



# Edge importance in complex networks

Silvia Noschese<sup>1</sup> · Lothar Reichel<sup>2</sup>

Received: 5 April 2024 / Accepted: 5 July 2024

© The Author(s) 2024

## Abstract

Complex networks are made up of vertices and edges. The latter connect the vertices. There are several ways to measure the importance of the vertices, e.g., by counting the number of edges that start or end at each vertex, or by using the subgraph centrality of the vertices. It is more difficult to assess the importance of the edges. One approach is to consider the line graph associated with the given network and determine the importance of the vertices of the line graph, but this is fairly complicated except for small networks. This paper compares two approaches to estimate the importance of edges of medium-sized to large networks. One approach computes partial derivatives of the total communicability of the weights of the edges, where a partial derivative of large magnitude indicates that the corresponding edge may be important. Our second approach computes the Perron sensitivity of the edges. A high sensitivity signals that the edge may be important. The performance of these methods and some computational aspects are discussed. Applications of interest include to determine whether a network can be replaced by a network with fewer edges with about the same communicability.

**Keywords** Network analysis · Sensitivity analysis · Edge importance

**Mathematics Subject Classification (2010)** 05C82 · 15A16 · 65F15

## 1 Introduction

Networks are helpful for modeling complex interactions between entities. A network can be represented by a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ , which consists of a set of *vertices* or *nodes*  $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ , a set of *edges*  $\mathcal{E}$ , with  $|\mathcal{E}| = m$ , that connect the vertices, and a set of nonnegative edge weights  $\mathcal{W} = \{w_{11}, w_{21}, \dots, w_{nn}\}$ . The weight  $w_{ij}$

---

✉ Silvia Noschese  
noschese@mat.uniroma1.it

Lothar Reichel  
reichel@math.kent.edu

<sup>1</sup> Dipartimento di Matematica, SAPIENZA Università di Roma, P.le Aldo Moro 5, 00185 Roma, Italy

<sup>2</sup> Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA

is positive if there is an edge pointing from vertex  $v_i$  to vertex  $v_j$ ; while  $w_{ij} = 0$  signifies that there is no such edge. Edges may be directed (and then model one-way streets) or undirected (and then model two-way streets). If the graph models a road network in which the vertices model intersections and the edges model roads, then the weights may, e.g., be proportional to the amount of traffic along each road. A network in which all positive weights are one is said to be *unweighted*. Descriptions and many applications of networks are provided by, e.g., Estrada [11] and Newman [19].

We say that vertex  $v_i$  is *directly connected* to vertex  $v_j$  if there is a single edge from vertex  $v_i$  pointing to vertex  $v_j$ . Vertex  $v_j$  is then said to be *adjacent* to vertex  $v_i$ . When the edge between these vertices is undirected, vertex  $v_j$  also is directly connected to vertex  $v_i$ , and  $v_i$  is adjacent to vertex  $v_j$ . Vertex  $v_i$  is said to be *indirectly connected* to vertex  $v_j$  if the latter vertex can be reached from the former by following at least two edges from  $v_i$ . We will consider graphs without multiple edges and without edges that start and end at the same vertex.

Let  $e(v_i \rightarrow v_j)$  denote an edge from vertex  $v_i$  to  $v_j$ . If there also is an edge  $e(v_j \rightarrow v_i)$  and  $w_{ij} = w_{ji} > 0$ , then the edge is said to be undirected and denoted by  $e(v_i \leftrightarrow v_j)$ . A sequence of edges (not necessarily distinct)

$$\{e(v_1 \rightarrow v_2), e(v_2 \rightarrow v_3), \dots, e(v_k \rightarrow v_{k+1})\}$$

forms a *walk*. The *length* of a walk is the sum of the weights of the edges that make up the walk, i.e.,  $\sum_{i=1}^k w_{i,i+1}$ . If the edges in a walk are distinct, then the walk is referred to as a *path*.

Introduce the adjacency matrix  $\mathbf{A} = [w_{ij}]_{i,j=1}^n \in \mathbb{R}^{n \times n}$  associated with the graph  $\mathcal{G}$ . The adjacency matrix  $\mathbf{A}$  is *sparse* in most applications, i.e., the matrix has many more zero entries than positive entries. The matrix is symmetric if for each edge there also is an edge in the opposite direction with the same weight. The graph determined by such an adjacency matrix is said to be *undirected*. If at least one edge of a graph is directed, or if  $w_{ij} \neq w_{ji}$  for at least one index pair  $\{i, j\}$ , then the graph is said to be *directed*. The adjacency matrix associated with a directed graph is nonsymmetric. Since we assume that there are no edges that start and end at the same vertex, the diagonal entries of the adjacency matrix  $\mathbf{A}$  vanish.

A graph is said to be *connected* if for every pair of vertices  $v_i$  and  $v_j$ , there is a path from vertex  $v_i$  to vertex  $v_j$  and from vertex  $v_j$  to vertex  $v_i$ . Directed graphs with this property are sometimes referred to as *strongly connected*. A directed graph is said to be *weakly connected* if the undirected graph that is obtained by replacing all directed edges by undirected ones is connected. A graph is strongly connected if and only if the adjacency matrix associated with the graph is irreducible. This provides a computational approach to determine whether a graph is strongly connected. Further, an undirected graph is connected if and only if the second smallest eigenvalue of the associated graph Laplacian is positive; see e.g., [11, 19].

Let  $\exp_0(t) = \exp(t) - 1$  and consider the power series expansion

$$\exp_0(\mathbf{A}) = \mathbf{A} + \frac{\mathbf{A}^2}{2!} + \frac{\mathbf{A}^3}{3!} + \dots \tag{1}$$

Let  $\mathbf{A}^k = [a_{ij}^{(k)}]_{i,j=1}^n$  for  $k = 1, 2, \dots$ , where  $a_{ij}^{(1)} = w_{ij}$  for  $1 \leq i, j \leq n$ . A nonvanishing entry  $a_{ij}^{(k)}$  for some  $k > 0$  indicates that there is at least one walk of  $k$  edges from vertex  $v_i$  to vertex  $v_j$ . In case of an unweighted network,  $a_{ij}^{(k)}$  represents the number of walks of length  $k$  from vertex  $v_i$  to vertex  $v_j$ . The denominators in the terms of the expansion (1) ensure that the expansion converges and that terms  $\frac{\mathbf{A}^k}{k!}$  with  $k$  large contribute only little to  $\exp_0(\mathbf{A})$ . It follows that short walks typically are more important than long ones, which is in agreement with the intuition that messages propagate better along short walks than along long ones. This led Estrada and Rodriguez-Velazquez [13] to use  $\exp(\mathbf{A})$  to study properties of a graph; we use the function  $\exp_0(\mathbf{A})$  because the term with the identity matrix  $\mathbf{I}$  in the expansion of  $\exp(\mathbf{A})$  has no natural interpretation in the context of network modeling. Other functions also can be used such as a resolvent or a Mittag-Leffler function; see Estrada and Higham [12] and Arrigo and Durastante [2] for discussions.

Estrada and Rodriguez-Velazquez [13] define for graphs with a symmetric adjacency matrix the communicability matrix  $\exp(\mathbf{A})$ ; we will use the matrix

$$\mathbf{C} = [c_{ij}]_{i,j=1}^n = \exp_0(\mathbf{A}).$$

The entry  $c_{ij}$ ,  $i \neq j$ , is referred to as the *communicability* between the vertices  $v_i$  and  $v_j$ . A relatively large value implies that it is easy for the vertices  $v_i$  and  $v_j$  to communicate. Estrada and Rodriguez-Velazquez [13] measure the importance of the vertex  $v_i$  of an undirected graph by the *subgraph centrality*  $c_{ii} + 1$ ; we will use  $c_{ii}$ . Related measures of communicability can be defined when the adjacency matrix  $\mathbf{A}$  is nonsymmetric; see [6].

We define the *total communicability* of the graph  $\mathcal{G}$  as

$$T_{\mathcal{G}}(w_{11}, w_{21}, \dots, w_{nn}) = e^T \exp_0(\mathbf{A})e, \tag{2}$$

where  $e = [1, 1, \dots, 1]^T \in \mathbb{R}^n$  denotes the vector with all entries 1 and the superscript  $T$  stands for transposition. Benzi and Klymko [3] introduced the related measure  $e^T \exp(\mathbf{A})e$ , which differs from (2) by the additive constant  $n$ . Note that the expression (2) is invariant under transposition. We have

$$T_{\mathcal{G}}(w_{11}, w_{21}, \dots, w_{nn}) = e^T \exp_0(\mathbf{A}^T)e = \frac{1}{2}e^T (\exp_0(\mathbf{A}) + \exp_0(\mathbf{A}^T))e.$$

The graph associated with the adjacency matrix  $\mathbf{A}^T$  is known as the *reverse graph* to  $\mathcal{G}$ . Thus, the total communicability of the graph  $\mathcal{G}$  and of the reverse graph are the same.

We are interested in investigating the importance of the edges of a graph  $\mathcal{G}$  and, in particular, in determining which edge weights can be reduced or set to zero without significantly affecting the total communicability. A possible approach to investigate the importance of edges is to consider the line graph associated with the graph. The edges of  $\mathcal{G}$  correspond to vertices in the associated line graph, and the importance of the edges in  $\mathcal{G}$  can be measured by the subgraph centrality of the vertices of the line graph. This

approach is investigated in [7, 8]. However, it is quite cumbersome to construct the line graph except for small graphs. A simple heuristic technique was proposed by Arrigo and Benzi [1], who define the edge total communicability and seek to remove edges whose removal does not reduce the edge total communicability much. This approach is quite easy to implement for networks that are small enough to allow the computation of the singular value decomposition of the associated adjacency matrix; however, it is not straightforward to use for large-scale networks. Moreover, the importance of an edge is assumed to depend on the importance of the vertices that it connects. This holds for some networks, but not for others. Therefore, this approach to edge removal may result in removals that are not in agreement with intuition. More recently an approach that uses the right and left Perron vectors of the adjacency matrix in combination with the Wilkinson perturbation to determine which weights to increase in order to increase the total communicability has been described in [4, 21]. An analogous technique is applied in [10] to discern which weights in a weighted multilayer network can be decreased without affecting the total communicability significantly.

Another approach to study the importance of edges is to evaluate the Fréchet derivatives of the total communicability (2) with respect to the weights. This approach was advocated by De la Cruz Cabrera et al. [4] and recently Schweitzer [27] described how to speed up the computations. Introduce the gradient

$$\nabla T_G(w_{11}, w_{21}, \dots, w_{nn}) = \left[ \frac{\partial}{\partial w_{11}} e^T \exp_0(\mathbf{A})e, \frac{\partial}{\partial w_{21}} e^T \exp_0(\mathbf{A})e, \dots, \frac{\partial}{\partial w_{nn}} e^T \exp_0(\mathbf{A})e \right]^T. \tag{3}$$

When the partial derivative

$$\frac{\partial}{\partial w_{ij}} e^T \exp_0(\mathbf{A})e \tag{4}$$

is (relatively) large, a small increase in the positive weight  $w_{ij}$  results in a substantial change in the total communicability. We will show below that the partial derivatives (4) are nonnegative.

For example, let the graph represent a road map, where the edges model roads and the vertices represent intersections of roads. Let vertex  $v_j$  be adjacent to vertex  $v_i$ . Then widening an existing road from  $v_i$  to  $v_j$  may result in a significant increase in the total communicability of the graph; the widening of this road is modeled by increasing the weight  $w_{ij}$ . Also, when (4) is relatively large, and vertex  $v_j$  is not adjacent to vertex  $v_i$ , i.e., there is no road from  $v_i$  to  $v_j$ , building such a road may increase the total communicability substantially. This is modeled by making the vanishing weight  $w_{ij}$  positive.

Conversely, if the partial derivative (4) is relatively small and the weight  $w_{ij} > 0$  is small, then setting  $w_{ij}$  to zero, i.e., removing the edge from vertex  $v_i$  to vertex  $v_j$ , will not affect the total communicability (2) much. This implies that blocking the road from vertex  $v_i$  to  $v_j$ , e.g., due to construction, does not change the total communicability of the network significantly.

This paper is organized as follows. Section 2 discusses how the gradient can be applied to assess the importance of edges. The first part of the section is concerned with small to medium-sized problems for which it is feasible to evaluate the gradient (3). The latter part of the section discusses the application of Krylov subspace methods to project large-scale problems to problems of fairly small size. The computations use a result by Schweitzer [27] on the evaluation of Fréchet derivatives, but differ in various aspects that speed up the computations. Section 3 reviews methods described in [4, 10, 21] based on evaluating the right and left Perron vectors and the Wilkinson perturbation to determine important and unimportant edges. One of the aims of this paper is to compare the methods discussed in Sections 2 and 3. This is done in Section 4, where we also report timings. Section 5 contains concluding remarks.

We conclude this section with comments on some related methods. A scheme that combines regression, soft-thresholding, and projection is applied in [5] to approximate an unweighted network by a simpler unweighted network. This scheme performs well but may be expensive and is restricted to unweighted networks. Massei and Tudisco [17] consider the problem of determining a low-rank perturbation  $\mathbf{E} \in \mathbb{R}^{n \times n}$  to the adjacency matrix  $\mathbf{A}$  so that the perturbed matrix  $\mathbf{A} + \mathbf{E}$  maximizes or minimizes the robustness of the network. For instance,  $\mathbf{E}$  may be chosen to maximize or minimize the trace of  $f(\mathbf{A} + \mathbf{E}) - f(\mathbf{A})$  for a user-specified matrix-valued function  $f$ . Thus, this method seeks to modify a few edge weights so that the trace is increased or decreased as much as possible. The perturbation  $\mathbf{E}$  is determined by a greedy algorithm for solving an optimization problem that can be quite expensive to solve. A careful comparison with this method is outside the scope of the present paper.

## 2 Network modifications based on the gradient

This section discusses methods for modifying, adding, or removing edges of a network by using information furnished by the gradient (3). We first describe methods for small to medium-sized networks for which all entries of the gradient (3) can be evaluated. Subsequently, we will consider Krylov subspace methods that can be applied to large-scale networks.

### 2.1 Methods for small to medium-sized networks

Let the matrix function  $f : \mathbf{A} \in \mathbb{R}^{n \times n} \rightarrow f(\mathbf{A}) \in \mathbb{R}^{n \times n}$  be continuously differentiable sufficiently many times in a region in the complex plane that contains all eigenvalues of  $\mathbf{A}$ . Then the function  $f$  has a Fréchet derivative  $L_f(\mathbf{A}, \mathbf{E}) \in \mathbb{R}^{n \times n}$  at  $\mathbf{A}$  in the direction  $\mathbf{E} \in \mathbb{R}^{n \times n} \setminus \{\mathbf{0}\}$ . The Fréchet derivative satisfies

$$f(\mathbf{A} + \mathbf{E}) = f(\mathbf{A}) + L_f(\mathbf{A}, \mathbf{E}) + o(\|\mathbf{E}\|), \quad (5)$$

as  $\|\mathbf{E}\| \rightarrow 0$ , where  $\|\cdot\|$  is any matrix norm; see, e.g., [15] for details. Schweitzer described an efficient approach to evaluate  $L_f(\mathbf{A}, \mathbf{E})$  in several directions  $\mathbf{E}$  simultaneously.

