Communication Network Design: Balancing Modularity and Mixing via Extremal Graph Spectra

BEN LUBIN,[a] JESSE SHORE[a], AND VATCHE ISHAKIAN[b]

[a]Boston University School of Management

[b]BBN

July 22, 2013

Abstract

By leveraging information technologies, organizations now have the ability to design their communication networks and crowdsourcing platforms to pursue various performance goals. However, research on network *design* has not incorporated any notion of teams, which are known to have many performance benefits in problem-solving networks, instead focusing only on designing networks for fast diffusion of information. Here, we fill this gap by providing a design framework and methodology that incorporates both modularity and mixing time. We take advantage of prior literature in the area of spectral graph theory and demonstrate how desirable aspects of organizational structure can be mapped parsimoniously onto the spectrum of the graph Laplacian derived from a matrix representation of that communication structure. We rely on recent advances in convex optimization to extremize our objective, defined in terms of elements of the graph spectra. Finally, we also present and discuss the resulting communications structures that balance modularity and mixing time.

1 Introduction

By leveraging modern information and communications technology, there is now the opportunity for organizations to go beyond *understanding* their networks to *designing* their networks. Much research has contributed to our understanding of how networks shape knowledge management, knowledge sharing, coordination, diffusion of technology and best practices, innovation, the success of IT-mediated collaboration, and overall performance (Bavelas 1950, McCubbins et al. 2009, Mason and Watts 2012, March 1991, Huang and Cummings 2011, Cummings and Cross 2003, Mason et al. 2008, McEvily and Marcus 2005, Sparrowe et al. 2001, Suri and Watts 2011, Lazer and Friedman 2007, Aral and Van Alstyne 2011, Bae et al. 2011, Borgatti and Cross 2003, Capaldo 2007, Reagans and Zuckerman 2001, Sundararajan et al. 2012). Moreover, with growing awareness of the importance of crowdsourcing and external innovation and human computation (Lakhani and Jeppesen 2007, Zheng et al. 2011), recent research has examined the effect of network structure (Kearns 2012) and group size (Boudreau et al. 2011) on performance in these platforms.

Unfortunately, however, there is little research specifically addressing the design of networks of communicating human beings. Rather, the design literature has focused on problems of minimal or optimally "efficient" networks, with applications in non-human infrastructure networks (Balakrishnan et al. 1989, Magnanti and Wong 1984, Minoux 1989, Guimerà et al. 2002, Donetti et al. 2005, Dionne and Florian 1979, Winter 1987, Kershenbaum et al. 1991, Lubotzky et al. 1988, Estrada 2007). The work of Lovejoy and Sinha (2010) is a notable exception in that it is concerned with social networks within organizations, but it is similar in its orientation toward efficiency and short paths between any given individuals in the network. There is indeed substantial theoretical justification for pursuing short paths as a design criterion in human as well as infrastructural networks that is generally understood in terms of two related ideas: that weak-ties enable rapid diffusion of information (Watts and Strogatz, 1998) and that bridging structural holes can be associated with innovation (Burt, 2004).

Although these are important issues, there are also advantages to modularity – having teams or groups that are separate but internally cohesive clusters in organizations – but this has to our knowledge been omitted as a network design criterion. Within organizations, internally cohesive groups tend to use similar language constructs, which enables high-bandwidth communication (Aral and Van Alstyne 2011) and increases their effectiveness (Hansen 1999, Reagans and Zuckerman 2001). Shore et al. (2013) show experimentally that clustering is beneficial for solving problems that require extensive information-space searching and/or coordination. Additionally, certain types of information and behaviors spread more easily within rather than between clusters (Centola

2010). Finally, real organizations are usually structured in divisions, work groups, or teams — lending an added importance to incorporating some notion of modularity into network design work. Despite all of this, the design literature has yet to address network contexts in which modularity is desirable.

Two major issues may have stood in the way of incorporating modularity into design work. First, obtaining modularity and short path lengths imply quite different network structures, making theoretical analysis that encompasses both properties difficult. Second, the space of all possible networks is combinatorially large, making the design problem formidably complex (for example, the number of possible undirected graphs with 16 nodes is 2^{120}, or approximately 1.3×10^{36} — far too many to evaluate individually by any known means). Here, we propose a design framework that addresses both issues simultaneously: we frame the network design problem in a way that lets the designer tradeoff between modularity and mixing time, and we propose an an algorithm that can find extremal graphs under these criteria. Specifically, for the design framework, we take advantage of prior literature in the area of spectral graph theory and demonstrate how desirable aspects of organizational structure can be mapped parsimoniously onto the spectrum of the graph Laplacian derived from a matrix representation of that communication structure. Recent advances in convex and non-convex optimization allow us to capture these spectral elements in an objective function to be extremized. Finally, we present examples of the communications structures produced under this method that balance modularity and mixing time and discuss the implications of their properties.

2 Spectral Theory Informs Design

Spectral graph theory (Cvetkovic and Sachs, 1998; Chung, 1997) is concerned with the relationships between the structure of a network and the eigenvalues, also called the "spectrum," of the matrix representation of the network. One major advantage of thinking of networks in terms of their spectra is that spectra are insensitive to permutations and labeling. All networks with the same structure have the same spectrum.This property lets us avoid having to deal with the so-called "graph isomorphism problem," where many equivalent representations for structurally isomorphic graphs exist, making search and classification in graph space difficult. In essence, working with the spectra lets us focus on a more tractable and compact object, and one which corresponds to a unique graph with high probability (see section 2.3). Moreover, the values of the spectra provide enormously useful information about graph structure in a compact and accessible way. These properties make spectra the ideal mathematical objects to use in formalizing desiderata and constraints in network design problems.

