Making Networks Stronger: Spectral and Combinatorial Graph Optimization

BAMS project - Algorithms and Graph Theory

Abstract.
A network may be connected and still be easy to break. Adding new links can make it stronger, but when the budget is limited we must decide which connections matter most. This project studies that choice through two mathematical viewpoints. *Spectral graph theory* represents a network by a matrix and studies its eigenvalues; *combinatorial graph theory* describes connectivity through paths, cuts and critical edges. The two viewpoints are related, but they do not measure exactly the same thing.

We will first develop the spectral theory in depth: the graph Laplacian, its quadratic form, algebraic connectivity and Fiedler vectors. Then we will use computation to compare algorithms that select new edges using spectral information with algorithms driven by combinatorial vulnerabilities. The main question is whether increasing spectral connectivity also improves resilience to link failures, and under which network structures the approaches disagree. The project is one coherent investigation, with the **possibility of preparing a public introductory lecture on Spectral Graph Theory** from its theoretical work.

Two ways to describe a well-connected graph.

Spectral viewpoint. From the adjacency matrix A and degree matrix D, form the Laplacian $L = D - A$. Its second-smallest eigenvalue, λ_2, reflects a global notion of connectivity: roughly, how difficult it is to assign very different values to parts of the network without a large variation across its edges. Eigenvectors reveal weakly joined regions and guide algorithms that seek to strengthen them.

Combinatorial viewpoint. A cut counts the links between two parts of the graph; a bridge is an edge whose removal disconnects the network; edge-connectivity is the smallest number of links whose removal can disconnect it. These quantities give direct worst-case guarantees against edge failures, but can remain unchanged even while many other aspects of the network improve.



Figure 1. A star and a six-cycle have the same λ_2 for the unnormalized Laplacian, yet the cycle tolerates any single edge failure while the star does not. The example motivates comparing, rather than equating, spectral and combinatorial connectivity.

What we will do.

The project has one mathematical foundation and one experimental investigation. Theory will focus primarily on spectral graph theory; combinatorial tools, many of which students already know from algorithms courses, will provide the alternative methods and the independent measures of resilience.

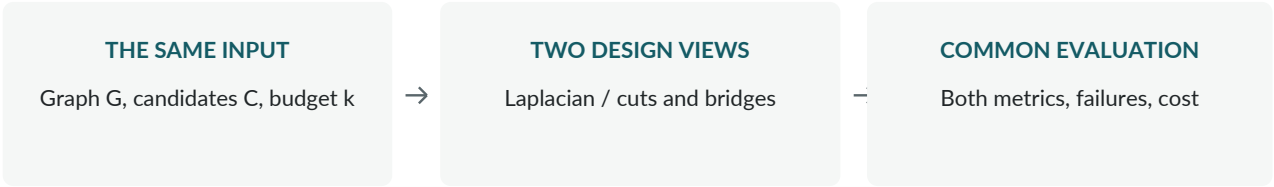
1. Build the spectral foundations. Begin with eigenvalues and eigenvectors of symmetric matrices, then derive the Laplacian and prove the identity $x^T L x = \Sigma_{\{u,v\} \in E} (x_u - x_v)^2$. This yields positive semidefiniteness and the fact that the dimension of the null space equals the number of connected components. Study the variational characterization of λ_2 through the Rayleigh quotient, and understand how a Fiedler vector assigns coordinates to vertices. We will also investigate why the spectrum is connected to graph cuts and how an eigenvector can suggest a partition.

2. Turn the theory into a precise optimization problem. Let $G = (V,E)$ be a connected, undirected, unweighted graph, C a set of permissible new edges, and k the maximum number of links that may be installed. The spectral design problem is to choose $S \subseteq C$, with $|S| \leq k$, in order to maximize λ_2 of the augmented graph. Prove that adding an edge cannot decrease this eigenvalue for the unnormalized Laplacian, but need not increase it strictly. On small graphs, exhaustive enumeration provides a verified optimum.

$$\max_{S \subseteq C,\ |S| \leq k} \lambda_2(L(G + S))$$

3. Compare spectral and combinatorial algorithms. Develop two spectral heuristics: one that evaluates the actual change in λ_2 for every candidate edge, and a faster method that ranks edges using $(v_u - v_v)^2$, where v is a Fiedler vector. Implement a contrasting combinatorial heuristic that prioritizes vulnerable cuts or bridge elimination, together with a random baseline. Each method must use the same initial graph, permissible edges and budget. The algorithms all make discrete choices; the distinction is *which mathematical information guides the choice*, not whether one is an algorithm and the other is not.

