Optimal network modification for spectral radius dependent phase transitions

Yonatan Rosen\textsuperscript{1,3}, Lior Kirsch\textsuperscript{2,3} and Yoram Louzoun\textsuperscript{1,2}

\textsuperscript{1} Department of Mathematics, Bar-Ilan University, Ramat Gan, Israel
\textsuperscript{2} The Gonda Multidisciplinary Brain Research Center, Bar-Ilan University, Ramat Gan, Israel
\textsuperscript{3} These authors have contributed equally.

E-mail: ylouzoun@gmail.com

Keywords: networks, contact process, percolation, synchronization, epidemics, spectral radius

Abstract

The dynamics of contact processes on networks is often determined by the spectral radius of the networks adjacency matrices. A decrease of the spectral radius can prevent the outbreak of an epidemic, or impact the synchronization among systems of coupled oscillators. The spectral radius is thus tightly linked to network dynamics and function. As such, finding the minimal change in network structure necessary to reach the intended spectral radius is important theoretically and practically.

Given contemporary big data resources such as large scale communication or social networks, this problem should be solved with a low runtime complexity. We introduce a novel method for the minimal decrease in weights of edges required to reach a given spectral radius. The problem is formulated as a convex optimization problem, where a global optimum is guaranteed. The method can be easily adjusted to an efficient discrete removal of edges. We introduce a variant of the method which finds optimal decrease with a focus on weights of vertices. The proposed algorithm is exceptionally scalable, solving the problem for real networks of tens of millions of edges in a short time.

1. Introduction

The relation between structure and dynamics in networks has been a subject of extensive research in recent years, extending over diverse domains such as diffusion, epidemiology, percolation, synchronization and random walks [1–6]. One of the most prominent insights is that the largest eigenvalue of a graph’s adjacency matrix, $\lambda_1$, can determine different modes of network dynamics. In two of the most fundamental models of epidemics spread, SIS and SIR, a network is infected if the infection rate $\tau$ exceeds the critical value of $\tau_c = \frac{1}{\lambda_1}$ [7–9]. The same relation of phase transition characterizes a system of coupled oscillators where the critical coupling strength for the emergence of coherence is proportional to $\frac{1}{\lambda_1}$ [10]. Additionally, in a percolation process a directed network disintegrates when the largest eigenvalue of the adjacency matrix, which is modulated by removal probabilities, is smaller than 1 [11]. For a review of further effects of the spectral radius in multiple other applications, see [12].

Dynamical processes and networks structure are also related to the spectral properties of matrices other than the adjacency matrix. For example, the emergence of synchronization in networks has been found to be linked to the eigenvalues of the networks Laplacian matrix [1, 13, 14]. As such, efficient ways to approximate this spectrum, in the case of small world networks, were given via mean field theory [15] or by recursive relations using the characteristic polynomial [16]. In another case, the spectra of the graph Laplacian was used to optimally reconstruct brain networks [17]. In the current analysis, we focus on the criticality of the largest eigenvalue of the adjacency matrix.
Motivated by the interplay between the spectral radius and dynamics, strategies for affecting phase transitions were first proposed in percolation problems. An early solution was a ranking of vertices by their connectivity [18, 19]. This ranking was improved by a centrality measure specifically designed for ranking edges or vertices via spectral radius reduction—the dynamical importance method [20]. Taking a perturbative approach, the dynamical importance method is defined as the first order approximation of the relative decrease in the radius when an edge or vertex is removed from the network. This work was further expanded to include structural perturbations of removing or adding a small subgraph and second order terms [21]. A different line of research is focused on a closely related problem, where minimization of the spectral radius is constrained by the number of edges allowed to be removed. Interestingly, the proposed algorithms for this problem [22, 23], offer similar ranking to that of the dynamical importance algorithm [20].

The aforementioned algorithms share a common strategy—they perform a full removal of edges, which results in discrete solutions. In the case of weighted networks, where it is possible to regulate the weights of edges continuously, this all-or-none discrete edge removal is a drastic measure which provides a suboptimal solution. This strategy of discrete removal has two major shortcomings.

First, in many real world applications, continuous reduction of weights of edges is favorable. One such example is found in epidemiological models on weighted networks. The aforementioned criticality of the spectral radius and the infection rate also applies to these models [5, 24]. Worldwide epidemic spread is mediated through transportation networks, e.g., the spread of Influenza, SARS or Ebola through the global flight network [25–28]. In these networks, issuing a full quarantine by closing flight routes is much harsher than reducing the frequencies of flights or imposing limitations on passengers, an operation that maintains the connectivity of the aerial transportation network. Another case where a more subtle manipulation of the network is preferable over removal strategy is found in neuronal networks. The network of neurons in the brain has been modeled as a system of coupled oscillators, momentarily synchronizing and desynchronizing [29]. Full removal of edges in this system results in the complete blockage of communication routes in the brain, while continuous decrease in weights of edges can be equivalent to fine tuning of this network by electric stimulation. A specific example can be found in epilepsy where seizures are characterized by periods of excessive synchronization [30]. During the onset of a seizure, a large increase of the spectral radius is conspicuous in a functional neural network constructed from EEG recordings [31]. A possible treatment of a weighted character of the brain’s relevant signaling network is accurate neuro-stimulation within responsive antiepileptic therapy [32].

The second shortcoming of the full removal approach is that removal of the smallest set of edges is an NP hard problem [23]. Previous methods [20–23] use heuristic approaches and their solutions are not necessarily optimal. Defining the problem in the continuous domain produces the globally optimal solution in undirected networks (see theorem below). Therefore, our approach resolves three important previous limitations by using: (A) a softer edge reduction method, (B) a more computationally efficient algorithm, (C) an optimal global solution providing better results than current heuristic approaches.