In this paper, we adopt a particular design objective: we aim to design networks that both manifest distinct subgroups and yet are still "sufficiently connected". As we have seen in the previous section, these are well motivated goals. However, it is not obvious how to formalize them. Spectral theory gives us a means to frame this precisely. Existing work has not not examined such an objective; we provide:

- A spectral formalization of our modularity and mixing objective (section 2.2)
- A novel optimization problem based on this formulation that captures our design objective (section 3)
- An algorithm for approximately solving this problem (section 3.1)
- A set of numerical experiments based on this algorithm, and their results and interpretation (sections 4 and 5).

2.1 Preliminaries

The standard matrix representation of a graph, where each entry represents the strength of the connection between the node indexed by the matrix row and column, is called the *adjacency* matrix. In this paper, we assume that each individual in the organization has equal communications capacity that they use fully. This implies that our matrix representations of the network must have rows and columns that can be normalized so that they all sum to 1 (such matrices are called "doubly stochastic"). Further, we assume that a given communication tie takes the same proportion of each connected individual's communication capacity. Together these properties imply that the matrix representation of the network must be symmetric about its diagonal.

Instead of working with the adjacency matrix, it can be useful to work with the graph *Laplacian* matrix given, for stochastic graphs, by $L = I - A$, where I is the identity matrix and A the adjacency matrix.[1] The spectrum of A and L are related but have distinct properties; those of the Laplacian match our needs and we consequently adopt it here. The matrix spectrum is simply the multiset of eigenvalues, sorted in decreasing order of magnitude.[2] Such a spectrum can be plotted as a set of points, as illustrated in figure 1 and elaborated upon below.

[1]In general the *Laplacian* is given by $L = D - A$, where D is the degree matrix, con-

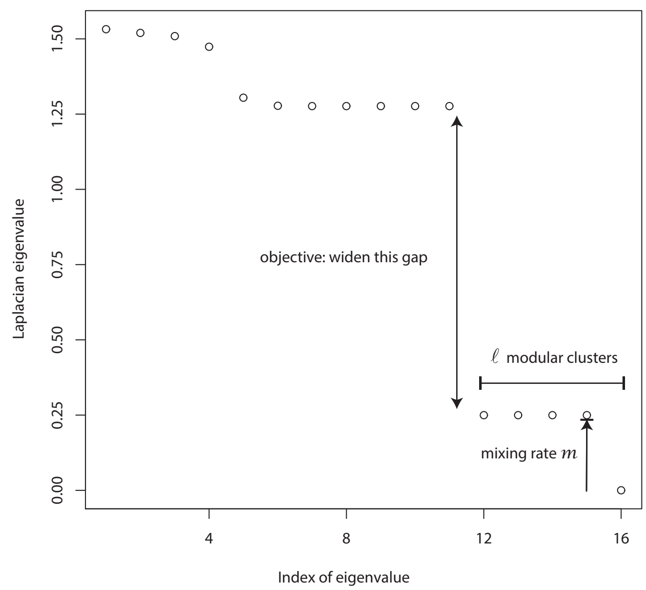

Figure 1: Illustration of the spectral framework, including objective and constraints

2.2 The Laplacian Spectrum and Network Structure

The relative magnitude of the various spectral values correspond to specific structural properties of the corresponding network. We describe those necessary for capturing our design objective below.

2.2.1 Bounding the mixing time with m

The magnitude of the smallest Laplacian eigenvalue (hereafter, just "eigenvalue" for brevity) is always zero, and therefore of little immediate interest. However, the magnitude of the second smallest eigenvalue is also the graph's "algebraic connectivity" (Fiedler 1973) and is inversely related to the mixing time for Markov chains (Mohar 1997). In short, the larger the second smallest eigenvalue, the faster we expect information to diffuse through the network (Donetti et al. 2006). Because of its known connection to mixing time, we refer to the magnitude of the second smallest eigenvalue as m (see Figure 1). By tuning m a network

structed by putting the row sums of A on the diagonal, with zeros elsewhere.

[2]The eigenvalues of a matrix M are given by $\{\lambda | M\boldsymbol{v} = \lambda\boldsymbol{v}, \boldsymbol{v} \neq 0\}$. The $\boldsymbol{v}$ are called the eigenvectors of the matrix: those vectors that when multiplied by the matrix yield a scaled copy of themselves. Each scale factor is a corresponding eigenvalue, λ.

designer has a spectral method for formalizing the idea of "sufficiently connected:" the larger the m, the more rapidly that communication structure is expected to diffuse information. However, raising m may come at the cost of other desirable features, such as the amount of modularity that is manifest in the network, as we shall see shortly.

2.2.2 Setting the number of modular clusters with ℓ

It is well known that the number of connected components of an undirected graph is equal to the number eigenvalues of the Laplacian that are equal to zero (Brouwer and Haemers 2011). For example, if there were four totally disconnected components, there would be four eigenvalues equal to zero. If, however, there existed weak connections among those distinct communities such that they are no longer disconnected components but rather modular clusters, then rather than having one zero for each cluster, we would have one small eigenvalue for each module (Donetti et al. 2006). Consequently, for a graph consisting of four modular clusters that are weakly connected to each other, the spectrum of the Laplacian (hereafter "spectrum") would contain four small eigenvalues, one of which would be zero (as there would be one component, and thus one eigenvalue equal to zero).