4. Explain where the two viewpoints agree or diverge. Measure the resulting λ_2, the global edge-connectivity, the number of bridges and the impact of selected edge failures. Compare the spectral heuristics with the exact spectral optimum whenever small instances permit it. Where feasible, also compute the best attainable edge-connectivity, since the two objectives have different optimal solutions. Seek examples in which a significant gain in one metric gives little immediate gain in the other, and investigate the reasons.



How we will investigate the algorithms.

The purpose of the experiments is not simply to identify a winning implementation, but to understand *why* a strategy works on some network structures and not others. We will construct small trees, paths, two-community graphs joined by bottlenecks, connected random graphs and geometric networks. For each graph, the candidate-edge set and budget will be fixed across methods; different budgets and graph sizes will be tested separately.

A central question is whether selecting the edge with the best predicted spectral effect also repairs the graph's most serious failure points. A spectral method may increase λ_2 without raising the size of the global minimum cut; conversely, a cut-focused method may protect a particular bottleneck without maximizing spectral connectivity. We will quantify these outcomes and compare runtime, repeatability and sensitivity to graph structure. When several eigenvectors correspond to λ_2, a Fiedler-based ranking may depend on the eigenvector chosen, so this numerical issue must be treated explicitly.

A small set of experiments should be checked against exact solutions and manually constructed examples. The project should report negative results, unstable behaviors and counterexamples, not only favorable cases. No approximation guarantee should be inferred merely from empirical performance.

Possible directions beyond the core.

Once the basic comparison is reliable, students may pursue one focused extension. A natural option is a **hybrid edge-selection strategy** that first repairs bridges or critical cuts and then uses spectral scores for the remaining choices. Another is to study families of graphs where spectral and combinatorial rankings can be characterized analytically, or to examine how eigenvalue multiplicity affects the Fiedler heuristic. These are questions to explore, not promised breakthroughs. More advanced techniques, such as semidefinite relaxations, belong outside the required core.

Why this project?

The project connects linear algebra, graph algorithms, optimization and computational experimentation. The spectral component offers substantial new mathematics beyond standard first-year graph algorithms, while the combinatorial component makes it possible to test the practical meaning of spectral improvement. Related network-augmentation problems are part of Laura Sanità's research, although the key spectral edge-addition paper for this project is by Ghosh and Boyd rather than Sanità.

Expected outcome.

Participants should be able to derive the main properties of the Laplacian, explain the meaning and limits of algebraic connectivity, and implement and assess both classes of network-augmentation heuristics. The work will be collected in a **short illustrated mathematical report** with proofs, examples, comparison plots and an explanation of the methods' successes and failures, alongside a tested Python repository. A successful outcome is a coherent and reproducible study; discovering a new state-of-the-art algorithm is not a requirement.

A possible introductory lecture on Spectral Graph Theory.

Because the project requires learning a new mathematical language, the group may turn part of this preparation into a **60–75 minute BAMS lecture** for students familiar with elementary linear algebra and graph algorithms. This is an optional outcome of the *same project*, not a second independent initiative. Its purpose would be to introduce spectral graph theory as a subject in its own right, rather than to present the team's code or experimental results.

The lecture should follow a coherent chain of ideas, with one complete mathematical argument and one worked example. It could be organized as follows:

From a graph to a matrix. Begin with a network containing two tightly connected groups joined by only a few links. Ask how we could detect its weak connection using algebra rather than graph search. Introduce adjacency and degree matrices and construct the Laplacian for a small example. Explain intuitively why the Laplacian captures variation of a numerical signal over neighboring vertices.