**Theorem 1** (Schweitzer [27, Theorem 2.3]) *Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  and  $u, v \in \mathbb{R}^n \setminus \{0\}$ , and assume that  $f$  is Fréchet differentiable at  $\mathbf{A}$ . Define  $\mathbf{E}_{ij} = e_i e_j^T$ , where  $e_k = [0, \dots, 0, 1, 0, \dots, 0]^T \in \mathbb{R}^n$  denotes the  $k$ th column of the identity matrix. Then*

$$u^T L_f(\mathbf{A}, \mathbf{E}_{ij})v = e_i^T L_f(\mathbf{A}^T, uv^T)e_j. \tag{6}$$

Thus, the entries of the matrix  $L_f(\mathbf{A}^T, uv^T)$  furnish Fréchet derivatives in all directions  $\mathbf{E}_{ij} = e_i e_j^T, 1 \leq i, j \leq n$ . We are primarily interested in the situation when  $f(t) = \exp_0(t)$  and

$$u = v = e = [1, 1, \dots, 1]^T \in \mathbb{R}^n. \tag{7}$$

**Theorem 2** *All entries of the gradient (3) are nonnegative.*

**Proof** Let  $\mathbf{E}_{ij} = e_i e_j^T$ . Then for  $f(t) = \exp_0(t)$ , we have

$$\frac{\partial}{\partial w_{ij}} e^T \exp_0(\mathbf{A})e = e^T L_f(\mathbf{A}, \mathbf{E}_{ij})e.$$

It follows from (5) that

$$L_f(\mathbf{A}, h\mathbf{E}_{ij}) = \exp_0(\mathbf{A} + h\mathbf{E}_{ij}) - \exp_0(\mathbf{A}) + o(h) \text{ as } h \searrow 0.$$

The power series expansion of  $f(t) = \exp_0(t)$  gives

$$\begin{aligned} L_f(\mathbf{A}, h\mathbf{E}_{ij}) &= h \left( \mathbf{E}_{ij} + \frac{\mathbf{A}\mathbf{E}_{ij} + \mathbf{A}\mathbf{E}_{ij}}{2!} + \frac{\mathbf{A}^2\mathbf{E}_{ij} + \mathbf{A}\mathbf{E}_{ij}\mathbf{A} + \mathbf{E}_{ij}\mathbf{A}^2}{3!} + \dots \right) + o(h) \\ &= h \left( \sum_{\ell=1}^{\infty} \frac{1}{\ell!} \sum_{k=0}^{\ell-1} \frac{1}{\ell!} \mathbf{A}^k \mathbf{E}_{ij} \mathbf{A}^{\ell-1-k} + o(1) \right). \end{aligned}$$

Each term in the above sum is a matrix with nonnegative entries. Hence, the sum is a matrix with nonnegative entries. The term  $o(1)$  vanishes as  $h \searrow 0$ . Since  $L_f(\mathbf{A}, h\mathbf{E}_{ij})$  is linear in  $h$ , we obtain

$$L_f(\mathbf{A}, \mathbf{E}_{ij}) = \lim_{h \searrow 0} \frac{L_f(\mathbf{A}, h\mathbf{E}_{ij})}{h} = \sum_{\ell=1}^{\infty} \frac{1}{\ell!} \sum_{k=0}^{\ell-1} \frac{1}{\ell!} \mathbf{A}^k \mathbf{E}_{ij} \mathbf{A}^{\ell-1-k}. \tag{8}$$

This completes the proof.

A possible way to evaluate the matrix  $L_f(\mathbf{A}^T, uv^T)$  in (6) when  $\mathbf{A}^T \in \mathbb{R}^{n \times n}$  is to use the relation

$$f \left( \begin{bmatrix} \mathbf{A}^T & uv^T \\ \mathbf{0} & \mathbf{A}^T \end{bmatrix} \right) = \begin{bmatrix} f(\mathbf{A}^T) & L_f(\mathbf{A}^T, uv^T) \\ \mathbf{0} & f(\mathbf{A}^T) \end{bmatrix}; \tag{9}$$

see, e.g., [15, p.253]. However, when  $f(t) = \exp_0(t)$ , the computation of  $f(\mathbf{A}^T)$  requires  $\mathcal{O}(n^3)$  arithmetic floating point operations (flops). Therefore, the evaluation

of the left-hand side of (9) demands about 8 times more flops than the calculation of  $f(\mathbf{A}^T)$ . It is cheaper to approximate  $L_f(\mathbf{A}^T, uv^T)$  by using the finite-difference approximation

$$L_f(\mathbf{A}^T, uv^T) \approx \frac{f(\mathbf{A}^T + huv^T) - f(\mathbf{A}^T - huv^T)}{2h} \tag{10}$$

for some  $h > 0$ . We will use  $h = \frac{2}{n} \cdot 10^{-4}$  in the computed examples in Section 4. This is suggested by the following simple computations. We have used the fact that  $\|uv^T\|_2 = n$ , which holds for the vectors (7). Here and throughout this paper  $\|\cdot\|_2$  denotes the spectral matrix norm or the Euclidean vector norm.

Example 2.1. Let  $f(t) = \exp_0(t)$  be evaluated with a relative error  $\delta_t$  bounded by  $\delta > 0$  and let  $h > 0$  be a small scalar. Then

$$\begin{aligned} \frac{f(t+h) - f(t-h)}{2h} &\approx \frac{f_{\text{exact}}(t+h) + \delta_{t+h}f_{\text{exact}}(t) - (f_{\text{exact}}(t-h) + \delta_{t-h}f_{\text{exact}}(t))}{2h} \\ &\approx f'_{\text{exact}}(t) + \frac{h^2}{6}f'''_{\text{exact}}(t) + \frac{\delta_{t+h} - \delta_{t-h}}{2h}f_{\text{exact}}(t). \end{aligned}$$

Thus, the error is bounded by about

$$\left(\frac{h^2}{6} + \frac{\delta}{h}\right)\exp(t).$$

Minimization over  $h > 0$  yields

$$h \approx (3\delta)^{1/3}.$$

The computation of the scalar exponential is carried out with high relative accuracy in MATLAB. However, evaluation of the matrix exponential  $\exp(\mathbf{A}^T)$  is more difficult. It can be computed in several ways; see, e.g., [15, Chapter 10] as well as [24, 30]. The accuracy achieved depends on the method used as well as on the size and properties of the matrix  $\mathbf{A}^T$ ; see, e.g., [15, Chapter 10] and [24] for computed examples. We therefore include a factor  $10^3$  in the bound  $\delta$  for the relative accuracy. This bound is valid for most matrices of sizes of interest to us. Letting  $\delta \approx 10^3\epsilon_{\text{mach}}$  with  $\epsilon_{\text{mach}} \approx 2 \cdot 10^{-16}$  gives  $h \approx 2 \cdot 10^{-4}$ .  $\square$

**Proposition 1** *Let  $f(t) = \exp_0(t)$ . Then*

$$\frac{f(\mathbf{A}^T + h\mathbf{E}_{ij}) - f(\mathbf{A}^T - h\mathbf{E}_{ij})}{2h} = L_f(\mathbf{A}^T, \mathbf{E}_{ij}) + O(h^2).$$

**Proof** The right-hand side of (10) with  $uv^T$  replaced with  $\mathbf{E}_{ij}$  can be expressed as

$$\begin{aligned} \frac{f(\mathbf{A}^T + h\mathbf{E}_{ij}) - f(\mathbf{A}^T - h\mathbf{E}_{ij})}{2h} &= \mathbf{E}_{ij} + \frac{1}{2!}(\mathbf{A}^T\mathbf{E}_{ij} + \mathbf{E}_{ij}\mathbf{A}^T) \\ &+ \frac{1}{3!}((\mathbf{A}^T)^2\mathbf{E}_{ij} + \mathbf{A}^T\mathbf{E}_{ij}\mathbf{A}^T + \mathbf{E}_{ij}(\mathbf{A}^T)^2) \\ &+ \frac{1}{4!}((\mathbf{A}^T)^3\mathbf{E}_{ij} + (\mathbf{A}^T)^2\mathbf{E}_{ij}\mathbf{A}^T + \mathbf{A}^T\mathbf{E}_{ij}(\mathbf{A}^T)^2 + \mathbf{E}_{ij}(\mathbf{A}^T)^3) \\ &+ \dots + O(h^2). \end{aligned} \tag{11}$$

The result now follows from (8) with  $\mathbf{A}$  replaced by  $\mathbf{A}^T$ .

The above proposition is a corollary of the well-known series representation of the Fréchet derivative; see, e.g., [15]. It can be stated for an arbitrary direction matrix  $\mathbf{E}$  by just replacing the last term  $O(h^2)$  in (11) with  $O(\|\mathbf{E}\|_2 h^2)$ . The evaluation of the right-hand side of (10) with  $f(t) = \exp_0(t)$  gives approximations of all the entries of the gradient (3). We will refer to  $\|\nabla T_{\mathcal{G}}(w_{11}, w_{21}, \dots, w_{nn})\|_2$  as the *total transmission* of the graph  $\mathcal{G}$ .

### 2.1.1 Network simplification by edge removal

One of the aims of this paper is to discuss how to reduce the complexity of a network by removing edges without changing the total transmission or total communicability significantly. A simple way to achieve the former is to set positive weights  $w_{ij}$  to zero when the associated entries of the gradient (4) are (relatively) small, thus removing the corresponding edges  $e(v_i \rightarrow v_j)$ . This determines a new network  $\tilde{\mathcal{G}}$  with fewer edges than  $\mathcal{G}$  with about the same total transmission. However, in order for the network  $\tilde{\mathcal{G}}$  also to have about the same total communicability as  $\mathcal{G}$ , we also have to require that the removed weights be small. We therefore introduce the vector  $\mathcal{E}_{L_f} \in \mathbb{R}^m$ , whose  $k$ th entry is the *edge importance* of  $e_k = e(v_i \rightarrow v_j)$ , defined as the product of the weight  $w_{ij}$  and the corresponding partial derivative (4) normalized by the total transmission. Observe that  $\|L_f(\mathbf{A}^T, ee^T)\|_F = \|\nabla T_{\mathcal{G}}\|_2$ , where  $\|\cdot\|_F$  stands for the Frobenius norm. We refer to the norm  $\|\mathcal{E}_{L_f}\|_2$  as the *total edge importance*.

The following simple procedure can be used to construct the edge importance vector  $\mathcal{E}_{L_f}$  for undirected graphs:

**Procedure 1:**

1. Multiply the adjacency matrix  $\mathbf{A}$  element by element by the matrix  $L_f(\mathbf{A}^T, ee^T)$ .
2. Divide the elements of the so obtained matrix that correspond to edges of  $\mathcal{G}$  by the total transmission  $\|\nabla T_{\mathcal{G}}\|_2$  and put them column by column into the vector  $\mathcal{E}_{L_f}$ .

If the graph is undirected, then the  $k$ th entry of the vector  $\mathcal{E}_{L_f} \in \mathbb{R}^m$  gives the importance of the edge  $e_k = e(v_i \leftrightarrow v_j)$ . We obtain the following procedure:



**Procedure 2:**

1. Extract the strictly lower triangular portion  $\mathbf{L}$  of the adjacency matrix  $\mathbf{A}$ , and multiply  $\mathbf{L}$  element by element by the strictly lower triangular portion of the matrix  $L_f(\mathbf{A}^T, ee^T)$ .
2. Divide the elements of the matrix so obtained that correspond to the edges  $e(v_i \rightarrow v_j)$  of  $\mathcal{G}$  with  $i > j$  by  $\frac{\|\nabla T_{\mathcal{G}}\|_2}{2}$  and put them column by column into the vector  $\mathcal{E}_{L_f}$ .

Consider the cone  $\mathcal{A}$  of all nonnegative matrices in  $\mathbb{R}^{n \times n}$  with the same sparsity structure as  $\mathbf{A}$  and let  $\mathbf{M}|_{\mathcal{A}}$  denote a matrix in  $\mathcal{A}$  that is closest to a given nonnegative matrix  $\mathbf{M} \in \mathbb{R}^{n \times n}$  with respect to the Frobenius norm. It is straightforward to verify that  $\mathbf{M}|_{\mathcal{A}}$  is obtained by setting all the entries outside the sparsity structure of  $\mathbf{A}$  to zero.

In the first step of the Procedure 1, one considers the matrix  $L_f(\mathbf{A}^T, ee^T)|_{\mathcal{A}}$ , whereas in the first step of the second procedure one considers the projected matrix  $L_f(\mathbf{A}^T, ee^T)|_{\mathcal{L}}$  with  $\mathcal{L}$  the cone of all nonnegative matrices in  $\mathbb{R}^{n \times n}$  with the same sparsity structure as  $\mathbf{L}$ . It follows that  $\mathbf{L} = \mathbf{A}|_{\mathcal{L}}$ .

In computations, we order the entries of  $\mathcal{E}_{L_f}$  from smallest to largest and set the weights  $w_{ij}$  (or  $w_{ij} = w_{ji}$  if the graph is undirected) associated with the first few of the ordered edge importances to zero. Some post-processing may be necessary if the reduced graph  $\tilde{\mathcal{G}}$  obtained by removing the edges associated with the weights that are set to zero is required to be connected.

**2.1.2 Network modification to increase or decrease total communicability**

We turn to the task of increasing or decreasing the total communicability of a network by changing a few weights. The weights to be changed are chosen with the aid of the entries of the vector  $\mathcal{E}_{L_f} \in \mathbb{R}^m$ . We obtain a relatively large increase/reduction in the total communicability by slightly increasing/reducing the weights associated with the largest entries of  $\mathcal{E}_{L_f}$ . To this end, we order the entries of  $\mathcal{E}_{L_f}$  from the largest to the smallest. More than one of the weights can be modified to achieve a desired increase or reduction in the total communicability.

Assume that the given graph  $\mathcal{G}$  is strongly or weakly connected, and that we would like the modified graph to have the same property. Consider the situation when removing an edge  $e_k$  associated with one of the first few of the ordered entries of the vector  $\mathcal{E}_{L_f}$  results in a graph that does not have this property. Then typically the total communicability can be decreased considerably by reducing the corresponding edge-weight  $w_{ij}$  to a small positive value (or both the weights  $w_{ij}$  and  $w_{ji}$  to the same small positive value if the graph is undirected), with the perturbed graph so obtained having the same connectivity property as the original graph.

**2.1.3 Network modification by inclusion of new edges**

The partial derivatives (4) reveal which edges would be important to add to a given graph to increase the communicability of the network significantly, namely nonexistent

edges, whose associated partial derivative is large. Let  $\widehat{\mathcal{A}}$  denote the cone of the nonnegative matrices in  $\mathbb{R}^{n \times n}$  with sparsity structure given by the zero entries of  $\mathbf{A}$  except for the diagonal entries. The *virtual importance* of the nonexistent edge  $e(v_i \rightarrow v_j) \notin \mathcal{E}$  is given by the corresponding entry of  $L_f(\mathbf{A}^T, ee^T)$  normalized by the total transmission. The construction of the virtual edge importance vector  $\widehat{\mathcal{E}}_{L_f} \in \mathbb{R}^{n^2-n-m}$ , which makes use of matrix entries in the sparsity structure associated with  $\widehat{\mathcal{A}}$ , can be summarized as follows:

**Procedure 3:**

1. Construct the matrix  $L_f(\mathbf{A}^T, ee^T)|_{\widehat{\mathcal{A}}}$ .
2. Divide the entries of the matrix so obtained that belong to the sparsity structure associated with  $\widehat{\mathcal{A}}$  by the total transmission  $\|\nabla T_G\|_2$  and put them column by column into the vector  $\widehat{\mathcal{E}}_{L_f}$ .