2. Problem formulation

Let $G = (V, E, Q)$ be an undirected graph where the weights of edges $Q$ are non-negative. If all weights are equal, $G$ is equivalent to an unweighted graph. Let $A$ mark the adjacency matrix of $G$, and $\beta$ the targeted spectral radius. The function $\lambda_1(W) : = \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$ returns the largest eigenvalue of a given symmetric adjacency matrix $W$, which in absolute value is the spectral radius. We seek the matrix $W$, with the minimal distance to $A$, whose spectral radius is bound by $\beta$ and has non-negative edge weights smaller or equal to the values of $A$. The optimization problem can be formally defined as:

$$\arg \min_W \|A - W\|$$

s.t.

$$|\lambda_1(W)| \leq \beta$$

$$\forall (i, j) \in E : 0 \leq W_{ij} \leq A_{ij}$$

$$W \in \text{Sym}_n(\mathbb{R})$$

$\text{Sym}_n(\mathbb{R})$ is the vector space of all symmetrical square matrices over $\mathbb{R}$ of dimension $n$. The norm in the objective function $\|A - W\|$ can be set as the $L_1$ norm or the Frobenius norm $L_2$, as shall be discussed in more detail in the algorithm description. We shall henceforth call $|\lambda_1| \leq \beta$ the spectral constraint, and mark by $S_\beta$ the set of all matrices of spectral radius smaller than or equal to $\beta$. The condition $\forall (i, j) \in E : 0 \leq W_{ij} \leq A_{ij}$ will be referred as the box constraint, and the set of matrices bounded by the hyper rectangle derived from these linear constraints will be marked by $S_{\text{box}}$. 


In undirected networks, the optimization problem in equation (1) is a convex problem. The objective function is the norm of a linear function, which is a convex function. The box constraint ($S_{\text{box}}$) is an intersection of half-planes and thus is a convex set. In symmetric matrices, the spectral radius is equal to the spectral norm, $\lambda_1(W) = \|W\|_2$. Hence, the epigraph of the spectral norm, $S_j \cap S_{\text{box}} \cap \text{Sym}_n(\mathbb{R})$ is a convex optimization problem.

In asymmetric matrices, the spectral constraint does not necessarily induce a convex set. In this paper, we focus on undirected networks, weighted and unweighted. Since the problem is convex, it follows immediately that the decrease in the solution of the optimization problem (1) is smaller than or equal to the decrease obtained from all current heuristic or perturbative algorithms.

3. Algorithms for optimal weight reduction

Following the optimization problem in equation (1), we name our algorithm SPITZ—SPectral lImited optimization (in addition, ‘Spitz’ in German means ‘Pointed’, which implies the convex character of the problem). When using a continuous reduction, the norm in the objective function, $\|A - W\|_i$, can be determined by the desired application. In this paper, we discuss two types of objective functions: the $L_1$ norm, where $\|A - W\|_1 = \sum_{i,j} |A_{ij} - W_{ij}|$, and the $L_2$ norm, where $\|A - W\|_2 = \sqrt{\sum_{i,j} (A_{ij} - W_{ij})^2}$ (the adjacency matrix Frobenius norm).

When one desires the main mass of the reduction to be concentrated in a small number of edges, the $L_1$ norm should be used for the objective. Alternatively, when a distributed reduction across many edges is required, the preferred objective function should use the $L_2$ norm.

For each objective function, we propose a different optimization method to reach the optimal solution. When the objective function is $\|A - W\|_1$, we use the interior point algorithm [33]. We denote the application of interior point method as Spitz $L_1$ algorithm. We have accelerated the interior point algorithm in this case, using the LBFGS method [34]. This method uses the gradients of the several last iterations to approximate the inverse of the hessian. The gradient of the spectral constraint function is given in the appendix along with a proof that allows us to dismiss the upper bounds and thus discard half of the constraints (appendices C and D) and thus also improve runtime. Still, in practice, the runtime of the interior point implementation increases substantially with the number of edges making this algorithm impractical in large scale networks.

When the objective function is $\|A - W\|_2$, we propose an algorithm based on a concept developed by Dykstra for projections onto convex sets [35] and denote it Spitz $L_2$. Our $L_2$ algorithm scales exceptionally well and in practice is only marginally affected by the size of the network, and can be used in networks with hundreds of millions edges and possibly beyond.

Dykstra’s method iterates by alternately projecting onto $S_j$ or $S_{\text{box}}$ (circle and the box respectively in figure 1), using intermediate auxiliary variables. The process is initialized by the original adjacency matrix and is guaranteed to converge into the closest matrix to $A$ in the intersection $S_j \cap S_{\text{box}}$ (see appendix J for additional details).
Dykstra’s algorithm assumes that each projection step (alternately onto $S_f$ and onto $S_{max}$) can be computed efficiently. The projection onto $S_{max}$ is done by substituting each coordinate that is outside the box boundaries by the boundary value. Projecting onto $S_f$ is less trivial but can also be calculated analytically, as follows:

Let $U'DU = A$ be the singular value decomposition of the non-negative symmetric matrix $A$. $D^β$ represents the matrix which is identical to the diagonal matrix $D$ except that in $D^β$ eigenvalues larger than $β$ are substituted by $β$. For example, when $D$ has $k$ eigenvalues larger than $β$, $D^β$ is

$$
D^β = \begin{pmatrix}
\beta & 0 & 0 & \cdots & 0 \\
0 & \beta & 0 & \cdots & 0 \\
0 & 0 & \beta & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & \beta
\end{pmatrix},
$$

(2)

where the eigenvalues are ordered from largest to smallest, $λ_1 ≥ λ_2 ≥ \cdots ≥ λ_n$. Let us denote $P_f(A)$ as the projection of $A$ onto $S_f$, $P_f(A) = \text{arg min}_{W \in S_f} \|A - W\|_F$. We shall now prove that $P_f(A) = U'D^βU$.

