

Cuts as a Lens on Algorithms

Randomization, approximation, and spectral relaxation

One framework, three course strands

This course is jointly taught by three professors, each responsible for one strand: randomized algorithms and randomization techniques; approximation algorithms for combinatorial optimization; and spectral algorithms and spectral graph theory. This shared opening lecture uses graph cuts as a common language: Global Min-Cut previews randomized survival and amplification, Max-Cut previews approximation and derandomization, and conductance previews spectral relaxation and sweep rounding.

Advanced Algorithms teaching team

Nanjing University

Lecture 1 - a shared opening lecture

Learning goals and reading routes

This opening class is a common doorway to the course's three instructor-led parts. After the lecture, you should be able to

1. explain how one graph primitive—a cut—leads to three different algorithmic goals and proof styles;
2. reproduce Karger's fixed-cut survival argument and amplify it;
3. derive and derandomize the $1/2$ -approximation for Max-Cut; and
4. trace spectral partitioning from a Rayleigh relaxation to an eigenvector and then to a certified sweep cut.

For a first pass, read [section 1](#), [section 2](#) through the elementary amplification argument, [section 3](#) through conditional expectation, and [section 4](#) through Cheeger's inequality. The Karger–Stein analysis, pairwise-independent seed search, and applications of spectral coordinates are enrichment, but remain in their algorithmic parts. The appendices contain only the probability and linear-algebra tools used by the main text.

Throughout, graphs are finite and undirected, and $n \geq 2$. Parallel edge copies are retained during contraction and self-loops are discarded. Edges have unit weight unless a weighted extension is stated explicitly. Runtime claims name their representation and cost model.

Contents

Learning goals and reading routes	i
1 Cuts as a common framework	1
I Randomized Algorithms and Randomization Techniques	2
2 Global Min-Cut by random contraction	2
II Approximation Algorithms for Combinatorial Optimization	6
3 Max-Cut: approximation and derandomization	6
III Spectral Algorithms and Spectral Graph Theory	10
4 Spectral partitioning and conductance	10
5 Exercises	16
A Probability and randomization toolkit	18
B Linear-algebra toolkit	19
Bibliographic notes	20

1 Cuts as a common framework

A graph cut is elementary enough to define in one line, yet rich enough to expose three distinct algorithmic cultures. This is why cuts serve as the unified example for the shared opening class. The boundaries between the strands are deliberately visible: the random-cut proof connects randomization to approximation, while relaxation and rounding connect approximation to spectral methods.

Let $G = (V, E)$ be an undirected, loopless multigraph with $n = |V|$ vertices and $m = |E|$ edge copies. For disjoint sets $S, T \subseteq V$, write

$$E(S, T) = \{uv \in E : u \in S, v \in T\}.$$

For $\emptyset \neq S \subsetneq V$, let $\bar{S} = V \setminus S$ and $\delta(S) = E(S, \bar{S})$. The unordered bipartition $\{S, \bar{S}\}$ is a *cut*, and $|\delta(S)|$ is its size. Notice that S and $\bar{S}$ describe the same cut.

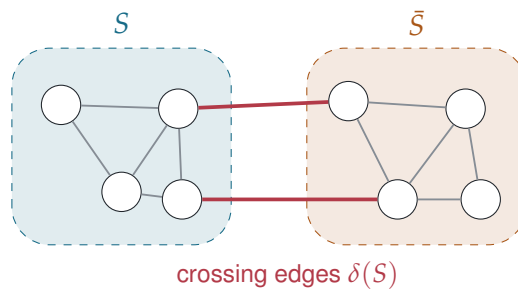


Figure 1: A cut is a vertex bipartition; its value is determined by the crossing edges. Different objectives reward or penalize the same boundary in different ways.