If the graph is undirected, then the virtual importance of the virtual edge  $e(v_i \leftrightarrow v_j) \notin \mathcal{E}$  is defined as twice the corresponding entry in  $L_f(\mathbf{A}^T, ee^T)$  normalized by the total transmission. Let  $\widehat{\mathcal{L}}$  be the cone of the nonnegative matrices in  $\widehat{\mathcal{A}}$ , where all the entries in the strictly upper triangular portion are set to zero. The procedure for the construction of the virtual edge importance vector  $\widehat{\mathcal{E}}_{L_f} \in \mathbb{R}^{(n^2-n-2m)/2}$ , which makes use of matrix entries in the sparsity structure associated with  $\widehat{\mathcal{L}}$ , becomes:

**Procedure 4:**

1. Construct the matrix  $L_f(\mathbf{A}^T, ee^T)|_{\widehat{\mathcal{L}}}$ .
2. Divide the entries of the matrix so obtained that belong to the sparsity structure associated with  $\widehat{\mathcal{L}}$  by  $\frac{\|\nabla T_G\|_2}{2}$  and put them column by column into the vector  $\widehat{\mathcal{E}}_{L_f}$ .

In case  $\|L_f(\mathbf{A}^T, ee^T)|_{\mathcal{A}}\|_F \ll \|\nabla T_G\|_2$  many large partial derivatives (4) correspond to zero extra-diagonal entries of  $\mathbf{A}$ . Then making suitable zero weights  $w_{ij}$  positive, or if the graph is undirected giving suitable pairs of zero weights  $\{w_{ij}, w_{ji}\}$  the same positive value, might be beneficial. We recall that replacing a zero weight  $w_{ij}$  with a positive weight  $w_{ij}$  is equivalent to including a weighted edge  $e(v_i \leftrightarrow v_j)$  into the graph.

In computations, we order the entries of  $\widehat{\mathcal{E}}_{L_f}$  from largest to smallest and add the positive weights  $w_{ij}$  (or pairs of positive entries  $w_{ij} = w_{ji}$  if the graph is undirected) associated with the first few of the virtual ordered edge importances.

**2.2 Methods for large networks**

Recently, Kandolf et al. [16] derived a method for evaluating approximations of the Fréchet derivative of a matrix function by Krylov subspace methods. Applications of this technique to network analysis have recently been discussed by De la Cruz Cabrera et al. [4] and Schweitzer [27]. We first outline this method and subsequently discuss some alternatives.

Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  and assume that the function  $f$  is analytic in an open simply connected set  $\Omega$  in the complex plane that contains the spectrum of  $\mathbf{A}$ . Then

$$f(\mathbf{A}) = \frac{1}{2\pi i} \int_{\Gamma} f(z)(z\mathbf{I} - \mathbf{A})^{-1} dz,$$

where  $\Gamma$  is a curve in  $\Omega$  that winds around the spectrum of  $\mathbf{A}$  exactly once and  $i = \sqrt{-1}$ . In this paper, we are primarily interested in the situation when  $f(z) = \exp_0(z)$ , but the techniques discussed apply to other analytic functions as well. Let  $u, v \in \mathbb{R}^n$  be nonvanishing vectors. The Fréchet derivative of  $f$  at  $\mathbf{A}$  in the direction  $uv^T$  can be expressed as

$$L_f(\mathbf{A}, uv^T) = \frac{1}{2\pi i} \int_{\Gamma} f(z)(z\mathbf{I} - \mathbf{A})^{-1} uv^T (z\mathbf{I} - \mathbf{A})^{-1} dz;$$

see, e.g., [15]. Kandolf et al. [16] determine an approximation of this expression by using Krylov subspace techniques to approximate the vectors

$$s(z) = (z\mathbf{I} - \mathbf{A})^{-1} u, \quad t(z) = (z\mathbf{I} - \mathbf{A})^{-H} v, \quad z \in \Gamma,$$

where the superscript  $H$  denotes transposition and complex conjugation. Specifically, Kandolf et al. [16] and Schweitzer [27] approximate the vectors  $s(z)$  and  $t(z)$  by a Krylov subspace technique based on the Arnoldi process. Application of  $1 \leq \ell \ll n$  steps of the Arnoldi process to  $\mathbf{A}$  with initial vector  $u$ , and to  $\mathbf{A}^T$  with initial vector  $v$ , generically, yields the Arnoldi decompositions

$$\mathbf{A}\mathbf{V}_\ell = \mathbf{V}_\ell \mathbf{G}_\ell + \tilde{v}_{\ell+1} e_\ell^T, \quad \mathbf{A}^T \mathbf{W}_\ell = \mathbf{W}_\ell \mathbf{M}_\ell + \tilde{w}_{\ell+1} e_\ell^T, \tag{12}$$

where the matrices  $\mathbf{G}_\ell, \mathbf{M}_\ell \in \mathbb{R}^{\ell \times \ell}$  are of upper Hessenberg form, the matrix  $\mathbf{V}_\ell \in \mathbb{R}^{n \times \ell}$  has orthonormal columns with initial column  $u/\|u\|_2$ , the vector  $\tilde{v}_{\ell+1} \in \mathbb{R}^n$  is orthogonal to the range of  $\mathbf{V}_\ell$ , the matrix  $\mathbf{W}_\ell \in \mathbb{R}^{n \times \ell}$  has orthonormal columns with initial column  $v/\|v\|_2$ , and the vector  $\tilde{w}_{\ell+1} \in \mathbb{R}^n$  is orthogonal to the range of  $\mathbf{W}_\ell$ ; see, e.g., Saad [26, Chapter 6] for details on the Arnoldi process. Here we only note that the computation of the decompositions (12) requires the evaluation of  $\ell$  matrix-vector products with the matrix  $\mathbf{A}$  and  $\ell$  matrix-vector products with the matrix  $\mathbf{A}^T$ . This is the dominating computational work when the matrix  $\mathbf{A}$  is large and the number of Arnoldi steps is fairly small. We assume here that the Arnoldi processes do not break down when computing (12); in case of breakdown, the formulas (12) simplify.

Kandolf et al. [16] propose to use the Arnoldi approximation

$$L_{f, \text{Arnoldi}}(\mathbf{A}, uv^T) = \mathbf{V}_\ell \mathbf{X}_\ell \mathbf{W}_\ell^T, \tag{13}$$

of  $L_f(\mathbf{A}, uv^T)$ , where  $\mathbf{X}_\ell$  is the upper right  $\ell \times \ell$  submatrix of the  $2\ell \times 2\ell$  matrix

$$f \left( \begin{bmatrix} \mathbf{G}_\ell & \|u\|_2 \|v\|_2 e_1 e_1^T \\ \mathbf{0} & \mathbf{M}_\ell^T \end{bmatrix} \right) = \begin{bmatrix} f(\mathbf{G}_\ell) & \mathbf{X}_\ell \\ \mathbf{0} & f(\mathbf{M}_\ell^T) \end{bmatrix} \tag{14}$$

with  $e_1 = [1, 0, \dots, 0]^T$ . Schweitzer [27] applies formulas (12), (13), and (14) with  $\mathbf{A}$  replaced by  $\mathbf{A}^T$  and  $u = v = e$ ; cf. (6). Convergence results are provided by Kandolf et al. [16].

An alternative approach to compute an approximation of the Fréchet derivative  $L_f(\mathbf{A}^T, ee^T)$  is to apply finite-difference approximations analogously as (10). Application of  $\ell$  steps of the Arnoldi process to the matrix  $\mathbf{A}^T$  with initial vector  $e$  gives the Arnoldi decomposition

$$\mathbf{A}^T \mathbf{U}_\ell = \mathbf{U}_\ell \mathbf{H}_\ell + \tilde{u}_{\ell+1} e_\ell^T,$$

where the columns of  $\mathbf{U}_\ell \in \mathbb{R}^{n \times \ell}$  are orthonormal and span the Krylov subspace

$$\mathbf{K}_\ell(\mathbf{A}^T, e) = \text{span}\{e, \mathbf{A}^T e, (\mathbf{A}^T)^2 e, \dots, (\mathbf{A}^T)^{\ell-1} e\}$$

and the first column of  $\mathbf{U}_\ell$  is  $e/\sqrt{n}$ . Moreover, the vector  $\tilde{u}_{\ell+1} \in \mathbb{R}^n$  is orthogonal to  $\mathbf{K}_\ell(\mathbf{A}^T, e)$  and  $\mathbf{H}_\ell \in \mathbb{R}^{\ell \times \ell}$  is an upper Hessenberg matrix. Then

$$\mathbf{U}_\ell^T \mathbf{A}^T \mathbf{U}_\ell = \mathbf{H}_\ell.$$

We will use the approximations

$$\mathbf{A}^T \approx \mathbf{U}_\ell \mathbf{H}_\ell \mathbf{U}_\ell^T \tag{15}$$

and

$$f(\mathbf{A}^T) \approx \mathbf{U}_\ell f(\mathbf{H}_\ell) \mathbf{U}_\ell^T. \tag{16}$$

The approximation (16) is quite accurate when  $f(\mathbf{A}^T)$  can be approximated well by a matrix of low rank. This is the case for many real-life undirected networks when  $f(t) = \exp(t)$ ; see [14].

It follows from (15) that

$$\mathbf{A}^T + h e e^T \approx \mathbf{U}_\ell \mathbf{H}_\ell \mathbf{U}_\ell^T + h e e^T = \mathbf{U}_\ell (\mathbf{H}_\ell + h n e_1 e_1^T) \mathbf{U}_\ell^T$$

and from (16) that

$$f(\mathbf{A}^T + h e e^T) \approx \mathbf{U}_\ell f(\mathbf{H}_\ell + h n e_1 e_1^T) \mathbf{U}_\ell^T,$$

where  $h$  is a scalar of small magnitude and  $e_1 = [1, 0, \dots, 0]^T \in \mathbb{R}^m$ . Hence,

$$\frac{f(\mathbf{A}^T + h e e^T) - f(\mathbf{A}^T - h e e^T)}{2h} \approx \mathbf{U}_\ell \frac{f(\mathbf{H}_\ell + h n e_1 e_1^T) - f(\mathbf{H}_\ell - h n e_1 e_1^T)}{2h} \mathbf{U}_\ell^T. \tag{17}$$

We will use the expression on the right-hand side as an approximation of  $L_f(\mathbf{A}^T, ee^T)$  in computed examples with  $h$  the same as in Example 2.1. Note that the evaluation of this expression only requires the computation of  $\ell$  matrix-vector products with the matrix  $\mathbf{A}^T$ . For large-scale problems for which the evaluation of matrix-vector products is the dominant computational work, under the assumption

that both methods require the same number of iterations, the use of the right-hand side of (17) halves the computational burden when compared with the evaluation of (13).

We turn to the situation when the matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is symmetric and first review the computations described by Kandolf et al. [16] of the analogue of the expression (13) when the direction is  $ee^T$ . Then the calculation of the Arnoldi decompositions (12) can be replaced by application of  $\ell$  steps of the symmetric Lanczos process to  $\mathbf{A}$  with initial vector  $e$ . Generically, we obtain

$$\mathbf{A}\mathbf{U}_\ell = \mathbf{U}_\ell\mathbf{T}_\ell + \tilde{u}_{\ell+1}e_\ell^T,$$

where the matrix  $\mathbf{T}_\ell \in \mathbb{R}^{\ell \times \ell}$  is symmetric and tridiagonal, the matrix  $\mathbf{U}_\ell \in \mathbb{R}^{n \times \ell}$  has orthonormal columns with initial column  $e/\|e\|_2$ , and the vector  $\tilde{u}_{\ell+1} \in \mathbb{R}^n$  is orthogonal to the range of  $\mathbf{U}_\ell$ ; see, e.g., Saad [25] for details on the symmetric Lanczos process.

The analogue of the expression (13) is given by

$$L_{f,\text{Lanczos}}(\mathbf{A}, ee^T) = \mathbf{U}_\ell\mathbf{X}_\ell\mathbf{U}_\ell^T, \tag{18}$$

where  $\mathbf{X}_\ell$  is the upper right  $\ell \times \ell$  submatrix of the  $2\ell \times 2\ell$  matrix

$$f\left(\begin{bmatrix} \mathbf{T}_\ell & ne_1e_1^T \\ \mathbf{0} & \mathbf{T}_\ell \end{bmatrix}\right) = \begin{bmatrix} f(\mathbf{T}_\ell) & \mathbf{X}_\ell \\ \mathbf{0} & f(\mathbf{T}_\ell) \end{bmatrix}; \tag{19}$$

see [16] for further details.

It remains to discuss how to determine approximations of the elements of  $L_f(\mathbf{A}^T, ee^T)$  of largest and smallest magnitude by using the right-hand sides of (13) or (17). We first consider the former. To determine an approximation of an entry of largest magnitude of  $L_f(\mathbf{A}^T, ee^T)$ , we first locate an entry  $x_{ij}$  of the matrix  $\mathbf{X}_\ell$  of largest magnitude and then determine entries of largest magnitude of columns  $i$  and  $j$  of the matrices  $\mathbf{V}_\ell$  and  $\mathbf{W}_\ell$ , respectively. The product of these entries furnishes an approximation of an entry of  $L_f(\mathbf{A}^T, ee^T)$  of largest magnitude. We proceed analogously to determine an approximation of an entry of  $L_f(\mathbf{A}^T, ee^T)$  of smallest magnitude. Other entries of closest to largest or smallest magnitudes can be computed similarly.

We turn to the use of the right-hand side of (17). To determine an approximation of an entry of largest magnitude of  $L_f(\mathbf{A}^T, ee^T)$ , we first locate an entry of the matrix  $\frac{f(\mathbf{H}_\ell+hne_1e_1^T)-f(\mathbf{H}_\ell-hne_1e_1^T)}{2h}$  of largest magnitude. Assume it is entry  $\{i, j\}$ . Then determine entries of largest magnitude of columns  $i$  and  $j$  of the matrix  $\mathbf{U}_\ell$ . The product of these entries yields an approximation of an entry of largest magnitude of  $L_f(\mathbf{A}^T, ee^T)$ .

### 3 Network modifications based on Perron root sensitivity

The methods of this section require right and left Perron vectors of the adjacency matrix  $\mathbf{A}$ . When the matrix  $\mathbf{A}$  is of small to moderate size, these vectors can be determined with the MATLAB function eig, which computes all eigenvalues and eigenvectors of

**A.** For large networks, we can compute the Perron vectors with the MATLAB function `eigs` or with the two-sided Arnoldi method. The latter method was introduced by Ruhe [23] and improved by Zwaan and Hochstenbach [31].

Our interest in the method of this section stems from the fact that it is easy to implement because the required computations are quite straightforward. However, the method does not identify edge weights whose modification yields a relatively large change in the total communicability (2). Instead, it identifies edge weights whose modification gives a relatively large change in the Perron root of the adjacency matrix. Computed examples in Section 4 indicate that modifications of edge weights identified by this method also results in relatively large changes in the total communicability.