We use the decomposition of $A$ to obtain a simpler interpretation of the objective function. First, in $\|A - W\|_F$, we substitute $A$ by its decomposition $U'DU$.

$$
\|A - W\|_F = \|U'DU - W\|_F = \|D - WUU^T\|_F,
$$

(3)

The second equality in (3) stems from the invariance of the Frobenius norm under multiplication by orthogonal matrices, following the cyclic property of the trace.

Therefore, in choosing $W$ two requirements must be answered: first, the distance from the diagonal matrix $D$ in equation (3) should be minimized. Second, the eigenvalues of $W$ must be smaller than or equal to $β$. Choosing $W = U'D^βU$ indeed satisfies the spectral constraint and yields

$$
\|D - UUU^TDD^βUU^T\|_F = \|D - D^β\|_F = \sqrt{\sum_{i=1}^{k} (λ_i - β)^2},
$$

(4)

where the sum in (4) is derived from the definition of $D^β$ in equation (2). This is minimal since all entries of the matrix $D - D^β$ equal zero, except the first $k$ entries of the diagonal. The $β$ elements in these $k$ entries of $D^β$ cannot be enlarged further to diminish the difference from their matching eigenvalues in $D$ since it would violate the spectral constraint.

4. Results

4.1. Continuous reduction in edges

We have shown in the previous section that the proposed algorithm produces the minimal reduction in the total weight that is needed in order to obtain a targeted spectral radius (both $L_1$ and $L_2$ objective functions are convex). We now evaluate the performance by measuring the change in the network necessary to reach a targeted $β$ with comparison to the state of the art methods for edge removal. The current best performing algorithm is the dynamical importance method by Restrepo et al [20]. We have also tested various other methods of edge reduction, but they all provided inferior results in comparison to dynamical importance in all cases, and much worse results than the current approach (appendix A). The dynamical importance method is based on first order perturbations. The dynamical importance method was later extended to use second order perturbations [21], but we found that it provides negligible differences from first order perturbations and is not discussed here.

We define the comparison criterion as the relative decrease, $\frac{\|A - W\|}{\|A\|}$, where $W$ is the resulting matrix following the application of an edge reduction/removal algorithm and $A$ is the original matrix. Note that this criterion is proportional to $\|A - W\|$, thus optimizing this value for a given $A$ is equivalent to optimizing the objective function defined above. The averages and standard deviations of the relative decreases resulting from the implementations of all of the following algorithms are shown in appendix H.

We first compared the methods on two types of random networks, configuration model scale free (SF) and Erdos–Renyi (ER) networks [36, 37]. In the SF networks studied here, the degree distributions follow a power law with slope $-2.5$, while the ER networks have a fixed density of 0.05. We further divided each of the two types of networks into unweighted networks and weighted networks where weights of edges are uniformly distributed between 0 and 1.

For each of these four cases, we generated 100 random networks in different sizes over a logarithmic scale of 100–1000 vertices. For each of the 10 network sizes, we repeated the generating process 10 times. For all
networks in the results section, the targeted $\beta$ was defined as 0.8 of $\lambda_1$ (the original spectral radius of $A$). We tested other values of $\beta$ in appendix B.

As expected by convexity, in all networks, our two algorithms outperform the dynamical importance algorithm. We observed very large differences when using the Frobenius norm (figures 2(a)–(d)). Using the $L_1$ criterion, smaller differences were found in the weighted networks (figures 2(e), (g)) and negligible ones for unweighted networks (figures 2(f), (h)).

Next, we tested a large number of weighted and unweighted real-world networks from diverse domains (appendix F), curated by the Koblenz repository [38]. As in the previous test (figure 3), we examine the networks using the $L_1$ and $L_2$ relative decreases. Similarly to the comparison using random networks, the solutions of the continuous algorithms are closer to the original matrix $A$ in all networks. We find a substantial difference in weighted networks using either an $L_1$ or $L_2$ criterions (figures 3(a), (c)) and also in unweighted networks using $L_1$ criterion (figure 3(b)). In unweighted networks and using the $L_1$ criterion, the dynamical importance and the Spitz $L_1$ algorithm have only marginal differences, (figure 3(d)). Note that $L_1$ optimization drives to a sparse solution, forcing the continuous optimization to approximate whole edge removal solution. This can explain the similar scores of our optimal solutions and the ones obtained by the dynamical importance algorithm, as found in the tested real and random networks (figures 2 and 3). A single exception was found in the 'Facebook (NIPS)' network where the two solutions differ considerably in favor of the continuous Spitz algorithm. The edges that were reduced in the 'Facebook (NIPS)' network emphasize some key points in the difference in behavior of the two methods (figure S10). The optimal continuous $L_1$ solution has a unique property—approximately half of the edges achieve the same optimal value (the optimal $L_1$ weights distribution shows two peaks at 0.805 and 0.835, figure S10 in appendix M). All these edges share the same influence on the spectral radius and have equal weights. The dynamical importance method, is not able approximate the solution of Spitz, which is the optimal solution. Due to its inability to continuously reduce the edges, it is forced to fully remove some of these edges resulting in a suboptimal solution, which is considerably different from the optimal solution.