From the design point of view, then, we observe that if one desires a communication network with some number, ℓ, distinct modular clusters, then one should construct a graph with a spectrum containing ℓ small eigenvalues, one of which is zero (see Figure 1).

2.2.3 The rest of the spectrum

We have just argued that we want $\lambda_k, k \leq \ell$ to be small. But small relative to what? To make λ_ℓ relatively small, we need $\lambda_{\ell+1}$ to be large, and this in turn will drive up all $\lambda_k, \ell < k \leq n$, giving us a graph that is as modular as possible. A theorem provided by Newman and Kel'mans enables us to interpret this more clearly (Newman 2000):

$$\lambda_k(G^C) = n - \lambda_{n+2-k}(G) \text{ for } 2 \leq k \leq n \tag{1}$$

Where G is a graph and G^C is its complement. This theorem provides that the kth largest eigenvalue is equivalent to the $k-1$ smallest eigenvalue of the complementary graph. So by driving the large eigenvalues up, we are driving down the small eigenvalues of the complementary graph. This makes the complementary graph have a long mixing time (via $\lambda_2(G^C) = \lambda_n(G)$), and more broadly have a large number $(n-\ell)$ of largely disconnected modules. When n is large relative to ℓ this implies a largely disconnected complementary graph, and therefore a (primary) graph that is highly connected, given its modular structure.

2.3 Co-spectral graphs

It is one thing to calculate the spectrum of a known graph and quite another to construct a graph with a given spectrum. Since we are using spectral properties to design networks with desirable structural properties, we are more concerned with the latter problem. The next section details our method for constructing matrices with desirable spectral properties. Before we do so, however, we must take note of the issue of co-spectral graphs, or graphs with the same spectrum (Harary et al. 1971, Godsil and McKay 1982).

Although at present relatively little is known about which graphs have co-spectral partners (Van Dam and Haemers 2003), we do not believe this presents a substantial impediment to the present undertaking. Most fundamentally, we are presenting a framework for designing communication networks with properties that have spectral correlates. If by chance we construct a graph for which there exists a co-spectral partner that we do not find, we have still achieved our design goal, because co-spectral graphs have similar structure with respect to the features captured by that spectrum.

Additionally, but less essentially, enumerations of unweighted graphs that are co-spectral with respect to their Laplacian (Haemers and Spence 2004, Brouwer and Spence 2009, Cvetković 2012) show that the proportion of graphs with co-spectral partners is highest at $n = 9$ and decreases as n and the number of edges increase. Halbeisen and Hungerbühler (2000) show that for weighted graphs — which we employ here — there are almost surely no co-spectral partners. Therefore, we assert that by constructing weighted networks according to spectral parameters, we are not leaving anything important to our aims on the table.

3 Methods

Spectral theory has given us the means to formalize both of our design objectives:

- Sufficient connectivity, by imposing a lower bound, m, on the second smallest eigenvalue λ_2, which ensures a fast enough mixing time.
- Modularity with ℓ clusters, by having ℓ small eigenvalues and $n-\ell$ large eigenvalues.

Our network design problem can then be cast the following non-linear optimization problem:

$$\max_{\boldsymbol{W}} \quad \lambda_{\ell+1}(\boldsymbol{W}) - \lambda_{\ell}(\boldsymbol{W}) \tag{2}$$

$$\text{s.t.} \quad \lambda_2 \geq m \tag{3}$$

$$\sum_j \boldsymbol{W}_{ij} = 1 \; \forall \; i \tag{4}$$

$$\boldsymbol{W}_{ij} = \boldsymbol{W}_{ji} \; \forall \; i, j \tag{5}$$

The objective, equation 2, maximizes the difference between the $\ell + 1$ and ℓ Laplacian eigenvalue. The constraint 3 ensures that the mixing time is at least m. Constraints 4 and 5 ensure stochasticity and symmetry respectively. Note that the variables in this formulation are the weights of matrix $\boldsymbol{W}$.

3.1 Optimization Algorithm

The "eigenvalue problem," that of computing the eigenvalues for a known matrix, can be calculated in closed form for small matrices, and for large matrices by numerical algorithms, e.g. QR, that have been known since the early sixties (Francis 1961, 1962). However, "inverse eigenvalue problems," those of finding the graph that corresponds to a specific spectrum or specific spectral characteristics have proven vastly harder to solve (Chu 1998). Most such problems admit no computationally tractable algorithm for obtaining a globally optimal solution.

Our formulation falls within this hard class, and thus the best we can hope for is a high-quality approximation algorithm. We are not aware of any existing work that has looked at solving our particular spectral objective and constraints. We have therefore constructed our own approximation method by leveraging recent advances in Semi-Definite Programming (SDP) and Difference in Convex (DC) programming, which we next describe.

3.1.1 Semi-Definite Programming

Semi-Definite Programming (SDP) is a type of convex optimization that operates over a matrix variable, instead of the scalar variables seen in other convex optimization methods (Vandenberghe and Boyd 1996). SDP objectives are specified as the inner-product of the matrix variable, with a user-specified constant matrix. Similarly, SDP constraints consist of a bound on the inner-product between the matrix variable and another user-specified constant matrix. The minimal value for the objective is found, where the matrix variable is drawn from the cone of semi-definite matrices. Many problems can be cast into this structure, and because the resulting formulation is convex, it can be solved efficiently by e.g. interior point methods (Todd 2001, Wolkowicz et al. 2000, Alizadeh 1995).