### 3.1 Perron communicability for small to medium-sized networks

Let the adjacency matrix  $\mathbf{A}$  for the graph  $\mathcal{G}$  be irreducible and let  $\rho$  be its Perron root. Then there are unique right and left eigenvectors  $x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$  and  $y = [y_1, y_2, \dots, y_n]^T \in \mathbb{R}^n$ , respectively, of unit Euclidean norm with positive entries associated with  $\rho$ , i.e.,

$$\mathbf{A}x = \rho x, \quad y^T \mathbf{A} = \rho y^T.$$

They are referred to as Perron vectors. Let  $\mathbf{F} \in \mathbb{R}^{n \times n}$  be a nonnegative matrix of unit spectral norm,  $\|\mathbf{F}\|_2 = 1$ . Introduce the small positive parameter  $\varepsilon$  and denote the Perron root of  $\mathbf{A} + \varepsilon \mathbf{F}$  by  $\rho + \delta\rho$ . Then

$$\delta\rho = \varepsilon \frac{y^T \mathbf{F} x}{y^T x} + \mathcal{O}(\varepsilon^2)$$

and

$$\frac{y^T \mathbf{F} x}{y^T x} \leq \frac{\|y\|_2 \|\mathbf{F}\|_2 \|x\|_2}{y^T x} = \frac{1}{\cos \theta}, \tag{20}$$

where  $\theta$  is the angle between  $x$  and  $y$ . The quantity  $1/\cos \theta$  is referred to as the *condition number* of  $\rho$  and denoted by  $\kappa(\rho)$ ; see [29, Section 2]. Note that when  $\mathbf{A}$  is symmetric, we have  $x = y$ , hence  $\theta = 0$ . Equality in (20) is attained when  $\mathbf{F}$  is the *Wilkinson perturbation*  $\mathbf{W} = yx^T$  associated with  $\rho$ ; see [18, 29] for details.

The total communicability (2) of the graph  $\mathcal{G}$  can be approximated by the *Perron communicability* of  $\mathcal{G}$  [4]:

$$P_{\mathcal{G}}(w_{11}, w_{21}, \dots, w_{nn}) = \exp(\rho) e^T y x^T e = \exp(\rho) e^T \mathbf{W} e \tag{21}$$

with

$$T_{\mathcal{G}}(w_{11}, w_{21}, \dots, w_{nn}) \approx \kappa(\rho) P_{\mathcal{G}}(w_{11}, w_{21}, \dots, w_{nn}).$$

Typically,  $\exp(\rho)$  is a fairly accurate indicator of the Perron communicability and, consequently, of the total communicability. In fact, one has [4]:

$$\exp(\rho) \leq P_{\mathcal{G}}(w_{11}, w_{21}, \dots, w_{nn}) \leq n \exp(\rho). \tag{22}$$

Perturb the entry  $w_{ij}$  with  $i \neq j$  of  $\mathbf{A}$  by  $\varepsilon \neq 0$  and let

$$\mathbf{F} = e_i e_j^T \in \mathcal{A} \tag{23}$$

for some index pair  $\{i, j\}$ . The perturbation  $\delta\rho$  of  $\rho$  due to the perturbation  $\varepsilon\mathbf{F}$  of  $\mathbf{A}$  is

$$\delta\rho = \varepsilon \frac{y_i x_j}{y^T x} + \mathcal{O}(\varepsilon^2). \tag{24}$$

### 3.1.1 Network simplification by edge removal

To reduce the complexity of a network by removing edges without changing the Perron communicability significantly, we choose the matrix (23) so that  $\rho$  (and hence  $\exp(\rho)$ ) changes as little as possible and, therefore, choose the indices  $i$  and  $j$  so that

$$w_{ij} y_i x_j = \min_{\substack{1 \leq h, k \leq n \\ w_{hk} > 0}} w_{hk} y_h x_k,$$

and use  $\mathbf{A} - \varepsilon\mathbf{F}$  with  $\varepsilon = w_{ij}$  and  $\mathbf{F}$  given by (23).

If the graph is undirected, then we choose the matrix

$$\mathbf{F} = \frac{e_i e_j^T + e_j e_i^T}{2} \in \mathcal{A},$$

with the indices  $i$  and  $j$  determined as above, and use  $\mathbf{A} - \varepsilon\mathbf{F}$  with  $\varepsilon = 2 w_{ij} = 2 w_{ji}$ .

Introduce the vector  $\mathcal{E}_\rho \in \mathbb{R}^m$ , whose  $k$ th entry is the *Perron edge importance* of the edge  $e_k = e(v_i \rightarrow v_j)$ , defined as the product of the edge-weight  $w_{ij}$  and the corresponding entry  $y_i x_j$  of  $\mathbf{W}$ . Observe that  $\|\mathbf{W}\|_F = \|\mathbf{W}\|_2 = 1$ . The procedure to construct the Perron edge importance vector  $\mathcal{E}_\rho$  consists of two steps:

**Procedure 5:**

1. Multiply the adjacency matrix  $\mathbf{A}$  element by element by  $\mathbf{W}|_{\mathcal{A}}$ .
2. Put column by column the  $m$  nonvanishing entries of the matrix so obtained into the vector  $\mathcal{E}_\rho$ .

If the graph is undirected, then the *Perron edge importance* of edge  $e_k = e(v_i \leftrightarrow v_j)$  is defined as twice the product of the edge-weight  $w_{ij}$  and the corresponding entry  $y_i x_j$  of  $\mathbf{W}$ , so that the procedure to construct the Perron edge importance vector  $\mathcal{E}_\rho$  becomes:

**Procedure 6:**

1. Multiply  $\mathbf{A}|_{\mathcal{L}}$  element by element by  $\mathbf{W}|_{\mathcal{L}}$ .
2. Multiply by 2 the  $m$  nonvanishing entries of the matrix so obtained and put them column by column into the vector  $\mathcal{E}_\rho$ .

In computations, we order the entries of  $\mathcal{E}_\rho$  from smallest to largest, and set the entries  $w_{ij}$  (or the pair of entries  $w_{ij} = w_{ji}$  if the graph is undirected) associated with the first few of the ordered edge importances to zero. As mentioned before some post-processing may be necessary if the reduced graph is required to be connected.

### 3.1.2 Network modification by edge-weight tuning

We describe how to increase the total communicability and use the notation of Subsection 3.1.1. The discussion follows [21]. We would like to choose a perturbation  $\varepsilon \mathbf{F}$  of  $\mathbf{A}$ , where  $\varepsilon > 0$  and  $\mathbf{F}$  is of the form (23), so that the Perron root  $\rho$  increases as much as possible. This suggests that we choose the indices  $i$  and  $j$  in (23) so that

$$w_{ij}y_i x_j = \max_{\substack{1 \leq h, k \leq n, \\ w_{hk} > 0}} w_{hk}y_h x_k.$$

Thus, we choose the weight associated with the largest entry of the vector  $\mathcal{E}_\rho$  that yields the Perron edge importance of each edge.

We turn to the reduction of the total communicability. Define the matrix  $\mathbf{F}$  as above and consider the perturbed matrix  $\mathbf{A} - \varepsilon \mathbf{F}$ . The parameter  $\varepsilon > 0$  should be chosen small enough so that this matrix has nonnegative entries only. Moreover, if removing an edge  $e_k = e(v_i \rightarrow v_j)$  associated with one of the first few of the ordered entries of  $\mathcal{E}_\rho$  results in a disconnected graph and this is undesirable, then we choose  $\varepsilon > 0$  so that  $0 < \varepsilon < w_{ij}$ . Analogously, if removing the edges  $e(v_i \leftrightarrow v_j)$  of an undirected graph  $\mathcal{G}$  makes the graph disconnected and this is undesirable, then we choose  $\varepsilon > 0$  so that  $\varepsilon < 2w_{ij} = 2w_{ji}$ .

### 3.1.3 Network modification by inclusion of new edges

Let  $\mathbf{F} \in \mathcal{A}$  be a nonnegative matrix with  $\|\mathbf{F}\|_F = 1$ , and let  $\varepsilon > 0$  be a small constant. Then

$$\frac{y^T \mathbf{F} x}{y^T x} \leq \frac{\|y\|_2 \|y x^T|_{\mathcal{A}}\|_F \|x\|_2}{y^T x} = \frac{\|\mathbf{W}|_{\mathcal{A}}\|_F}{\cos \theta},$$

with equality for the  $\mathcal{A}$ -structured analogue of the Wilkinson perturbation,

$$\mathbf{F} = \frac{\mathbf{W}|_{\mathcal{A}}}{\|\mathbf{W}|_{\mathcal{A}}\|_F}.$$

This is the maximal perturbation for the Perron root  $\rho$  induced by a unit norm matrix  $\mathbf{F} \in \mathcal{A}$ ; see [10, 20]. The quantity

$$\frac{\|\mathbf{W}|_{\mathcal{A}}\|_F}{\cos \theta} = \kappa(\rho) \|\mathbf{W}|_{\mathcal{A}}\|_F$$

is referred to as the  $\mathcal{A}$ -structured condition number of  $\rho$  and denoted by  $\kappa_{\mathcal{A}}(\rho)$ . Thus,  $\kappa_{\mathcal{A}}(\rho) \leq \kappa(\rho)$ .

To increase the Perron communicability, we would like to modify the edges of the graph  $\mathcal{G}$  so that the Perron root is increased as much as possible; cf. (22). In case the  $m$  edges of  $\mathcal{G}$  are such that

$$\|\mathbf{W}|_{\mathcal{A}}\|_F \approx \|\mathbf{W}\|_F = 1, \tag{25}$$



i.e., when  $\kappa_{\mathcal{A}}(\rho) \approx \kappa(\rho)$ , increasing positive entries of  $\mathbf{A}$  should be a successful strategy to increase the Perron communicability. In fact, the matrix  $\mathbf{S} = [s_{ij}]_{i,j=1}^n \in \mathcal{A}$ , with entries  $s_{ij} = \frac{y_i x_j}{y^T x}$ , if  $w_{ij} > 0$  and  $s_{ij} = 0$  otherwise, referred to as the *structured Perron sensitivity matrix*, is such that

$$\mathbf{S} = \kappa(\rho)\mathbf{W}|_{\mathcal{A}} = \kappa(\rho)\|\mathbf{W}|_{\mathcal{A}}\|_F \frac{\mathbf{W}|_{\mathcal{A}}}{\|\mathbf{W}|_{\mathcal{A}}\|_F},$$

so that  $\|\mathbf{S}\|_F = \kappa_{\mathcal{A}}(\rho) \approx \kappa(\rho)$ . If  $\mathbf{F}$  is of the form (23), the perturbation (24) of  $\rho$  induced by  $\varepsilon\mathbf{F}$  can be written as  $\delta\rho = \varepsilon s_{ij} + \mathcal{O}(\varepsilon^2)$ .

Conversely, if  $\kappa_{\mathcal{A}}(\rho) \ll \kappa(\rho)$ , then the addition of a suitable edge with weight  $w_{ij} > 0$  (or a suitable pair of edges with weights  $w_{ij} = w_{ji} > 0$ , if the graph is undirected) that increases the ratio  $\kappa_{\mathcal{A}}(\rho)/\kappa(\rho)$  may be an appropriate strategy to increase the Perron communicability. Recall that  $\widehat{\mathcal{A}}$  denotes the cone of the nonnegative matrices in  $\mathbb{R}^{n \times n}$  whose sparsity structure is given by the zero entries of  $\mathbf{A}$  except for the diagonal entries. Perturb the entry  $w_{ij}$  with  $i \neq j$  of  $\mathbf{A}$  and assume that

$$\mathbf{F} = e_i e_j^T \in \widehat{\mathcal{A}}$$

for some index pair  $\{i, j\}$ . The entries of the Wilkinson perturbation  $\mathbf{W}$  reveal which edges should be added to the network to increase the communicability, namely edges whose associated entries of the matrix  $\mathbf{W}$  are large. The procedure for the construction of the vector  $\widehat{\mathcal{E}}_{\rho} \in \mathbb{R}^{n^2-n-m}$  that gives the *Perron virtual importance* of the virtual edges is the following:

**Procedure 7:**

1. Construct the matrix  $\mathbf{W}|_{\widehat{\mathcal{A}}}$ .
2. Put column by column the entries of the matrix so obtained that belong to the sparsity structure associated with  $\widehat{\mathcal{A}}$  into the vector  $\widehat{\mathcal{E}}_{\rho}$ .

If the graph is undirected, then the Perron virtual importance of the nonexistent edge  $e(v_i \leftrightarrow v_j) \notin \mathcal{E}$  is defined to be twice the corresponding entry in  $\mathbf{W}$ . The procedure for the construction of the Perron virtual edge importance vector  $\widehat{\mathcal{E}}_{\rho} \in \mathbb{R}^{(n^2-n-2m)/2}$  is given by:

**Procedure 8:**

1. Construct the matrix  $\mathbf{W}|_{\widehat{\mathcal{L}}}$ .
2. Multiply by 2 the entries of the matrix so obtained that belong to the sparsity structure associated with  $\widehat{\mathcal{L}}$  and put them column by column into the vector  $\widehat{\mathcal{E}}_{\rho}$ .

**3.2 Network modification criteria for large-scale networks**

Introduce the *structured Perron communicability* of  $\mathcal{G}$ :

$$P_{\mathcal{G}}^{\mathcal{A}}(w_{11}, w_{21}, \dots, w_{nm}) = \exp(\rho)e^T \mathbf{W}|_{\mathcal{A}} e.$$

One has, entry-wise,  $\mathbf{W}|_{\mathcal{A}} \leq \mathbf{W}$ , so that the structured Perron communicability  $P_{\mathcal{G}}^{\mathcal{A}}(w_{11}, w_{21}, \dots, w_{nn})$  is a lower bound for the Perron communicability (21). When  $\kappa_{\mathcal{A}}(\rho) \approx \kappa(\rho)$ , i.e., when (25) holds, the two measures are very close.

Additionally, if  $\mathcal{G}$  is undirected, then one has

$$P_{\mathcal{G}}^{\mathcal{A}}(w_{11}, w_{21}, \dots, w_{nn}) = 2 \exp(\rho) e^T \mathbf{W}|_{\mathcal{L}} e,$$

with  $\mathcal{L}$  the cone of all nonnegative matrices in  $\mathbb{R}^{n \times n}$  with the same sparsity structure as the strictly lower triangular portion of  $\mathbf{A}$ .

If our aim is to perturb or set to zero suitable positive entries of  $\mathbf{A} \in \mathcal{A}$ , then the Wilkinson perturbation  $\mathbf{W}$  does not have to be constructed, since one only needs the entries of  $\mathbf{W}|_{\mathcal{A}} \in \mathcal{A}$ . The Perron edge importance vector  $\mathcal{E}_{\rho} \in \mathbb{R}^m$ , associated with the  $m$  edges of  $\mathcal{G}$ , can be evaluated as discussed in Subsection 3.1.1.

### 4 Computed examples

The numerical tests reported in this section have been carried out using MATLAB R2023a on a 3.2 GHz Intel Core i7 6 core iMac.