4.2. Discrete reduction in edges

In some scenarios, it is not possible to reduce edges in a continuous manner. In these scenarios the removal of entire edges is the only viable option. For example, in an extremely contagious disease, when all other options are
exhausted, quarantine actions are taken by closing all routes to avoid mass infection. We define discrete solutions as edge removal strategies where there are only two options: leave the edge intact ($W_{ij} = A_{ij}$) or remove it completely ($W_{ij} = 0$). Continuous solutions are strategies which allow a continuous tuning of the weights ($0 \leq W_{ij} \leq A_{ij}$).

The formulation of the optimization problem in equation (1) was developed for the continuous domain. However, it is easily extended to the discrete case using a two-steps process. First, we rank the edges using their scores from the continuous optimal solution $W$. The edges are ranked by $-A_{ij}W_{ij}$. In this score, edges in $W$ that deviated strongly from $A$ are ranked high. Then, using the derived order of the edges, we apply a binary search to remove the minimal set of edges that results in a network whose spectral radius is below $\beta$. The discrete extension comes almost without any substantial computation cost since binary search is an extremely fast operation with only $\log(E)$ comparisons, as shall be discussed shortly in the complexity analysis. Note that the problem of discretely removing edges can be formalized as an integer programing problem—a challenging domain with many open problems. As such, both the dynamical importance and the discrete Spitz method offer heuristic algorithms and neither are guaranteed to provide the optimal discrete solution.

We compare the results of our discrete Spitz $L_1$ and Spitz $L_2$ algorithms with the dynamical importance algorithm, which is inherently designed for discrete solutions. In real-world weighted networks (figure 4(a)), the Spitz $L_1$ discrete algorithm produces solutions that are similar, or with a slight improvement, comparing to the solutions of the dynamical importance method. The two methods also show similar performance in real-world unweighted networks (figure 4(c)), with a slight better performance to the dynamical importance method. Interestingly, in the unweighted case, the performance in discrete removal (figure 4(c)) is the mirror image of the performance in continuous reduction (figure 3(d)). Since the dynamical importance discrete solutions are close to the optimal $L_1$ solution, it is not surprising that the two algorithms also perform similarly in discrete removal of edges. A single exception is the $L_1$ method reduction to the ‘Facebook (NIPS)’ network. This was the only network where the dynamical importance was substantially outperformed by the continuous unweighted $L_1$ case (figure 3(d)). As discussed in the continuous reduction analysis, in this network, there is a set of edges with a shared influence level on the spectral radius (half of the edges had equal weight—two peaks at 0.805 and 0.835, figure S10, appendix M). This equal-weights scenario does not contribute ranking information to the discrete $L_1$ removal due to the large number of ties in weights. On the other hand, the dynamical importance benefits from its greedy nature. After choosing an edge from a set of edges with similar importance (the dynamical importance analysis)}
The importance of edges are recalculated following a removal. In this process, the uniformity in the dynamical importance scores is gradually broken and an order is formed, which enables better discrete ranking. We also used the Spitz $L_2$ algorithm for discrete reduction. In the weighted networks (figure 4(a)), we find that the Spitz $L_2$ performs similarly to the Spitz $L_1$ and to dynamical importance. The fact that in the discrete Spitz $L_1$ and $L_2$ algorithms perform slightly better in weighted networks comparing to unweighted networks may be related again to the frequency of ties. In weighted networks, which are heterogeneous in weights, the occurrences of ties which impair the ranking are unlikely.

The continuous $L_1$ and $L_2$ solutions are highly correlated (figure 4(b)) and thus in general should derive similar ranking and consequently a similar discrete solution. Despite the similarity in ranking, we found that in a few cases, especially in unweighted networks (figure 4(c)), the Spitz $L_1$ method is a better proxy for discrete removal and performs better than the Spitz $L_2$ algorithm. However, the Spitz $L_2$ method is considerably faster than the other methods and in practice it is not limited by network’s size. This enabled us to scale the problem from medium size networks (figure 4(c)) to large-scale networks (figure 4(d)). The Spitz $L_2$ method is also able to scale the discrete reduction to networks with tens of millions of edges, for which the dynamical importance was unable to reach the required spectral radius after one day of computation as demonstrated in the following runtime analysis section.

### 4.3. Complexity analysis

Apart from the measured minimal change derived from the algorithms’ use, we are also interested in the scalability of the studied algorithms. For both Spitz and dynamical importance, the main operation performed in each iteration is the computation of the dominant eigenvector (the eigenvector of the spectral radius). This operation depends on the number of edges in the network, $|E|$, and on the eigengap, $\gamma$. A computation of the dominant eigenvector requires $O\left(\frac{|E|}{\log(1/\epsilon_{\text{eq}})}\right)$ operations [39], where $\epsilon_{\text{eq}}$ is the desired distance from the optimal solution.

At each iteration of dynamical importance, it computes the dominant eigenvector in order to find the edge with maximal importance. Then, this edge, $e$, is removed. This procedure is repeated in a new graph whose edges
are $E/ε$, until $β$ is reached. This results in a sequential deletion of edges one by one. In total, after removing $m$ edges, it performs $O\left(m\left(|E| - \frac{m}{2}\right) \log \left(\frac{1}{\varepsilon_{\text{Spitz}}}\right)\right)$ operations (further details in appendix N).