For the present work, the key property of SDP is its ability to capture the sum of the k smallest Laplacian eigenvalues, S_k, as a concave function, the maximization of which is a convex optimization. Boyd et. al. have used this capability to solve certain Laplacian inverse eigenvalue problems directly (Boyd et al. 2004, Boyd 2006). For example, they formulate S_2 as a concave function which they can then maximize via SDP to efficiently solve for the Markov process with the fastest mixing time. We leverage their result by moving their objective formulation to a constraint, obtaining a convex form for equation 3. Further, as the remaining constraints are linear, only our objective 2 fails to be directly representable as an SDP, which we address next.

We start by noting that $\lambda_\ell = S_\ell - S_{\ell-1}$ and $\lambda_{\ell+1} = S_{\ell+1} - S_\ell$. And thus our objective in equation 2 can be rewritten as:

$$\lambda_{\ell+1} - \lambda_\ell = (S_{\ell+1} - S_\ell) - (S_\ell - S_{\ell-1}) = S_{\ell-1} + S_{\ell+1} - 2S_\ell \quad (6)$$

This objective captures our intent, and can be formulated by known SDP-style expressions. However, it can not be directly solved because, as a maximization, the third term is non-convex.

3.1.2 Difference in Convex Programming

As we have seen, the formulation of equation 2 given in equation 6, is almost convex and solvable as and SDP, but not quite. Consequently, and as expected, we are not going to be able to directly use convex optimization, and the best we can hope for is an approximately optimal algorithm. However, equation 6 is a *difference of convex functions* and, as such, is amenable to an algorithm known as the Concave-Convex Procedure (Yuille et al. 2002, Yuille and Rangarajan 2003). This is an iterative method for obtaining approximate solutions to problems with convex and concave components in the objective that has good convergence properties (Sriperumbudur and Lanckriet 2009). Our approach is to implement the Concave-Convex Procedure over our SDP formulation[3]. Our approach is as follows:

[3]The Concave-Convex Procedure has generally be used for simpler optimization formalisms in the literature, here we adapt it to the more expressive SDP context.

We start with a random initial graph $\widehat{\boldsymbol{W}}$. We then form a first-order Taylor expansion of the concave portion of the objective around $\widehat{\boldsymbol{W}}$. Using this linear form, we can then approximate the objective as:

$$S_{\ell-1}(\boldsymbol{W}) + S_{\ell+1}(\boldsymbol{W}) - 2\left(S_\ell(\widehat{\boldsymbol{W}}) + (\boldsymbol{W} - \widehat{\boldsymbol{W}})S'_\ell(\widehat{\boldsymbol{W}})\right) \tag{7}$$

This then, is directly solvable as an SDP, which we obtain using the CVX package (Grant and Boyd 2012, 2008). We then set $\widehat{\boldsymbol{W}} \leftarrow \boldsymbol{W}$ and repeat until convergence.

3.2 Bounding the Objective Value

Because our optimization algorithms may be only locally optimal, it is useful to have a theoretical upper bound on the objective value in equation 2. When the objective value of the solution found by our numerical calculations approach the bound, we have found an approximately optimal graph[4] Accordingly, we can take advantage of the following:

Theorem 1. $\frac{n-m(\ell-1)}{n-\ell} - m$ *gives an upper bound on the non-convex objective in equation 2.*

Proof. $\lambda_1 = 0$ always and $\lambda_k \geq m, 2 \leq k \leq \ell$ by constraint 3, a lower bound on each of these eigenvalues. This implies $\sum_{1\leq k\leq \ell} \lambda_k \geq m(\ell-1)$. There is a known result that $\sum_k \lambda_k \leq n$ (Chung 1996). Subtracting the first from the second yields $\sum_{\ell+1\leq k\leq n} \lambda_k \leq n - m(\ell-1)$, an upper bound on the large eigenvalues. The smallest of these, $\lambda_{\ell+1}$, is made maximal at this bound and when these eigenvalues are of equal size, giving it a value of $\frac{n-m(\ell-1)}{\ell-n}$. Subtracting our upper bound on $\lambda_{\ell+1}$ from our lower bound on λ_ℓ gives an overall objective upper bound of: $\frac{n-m(\ell-1)}{n-\ell} - m$. □

4 Experiment and Results

We next describe several experiments we have conducted to find approximately optimal graphs according to our spectral design framework.

4.1 Properties of Spectrally Designed Communication Networks

Figure 2 shows two examples of networks produced by our framework, with the weakest ties omitted for clarity. Several features are imme-

[4]However, the converse does not necessarily follow: solutions far from the bound may still be near-optimal when the bound is loose.