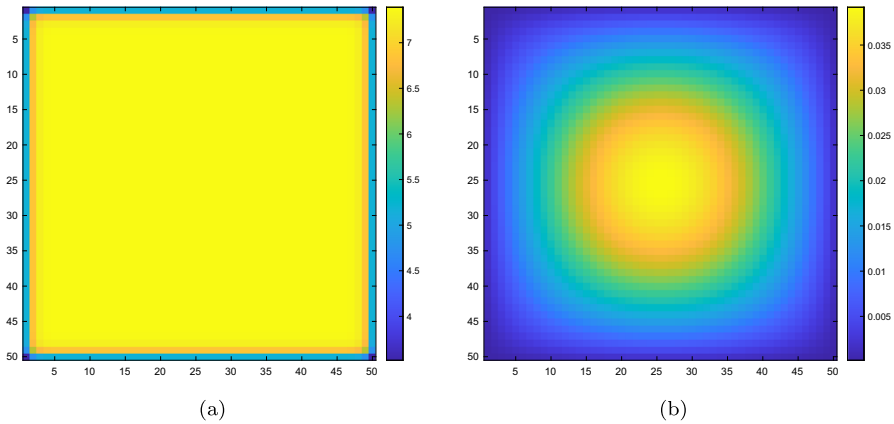
#### 4.1 A synthetic example

The following example explains why we might be interested in estimating the total communicability without evaluating the exponential of the adjacency matrix associated with the given network.

**Example 1** Consider the undirected and connected graph where each vertex represents a person in a line. Each person can communicate only with the following and the preceding persons. The adjacency matrix associated with such a network is the symmetric tridiagonal Toeplitz matrix

$$\mathbf{A} = \begin{bmatrix} & \sigma & & & & \\ \sigma & & \sigma & & & \\ & \sigma & & \cdot & & \\ & & \cdot & & \cdot & \\ & & & \cdot & & \sigma \\ & & & & \sigma & \end{bmatrix} \in \mathbb{R}^{n \times n}, \tag{26}$$

with  $\sigma > 0$ . It is intuitive that the closer the vertices are to the center of the graph, the better connected they are and, therefore, the more important. Also, it is immediate to observe that the best strategy to improve communication in this network is to make the two vertices at the ends adjacent by adding the new undirected edge  $e(v_1 \leftrightarrow v_n)$  with weight  $\sigma$ , which makes the adjacency matrix a circulant. This scenario is consistent with the structure of the Wilkinson matrix  $\mathbf{W}$  associated with the Perron root of  $\mathbf{A}$ , which we show in Fig. 1(b). This matrix is independent of  $\sigma$ ; see [20, Proposition 1].



**Fig. 1** Example 1. Structure of  $L_{\text{exp}_0}(\mathbf{A}, ee^T)$  (left picture (a)) and of  $\mathbf{W}$  (right picture (b)) for the matrix  $\mathbf{A} \in \mathbb{R}^{50 \times 50}$  in (26) with  $\sigma = 1$

Therefore the network modifications based on the Perron root sensitivity described in Section 3 are in agreement with the previous observations and considerations.

Instead, due to round-off errors introduced in the representation and evaluation of the matrix exponential, we have that the the structure of the matrix  $L_{\text{exp}_0}(\mathbf{A}^T, ee^T)$  (see Fig. 1(a)) does not completely reflect the scenario described above. In fact, the techniques based on the gradient (3) described in Section 2 do not identify the same edge as the Wilkinson perturbation. For instance, let  $n$  be even and  $\sigma = 1$ . Then we obtain for  $n \geq 36$  that the computed gradient does not identify the undirected edge  $e(v_{n/2} \leftrightarrow v_{n/2+1})$  as the most important one.

### 4.2 Medium-sized networks

**Example 2** Consider the adjacency matrix  $\mathbf{A} \in \mathbb{R}^{500 \times 500}$  for the network *Air500* in [9]. This data set describes flight connections for the top 500 airports worldwide based on total passenger volume. The flight connections between airports are for the year from 1 July 2007 to 30 June 2008, and the network is represented by a directed unweighted connected graph  $\mathcal{G}$  with  $n = 500$  vertices and  $m = 24009$  directed edges. Vertices of the network are airports and edges represent direct flight routes between two airports.

The total communicability in (2) is  $T_{\mathcal{G}} = 1.9164 \cdot 10^{38}$ . The gradient  $\nabla T_{\mathcal{G}}$  in (3) has been computed by evaluating the matrix  $L_f(\mathbf{A}^T, ee^T)$  in (8) using (9). The total transmission is  $\|\nabla T_{\mathcal{G}}\|_2 = 1.9205 \cdot 10^{38}$ . Also, the gradient  $\nabla T_{\mathcal{G}}$  has been approximated by evaluating  $L_f(\mathbf{A}^T, ee^T)$  using (10) with  $h = \frac{2}{n} \cdot 10^{-4} = 4 \cdot 10^{-7}$ , obtaining  $\widetilde{\nabla T}_{\mathcal{G}}$ . The resulting total transmission is  $\|\widetilde{\nabla T}_{\mathcal{G}}\|_2 = 1.9205 \cdot 10^{38}$  with

$$\frac{\|\nabla T_{\mathcal{G}} - \widetilde{\nabla T}_{\mathcal{G}}\|_2}{\|\nabla T_{\mathcal{G}}\|_2} = 1.9688 \cdot 10^{-9}.$$

For both the edge importance vector  $\mathcal{E}_{L_f}$  and the virtual edge importance vector  $\widehat{\mathcal{E}}_{L_f}$ , the same results are obtained regardless of whether  $\nabla T_G$  or  $\widehat{\nabla} T_G$  is used; see below. We remark that evaluating  $L_f(\mathbf{A}^T, ee^T)$  by (9) required the average time  $t_1 \approx 3.4 \cdot 10^{-1}$  in  $10^4$  tests, while evaluating  $L_f(\mathbf{A}^T, ee^T)$  using (10) required the average time  $t_2 \approx 1.5 \cdot 10^{-1}$  in the same tests, with a relative average time saving

$$\frac{t_1 - t_2}{t_1} \approx 5.6 \cdot 10^{-1}.$$

The Perron communicability in (21) is  $P_G = 1.9132 \cdot 10^{38}$ . The Perron root and left and right Perron vectors were evaluated with the MATLAB function eig.

In Table 1 the 10 smallest entries of both the edge importance vector  $\mathcal{E}_{L_f}$  and the Perron edge importance vector  $\mathcal{E}_\rho$  are shown along with the corresponding edges. This is useful for determining which edges to remove in order to reduce the complexity of the *Air500* network (cf. Subsections 2.1.1 and 3.1.1). One can observe that the elimination of the connection from Santos Dumont Airport, Rio de Janeiro, Brazil (SDU) to Congonhas Airport, S. Paulo, Brazil (CGH), at the third position in the ranking given by  $\mathcal{E}_\rho$ , would disconnect the network. We therefore only remove the two edges in bold face in Table 1, i.e., the two most irrelevant edges - according to edge importance determination based on both gradient and Perron root sensitivity. These edges correspond to the flight connection between San Antonio International Airport, San Antonio, Texas (TSA) and Penghu Airport, Taiwan (MZG). This results in the network  $\mathcal{G}_1$ , for which we have:

$$T_{\mathcal{G}_1} = 1.9164 \cdot 10^{38}; \quad \frac{T_G - T_{\mathcal{G}_1}}{T_G} = 1.8014 \cdot 10^{-7}; \quad P_{\mathcal{G}_1} = 1.9132 \cdot 10^{38}; \quad \frac{P_G - P_{\mathcal{G}_1}}{P_G} = 1.8014 \cdot 10^{-7}.$$

**Table 1** Example 2

$\mathcal{E}_{L_f}$	$e(v_i \rightarrow v_j)$	$\mathcal{E}_\rho$	$e(v_i \rightarrow v_j)$
$9.0980 \cdot 10^{-8}$	<b>TSA → MZG</b>	$3.1829 \cdot 10^{-9}$	<b>TSA → MZG</b>
$9.0980 \cdot 10^{-8}$	<b>MZG → TSA</b>	$3.1829 \cdot 10^{-9}$	<b>MZG → TSA</b>
$5.1931 \cdot 10^{-7}$	UKB → ISG	$1.5705 \cdot 10^{-8}$	SDU → CGH
$5.8446 \cdot 10^{-7}$	ISG → UKB	$1.5705 \cdot 10^{-8}$	CGH → SDU
$8.8771 \cdot 10^{-7}$	SDU → CGH	$6.9399 \cdot 10^{-8}$	UKB → ISG
$9.2224 \cdot 10^{-7}$	CGH → SDU	$8.5792 \cdot 10^{-8}$	ISG → UKB
$9.6419 \cdot 10^{-7}$	GMP → HND	$1.2677 \cdot 10^{-7}$	HND → GMP
$9.7169 \cdot 10^{-7}$	HND → GMP	$1.2979 \cdot 10^{-7}$	GMP → HND
$1.0369 \cdot 10^{-6}$	DUR → PLZ	$1.7020 \cdot 10^{-7}$	UKB → HND
$1.0376 \cdot 10^{-6}$	PLZ → DUR	$1.9511 \cdot 10^{-7}$	HND → UKB

The first 10 flight connections to remove in order to reduce the complexity of *Air500* without changing the network communicability significantly, according to the determination of the edge importance based on gradient and on Perron root sensitivity, respectively

Table 2 shows the 10 largest entries of both the edge importance vector  $\mathcal{E}_{L_f}$  and the Perron edge importance vector  $\mathcal{E}_\rho$  along with the corresponding edges. In order to obtain a relatively large reduction in the total communicability, we set to zero the weights associated with the two largest entries of both  $\mathcal{E}_{L_f}$  and  $\mathcal{E}_\rho$ . This means we remove the edges that represent the air route between John F. Kennedy International Airport, New York City (JFK) and Atlanta Hartsfield-Jackson Airport, Georgia (ATL). This results in the network  $\mathcal{G}_2$ , for which, as it is apparent, the reduction in the total communicability is much larger than in  $\mathcal{G}_1$ :

$$T_{\mathcal{G}_2} = 1.8423 \cdot 10^{38}; \quad \frac{T_{\mathcal{G}} - T_{\mathcal{G}_2}}{T_{\mathcal{G}}} = 3.8680 \cdot 10^{-2}; \quad P_{\mathcal{G}_2} = 1.8392 \cdot 10^{38}; \quad \frac{P_{\mathcal{G}} - P_{\mathcal{G}_2}}{P_{\mathcal{G}}} = 3.8689 \cdot 10^{-2}.$$

Following the discussion in Subsections 2.1.2 and 3.1.2, in order to obtain a relatively large increase in the total communicability, we increase by 1 the edge-weights associated with the two largest entries of both  $\mathcal{E}_{L_f}$  and  $\mathcal{E}_\rho$  (i.e., the air route between John F. Kennedy International Airport, New York City (JFK), and Atlanta Hartsfield-Jackson Airport, Georgia (ATL)). For the so obtained network  $\mathcal{G}_3$ , one has

$$T_{\mathcal{G}_3} = 1.9943 \cdot 10^{38}; \quad \frac{T_{\mathcal{G}_3} - T_{\mathcal{G}}}{T_{\mathcal{G}}} = 4.0658 \cdot 10^{-2}; \quad P_{\mathcal{G}_3} = 1.9910 \cdot 10^{38}; \quad \frac{P_{\mathcal{G}_3} - P_{\mathcal{G}}}{P_{\mathcal{G}}} = 4.0661 \cdot 10^{-2}.$$

Finally, following the discussion in Subsections 2.1.3 and 3.1.3, we display in Table 3 the 10 largest entries of the total virtual edge importance vector  $\widehat{\mathcal{E}}_{L_f}$  and the 10 largest entries of the Perron virtual edge importance vector  $\widehat{\mathcal{E}}_\rho$ , along with the corresponding *nonexistent* edges. Notice that the edges associated with the two largest entries of both  $\widehat{\mathcal{E}}_{L_f}$  and  $\widehat{\mathcal{E}}_\rho$  cannot be considered because they model the missing air

**Table 2** Example 2

$\mathcal{E}_{L_f}$	$e(v_i \rightarrow v_j)$	$\mathcal{E}_\rho$	$e(v_i \rightarrow v_j)$
$1.9810 \cdot 10^{-2}$	<b>JFK → ATL</b>	$2.0038 \cdot 10^{-2}$	<b>JFK → ATL</b>
$1.9757 \cdot 10^{-2}$	<b>ORD → JFK</b>	$1.9987 \cdot 10^{-2}$	<b>ORD → JFK</b>
$1.9660 \cdot 10^{-2}$	JFK → ORD	$1.9882 \cdot 10^{-2}$	JFK → ORD
$1.9625 \cdot 10^{-2}$	ATL → JFK	$1.9861 \cdot 10^{-2}$	ATL → JFK
$1.9152 \cdot 10^{-2}$	JFK → LAX	$1.9369 \cdot 10^{-2}$	JFK → LAX
$1.9068 \cdot 10^{-2}$	EWR → JFK	$1.9280 \cdot 10^{-2}$	EWR → JFK
$1.8959 \cdot 10^{-2}$	JFK → EWR	$1.9239 \cdot 10^{-2}$	ORD → ATL
$1.8945 \cdot 10^{-2}$	ORD → ATL	$1.9165 \cdot 10^{-2}$	JFK → EWR
$1.8727 \cdot 10^{-2}$	LAX → JFK	$1.8970 \cdot 10^{-2}$	ATL → ORD
$1.8677 \cdot 10^{-2}$	ATL → ORD	$1.8936 \cdot 10^{-2}$	LAX → JFK

The first 10 flight connections to increase/decrease in order to increase/decrease the network communicability in *Air500* according to the determination of the edge importance based on gradient and on Perron root sensitivity, respectively

**Table 3** Example 2

$\widehat{\mathcal{E}}_{L_f}$	$e(v_i \rightarrow v_j)$	$\widehat{\mathcal{E}}_\rho$	$e(v_i \rightarrow v_j)$
$1.3153 \cdot 10^{-2}$	JFK $\rightarrow$ LGA	$1.3385 \cdot 10^{-2}$	JFK $\rightarrow$ LGA
$1.3102 \cdot 10^{-2}$	LGA $\rightarrow$ JFK	$1.3328 \cdot 10^{-2}$	LGA $\rightarrow$ JFK
$1.2402 \cdot 10^{-2}$	<b>LHR <math>\rightarrow</math> ATL</b>	$1.2383 \cdot 10^{-2}$	<b>MDW <math>\rightarrow</math> JFK</b>
$1.2276 \cdot 10^{-2}$	<b>AMS <math>\rightarrow</math> DFW</b>	$1.2278 \cdot 10^{-2}$	<b>LHR <math>\rightarrow</math> ATL</b>
$1.2252 \cdot 10^{-2}$	ATL $\rightarrow$ LHR	$1.2264 \cdot 10^{-2}$	JFK $\rightarrow$ MDW
$1.2160 \cdot 10^{-2}$	MDW $\rightarrow$ JFK	$1.2212 \cdot 10^{-2}$	AMS $\rightarrow$ DFW
$1.2039 \cdot 10^{-2}$	JFK $\rightarrow$ MDW	$1.2136 \cdot 10^{-2}$	ABQ $\rightarrow$ JFK
$1.1915 \cdot 10^{-2}$	ABQ $\rightarrow$ JFK	$1.2093 \cdot 10^{-2}$	ATL $\rightarrow$ LHR
$1.1581 \cdot 10^{-2}$	DFW $\rightarrow$ AMS	$1.1775 \cdot 10^{-2}$	ORD $\rightarrow$ MDW
$1.1525 \cdot 10^{-2}$	ORD $\rightarrow$ LGW	$1.1472 \cdot 10^{-2}$	DFW $\rightarrow$ AMS