We can derive an exact expression for the dynamical importance algorithm complexity in terms of the number of edges. In contrast to sequential deletion in dynamical importance method, the Spitz algorithm tunes the weights of all the edges jointly and continuously. The complexity of Spitz algorithm is measured by its rate of convergence.

We found that the Spitz method converges linearly to the solution with a network-dependent constant, $C$. It performs $O\left(C \log \left(\frac{1}{\varepsilon_{\text{Spitz}}}\right)\right)$ iterations where $\varepsilon_{\text{Spitz}}$ is the distance from the optimal solution (figure S11 in appendix N). In each iteration, the main operation is the computation of the dominant eigenvector for the spectral projection. In total, the Spitz method performs $O\left(C \log \left(\frac{1}{\varepsilon_{\text{Spitz}}}\right)|E| \log \left(\frac{1}{\varepsilon_{\text{Spitz}}}\right)\right)$ operations. In our discrete extension, the binary search requires an additional $\log (\log |E|)$ comparisons. In general, this is negligible compared to the runtime of the optimization procedure. The total complexity of the Spitz discrete extension is $O\left(C \log \left(\frac{1}{\varepsilon_{\text{Spitz}}}\right)|E| \log \left(\frac{1}{\varepsilon_{\text{Spitz}}}\right) + \log (|E|)|E| \log \left(\frac{1}{\varepsilon_{\text{Spitz}}}\right)\right)$.

Since computing the largest eigenvalue can be costly in large networks, we have further accelerated its computation using specific initialization. At each iteration, we initialize the call to the power method with the cached eigenvector of the previous iteration instead of a random vector. This has improved the runtime of dynamical importance dramatically and enabled us to apply dynamical importance to networks of a larger scale. Moreover, this caching mechanism accelerates a single iteration of dynamical importance more than a single iteration of Spitz. In dynamical importance, the removal of a single edge causes a limited change in the dominant eigenvector in general. In Spitz, on the other hand, we also see improvement in performance but it is less drastic because the power method works on the corrected matrices (rows 4–7 in the pseudo code in appendix J). However, even with this caching mechanism, the total number of iteration performed by the dynamical importance algorithm is not changed—it is still forced to repeat the ranking computation following the removal of each single edge.

The number of iterations of the dynamical importance method is determined by the number of edges removed. In contrast, the number of iterations in the Spitz method does not depend directly on the number of reduced or removed edges. In practice, the Spitz $L_2$ algorithm converges after several hundred iterations, independently of the networks size (figure S12 in appendix N). This explains well the observed differences in runtime, as follows.

In medium networks, the total number of iterations performed by dynamical importance is small since the overall number of edges in the network is not of considerable size. This allows the dynamical importance method to achieve the best results in terms of runtime. The runtime difference in the medium scale is in the order of several minutes for Spitz $L_2$ and dynamical importance. The Spitz $L_1$ does not scale well even for medium scale networks. In large networks, we see a considerable difference in runtime, mainly due to the number of iterations taken by each method. When millions of edges are removed dynamical importance requires millions of iterations. On the other hand, the number of iterations in the Spitz $L_2$ algorithm is considerably smaller than the number of edges that are removed. We find that the Spitz $L_2$ algorithm runs more than two orders of magnitude faster than dynamical importance (figure 5(b)). In addition, the dynamical importance failed to terminate in the seven largest networks after a single day. Therefore, using dynamical importance is impractical in such large scales. In contrast, the Spitz $L_2$ method is able to scale and to compute the optimal reduction even for networks with tens of millions of edges.

4.4. The vertices problem

In some applications, the dynamics of the network can only be affected by manipulating the vertices. For example, in computer networks the propagation of viruses can be contained by controlling central servers in a communication network. The effect of vertices on network structure can be viewed as the combined contribution of adjacent vertices to the strength of their mutual edge.

We reformulate the optimization problem as minimization of vertices weights. In this formulation, each edge of the optimal network is determined by a combination of values from its two adjacent vertices. For example, in a social network, the frequency of interactions between two people is affected by the motivation of each one. The simplest case is when a pair of people is equally motivated to interact. This, however, is not always the case and sometimes one party contributes more to the connection than the other.

Formally, each vertex is now assigned an optimization variable $v_i$, where $0 \leq v_i \leq 1$. We use the $v$ variables to tune edges and construct a new adjacency matrix, marked now by $W'$ (the vertices equivalent to $W$ of the edges problem). In the case of equal contribution of two vertices to their mutual edge, we formulate...
The values of $v_i$ are tuned to optimally control the participation of each vertex in its interactions. In the extreme case where the optimal value of $v_i$ is zero the strength of all interactions of a vertex $i$ is determined solely by its neighbors. At the other extreme end, when the value of $v_i$ is 1, its contribution to its interactions remains unchanged following the optimization.

When the contribution of nodes to an interaction is a priori uneven, we extend the formulation to include such biases. For each edge, we predefine a constant $p_{ij}$ where $0 \leq p_{ij} \leq 1$. A $p_{ij}$ constant controls in a specific interaction of vertices $i$ and $j$ the extent to which each of the two vertices can contribute to their mutual edge. We now define the matrix $W'$ as

$$W'_{ij} = \frac{(v_i + v_j)}{2} A_{ij}. \quad (5)$$

The values of $v_i$ are tuned to optimally control the participation of each vertex in its interactions. In the extreme case where the optimal value of $v_i$ is zero the strength of all interactions of a vertex $j$ is determined solely by its neighbors. At the other extreme end, when the value of $v_i$ is 1, its contribution to its interactions remains unchanged following the optimization.

