartier, K. R. Hutson, and A. N. Langville, *Identifying influential edges in a directed network: Big events, upsets and non-transitivity*, J. Complex Networks, 2 (2013), pp. 87–109.

[10] A. Beygelzimer, G. Grinstein, R. Linsker, and I. Rish, *Improving network robustness by edge modification*, Phys. A, 357 (2005), pp. 593–612.

[11] U. Brandes and T. Erlebach, eds., *Network Analysis: Methodological Foundations*, Lecture Notes in Comput. Sci. 3418, Springer, New York, 2005.

[12] H. Chan, L. Akoglu, and H. Tong, *Make it or break it: Manipulating robustness in large networks*, in Proceedings of the 2014 SIAM International Conference on Data Mining, SIAM, Philadelphia, pp. 325–333.

[13] R. Cohen and S. Havlin, *Complex Networks: Structure, Robustness and Function*, Cambridge University Press, Cambridge, UK, 2010.

[14] J. J. Crofts and D. J. Higham, *A weighted communicability measure applied to complex brain networks*, J. R. Soc. Interface, 6 (2009), pp. 411–414.

[15] T. Davis and Y. Hu, *The University of Florida Sparse Matrix Collection*, http://www.cise.ufl.edu/research/sparse/matrices/.

[16] E. Estrada, *Characterization of 3D molecular structure*, Chem. Phys. Lett., 319 (2000), 713–718.

[17] E. Estrada, *Spectral scaling and good expansion properties in complex networks*, Europhys. Lett., 73 (2006), pp. 649–655.

[18] E. Estrada, *The Structure of Complex Networks. Theory and Applications*, Oxford University Press, Oxford, UK, 2012.

[19] E. Estrada and N. Hatano, *Statistical–mechanical approach to subgraph centrality in complex networks*, Chem. Phys. Lett., 439 (2007), pp. 247–251.

[20] E. Estrada and N. Hatano, *Communicability in complex networks*, Phys. Rev. E, 77 (2008), 036111.

[21] E. Estrada, N. Hatano, and M. Benzi, *The physics of communicability in complex networks*, Phys. Rep., 514 (2012), pp. 89–119.

[22] E. Estrada and J. A. Rodríguez-Velázquez, *Subgraph centrality in complex networks*, Phys. Rev. E, 71 (2005), 056103.

[23] C. Fenu, D. Martin, L. Reichel, and G. Rodriguez, *Network analysis via partial spectral factorization and Gauss quadrature*, SIAM J. Sci. Comput., 35 (2013), pp. A2046–A2068.

[24] H. Frank and I. Frisch, *Analysis and design of survivable networks*, IEEE Trans. Comm. Tech., 18 (1970), pp. 501–519.

[25] G. H. Golub and G. Meurant, *Matrices, Moments and Quadrature with Applications*, Princeton University Press, Princeton, NJ, 2010.

[26] S. Güttel, `funm_kryl`: *A Restart Code for the Evaluation of Matrix Functions*, http://guettel.com/funm_kryl/.

[27] N. J. Higham, *Function of Matrices. Theory and Computation*, SIAM, Philadelphia, 2008.

[28] S. Hoory, N. Linial, and A. Wigderson, *Expander graphs and their applications*, Bull. Amer. Math. Soc., 43 (2006), pp. 439–561.

[29] R. A. Horn and C. R. Johnson, *Matrix Analysis,* 2nd ed., Cambridge University Press, Cambridge, UK, 2013.

[30] L. Katz, *A new status index derived from sociometric data analysis*, Psychometrika, 18 (1953), pp. 39–43.

[31] V. H. P. Louzada, F. Daolio, H. J. Herrmann, and M. Tomassini, *Smart rewiring for network robustness*, J. Complex Networks, 1 (2013), pp. 150–159.

[32] G. Meurant, *MMQ Toolbox for MATLAB*, http://gerard.meurant.pagesperso-orange.fr/.

[33] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2000.

[34] M. E. J. Newman, *Networks. An Introduction*, Oxford University Press, Oxford, UK, 2010.

[35] A. Pinar, J. Meza, V. Donde, and B. Lesieutre, *Optimization strategies for the vulnerability analysis of the electric power grid*, SIAM J. Optim., 20 (2010), pp. 1786–1810.

[36] D. Puder, *Expansion of Random Graphs: New Proofs, New Results*, preprint, arXiv:1212.5216v2, 2013.

[37] B. Shargel, H. Sayama, I. R. Epstein, and Y. Bar-Yam, *Optimization of robustness and connectivity in complex networks*, Phys. Rev. Lett., 90 (2003), 068701.

[38] A. Taylor and D. J. Higham, *CONTEST: Toolbox Files and Documentation*, http://www.mathstat.strath.ac.uk/research/groups/numerical_analysis/contest/toolbox.

[39] A. Taylor and D. J. Higham, *CONTEST: A controllable test matrix toolbox for MATLAB*, ACM Trans. Math. Softw., 35 (2009), Art. 26.

[40] H. Tong, B. A. Prakash, T. Eliassi–Rad, M. Faloutsos, and C. Faloutsos, *Gelling, and melting, large graphs by edge manipulation*, in Proceedings of the 21st ACM International Conference on Information and Knowledge Management, ACM, New York, 2012, pp. 245–254.

[41] P. Van Mieghem, D. Stevanović, F. Kuipers, C. Li, R. van de Bovenkamp, D. Liu, and H. Wang, *Decreasing the spectral radius of a graph by link removals*, Phys. Rev. E, 84 (2011), 016101.

[42] D. J. Watts and S. H. Strogatz, *Collective dynamics of 'small-world' networks*, Nature, 393 (1998), pp. 440–442.

[43] J. Wu, M. Barahona, Y. Tan, and H. Deng, *Natural connectivity of complex networks*, Chinese Phys. Lett., 27 (2010), 078902.

[44] J. Wu, M. Barahona, Y. Tan, and H. Deng, *Robustness of random graphs based on graph spectra*, Chaos, 22 (2012), 043101.