The first 10 flight connections that should be added, in order to enhance the communicability in *Air500*, according to determination of the virtual edge importance based on gradient and on Perron root sensitivity, respectively

route between John F. Kennedy International Airport, New York City, (JFK), and La Guardia Airport, New York City, (LGA), and are too close to justify a flight route. The entries of the table suggest that there should be a shuttle service between these vertices and, indeed, such a shuttle service exists. We proceed to consider the third and fourth best nonexistent edges according to the edge importance based on the gradient, that is we consider the routes from Heathrow Airport, London, England, (LHR) to Atlanta Hartsfield-Jackson Airport, Georgia, (ATL) and from the Amsterdam Schiphol Airport, Netherlands, (AMS) to Dallas/Fort Worth International Airport, Texas, (DFW), and set their weights to 1. This way, we obtain the network  $\mathcal{G}_4$ , for which one has

$$T_{\mathcal{G}_4} = 1.9643 \cdot 10^{38}; \quad \frac{T_{\mathcal{G}_4} - T_{\mathcal{G}}}{T_{\mathcal{G}}} = 2.5020 \cdot 10^{-2}.$$

Conversely, setting to 1 the weight of the third and the fourth best nonexistent edges according to  $\widehat{\mathcal{E}}_\rho$ , i.e., the edges that represent flights from Midway International Airport, Chicago, Illinois, (MDW) to John F. Kennedy International Airport, New York City, (JFK) and from Heathrow Airport, London, England, (LHR) to Hartsfield-Jackson Airport, Atlanta, Georgia (ATL), we obtain the network  $\mathcal{G}_5$  with

$$P_{\mathcal{G}_5} = 1.9609 \cdot 10^{38}; \quad \frac{P_{\mathcal{G}_5} - P_{\mathcal{G}}}{P_{\mathcal{G}}} = 2.4915 \cdot 10^{-2}; \quad T_{\mathcal{G}_5} = 1.9641 \cdot 10^{38}; \quad \frac{T_{\mathcal{G}_5} - T_{\mathcal{G}}}{T_{\mathcal{G}}} = 2.4908 \cdot 10^{-2}.$$

In this example, edge addition is less effective than increasing the weights of existing edges. We observe that the matrix in  $\mathcal{A}$  that is closest to the Wilkinson perturbation  $\mathbf{W}$  associated to the Perron root  $\rho$  of  $\mathbf{A}$  with respect to the Frobenius norm, i.e.,  $\mathbf{W}|_{\mathcal{A}}$ , has Frobenius norm  $\|\mathbf{W}|_{\mathcal{A}}\|_F = 7.5920 \cdot 10^{-1}$ , meaning that the  $\mathcal{A}$ -structured condition number  $\kappa_{\mathcal{A}}(\rho)$  is approximately 76% of the condition number  $\kappa(\rho)$ .

**Example 3** Consider the undirected unweighted graph  $\mathcal{G}$  that represents the German highway system network *Autobahn*. The graph, which is available at [9], has  $n = 1168$  vertices representing German locations and  $m = 1243$  edges that represent highway segments connecting them. Therefore, the adjacency matrix  $\mathbf{A} \in \mathbb{R}^{1168 \times 1168}$  for this network has 2486 nonvanishing entries.

The total communicability in (2) is  $T_{\mathcal{G}} = 1.2563 \cdot 10^4$ . The gradient  $\nabla T_{\mathcal{G}}$  in (3) has been computed by evaluating the matrix  $L_f(\mathbf{A}^T, ee^T)$  in (8) using (9). The total transmission is  $\|\nabla T_{\mathcal{G}}\|_2 = 1.4464 \cdot 10^4$ . The gradient  $\nabla T_{\mathcal{G}}$  has been approximated by evaluating  $L_f(\widetilde{\mathbf{A}}^T, ee^T)$  using (10) with  $h = \frac{2}{n} \cdot 10^{-4} = 1.7123 \cdot 10^{-7}$ , obtaining  $\widetilde{\nabla T}_{\mathcal{G}}$ . The resulting total transmission is  $\|\widetilde{\nabla T}_{\mathcal{G}}\|_2 = 1.4464 \cdot 10^4$  with

$$\frac{\|\nabla T_{\mathcal{G}} - \widetilde{\nabla T}_{\mathcal{G}}\|_2}{\|\nabla T_{\mathcal{G}}\|_2} = 5.4537 \cdot 10^{-9}.$$

Notice that evaluating the matrix  $L_f(\mathbf{A}^T, ee^T)$  in  $10^4$  tests required the average time  $t_1 \approx 1.8 \cdot 10^0$  using (9) and the average time  $t_2 \approx 3.2 \cdot 10^{-1}$  using (10), with a relative average time saving

$$\frac{t_1 - t_2}{t_1} \approx 8.2 \cdot 10^{-1}.$$

The slight difference in the edge importance vectors computed using (9) or (10) gives rise to different orderings of the edges corresponding to the 10 smallest entries (which in fact differ about  $\mathcal{O}(10^{-12})$ ). This is displayed in Table 4. Removing the two edges in bold face in the second column of Table 4 results in the (disconnected) network  $\mathcal{G}_1$  (see Fig. 2(a)), while removing the two edges in bold face in the third column of Table 4 yields the disconnected network  $\widetilde{\mathcal{G}}_1$  (see Fig. 2(b)). Nevertheless,

**Table 4** Example 3

$\mathcal{E}_{L_f}$ [or $\widetilde{\mathcal{E}}_{L_f}$ ]	$e(v_i \leftrightarrow v_j)$ associated with $\mathcal{E}_{L_f}$	$e(v_i \leftrightarrow v_j)$ associated with $\widetilde{\mathcal{E}}_{L_f}$
$6.5794 \cdot 10^{-4}$	<b>Allershausen</b> $\leftrightarrow$ <b>Allersberg</b>	<b>Wüstenbrand</b> $\leftrightarrow$ <b>Wommen</b>
$6.5794 \cdot 10^{-4}$	<b>Bünde</b> $\leftrightarrow$ <b>Bissendorf</b>	<b>Thiendorf</b> $\leftrightarrow$ <b>Teupitz</b>
$6.5794 \cdot 10^{-4}$	Zarrentin $\leftrightarrow$ Witzhave	Zrbig $\leftrightarrow$ Wiedemar
$6.5794 \cdot 10^{-4}$	Wunsiedel $\leftrightarrow$ Wolnzach	Zarrentin $\leftrightarrow$ Witzhave
$6.5794 \cdot 10^{-4}$	Thiendorf $\leftrightarrow$ Teupitz	Aitrach $\leftrightarrow$ Aichstetten
$6.5794 \cdot 10^{-4}$	Wüstenbrand $\leftrightarrow$ Wommen	Wesuwe $\leftrightarrow$ Weener
$6.5794 \cdot 10^{-4}$	Alsfeld $\leftrightarrow$ Achern	Wunsiedel $\leftrightarrow$ Wolnzach
$6.5794 \cdot 10^{-4}$	Wesuwe $\leftrightarrow$ Weener	Zwingenberg $\leftrightarrow$ Zeppelinheim
$6.5794 \cdot 10^{-4}$	Zwingenberg $\leftrightarrow$ Zeppelinheim	Bünde $\leftrightarrow$ Bissendorf
$6.5794 \cdot 10^{-4}$	Zrbig $\leftrightarrow$ Wiedemar	Alsfeld $\leftrightarrow$ Achern

The first 10 highway segments that could be removed in order to reduce the complexity of *Autobahn* without changing the network communicability significantly, according to the determination of the edge importance based on the gradient. The edges in the second and third columns are determined by (9) and (10), respectively

we have the same result for both the graphs  $\mathcal{G}_1$  and  $\tilde{\mathcal{G}}_1$ :

$$T_{\mathcal{G}_1} = 1.2550 \cdot 10^4; \quad \frac{T_{\mathcal{G}} - T_{\mathcal{G}_1}}{T_{\mathcal{G}}} = 1.0171 \cdot 10^{-3}; \quad T_{\tilde{\mathcal{G}}_1} = 1.2550 \cdot 10^4; \quad \frac{T_{\mathcal{G}} - T_{\tilde{\mathcal{G}}_1}}{T_{\mathcal{G}}} = 1.0171 \cdot 10^{-3}.$$

Conversely, the largest entries of the edge importance vectors computed using (9) or (10) correspond to the same edges displayed in the first column of Table 5. Setting to zero the weights associated with the two highway segments Duisburg - Düsseldorf and München - Kirchheim (both in bold face in the first column of Table 5) results in the network  $\mathcal{G}_2$ , for which one has

$$T_{\mathcal{G}_2} = 1.2108 \cdot 10^4; \quad \frac{T_{\mathcal{G}} - T_{\mathcal{G}_2}}{T_{\mathcal{G}}} = 3.6214 \cdot 10^{-2},$$

while increasing by one the weights associated with the same edges results in the network  $\mathcal{G}_3$ , for which one has

$$T_{\mathcal{G}_3} = 1.3359 \cdot 10^4; \quad \frac{T_{\mathcal{G}_3} - T_{\mathcal{G}}}{T_{\mathcal{G}}} = 6.3330 \cdot 10^{-2}.$$

As for Perron communicability in *Autobahn*, one has  $P_{\mathcal{G}} = 2.2448 \cdot 10^3$ . Although the graph  $\mathcal{G}$  is irreducible, some entries of the Perron vector  $x$  are close to machine precision. In particular, the edges to remove in order to simplify the network, that are associated with the two smallest entries of the Perron edge importance vector, are the highway segments Wildsdruff - Wildeck and Wüstenbrand - Wommen, which connect four vertices associated with such quasi-zero components of  $x$  (see Fig. 2(c)). This results in the (disconnected) network  $\mathcal{G}_4$ , for which we have

$$P_{\mathcal{G}_4} = 2.2448 \cdot 10^3; \quad \frac{P_{\mathcal{G}} - P_{\mathcal{G}_4}}{P_{\mathcal{G}}} = -4.8619 \cdot 10^{-15}; \quad T_{\mathcal{G}_4} = 1.2547 \cdot 10^4; \quad \frac{T_{\mathcal{G}} - T_{\mathcal{G}_4}}{T_{\mathcal{G}}} = 1.2521 \cdot 10^{-3}.$$

Hence, this simplification is slightly less satisfactory than the ones described by  $\mathcal{G}_1$  and  $\tilde{\mathcal{G}}_1$ . Elimination of the edges associated with the two highway segments Duisburg - Düsseldorf and Essen - Duisburg (both in bold face in the second column of Table 5) gives the network  $\mathcal{G}_5$ , for which we have

$$P_{\mathcal{G}_5} = 1.2677 \cdot 10^3; \quad \frac{P_{\mathcal{G}} - P_{\mathcal{G}_5}}{P_{\mathcal{G}}} = 4.3527 \cdot 10^{-1}; \quad T_{\mathcal{G}_5} = 1.2155 \cdot 10^4; \quad \frac{T_{\mathcal{G}} - T_{\mathcal{G}_5}}{T_{\mathcal{G}}} = 3.2513 \cdot 10^{-2}.$$

Again the decrease is less than for  $\mathcal{G}_2$ . Conversely, increasing by one the weights associated with the same edges results in the network  $\mathcal{G}_6$ , for which one has

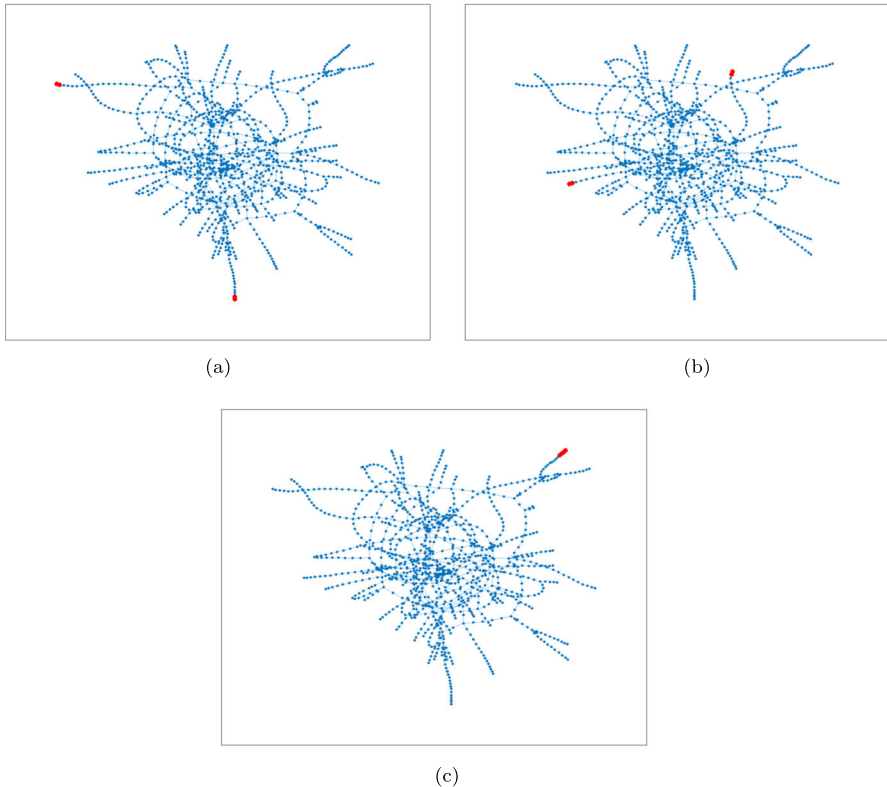
$$P_{\mathcal{G}_6} = 2.5248 \cdot 10^3; \quad \frac{P_{\mathcal{G}_6} - P_{\mathcal{G}}}{P_{\mathcal{G}}} = 1.2472 \cdot 10^{-2}; \quad T_{\mathcal{G}_6} = 1.3480 \cdot 10^4; \quad \frac{T_{\mathcal{G}_6} - T_{\mathcal{G}}}{T_{\mathcal{G}}} = 7.2982 \cdot 10^{-2}.$$



**Table 5** Example 3

$\mathcal{E}_{L_f}$	$e(v_i \leftrightarrow v_j)$	$\mathcal{E}_\rho$	$e(v_i \leftrightarrow v_j)$
$2.1662 \cdot 10^{-2}$	<b>Duisburg</b> $\leftrightarrow$ <b>Düsseldorf</b>	$3.2881 \cdot 10^{-1}$	<b>Duisburg</b> $\leftrightarrow$ <b>Düsseldorf</b>
$1.9101 \cdot 10^{-2}$	<b>München</b> $\leftrightarrow$ <b>Kirchheim</b>	$2.9148 \cdot 10^{-1}$	<b>Essen</b> $\leftrightarrow$ <b>Duisburg</b>
$1.8792 \cdot 10^{-2}$	Essen $\leftrightarrow$ Duisburg	$2.2093 \cdot 10^{-1}$	Duisburg $\leftrightarrow$ Dinslaken
$1.8106 \cdot 10^{-2}$	Duisburg $\leftrightarrow$ Dortmund	$2.1899 \cdot 10^{-1}$	Duisburg $\leftrightarrow$ Dortmund
$1.7092 \cdot 10^{-2}$	Krefeld $\leftrightarrow$ Duisburg	$1.8552 \cdot 10^{-1}$	Düsseldorf $\leftrightarrow$ Dinslaken
$1.6862 \cdot 10^{-2}$	Hamburg $\leftrightarrow$ Hagen	$1.8126 \cdot 10^{-1}$	Krefeld $\leftrightarrow$ Duisburg
$1.4823 \cdot 10^{-2}$	Duisburg $\leftrightarrow$ Dinslaken	$1.5044 \cdot 10^{-1}$	Gelsenkirchen $\leftrightarrow$ Essen
$1.4348 \cdot 10^{-2}$	Gelsenkirchen $\leftrightarrow$ Essen	$1.4553 \cdot 10^{-1}$	Flughafen $\leftrightarrow$ Duisburg
$1.4105 \cdot 10^{-2}$	Flughafen $\leftrightarrow$ Duisburg	$1.3617 \cdot 10^{-1}$	Elmpt $\leftrightarrow$ Düsseldorf
$1.3341 \cdot 10^{-2}$	Hagen $\leftrightarrow$ Groá	$1.2071 \cdot 10^{-1}$	Essen $\leftrightarrow$ Elmpt

The first 10 highway segments that should be widened/narrowed in order to increase/decrease the network communicability in *Autobahn* according to the edge importance vector  $\mathcal{E}_{L_f}$  (in the first column) and according to the Perron edge importance vector  $\mathcal{E}_\rho$  (in the second column), respectively



**Fig. 2** Example 3. The vertices (marked in red) that are connected by the edges to be removed in order to simplify the network according to  $\mathcal{E}_{L_f}$  for (a), according to  $\tilde{\mathcal{E}}_{L_f}$  for (b), and according to  $\mathcal{E}_\rho$  for (c)

The increase in this case is greater than for  $\mathcal{G}_3$ .