When the contribution of nodes to an interaction is a priori uneven, we extend the formulation to include such biases. For each edge, we predefine a constant $p_{ij}$ where $0 \leq p_{ij} \leq 1$. A $p_{ij}$ constant controls in a specific interaction of vertices $i$ and $j$ the extent to which each of the two vertices can contribute to their mutual edge. We now define the matrix $W'$ as

$$W'_{ij} = (p_{ij} v_i + (1 - p_{ij}) v_j) A_{ij}. \quad (6)$$

For example, a zero value in $p_{ij}$ means that node $i$ has no contribution to the edge with $j$ and a value of one means that node $i$ is the sole contributor to this specific connection. We preserve the symmetry of the network by requiring that $p_{ij} = 1 - p_{ji}$. This vertices-targeted formulation is a natural variant of the edges optimization problem in equation (1). The defined bounds of the optimization variables $v_i$ and the constants $p_{ij}$ in the interval $[0, 1]$ lead to $0 \leq W'_{ij} \leq A_{ij}$, as was the case in the edges problem formulation. Therefore, the resulting matrix $W'$ is also a feasible point in the space of the edges optimization problem. The optimization problem in vertices is now formulated as follows:

$$\arg \min_v \| A - W' \|$$

s.t.

$$\| \lambda_1(W') \| \leq \beta$$

$$\forall \; i = 1, \ldots, n : 0 \leq v_i \leq 1$$

$$\forall \; (i, j) \in E : W'_{ij} = (p_{ij} v_i + (1 - p_{ij}) v_j) A_{ij}.$$

The $p_{ij}$ constants can be given a priori or alternatively derived from the network structure. For example, the degree of a vertex can determine its relative weight in setting edge strength. We can prioritize hubs by setting...
where $k_i$ is the degree of vertex $i$. In this case, $p_{ij} = \frac{k_i}{k_i + k_j}$ and when $k_i > k_j$ the relative weight of vertex $i$ in determining $W'_{ij}$ is higher. Using this model, the optimal solution focuses on the reduction of weights of hubs. On the other hand, we can prioritize vertices with few neighbors by choosing

$$W'_{ij} = \left( \frac{k_i}{k_i + k_j} \nu_i + \frac{k_j}{k_i + k_j} \nu_j \right) A_{ij},$$

where $k_i$ and $k_j$ are the degrees of vertices $i$ and $j$, respectively.

Here, the reduction in weights of vertices is likely to be distributed among vertices with a low degree. These different strategies for selecting $p_{ij}$ are further discussed and compared in appendix E.

The optimization problem of vertices in equation (7) is convex in undirected networks. We have shown in section 2 that when $A$ is symmetric the problem in edges is convex. Here, $W'$ is an affine transformation of $\nu$ and affine transformations are operations which preserve convexity under composition.

In order to evaluate performance in the vertices reduction problem, we conducted similar tests to the edges reduction problem. We compared the simple strategy of equal contribution ($p_{ij} = \frac{1}{2}$, equation (5)) to the dynamical importance vertices solution. We observe substantial differences between the two algorithms (figure 6). The differences are more salient than in the case of edges, since vertex removal is a more severe action. In terms of runtime performance, our vertices algorithms are slower than the dynamical importance method in the studied networks, in contrast to the edges case. To solve the vertices problem in $L_2$, we also used Dykstra projections method, only this time instead of projecting on the set of edges that are non-negative we find the closest adjacency matrix that is the linear combination of bounded vertex weights (appendix K). For $L_1$, we used again the interior point algorithm. The gradient of the spectral constraint in the case of $W'$ is given in appendix C.

Similarly to the edge discrete solution, we use the ranking provided by the continuous-reduction of vertices to discretely remove vertices, using binary search. The results in vertices (figure 7) have the same trend as in the edge case. In weighted networks, our discrete Spitz $L_1$ algorithm achieved better or equal reductions when compared to the dynamical importance method in ten out of eleven networks. In unweighted networks, mixed results were found with small differences between the two algorithms.
4.5. Application of the algorithm in a real world scenario

We conclude the results section by a practical use of our algorithm which demonstrates the fundamental difference between the discrete greedy solution and the continuous solution. In recent years, flights facilitate and accelerate the spread of infectious diseases across distant regions. In this test case, we studied the network of air traffic in Australia through construction of a weighted network from flight routes data obtained from www.openflights.org. The vertices represent airports and the weighted edges represent flight frequency between airports (figure 8(a)). Using this weighted network, we search for the optimal change in flight frequency which is needed to stop an epidemic from spreading. Setting $\beta$ as half of the spectral radius of the original flight network, the dynamical importance and Spitz $L_2$ were applied to this network. The dynamical importance results in drastic full closure of the main airways between Australia’s five largest cities: Sydney, Melbourne, Brisbane, Perth and Adelaide (figure 8(b)). On the contrary, Spitz $L_2$ produces a milder solution that maintains the network routes by reducing the transportation frequencies (figure 8(c)). The Spitz $L_2$ solution does not shut down any of the major routes. Instead, it distributes small changes in flight frequency across the entire network. The change in the overall transportation volume by our algorithm (figure 8(d)) is also minimal; thus, the two solutions differ both quantitatively and qualitatively.

