

The sparsest cut problem

Bogdan Grechuk • 28 Aug 2026

This post continues my series on favorite theorems of the twenty-first century. For an overview of the categories and my earlier selections, see [this introductory post](#).

My choice for 2004 in Algorithms and Complexity is the theorem of Sanjeev Arora, Satish Rao, and Umesh Vazirani giving a polynomial-time $O(\sqrt{\log n})$ -approximation algorithm for the sparsest cut problem. A conference version of their result appeared in the proceedings of STOC 2004, and the full paper was published in the *Journal of the ACM* in 2009 (Arora et al. 2009).

Let $G = (V, E)$ be a finite undirected graph with n vertices. Every nonempty proper subset $S \subset V$ determines a partition

$$V = S \sqcup (V \setminus S)$$

and hence a *cut*. Its edge boundary is

$$\partial S = \{\{u, v\} \in E : u \in S, v \notin S\}.$$

We measure the quality of the cut by the ratio

$$\phi_G(S) = \frac{|\partial S|}{\min\{|S|, |V \setminus S|\}}.$$

Thus, we want to separate many vertices while cutting relatively few edges. The minimum possible value

$$\alpha(G) = \min_{\emptyset \neq S \subsetneq V} \phi_G(S)$$

is called the *edge expansion* of G . Finding a set S that attains this minimum is one of the standard forms of the *sparsest cut problem*.

The sparsest cut problem occurs naturally whenever a network must be divided into two parts while disrupting as few connections as possible. It also plays an important role in the design of divide-and-conquer algorithms, graph clustering, routing, and the study of expander graphs.

Computing $\alpha(G)$ exactly is NP-hard. We may therefore ask for an efficient approximation algorithm. An algorithm has approximation ratio $\rho(n)$ if it always produces a cut S satisfying

$$\phi_G(S) \leq \rho(n)\alpha(G).$$

In 1999, Leighton and Rao (Leighton and Rao 1999) gave a polynomial-time algorithm with approximation ratio

$$O(\log n).$$

Their proof used linear programming and a close relationship between graph cuts and multicommodity flows.

Arora, Rao, and Vazirani improved the approximation ratio from $O(\log n)$ to $O(\sqrt{\log n})$.

Theorem 1 (Arora–Rao–Vazirani) *There exists a universal constant C and a polynomial-time algorithm with the following property. Given any graph G with n vertices, the algorithm produces a nonempty proper subset $S \subset V$ such that*

$$\phi_G(S) \leq C\sqrt{\log n}\alpha(G).$$

The proof begins with a semidefinite programming relaxation of the sparsest cut problem. Instead of immediately assigning every vertex to one of two classes, the relaxation assigns to each vertex v a vector x_v in a Euclidean space. The squared distance

$$\|x_u - x_v\|^2$$

is intended to measure the extent to which u and v are separated.

The semidefinite program minimizes the sum of these squared distances over the edges of the graph. It also imposes triangle inequalities and a spreading condition that prevents all the vectors from collapsing near one another. Every actual cut gives a feasible vector configuration, so the value of the relaxation is a lower bound for the value of the optimal cut. Moreover, semidefinite programs can be solved in polynomial time to any required precision.

The geometric heart of the proof is the following remarkable fact. In every sufficiently spread-out configuration of n vectors satisfying the required triangle inequalities, one can find two sets of vectors A and B , each containing a fixed positive proportion of all the vectors, such that

$$\|x_a - x_b\|^2 \geq \frac{c}{\sqrt{\log n}} \quad \text{for every } a \in A, b \in B,$$

where $c > 0$ is an absolute constant. The proof of this statement uses random projections and concentration of measure in high-dimensional Euclidean spaces.

The vector solution is then rounded to an actual cut. Roughly speaking, one considers the distance of each vector from the set A and separates the vertices at a suitable distance threshold. The number of edges crossing the resulting cut is controlled by the value of the semidefinite program, while the separation between A and B guarantees that both sides contain many vertices. The loss incurred in this rounding procedure is the reciprocal of the separation, producing the factor $O(\sqrt{\log n})$.

Theorem 1 was important not only because it improved the previous approximation ratio. It introduced a powerful geometric method for rounding semidefinite programs: first prove that a complicated vector configuration contains two large, well-separated sets, and then use this separation to construct a combinatorial solution. This method has subsequently influenced many approximation algorithms and results about embeddings of finite metric spaces.

There is also a more general weighted version of the sparsest cut problem, see [this previous blog post](#) for details.

References

- Arora, Sanjeev, Satish Rao, and Umesh V. Vazirani. 2009. “Expander Flows, Geometric Embeddings and Graph Partitioning.” *Journal of the ACM* 56 (2): 5:1–37. <https://doi.org/10.1145/1502793.1502794>.
- Leighton, F. Thomson, and Satish B. Rao. 1999. “Multicommodity Max-Flow Min-Cut Theorems and Their Use in Designing Approximation Algorithms.” *Journal of the ACM* 46 (6): 787–832. <https://doi.org/10.1145/331524.331526>.