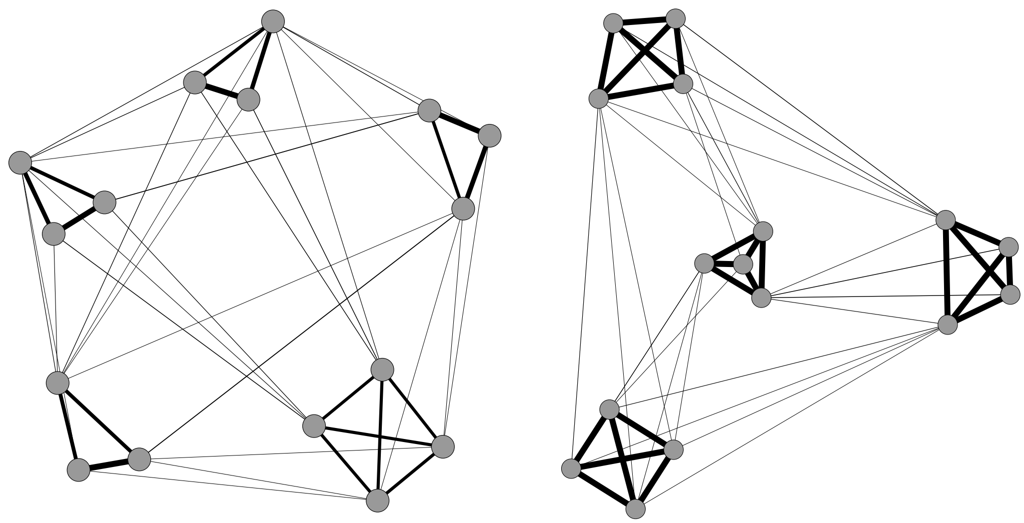

Figure 2: Two examples of sixteen person communication networks produced by the spectral design framework. The left hand network has $\ell = 5$ and the right hand $\ell = 4$ teams, both with mixing rate $m = 0.25$

diately apparent. As expected, these networks have a clear modular structure, with strong intra-team connections. Additionally, there are weak ties connecting the teams in patterns that appear in the visualization as "fans." Intuitively, one could think of these fans as ties from one representative of a team to (usually) all the members of another team — more of a "liaison" than a broker. Although such an organization structure is reminiscent of the "matrix organization" (Galbraith 1971), we are not aware any appearance of similar graphs in the literature on network structure *per se*.

The disposition of the inter-clique liaisons has a definite structure, suggestive of a hierarchical "spiral" in the visualizations. In the right hand side of Figure 2, the central team has three "outgoing" liaisons; the team to the right has two; the team at the top has one; and the team at the bottom has none. A minority of the weak ties are not part of a liaison's fan structure: in this network, the most visually central individual has singleton ties to two individuals in other teams.

Results for 32-person networks are similar to those for 16-person networks. Figure 3 shows a network comparable to the 16-person network on the right-hand side of Figure 2, displaying the same hierarchical spiraling structure.

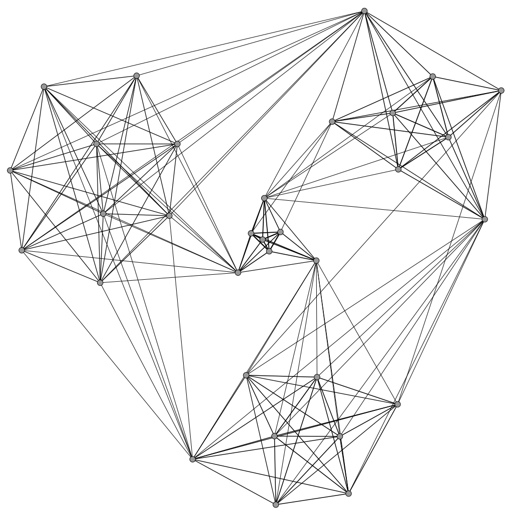

Figure 3: A 32-person network with $\ell = 4$ and $m = 0.15$

4.2 Optimality of results

Table 1 gives the best objective values we achieved, divided by the theoretical upper bound calculated for each set of parameter values, as described in section 3.2. Each data point represents the best of 2000 independent random starting points of our algorithm, obtained by running each in parallel on a 1000 node computational cluster. Each such pass through the algorithm generally completes in less then 2 hours of CPU time on modern Xeon-class hardware. From the table we see that our algorithm is finding answers that are very near our bound in most of the cases. Where some gap remains, it is unknown if this is due to the bound being loose, or the algorithm failing to find a sufficiently global optimum.

4.3 Spectral impact of inter-clique connection types

In order to understand how the liaison pattern of ties produced by the algorithm differs from other possible configurations of ties, we plot the spectra of similar graphs for inspection below. First, we compare the truncated result in 2 to the full algorithm output to assess the impact of such simplification. Second, we compare the liaison structure to two hand-constructed matrices that use single “broker”-type ties between cliques.

The full output is doubly stochastic, but the truncated output and

Table 1: Degree of optimality achieved

n	ℓ	m	optimality
16	6	0.15	0.938
16	6	0.20	0.938
16	6	0.25	0.938
16	5	0.15	0.917
16	5	0.20	0.919
16	5	0.25	0.922
16	4	0.15	1.000
16	4	0.20	1.000
16	4	0.25	0.999
32	8	0.15	0.937
32	8	0.20	0.939
32	6	0.20	0.975
32	4	0.15	0.984
32	4	0.20	0.978

hand-constructed networks are not necessarily so. In order to make the latter networks doubly-stochastic such that their spectra are comparable to the full result, we re-normalized the weights of the edges by iteratively row-normalizing then averaging the resulting matrix with its transpose until the matrix is doubly-stochastic.

4.3.1 Full versus truncated result

As noted above, the visualizations in Figure 2 truncate the algorithm output such that the very weakest links are not drawn. In addition to making the structure of the networks more apparent to the eye, simplified versions of the full result would certainly be easier to implement in practice than the full result. The two left-hand networks in Figure 5 visualize the difference between the full and truncated results.

Figure 4 shows the effect of such truncation on the spectrum. The full solution is essentially optimal; truncation produces a slightly sub-optimal spectrum, but the deviation is relatively minor. The right-hand side of the truncated spectrum shows that the mixing rate is lower than the full result, and the left hand side shows slight deviation from maximum modularity. In sum, although truncation moves us a step from the theoretical optimum, is is a small step and a more pragmatic alternative for situations in which implementation is important.