5. Discussion

The spectral radius is one of the fundamental properties of networks, merging dynamics and structure. In the dynamics of Markov processes, a crucial measure is the spectral radius of the stochastic transition matrix [40]. A special case that drew a large amount of interest is the contact process on networks and the spread of epidemics [41]. In SIR/SIS models, the threshold for the emergence of an epidemic is proportional to the spectral radius. Indeed de-facto reducing the spectral radius is one of the main methods to prevent the spread of epidemics [42].

The spectral radius of a network is not only determined by the presence or absence of edges, but by their weight. In models of contact process, weighted edges represent transmission rates, which vary among interactions. Neglecting the weights of edges leads to a uniform transmission rate, which in many cases is unrealistic. In these weighted networks, the continuous solution provides an essentially different alternative to quarantine strategies. Instead of completely preventing the transmission between vertices, the transmission rate is differentially decreased across the edges of the network. The continuous decrease in weights of edges is distributed over the network, thus preserving the overall network’s connectivity and function. The main
The difference between the Spitz approach developed here and existing algorithms is that other approaches [20–23] enforce a removal of full edges, while we allow for a soft solution. This solution is always better in terms of the minimal reduction than all full removal solutions.

The spectral radius is a macroscale measure that should be optimized at the full network level. The main mechanism proposed by other algorithms [20–23] is the iterative removal of edges according to a scoring function based on the contribution of edges (or vertices) to the spectral radius. Instead, we propose a formal optimization framework and prove that the change in the network is guaranteed to be the global minimum following the convexity of the domain and the target functions. In addition, the optimal weights of edges can be viewed as a centrality measure which depends on the targeted spectral radius. We plan to test in detail, what properties of edges are correlated with the optimal weights of edges.

We have compared the continuous Spitz performance with the dynamical importance algorithm, which proved to be the best contender in terms of the reduction in relation to applications of other centralities. When an $L_2$ objective function is used, we observed that Spitz substantially outperformed dynamical importance, both in weighted and unweighted networks. When an $L_1$ objective function is used, the result approximates an all-or-none solution, as is the case for other optimization problems [43]. In the weighted case, the performance of the continuous reduction is considerably better than the discrete solution of the dynamical importance algorithm (figures 3(a) and (c)). In unweighted networks, the Spitz $L_1$ solution is close to the solution of dynamical importance.

When optimizing over vertices instead of edges, we find that the Spitz algorithm significantly outperforms the dynamical importance method for both weighted and unweighted and for $L_1$ and $L_2$ criterions. In contrast to the edge problem, the vertices-Spitz $L_1$ algorithm significantly performs better also for unweighted networks. This boost in performance can be explained by the fact that reduction in vertices can always be viewed as weighted reduction. For example, when we reduce a vertex with 100 edges it induces 100 times more reduction to the overall network than the same reduction in a leaf-vertex and its single edge. In other words, our reduction

Figure 8. Analysis of different strategies for prevention of epidemic spread through Australia’s air travel network (A) Australia’s flights network, color bar represents flights frequency between destinations. Subplots B and C display the amount of transportation that was reduced from the network. (B) The dynamical importance algorithm closes flight routes between all major cities in Australia. (C) Our algorithm distributes small changes in frequency across the entire network and does not shut down any of the routes completely. (D) Comparison of the induced change in the network by the dynamical importance and the Spitz $L_2$ method over a range of targeted thresholds ($\beta$).
criterion, \[ \frac{\| A - W \|_{\infty}}{\| A \|_{\infty}} \], is in terms of edges and each vertex is intrinsically weighted by its degree. Thus, as in the edge reduction in weighted networks, when we tune the solution in a continuous manner, we observe enhanced performance.

Therefore, we found that in networks with high heterogeneity in the degree of nodes or in the weights of edges the optimal solution by Spitz is considerably better than the one provided by the Dynamical importance method. Given that real world networks are characterized by heavy tail degree distribution and wide weights distribution, partial removal using the Spitz method becomes even more crucial in the effort to limit the spread of contact processes.

Beyond the advantage of providing an optimal continuous solution, the global optimization also produces a ranking of the edges according to the fraction of their weight that should be removed. Using this ranking, we propose a solution to the discrete edge removal and vertices removal problems. Using the discrete extension, the solutions obtained using Spitz \( L_2 \) are typically as good in as the solutions from the dynamical importance method both for edges removal and for vertices removal. Even in the discrete Spitz \( L_2 \), the solution is often similar to the best existing discrete solutions.

The \( L_2 \) solution has two prominent advantages. First, in some scenarios the preferred course of action is to perform a decentralized change in a network which is characterized by small reductions across numerous edges. To demonstrate this, we examined a practical test case from the domain of disease propagation and studied the necessary change in the weighted network of flight routes in Australia. In this case, lowering the frequencies of flights is a preferred alternative to closing flight routes. The \( L_2 \) continuous objective function indeed distributes small changes in flight frequencies across the entire network and allows for the flow of flights to continue while maintaining connectivity of the entire network. Second, by using an adaptation of Dykstra’s algorithm in the Spitz \( L_2 \) method we are able to compute the optimal solution efficiently. This allows us to scale the problem to networks of hundreds of millions of edges, both for continuous and discrete problems. As a result, we can find the globally optimal network structure that achieves phase transitions even in modern large scale networks.