Why the spectrum contains structural information. Derive the quadratic-form identity and use it to prove that the Laplacian is positive semidefinite. Show that vectors with zero energy are constant on every connected component; this proves that the multiplicity of the zero eigenvalue counts connected components. This proof should be the central piece of mathematics, rather than a statement taken on trust from software.

Algebraic connectivity and the Fiedler vector. Introduce the Rayleigh quotient and characterize λ_2 as the minimum variation over vectors perpendicular to the constants. Explain how a small λ_2 indicates the existence of a slowly varying graph signal, and why the corresponding eigenvector frequently separates weakly linked communities. Work through a small two-cluster example with a plotted eigenvector. Distinguish rigorous statements from informal interpretations of how large λ_2 ought to be.

From eigenvectors to algorithms. Motivate spectral graph partitioning as a continuous relaxation of a discrete cut problem, showing why thresholding an eigenvector produces a candidate partition. Briefly indicate other applications, such as diffusion, random walks and network design, before introducing the project's central question: can the same spectral information guide the choice of useful new links? This is the natural bridge from the lecture back to the group project, not the lecture's main subject.

The lecturer should prepare two accurately computed graphs, a complete proof of the quadratic-form/kernel results, one clear Fiedler-vector visualization and a brief handout with definitions and sources. A short computational demonstration may help, but should not replace the mathematical derivation. Cheeger inequalities, spectral sparsification and advanced perturbation theory are valuable follow-ups; they should not be presented as material that can all be proved in this introductory lecture.

What the audience should take away.

Students should leave understanding how one converts a network into a Laplacian, why eigenvalues encode connectivity, and how a continuous linear-algebraic problem can inform a discrete graph algorithm. The seminar should feel like a genuine introduction to a new area of mathematics, not a review of minimum spanning trees or a progress presentation about the project.

Prerequisites / Difficulty.

Students should be comfortable with matrices, eigenvalues, eigenvectors, quadratic forms and elementary proofs in linear algebra. Familiarity with paths, cuts, spanning trees and graph algorithms at the level of *Algorithms* by Dasgupta, Papadimitriou and Vazirani is sufficient for the combinatorial background. The implementation requires basic Python, NumPy and graph manipulation. No previous knowledge of spectral graph theory or research-level approximation algorithms is assumed.

The mathematical starting point is accessible to strong undergraduate students; the more demanding parts are interpreting the variational results, designing fair comparisons and understanding numerical pitfalls. The optional lecture can be prepared from the core theory without completing the computational extensions.

Starting materials.

[1] S. Dasgupta, C. Papadimitriou and U. Vazirani, Algorithms (2006). Course-level reference for graph algorithms, cuts and algorithmic reasoning. Use relevant sections of chapters 3, 5 and 7 as needed. Open resource.

[2] D. A. Spielman, Spectral Graph Theory, Chapter 16. The main mathematical introduction: Laplacians, quadratic forms, spectra and eigenvalue interpretations. Study the elementary sections before approaching the paper. Open resource.

[3] A. Ghosh and S. Boyd, Growing Well-Connected Graphs (2006). The main research reference. Read the introduction, the Fiedler-vector greedy heuristic in Section III and the experimental discussion in Section IV. The semidefinite relaxation is optional. Open resource.

[4] U. von Luxburg, A Tutorial on Spectral Clustering (2007). Useful for the optional lecture, especially the connection between the graph Laplacian, spectral embeddings and discrete partitioning. Open resource.

[5] NetworkX, graph connectivity and linear algebra documentation. Use for independent numerical checks and implementation reference; mathematical definitions and explanations should come from the readings above. Open resource.

Related research by Laura Sanità.

[6] H. Angelidakis, D. Hyatt-Denesik and L. Sanità, Node Connectivity Augmentation via Iterative Randomized Rounding (2023). Open access. The introduction motivates resilience and combinatorial connectivity augmentation; this is not a paper on optimizing the Laplacian spectrum. Open resource.

[7] A. E. Feldmann, J. Könemann, K. Pashkovich and L. Sanità, Fast Approximation Algorithms for the Generalized Survivable Network Design Problem (2016). Open access. Gives a broader picture of network design with connectivity requirements; advanced linear-programming arguments are not needed for this project. Open resource.