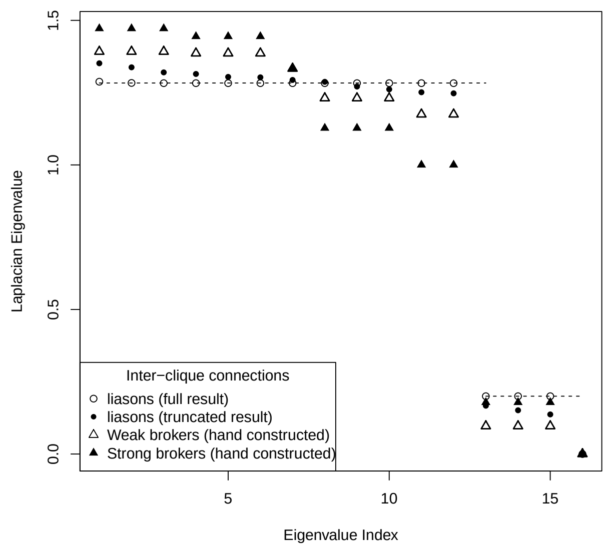

Figure 4: Comparison of spectra of related graphs. NB for the truncated version of the results and the hand-constructed matrices, weights were renormalized for double-stochasticity. The dotted line represents the optimal spectrum as we define it here.

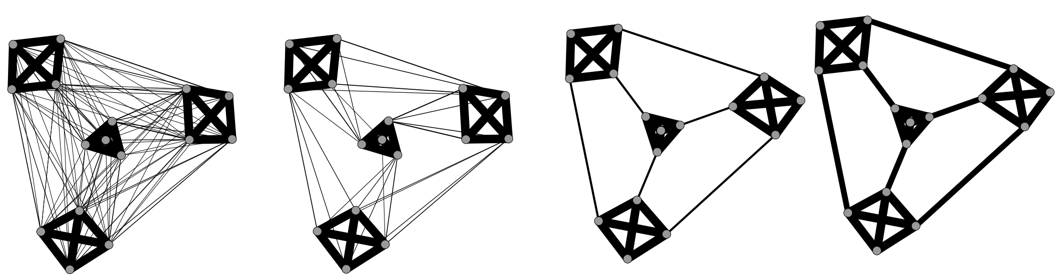

Figure 5: The four networks compared in Figure 4. From left: the full result of the optimization algorithm, the result truncated such that the weakest links are omitted, a hand-constructed network with brokers with single ties between cliques with weight equal to the sum of the weights of the fans, a hand constructed network with stronger ties between cliques.

4.3.2 Liaisons versus brokers

Literature on social networks has tended to focus on the role of brokers in organizations (e.g. Burt 2001), rather than the liaisons we describe here. In order to compare networks connected by brokers, rather than liaisons, we hand-constructed two such networks. In both cases, we replaced inter-clique liaisons with brokers. In the first (third from the left in Figure 5 and the open triangles in Figure 4), we set the weight of the inter-clique ties to be equal to the sum of the weights in a liaison's fan of ties. In the second (right-hand side of Figure 5 and the filled triangles in Figure 4), we set all ties to equal weight before normalizing for double-stochasticity, resulting in stronger brokerage ties.

We find that the network with stronger brokers has a similar mixing time to the result of our algorithm, but it is much farther from optimum modularity than our result. The network with weaker brokers (equal to the weight of the liaison's ties) has slower mixing time than our result, but it is closer to maximum modularity than the network with strong brokers.

Overall, for given rate of mixing, brokers produce a less modular network than liaisons. Alternatively, for a certain amount of weight on inter-clique ties, liaisons achieve a faster mixing rate than brokers.

5 Discussion

5.1 Communication Network design

Information systems are increasingly used to accomplish strategic goals; for knowledge-based organizations and crowdsourcing platforms, information system structure is critical. In such knowledge-intensive work, social network structure is known to be a major driver of outcomes from extensive research that has described the effects of structural features on performance. Given this, what network structure should an organization adopt to maximize its performance in knowledge-intensive work?

Description does not suffice for prescription. Here, we close the gap by developing a design framework for finding networks that maximize modularity — the "team-ness" of a network structure — given a certain desired speed of information diffusion. By adopting a design mentality, we find novel structures that meet our criteria, and also specify a framework that can be extended to specify different structural objectives.

In taking this approach, we create networks with novel structural features. In particular, what we call "liaisons" — individuals with strong connections in one team and weaker connections to multiple (usually all) members of another team — allow inter-team connectivity, while maintaining a high degree of modularity.

As for the global structure of these networks, the hierarchical "spirals" also permit speculative interpretation along these lines. On the right side of Figure 2, each team has a different combination of outgoing and incoming fans. The central team sends representatives to each of the other three divisions of the organization; this could be the leadership team, for example. At the bottom of the figure is a team that receives representatives from all the other divisions, suggesting a function depended on by all: perhaps an infrastructure or operations team.

These communication structures are finely articulated, but this need not present a barrier to implementation on a computer mediated communication platform. One plausible implementation of tie strength would be as a fraction of the problem solving time spent "together," with the opportunity to exchange ideas or observe the progress of others. On such a platform it would also be easy to tune the importance of the weaker ties to increase either the speed of information diffusion (with stronger ties), or the importance of separate teams (with weaker ties), even over time to respond to the collective progress within the network.