We do not address the issue of adding new highway segments, because the feasibility of building new highway segments depends on many issues that are not included in our model such as the length of the new segments and properties of the regions, e.g., the presence of mountains, valleys and lakes, that the new segments would traverse.

**Example 4** Consider the directed weighted graph  $\mathcal{G}$  that represents the network *C.elegans* available at [28], i.e., the metabolic network of the *Caenorhabditis elegans* worm. The network contains  $n = 306$  vertices that represent neurons and  $m = 2345$  edges. Two neurons are connected if at least one synapse exists between them and the associated edge-weight is the number of synapses. The network is disconnected.

The total communicability (2) is  $T_{\mathcal{G}} = 3.3401 \cdot 10^6$ . The gradient  $\nabla T_{\mathcal{G}}$  (3) has been computed by evaluating the matrix  $L_f(\mathbf{A}^T, uv^T)$  in (8) using (9). The total transmission is  $\|\nabla T_{\mathcal{G}}\|_2 = 7.9032 \cdot 10^6$ . The gradient  $\nabla T_{\mathcal{G}}$  has been approximated by evaluating  $L_f(\mathbf{A}^T, uv^T)$  using (10) with  $h = \frac{2}{n} \cdot 10^{-4} = 6.5359 \cdot 10^{-7}$  to obtain

$\widetilde{\nabla T}_{\mathcal{G}}$ . The resulting total transmission is  $\|\widetilde{\nabla T}_{\mathcal{G}}\|_2 = 7.9032 \cdot 10^6$ , having

$$\frac{\|\nabla T_{\mathcal{G}} - \widetilde{\nabla T}_{\mathcal{G}}\|_2}{\|\nabla T_{\mathcal{G}}\|_2} = 4.4914 \cdot 10^{-9}.$$

As for both the edge importance vector  $\mathcal{E}_{L_f}$  and the virtual edge importance vector  $\widehat{\mathcal{E}}_{L_f}$ , one obtains the same results, displayed in Table 6, regardless of whether  $\nabla T_{\mathcal{G}}$  or  $\widetilde{\nabla T}_{\mathcal{G}}$  is used. We remark that evaluating  $L_f(\mathbf{A}^T, ee^T)$  by (9) required the average time  $t_1 \approx 1.3 \cdot 10^{-1}$  in  $10^4$  tests, while evaluating  $L_f(\mathbf{A}^T, ee^T)$  using (10) required the average time  $t_2 \approx 1.7 \cdot 10^{-2}$  in the same tests. The relative average time saving is

$$\frac{t_1 - t_2}{t_1} \approx 8.6 \cdot 10^{-1}.$$

Removing the two edges in bold face in the second column of Table 6, associated with the smallest entries of the edge importance, results in the network  $\mathcal{G}_1$  for which

$$T_{\mathcal{G}_1} = 3.3401 \cdot 10^6; \quad \frac{T_{\mathcal{G}} - T_{\mathcal{G}_1}}{T_{\mathcal{G}}} = 5.9879 \cdot 10^{-7}.$$

Setting to zero the weights associated with the two edges in bold face in the fourth column of Table 6 results in the network  $\mathcal{G}_2$ , for which one has

$$T_{\mathcal{G}_2} = 2.9282 \cdot 10^6; \quad \frac{T_{\mathcal{G}} - T_{\mathcal{G}_2}}{T_{\mathcal{G}}} = 1.2332 \cdot 10^{-1},$$

while increasing by one the weights associated with the same edges results in the network  $\mathcal{G}_3$  with

$$T_{\mathcal{G}_3} = 3.8047 \cdot 10^6; \quad \frac{T_{\mathcal{G}_3} - T_{\mathcal{G}}}{T_{\mathcal{G}}} = 1.3909 \cdot 10^{-1}.$$

Finally, setting to one the (vanishing) entries of  $A$  associated with the two virtual edges in bold face in the sixth column of Table 6 yields the network  $\mathcal{G}_4$ , for which

$$T_{\mathcal{G}_4} = 7.3327 \cdot 10^6; \quad \frac{T_{\mathcal{G}_4} - T_{\mathcal{G}}}{T_{\mathcal{G}}} = 1.1954 \cdot 10^0.$$

Turning to network modifications based on Perron root sensitivity in *C.elegans*, one has  $P_{\mathcal{G}} = 9.1975 \cdot 10^5$ . The graph  $\mathcal{G}$  is reducible, some entries of the Perron vector  $x$  are vanishing. In particular, removing the edges in bold face in the second column of Table 7, which are associated with two vanishing entries of the Perron edge importance vector, results in the network  $\mathcal{G}_5$ , for which one has

$$P_{\mathcal{G}_5} = 9.1931 \cdot 10^5; \quad \frac{P_{\mathcal{G}} - P_{\mathcal{G}_5}}{P_{\mathcal{G}}} = 4.7732 \cdot 10^{-4}; \quad T_{\mathcal{G}_5} = 3.3381 \cdot 10^6; \quad \frac{T_{\mathcal{G}} - T_{\mathcal{G}_5}}{T_{\mathcal{G}}} = 6.0666 \cdot 10^{-4}.$$

**Table 6** Example 4

$\mathcal{E}_{L_f}$	$e(v_i \rightarrow v_j)$	$\mathcal{E}_{L_f}$	$e(v_i \rightarrow v_j)$	$\hat{\mathcal{E}}_{L_f}$	$e(v_i \rightarrow v_j)$
$1.2653 \cdot 10^{-7}$	$v_{53} \rightarrow v_{303}$	$2.8888 \cdot 10^{-2}$	$v_{71} \rightarrow v_{217}$	$1.1107 \cdot 10^{-1}$	$v_{305} \rightarrow v_{149}$
$1.2653 \cdot 10^{-7}$	$v_{151} \rightarrow v_{305}$	$2.6446 \cdot 10^{-2}$	$v_{72} \rightarrow v_{216}$	$1.0213 \cdot 10^{-1}$	$v_{305} \rightarrow v_{219}$
$1.2653 \cdot 10^{-7}$	$v_{191} \rightarrow v_{305}$	$2.5673 \cdot 10^{-2}$	$v_{73} \rightarrow v_{178}$	$9.6897 \cdot 10^{-2}$	$v_{305} \rightarrow v_{218}$
$1.2653 \cdot 10^{-7}$	$v_{243} \rightarrow v_{305}$	$2.2681 \cdot 10^{-2}$	$v_{72} \rightarrow v_{144}$	$8.9795 \cdot 10^{-2}$	$v_{305} \rightarrow v_{216}$
$1.2653 \cdot 10^{-7}$	$v_{259} \rightarrow v_{305}$	$2.1866 \cdot 10^{-2}$	$v_{76} \rightarrow v_{217}$	$8.9722 \cdot 10^{-2}$	$v_{305} \rightarrow v_{217}$
$1.2653 \cdot 10^{-7}$	$v_{267} \rightarrow v_{305}$	$2.1271 \cdot 10^{-2}$	$v_{78} \rightarrow v_{217}$	$8.9222 \cdot 10^{-2}$	$v_{305} \rightarrow v_{178}$
$1.2653 \cdot 10^{-7}$	$v_{291} \rightarrow v_{305}$	$2.0888 \cdot 10^{-2}$	$v_{75} \rightarrow v_{216}$	$8.5450 \cdot 10^{-2}$	$v_{305} \rightarrow v_{174}$
$1.2653 \cdot 10^{-7}$	$v_{292} \rightarrow v_{305}$	$2.0012 \cdot 10^{-2}$	$v_{77} \rightarrow v_{216}$	$8.2518 \cdot 10^{-2}$	$v_{305} \rightarrow v_{81}$
$1.2653 \cdot 10^{-7}$	$v_{293} \rightarrow v_{305}$	$1.9903 \cdot 10^{-2}$	$v_{71} \rightarrow v_{47}$	$7.9914 \cdot 10^{-2}$	$v_{305} \rightarrow v_{82}$
$1.2653 \cdot 10^{-7}$	$v_{294} \rightarrow v_{305}$	$1.9752 \cdot 10^{-2}$	$v_{71} \rightarrow v_{72}$	$7.7984 \cdot 10^{-2}$	$v_{305} \rightarrow v_{198}$

The smallest entries of the edge importance vector and the relevant junctions that could be removed without changing the network communicability in *C.elegans* significantly (displayed in the first two columns). The largest entries of the edge importance vector and the relevant junctions to increase/decrease in order to increase/decrease the network communicability (in the third and fourth columns). The largest entries of the virtual edge importance vector and the relevant junctions to add in order to increase the communicability in *C.elegans* (in the fifth and sixth columns)

**Table 7** Example 4

$\mathcal{E}_\rho$	$e(v_i \rightarrow v_j)$	$\mathcal{E}_\rho$	$e(v_i \rightarrow v_j)$	$\widehat{\mathcal{E}}_\rho$	$e(v_i \rightarrow v_j)$
0	<b><math>v_{53} \rightarrow v_1</math></b>	$2.7208 \cdot 10^{-2}$	<b><math>v_{73} \rightarrow v_{178}</math></b>	$1.3484 \cdot 10^{-1}$	<b><math>v_{305} \rightarrow v_{149}</math></b>
0	<b><math>v_{11} \rightarrow v_5</math></b>	$2.4126 \cdot 10^{-2}$	<b><math>v_{71} \rightarrow v_{217}</math></b>	$1.3000 \cdot 10^{-1}$	<b><math>v_{305} \rightarrow v_{219}</math></b>
0	$v_{12} \rightarrow v_6$	$2.2348 \cdot 10^{-2}$	$v_{72} \rightarrow v_{216}$	$1.2205 \cdot 10^{-1}$	$v_{305} \rightarrow v_{218}$
0	$v_{11} \rightarrow v_{19}$	$2.0573 \cdot 10^{-2}$	$v_{72} \rightarrow v_{144}$	$1.1067 \cdot 10^{-1}$	$v_{305} \rightarrow v_{178}$
0	$v_{11} \rightarrow v_{23}$	$1.8951 \cdot 10^{-2}$	$v_{76} \rightarrow v_{217}$	$1.0811 \cdot 10^{-1}$	$v_{305} \rightarrow v_{174}$
0	$v_{12} \rightarrow v_{24}$	$1.8489 \cdot 10^{-2}$	$v_{71} \rightarrow v_{47}$	$1.0128 \cdot 10^{-1}$	$v_{305} \rightarrow v_{81}$
0	$v_{12} \rightarrow v_{25}$	$1.8419 \cdot 10^{-2}$	$v_{75} \rightarrow v_{216}$	$9.7793 \cdot 10^{-2}$	$v_{305} \rightarrow v_{82}$
0	$v_8 \rightarrow v_{26}$	$1.7529 \cdot 10^{-2}$	$v_{74} \rightarrow v_{177}$	$9.5414 \cdot 10^{-2}$	$v_{305} \rightarrow v_{157}$
0	$v_{11} \rightarrow v_{26}$	$1.7392 \cdot 10^{-2}$	$v_{216} \rightarrow v_{81}$	$8.7441 \cdot 10^{-2}$	$v_{305} \rightarrow v_{216}$
0	$v_{12} \rightarrow v_{26}$	$1.7269 \cdot 10^{-2}$	$v_{78} \rightarrow v_{217}$	$8.6713 \cdot 10^{-2}$	$v_{305} \rightarrow v_{217}$

The smallest entries of the Perron edge importance vector and the relevant junctions that could be removed without changing the communicability in *C.elegans* significantly (displayed in the first two columns). The largest entries of the Perron edge importance vector and the relevant junctions to increase/decrease in order to increase/decrease the network communicability in *C.elegans* (in the third and fourth columns). The largest entries of the virtual Perron edge importance vector and the relevant junctions to add in order to increase the network communicability (displayed in the fifth and sixth columns)

Hence, this simplification is less satisfactory than  $\mathcal{G}_1$ . Elimination of the edges in bold face in the fourth column of Table 7 gives the network  $\mathcal{G}_6$ , for which one has

$$P_{\mathcal{G}_6} = 7.4856 \cdot 10^5; \quad \frac{P_{\mathcal{G}} - P_{\mathcal{G}_6}}{P_{\mathcal{G}}} = 1.8613 \cdot 10^{-1}; \quad T_{\mathcal{G}_6} = 2.9298 \cdot 10^6; \quad \frac{T_{\mathcal{G}} - T_{\mathcal{G}_6}}{T_{\mathcal{G}}} = 1.2284 \cdot 10^{-1}.$$

The decrease is less than that for  $\mathcal{G}_2$ . Conversely, increasing by one the weights associated with the same edges results in the network  $\mathcal{G}_7$ , for which one has

$$P_{\mathcal{G}_7} = 1.1140 \cdot 10^6; \quad \frac{P_{\mathcal{G}_7} - P_{\mathcal{G}}}{P_{\mathcal{G}}} = 2.1122 \cdot 10^{-1}; \quad T_{\mathcal{G}_7} = 3.7933 \cdot 10^6; \quad \frac{T_{\mathcal{G}_7} - T_{\mathcal{G}}}{T_{\mathcal{G}}} = 1.3569 \cdot 10^{-1}.$$

The increase is less than that for  $\mathcal{G}_3$ . Finally, setting to one the (vanishing) entries of  $A$  associated with the two virtual edges in bold face in the sixth column of Table 7 results in the network  $\mathcal{G}_8$ , having

$$P_{\mathcal{G}_8} = 3.5870 \cdot 10^6; \quad \frac{P_{\mathcal{G}_8} - P_{\mathcal{G}}}{P_{\mathcal{G}}} = 2.8999 \cdot 10^0; \quad T_{\mathcal{G}_8} = 7.3364 \cdot 10^6; \quad \frac{T_{\mathcal{G}_8} - T_{\mathcal{G}}}{T_{\mathcal{G}}} = 1.1965 \cdot 10^0.$$

Therefore the result is more satisfactory than that obtained for the network  $\mathcal{G}_4$ . Notice that  $\|\mathbf{W}|_{\mathcal{A}}\|_F = 1.2817 \cdot 10^{-1}$ .