References

[1] Arenas A et al 2008 Synchronization in complex networks Phys. Rep. 469 93–153
[2] Barahona M and Pecora L M 2002 Synchronization in small-world systems Phys. Rev. Lett. 89 054101
[3] Barzel B and Barabási A-L 2013 Universality in network dynamics Nat. Phys. 9 673–81
[4] Castellano C and Pastor–Satorras R 2010 Thresholds for epidemic spreading in networks Phys. Rev. Lett. 105 218701
[5] Goltsev A V et al 2012 Localization and spreading of diseases in complex networks Phys. Rev. Lett. 109 128702
[6] Noh J D and Rieger H 2004 Random walks on complex networks Phys. Rev. Lett. 92 118701
[7] Newman M 2010 Networks: An Introduction, (Oxford: Oxford University Press)
[8] Van Mieghem P, Omic J and Kooij R 2009 Virus spread in networks IEEE/ACM Trans. Netw. 17 1–14
[9] Wang Y et al 2003 Epidemic spreading in real networks: an eigenvalue viewpoint 2003. Proc. 22nd Int. Symp. on Reliable Distributed Systems (Piscataway, NJ: IEEE)
[10] Restrepo J G, Ott E and Hunt B R 2005 Onset of synchronization in large networks of coupled oscillators Phys. Rev. E 71 036151
[11] Restrepo J G, Ott E and Hunt B R 2008 Weighted percolation on directed networks Phys. Rev. Lett. 100 058701
[12] MacCuer C R 2000 The many proofs and applications of Perron’s theorem SIAM Rev. 42 487–98
[13] Grabow C, Grosskinsky S and Timme M 2011 Speed of complex network synchronization Eur. Phys. J. B 84 613–26
[14] Grabow C et al 2010 Do small worlds synchronize fastest? Europhys. Lett. 90 48002
[15] Grabow C, Grosskinsky S and Timme M 2012 Small-world network spectra in mean-field theory Phys. Rev. Lett. 108 218701
[16] Liu H et al 2015 Laplacian spectra of a class of small-world networks and their applications Sci. Rep. 5 9024
[17] Hu C et al 2015 A spectral graph regression model for learning brain connectivity of Alzheimer’s disease PloS One 10 e0128136
[18] Albert R, Jeong H and Barabási A-L 2000 Error and attack tolerance of complex networks Nature 406 378–82
[19] Cohen R et al 2001 Breakdown of the Internet under intentional attack Phys. Rev. Lett. 86 3682
[20] Restrepo J G, Ott E and Hunt B R 2006 Characterizing the dynamical importance of network nodes and links Phys. Rev. Lett. 97 094102
[21] Milanese A, Sun J and Nishikawa T 2010 Approximating spectral impact of structural perturbations in large networks Phys. Rev. E 81 046112
[22] Tong H et al 2012 Gelling, and melting, large graphs by edge manipulation Proc. 11st ACM Int. Conf. on Information and Knowledge Management (New York: ACM)
[23] Van Mieghem P et al 2011 Decreasing the spectral radius of a graph by link removals Phys. Rev. E 84 016101
[24] Odor G 2013 Spectral analysis and slow spreading dynamics on complex networks Phys. Rev. E 88 032109
[25] Balcan D et al 2009 Multiscale mobility networks and the spatial spreading of infectious diseases Proc. Natl Acad. Sci. 106 21484–9
[26] Bogoch I I et al 2015 Assessment of the potential for international dissemination of Ebola virus via commercial air travel during the 2014 west African outbreak Lancet 385 29–35
[27] Colizza V et al 2007 Modeling the worldwide spread of pandemic influenza: baseline case and containment interventions PLoS Med. 4 95
[28] Mangili A and Gendreau M A 2005 Transmission of infectious diseases during commercial air travel Lancet 365 989–96
[29] Buzsáki G and Draguhn A 2004 Neural oscillations in cortical networks Science 304 1926–9
[30] Chakravarthy N et al 2007 Controlling synchronization in a neuron-level population model Int. J. Neural Syst. 17 123–38
[31] Schindler K A et al 2008 Evolving functional network properties and synchronizability during human epileptic seizures Chaos 18 033119
[32] Santienello S et al 2011 Quickest detection of drug-resistant seizures: an optimal control approach Epilepsy Behav. 22 S49–60
[33] Byrd R H, Hribar M E and Nocedal J 1999 An interior point algorithm for large-scale nonlinear programming SIAM J. Optim. 9 877–900
[34] Nocedal J 1980 Updating quasi-Newton matrices with limited storage Math. Comput. 35 773–82
[35] Boyle J P and Dykstra R L 1986 A method for finding projections onto the intersection of convex sets in Hilbert spaces Advances in Order Restricted Statistical Inference (Berlin: Springer) pp 28–47
[36] Barabási A-L and Albert R 1999 Emergence of scaling in random networks Science 286 509–12
[37] Erdős P and Rényi A 1960 On the evolution of random graphs Publ. Math. Inst. Hung. Acad. Sci. 5 17–61
[38] Kunegis J 2013 Konect: the koblenz network collection Proc. 22nd Int. Conf. on World Wide Web Companion (International World Wide Web Conferences Steering Committee)
[39] Golub G H and Van Loan C F 2012 Matrix Computations vol 3 (Baltimore, MD: Johns Hopkins University Press) pp 21218–4363
[40] Motter A E, Zhou C and Kurths J 2005 Network synchronization, diffusion, and the paradox of heterogeneity Phys. Rev. E 71 016116
[41] Van den Driessche P and Watmough J 2002 Reproduction numbers and sub-threshold endemic equilibria for compartmental models of disease transmission Math. Biosci. 180 29–48
[42] Hethcote H W and Van Ark J W 2013 Modeling HIV Transmission and AIDS in the United States vol 93 (Dordrecht, Netherlands: Springer)
[43] Tibshirani R 1996 Regression shrinkage and selection via the lasso J. R. Stat. Soc. B 58 267–88