5.2 Future Research

By specifying a spectral interpretation of network properties relevant to the design of communication networks, we hope we have opened doors for future design research. Future computational experiments could extend this framework to networks of individuals with different communications capacities, thereby dropping the requirement that networks be representable by doubly-stochastic graphs. Additionally, by specifying more complex constraints, for example, by including more than one "step" in the spectrum, networks with multiple levels of structure could be generated.

Undoubtedly, much would be learned if these results were tested experimentally with human problem solvers to assess their performance in crowdsourcing tasks and knowledge management settings. These results could in turn inform future improvements to design methodology.

5.3 Conclusion

Beyond describing networks, many practical settings call for network design. For networks of humans, the benefits of network modularity have been been well documented in research on networked problem solving, above and beyond the well-explored benefits of short average path lengths between all members of an organization. However, prior work on network design has not incorporated these insights. Our contribution has been to fill this gap, drawing connections between research on networked problem solving, spectral graph theory, and combinatorial optimization to both construct a design methodology and use that methodology to generate novel structures for communication networks.

References

Alizadeh, Farid. 1995. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization* **5**(1) 13–51.

Aral, Sinan, Marshall Van Alstyne. 2011. The diversity-bandwidth trade-off1. *American Journal of Sociology* **117**(1) 90–171.

Bae, Jonghoon, Filippo Carlo Wezel, Jun Koo. 2011. Cross-cutting ties, organizational density, and new firm formation in the us biotech industry, 1994–98. *Academy of Management Journal* **54**(2) 295–311.

Balakrishnan, Anantaram, Thomas L Magnanti, Richard T Wong. 1989. A dual-ascent procedure for large-scale uncapacitated network design. *Operations Research* **37**(5) 716–740.

Bavelas, Alex. 1950. Communication patterns in task-oriented groups. *The Journal of the Acoustical Society of America* **22**(6) 725–730.

Borgatti, Stephen P, Rob Cross. 2003. A relational view of information seeking and learning in social networks. *Management science* **49**(4) 432–445.

Boudreau, Kevin J, Nicola Lacetera, Karim R Lakhani. 2011. Incentives and problem uncertainty in innovation contests: An empirical analysis. *Management Science* **57**(5) 843–863.

Boyd, Stephen. 2006. Convex optimization of graph laplacian eigenvalues. *Proceedings oh the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures*. 1311–1320.

Boyd, Stephen, Persi Diaconis, Lin Xiao. 2004. Fastest mixing markov chain on a graph. *SIAM review* **46**(4) 667–689.

Brouwer, Andries E, Willem H Haemers. 2011. *Spectra of graphs*. Springer.

Brouwer, Andries E, Edward Spence. 2009. Cospectral graphs on 12 vertices. *the electronic journal of combinatorics* **16**(20) 1.

Burt, Ronald S. 2001. Structural holes versus network closure as social capital. *Social capital: Theory and research* 31–56.

Capaldo, Antonio. 2007. Network structure and innovation: The leveraging of a dual network as a distinctive relational capability. *Strategic Management Journal* **28**(6) 585–608.

Centola, Damon. 2010. The spread of behavior in an online social network experiment. *science* **329**(5996) 1194–1197.

Chu, Moody T. 1998. Inverse eigenvalue problems. *SIAM review* **40**(1) 1–39.

Chung, Fan RK. 1996. Lectures on spectral graph theory. *CBMS Lectures, Fresno* .

Cummings, Jonathon N, Rob Cross. 2003. Structural properties of work groups and their consequences for performance. *Social Networks* **25**(3) 197–210.

Cvetković, Dragoš. 2012. Spectral recognition of graphs. *The Yugoslav Journal of Operations Research* **22**(2).

Dionne, René, Michael Florian. 1979. Exact and approximate algorithms for optimal network design. *Networks* **9**(1) 37–59.

Donetti, Luca, Pablo I Hurtado, Miguel A Munoz. 2005. Entangled networks, synchronization, and optimal network topology. *Physical Review Letters* **95**(18) 188701.

Donetti, Luca, Franco Neri, Miguel A Muñoz. 2006. Optimal network topologies: Expanders, cages, ramanujan graphs, entangled networks and all that. *Journal of Statistical Mechanics: Theory and Experiment* **2006**(08) P08007.

Estrada, Ernesto. 2007. Graphs (networks) with golden spectral ratio. *Chaos, Solitons & Fractals* **33**(4) 1168–1182.

Fiedler, Miroslav. 1973. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal* **23**(2) 298–305.

Francis, John GF. 1961. The qr transformation a unitary analogue to the lr transformationpart 1. *The Computer Journal* **4**(3) 265–271.

Francis, John GF. 1962. The qr transformationpart 2. *The Computer Journal* **4**(4) 332–345.

Galbraith, Jay R. 1971. Matrix organization designs how to combine functional and project forms. *Business Horizons* **14**(1) 29–40.

Godsil, Chris D, BD McKay. 1982. Constructing cospectral graphs. *Aequationes Mathematicae* **25**(1) 257–268.

Grant, Michael, Stephen Boyd. 2008. Graph implementations for nonsmooth convex programs. V. Blondel, S. Boyd, H. Kimura, eds., *Recent Advances in Learning and Control*. Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, 95–110. `http://stanford.edu/~boyd/graph_dcp.html`.

Grant, Michael, Stephen Boyd. 2012. CVX: Matlab software for disciplined convex programming, version 2.0 beta. `http://cvxr.com/cvx`.

Guimerà, Roger, Albert Díaz-Guilera, Fernando Vega-Redondo, Antonio Cabrales, Alex Arenas. 2002. Optimal network topologies for local search with congestion. *Physical Review Letters* **89**(24) 248701.