### 4.3 Large networks

**Example 5** Consider the unweighted undirected graph  $\mathcal{G}$  that represents the continental US road network *Usroads-48*. The graph  $\mathcal{G}$ , which is available at [28], has  $n = 126146$  vertices, which represent intersections and road endpoints. The  $m = 161950$  edges represent roads that connect the intersections and endpoints. We analyze the network *Usroads-48* with the tools discussed in Subsections 2.2 and 3.2 for large networks.

We would like to determine approximations of the smallest and largest elements of  $L_f(A, ee^T)|_{\mathcal{L}}$  that correspond to edges that should be removed to simplify the network or whose edge-weight should be modified to increase or decrease the total communicability. Moreover, we would like to determine approximations of the smallest and largest elements of  $L_f(A, ee^T)|_{\widehat{\mathcal{L}}}$  that correspond to edges that should be added to increase total communicability. We first carry out 5 steps of the symmetric Lanczos process, computing the matrices  $\mathbf{U}_5$  and  $\mathbf{T}_5$ , and make use of the latter to construct both the matrix  $\mathbf{X}_5$  in (18) by using (19), and the matrix

$$\tilde{\mathbf{X}}_5 = \frac{f(\mathbf{T}_5 + hne_1e_1^T) - f(\mathbf{T}_5 - hne_1e_1^T)}{2h},$$

with  $h = \frac{2}{n} \cdot 10^{-4}$ . Then, proceeding as described in Subsection 2.2, we found that both the matrices  $\mathbf{X}_5$  and  $\tilde{\mathbf{X}}_5$  determined the same edges. Regarding the timings (in seconds) of the two procedures, we remark that, having available the symmetric and tridiagonal matrix  $\mathbf{T}_5$  determined by  $\ell = 5$  steps of the Lanczos process, evaluating the matrix  $\mathbf{X}_5$  in (19) required the average time  $t_1 \approx 2.9 \cdot 10^{-5}$  over  $10^4$  tests, while evaluating the matrix  $\tilde{\mathbf{X}}_5$  required the average time  $t_2 \approx 2.7 \cdot 10^{-5}$  in the same tests, with a relative average time saving

$$\frac{t_1 - t_2}{t_1} \approx 7.0 \cdot 10^{-2}.$$

The smallest element of the computed approximation of  $L_f(A, ee^T)|_{\mathcal{L}}$  is  $9.7462 \cdot 10^{-7}$  and is associated with the edge  $e(v_{123259} \leftrightarrow v_{123258})$ , while the largest element is  $1.5008 \cdot 10^1$  and is associated with the edge  $e(v_{19694} \leftrightarrow v_{19186})$ ; the smallest element of the computed approximation of  $L_f(A, ee^T)|_{\widehat{\mathcal{L}}}$  is  $2.2047 \cdot 10^{-9}$ ; it is associated with edge the  $e(v_{25416} \leftrightarrow v_{11651})$ . The largest element is  $1.9380 \cdot 10^1$  and is associated with the edge  $e(v_{58080} \leftrightarrow v_1)$ .

Turning to the the structured Perron communicability, one has  $P_{\mathcal{G}}^A = 1.9138 \cdot 10^2$ . The smallest entries of the Perron edge importance vector  $\mathcal{E}_\rho$  and the relevant edges are displayed in Table 8 in the first and second columns, while the largest entries of  $\mathcal{E}_\rho$  and the relevant edges are shown in the third and fourth columns. In order to increase the network communicability one should add edge  $e(v_{44182} \leftrightarrow v_{44035})$ , which is associated with the largest entry of the  $\widehat{\mathcal{L}}$ -analogue of the Wilkinson perturbation.

Removing the two edges in bold face in the second column of Table 8, associated with the smallest entries of the edge importance, gives the network  $\mathcal{G}_1$ , for which one

**Table 8** Example 5

$\mathcal{E}_\rho$	$e(v_i \leftrightarrow v_j)$	$\mathcal{E}_\rho$	$e(v_i \leftrightarrow v_j)$
$1.2197 \cdot 10^{-40}$	<b><math>v_{105751} \leftrightarrow v_{105743}</math></b>	$2.2346 \cdot 10^{-1}$	<b><math>v_{44182} \leftrightarrow v_{44067}</math></b>
$3.3336 \cdot 10^{-40}$	<b><math>v_{42664} \leftrightarrow v_{42479}</math></b>	$1.8605 \cdot 10^{-1}$	<b><math>v_{44182} \leftrightarrow v_{44154}</math></b>
$5.0488 \cdot 10^{-40}$	$v_{114032} \leftrightarrow v_{42664}$	$1.8090 \cdot 10^{-1}$	$v_{44182} \leftrightarrow v_{44087}$
$7.3043 \cdot 10^{-40}$	$v_{44150} \leftrightarrow v_{44015}$	$1.5882 \cdot 10^{-1}$	$v_{44154} \leftrightarrow v_{44067}$
$8.8379 \cdot 10^{-40}$	$v_{68387} \leftrightarrow v_{68213}$	$1.5443 \cdot 10^{-1}$	$v_{44087} \leftrightarrow v_{44067}$
$1.0103 \cdot 10^{-39}$	$v_{39830} \leftrightarrow v_{39787}$	$1.5077 \cdot 10^{-1}$	$v_{44323} \leftrightarrow v_{44182}$
$1.0695 \cdot 10^{-39}$	$v_{29088} \leftrightarrow v_{29056}$	$1.4155 \cdot 10^{-1}$	$v_{44255} \leftrightarrow v_{44182}$
$1.2043 \cdot 10^{-39}$	$v_{90123} \leftrightarrow v_{89379}$	$1.4099 \cdot 10^{-1}$	$v_{44356} \leftrightarrow v_{44182}$
$1.2478 \cdot 10^{-39}$	$v_{78533} \leftrightarrow v_{78388}$	$1.1663 \cdot 10^{-1}$	$v_{44067} \leftrightarrow v_{44035}$
$1.2821 \cdot 10^{-39}$	$v_{35630} \leftrightarrow v_{35115}$	$9.2663 \cdot 10^{-2}$	$v_{44067} \leftrightarrow v_{44019}$

The smallest entries of the Perron edge importance vector and the edges associated with the roads that could be removed in order to reduce the complexity of the network without changing the communicability in *Usroads-48* significantly (displayed in the first two columns). The largest entries of the Perron edge importance vector and the edges associated with the first ten roads that should be widened/narrowed in order to increase/decrease the network communicability the most (in the third and fourth columns)

has

$$P_{\mathcal{G}_1}^A = 1.9138 \cdot 10^2; \quad \frac{P_{\mathcal{G}}^A - P_{\mathcal{G}_1}^A}{P_{\mathcal{G}}^A} = 1.4851 \cdot 10^{-16}.$$

Setting to zero the weights associated with the two edges in bold face in the fourth column of Table 8 results in the network  $\mathcal{G}_2$  with

$$P_{\mathcal{G}_2}^A = 1.7487 \cdot 10^2; \quad \frac{P_{\mathcal{G}}^A - P_{\mathcal{G}_2}^A}{P_{\mathcal{G}}^A} = 8.6228 \cdot 10^{-2},$$

while increasing the weights associated with the same edges by one results in the network  $\mathcal{G}_3$ , for which one has

$$P_{\mathcal{G}_3}^A = 3.6445 \cdot 10^2; \quad \frac{P_{\mathcal{G}}^A - P_{\mathcal{G}_3}^A}{P_{\mathcal{G}}^A} = 9.0435 \cdot 10^{-1}.$$

Finally, setting to one the (vanishing) entry  $w_{44182,44035}$  of the adjacency matrix **A** associated with the virtual edge  $e(v_{44182} \leftrightarrow v_{44035})$  returns the network  $\mathcal{G}_4$ , with

$$P_{\mathcal{G}_4}^A = 2.4540 \cdot 10^2; \quad \frac{P_{\mathcal{G}_4}^A - P_{\mathcal{G}}^A}{P_{\mathcal{G}}^A} = 2.8228 \cdot 10^{-1}.$$

Notice that one has  $\|\mathbf{W}|_{\mathcal{A}}\|_F = \sqrt{2}\|\mathbf{W}|_{\mathcal{L}}\|_F = 2.9318 \cdot 10^{-1}$ .

## 5 Conclusion and comments on related work

The identification of important and unimportant edges is a fundamental problem in network analysis. Several techniques for this purpose have been described in the literature; see, e.g., [5, 7, 8, 10, 21, 22]. In [4] the authors propose a method that uses the gradient of the total communicability, and Schweitzer [27] recently described how the computational effort required by this method can be reduced. Section 2 of this paper reviews this method and discusses computational aspects when this technique is applied to small and medium-sized networks, as well as to large-scale networks. In particular, further ways to speed up the computations when the method is applied to large-scale networks are described.

Another approach to identify important and unimportant edges is to determine edge weights whose modification yields a relatively large change in the Perron root of the adjacency matrix. This is described in [10, 21]. The computations required are quite straightforward and the method discussed in the latter reference is easy to implement also for large-scale problems. We therefore are interested in whether modifications of the weights of the edges identified by this technique give a relatively large change in the total communicability. Section 3 reviews the method described in [21] and extends it to include edge removal. Computed examples reported in Section 4 show that, indeed, modifications of edge weights identified by the technique discussed in [21] yield relatively large changes in the total communicability.

**Acknowledgements** The authors would like to thank an anonymous referee for useful suggestions, which resulted in clarifications of the presentation. Research by SN was partially supported by a grant from SAPIENZA Università di Roma, by INdAM-GNCS, and by the PRIN 2022 research project “Inverse Problems in the Imaging Sciences (IPIS)”, grant n. 2022ANC8HL.

**Author Contributions** All authors contributed equally to the paper.

**Funding** Open access funding provided by Università degli Studi di Roma La Sapienza within the CRUI-CARE Agreement. See acknowledgement.

**Data Availability Statement** Data sharing not applicable to this article as no new data sets were generated during the current study.

## Declarations

**Conflicts of interest** The authors declare that they have no conflict of interest.

**Ethical standard** Not Applicable.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.



## References

1. Arrigo, F., Benzi, M.: Edge modification criteria for enhancing the communicability of digraphs. *SIAM J. Matrix Anal. Appl.* **37**, 443–468 (2016)
2. Arrigo, F., Durastante, F.: Mittag-Leffler functions and their applications in network science. *SIAM J. Matrix Anal. Appl.* **42**, 1581–1601 (2021)
3. Benzi, M., Klymko, C.: Total communicability as a centrality measure. *J. Complex Netw.* **1**, 124–149 (2013)
4. De la Cruz Cabrera, O., Jin, J., Noschese, S., Reichel, L.: Communication in complex networks. *Appl. Numer. Math.* **172**, 186–205 (2022)
5. De la Cruz Cabrera, O., Jin, J., Reichel, L.: Sparse approximation of complex networks. *Appl. Numer. Math.* in press. (2024)
6. De la Cruz Cabrera, O., Matar, M., Reichel, L.: Analysis of directed networks via the matrix exponential. *J. Comput. Appl. Math.* **355**, 182–192 (2019)
7. De la Cruz Cabrera, O., Matar, M., Reichel, L.: Edge importance in a network via line graphs and the matrix exponential. *Numer. Algorithms* **83**, 807–832 (2020)
8. De la Cruz Cabrera, O., Matar, M., Reichel, L.: Centrality measures for node-weighted networks via line graphs and the matrix exponential. *Numer. Algorithms* **88**, 583–614 (2021)
9. Dynamic Connectome Lab - Data Sets. <https://sites.google.com/view/dynamicconnectomelab> (2024)
10. El-Halouy, S., Noschese, S., Reichel, L.: Perron communicability and sensitivity of multilayer networks. *Numer. Algorithms* **92**, 597–617 (2023)
11. Estrada, E.: *The Structure of Complex Networks: Theory and Applications*. Oxford University Press, Oxford (2011)
12. Estrada, E., Higham, D.J.: Network properties revealed through matrix functions. *SIAM Rev.* **52**, 696–714 (2010)
13. Estrada, E., Rodriguez-Velazquez, J.A.: Subgraph centrality in complex networks. *Phys. Rev. E* **71**, Art. 056103 (2005)
14. Fenu, C., Martin, D., Reichel, L., Rodriguez, G.: Network analysis via partial spectral factorization and Gauss quadrature. *SIAM J. Sci. Comput.* **35**, A2046–A2068 (2013)
15. Higham, N.J.: *Functions of Matrices: Theory and Computation*. SIAM, Philadelphia (2008)
16. Kandolf, P., Koskela, A., Relton, S.D., Schweitzer, M.: Computing low-rank approximations of the Fréchet derivative of a matrix function using Krylov subspace methods. *Numer. Linear Algebra Appl.* **28**, Art. 2401 (2021)
17. Massei, S., Tudisco, F.: Optimizing network robustness via Krylov subspaces. [arXiv:2303.04971](https://arxiv.org/abs/2303.04971) (2024)
18. Milanese, A., Sun, J., Nishikawa, T.: Approximating spectral impact of structural perturbations in large networks. *Phys. Rev. E* **81**, Art. 046112 (2010)
19. Newman, M.E.J.: *Networks: An Introduction*. Oxford University Press, Oxford (2010)
20. Noschese, S., Pasquini, L.: Eigenvalue condition numbers: Zero-structured versus traditional. *J. Comput. Appl. Math.* **185**, 174–189 (2006)
21. Noschese, S., Reichel, L.: Estimating and increasing the structural robustness of a network. *Numer. Linear Algebra Appl.* **29**, Art. e2418 (2022)
22. Noschese, S., Reichel, L.: Network analysis with the aid of the path length matrix. *Numer. Algorithms* **95**, 451–470 (2024)
23. Ruhe, A.: The two-sided Arnoldi algorithm for nonsymmetric eigenvalue problems. In *Matrix Pencils*, Kågström, B., Ruhe, A. (eds.) *Lecture Notes in Mathematics #973*, Springer, Berlin, pp. 104–120 (1983)
24. Ruiz, P., Sastre, J., Ibáñez, J., Defez, E.: High performance computing of the matrix exponential. *J. Comput. Appl. Math.* **291**, 370–379 (2016)
25. Saad, Y.: *Numerical Methods for Large Eigenvalue Problems*, revised SIAM, Philadelphia (2011)
26. Saad, Y.: *Iterative Methods for Sparse Linear Systems*, 2nd edn. SIAM, Philadelphia (2003)
27. Schweitzer, M.: Sensitivity of matrix function based network communicability measures: Computational methods and a priori bounds. *SIAM J. Matrix Anal. Appl.* **44**, 1321–1348 (2023)
28. SuiteSparse Matrix Collection. <https://sparse.tamu.edu> (2024)
29. Wilkinson, J.H.: Sensitivity of eigenvalues II. *Util. Math.* **30**, 243–286 (1986)
30. Xu, S., Xue, F.: Inexact rational Krylov subspace methods for approximating the action of functions of matrices. *Electron. Trans. Numer. Anal.* **58**, 538–567 (2023)

31. Zwaan, I.N., Hochstenbach, M.E.: Krylov–Schur–type restarts for the two-sided Arnoldi method. *SIAM J. Matrix Anal. Appl.* **38**, 297–321 (2017)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.