Haemers, Willem H, Edward Spence. 2004. Enumeration of cospectral graphs. *European Journal of Combinatorics* **25**(2) 199–211.

Halbeisen, Lorenz, Norbert Hungerbühler. 2000. Reconstruction of weighted graphs by their spectrum. *European Journal of Combinatorics* **21**(5) 641–650.

Hansen, Morten T. 1999. The search-transfer problem: The role of weak ties in sharing knowledge across organization subunits. *Administrative science quarterly* **44**(1) 82–111.

Harary, Frank, Clarence King, Abbe Mowshowitz, Ronald C Read. 1971. Cospectral graphs and digraphs. *Bulletin of the London Mathematical Society* **3**(3) 321–328.

Huang, Siyuan, Jonathon N Cummings. 2011. When critical knowledge is most critical centralization in knowledge-intensive teams. *Small Group Research* **42**(6) 669–699.

Kearns, Michael. 2012. Experiments in social computation. *Communications of the ACM* **55**(10) 56–67.

Kershenbaum, Aaron, Parviz Kermani, George A Grover. 1991. Mentor: an algorithm for mesh network topological optimization and routing. *Communications, IEEE Transactions on* **39**(4) 503–513.

Lakhani, Karim R, Lars Bo Jeppesen. 2007. R&d: Getting unusual suspects to solve r&d puzzles. *Harvard Business Review* **85**(5) 30–32.

Lazer, David, Allan Friedman. 2007. The network structure of exploration and exploitation. *Administrative Science Quarterly* **52**(4) 667–694.

Lovejoy, William S, Amitabh Sinha. 2010. Efficient structures for innovative social networks. *Management Science* **56**(7) 1127–1145.

Lubotzky, Alexander, Ralph Phillips, Peter Sarnak. 1988. Ramanujan graphs. *Combinatorica* **8**(3) 261–277.

Magnanti, Thomas L, Richard T Wong. 1984. Network design and transportation planning: Models and algorithms. *Transportation Science* **18**(1) 1–55.

March, James G. 1991. Exploration and exploitation in organizational learning. *Organization science* **2**(1) 71–87.

Mason, Winter, Duncan J Watts. 2012. Collaborative learning in networks. *Proceedings of the National Academy of Sciences* **109**(3) 764–769.

Mason, Winter A, Andy Jones, Robert L Goldstone. 2008. Propagation of innovations in networked groups. *Journal of Experimental Psychology: General* **137**(3) 422.

McCubbins, Mathew D, Ramamohan Paturi, Nicholas Weller. 2009. Connected coordination network structure and group coordination. *American Politics Research* **37**(5) 899–920.

McEvily, Bill, Alfred Marcus. 2005. Embedded ties and the acquisition of competitive capabilities. *Strategic Management Journal* **26**(11) 1033–1055.

Minoux, Michel. 1989. Networks synthesis and optimum network design problems: Models, solution methods and applications. *Networks* **19**(3) 313–360.

Mohar, Bojan. 1997. *Some applications of Laplace eigenvalues of graphs*. Springer.

Newman, Michael William. 2000. The laplacian spectrum of graphs. Ph.D. thesis, University of Manitoba.

Reagans, Ray, Ezra W Zuckerman. 2001. Networks, diversity, and productivity: The social capital of corporate r&d teams. *Organization science* **12**(4) 502–517.

Shore, Jesse, Ethan Bernstein, David Lazer. 2013. Exploration, exploitation and explication in networked learning. *Presented at Organization Science Winter Conference*. INFORMS.

Sparrowe, Raymond T, Robert C Liden, Sandy J Wayne, Maria L Kraimer. 2001. Social networks and the performance of individuals and groups. *Academy of management journal* **44**(2) 316–325.

Sriperumbudur, Bharath, Gert Lanckriet. 2009. On the convergence of the concave-convex procedure. *Advances in neural information processing systems* **22** 1759–1767.

Sundararajan, Arun, Foster Provost, Gal Oestreicher-Singer, Sinan Aral. 2012. Information in digital, economic and social networks. *Information Systems Research* .

Suri, Siddharth, Duncan J Watts. 2011. Cooperation and contagion in web-based, networked public goods experiments. *PLoS One* **6**(3) e16836.

Todd, Michael J. 2001. Semidefinite optimization. *Acta Numerica* **10**(515-560) 126.

Van Dam, Edwin R, Willem H Haemers. 2003. Which graphs are determined by their spectrum? *Linear Algebra and its Applications* **373** 241–272.

Vandenberghe, Lieven, Stephen Boyd. 1996. Semidefinite programming. *SIAM review* **38**(1) 49–95.

Winter, Pawel. 1987. Steiner problem in networks: a survey. *Networks* **17**(2) 129–167.

Wolkowicz, Henry, Romesh Saigal, Lieven Vandenberghe. 2000. *Handbook of semidefinite programming: theory, algorithms, and applications*, vol. 27. Kluwer Academic Pub.

Yuille, Alan L, Anand Rangarajan. 2003. The concave-convex procedure. *Neural Computation* **15**(4) 915–936.

Yuille, Alan L, Anand Rangarajan, AL Yuille. 2002. The concave-convex procedure (cccp). *Advances in neural information processing systems* **2** 1033–1040.

Zheng, Haichao, Dahui Li, Wenhua Hou. 2011. Task design, motivation, and participation in crowdsourcing contests. *International Journal of Electronic Commerce* **15**(4) 57–88.