The three objectives in this lecture are:

$$\begin{aligned} \lambda(G) &= \min_{\emptyset \neq S \subsetneq V} |\delta(S)| && \text{(global minimum cut),} \\ \text{OPT}_{\text{MC}}(G) &= \max_{S \subseteq V} |\delta(S)| && \text{(maximum cut),} \\ h(G) &= \min_{0 < |S| \leq n/2} \frac{|\delta(S)|}{|S|} && \text{(edge expansion).} \end{aligned}$$

If G is d -regular, its conductance is

$$\phi(G) = \frac{h(G)}{d} = \min_{\emptyset \neq S \subsetneq V} \frac{|\delta(S)|}{d \min\{|S|, |\bar{S}|\}}.$$

For an irregular graph, the correct denominator is volume, $\text{vol}(S) = \sum_{v \in S} \deg(v)$; see section 4.4.

Problem	Objective	Course module and algorithmic lens
Global Min-Cut	$\min_S \delta(S) $	Part I: exact randomized algorithm; fixed-witness survival
Max-Cut	$\max_S \delta(S) $	Part II: approximation for NP-hard optimization; derandomization
Conductance	$\min_S \phi(S)$	Part III: eigenvector relaxation and certified sweep rounding

Table 1: The same cut primitive previews the three instructor-led course modules.

Part I

Randomized Algorithms and Randomization Techniques

2 Global Min-Cut by random contraction

This first showcase asks for an exact optimum, but reaches it through a randomized process. Its proof introduces two themes developed later in Part I of the course: track a fixed witness through an adaptive random experiment, then amplify a modest success probability. The needed chain rule and repetition bound are reviewed in [appendix A](#).

If G is disconnected, then $\lambda(G) = 0$ and any connected component gives a minimum cut. We can find this case first by a graph search. Hence this section assumes that G is connected; Karger's insight is then to destroy the graph while trying not to destroy one fixed minimum cut.

2.1 Contraction and the survival invariant

To contract an edge $e = uv$, replace u and v by a single supervertex, reattach every edge formerly incident to either endpoint, retain parallel edges, and delete the self-loops created from edges between u and v . A supervertex represents the set of original vertices merged into it.

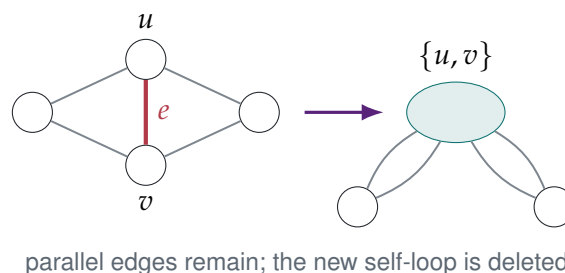


Figure 2: Contracting uv merges its endpoints. Keeping the supervertex as a set of original vertices makes every later cut easy to lift back to G .

Lemma 2.1 (Cuts under contraction). *Let G/e be obtained by contracting $e = uv$.*

- (i) *Every cut of G/e lifts to a cut of G in which u and v lie on the same side, with exactly the same crossing edge copies.*
- (ii) *A cut $C = \delta_G(S)$ survives as a cut of G/e if and only if $e \notin C$.*
- (iii) *If C is a minimum cut of G and $e \notin C$, then the surviving C is a minimum cut of G/e .*

Proof. A vertex of G/e is a block of original vertices. Expanding every block on one side of a cut gives a cut of G ; contraction changes neither the edge copies running between distinct blocks nor which of them cross. This proves (i). The endpoints of e occupy the same block after contraction, so a cut can represent C precisely when C does not separate them, proving (ii). For (iii), if G/e had a cut smaller than C , its lift from (i) would be a smaller cut of G , a contradiction. $\square$

RandomContraction(G) - Karger (1993)

1. While more than two supervertices remain, choose an edge copy uniformly from the current multiset and contract it.
2. Return the parallel edge copies between the final two supervertices, lifted to the corresponding cut of the original graph.

The output is always a cut of the original graph. Correctness is probabilistic: we must show that a minimum cut has a reasonable chance to survive.

Fresh randomness is not the same as independent outputs

In Karger's algorithm, every step uses fresh randomness to choose uniformly from the *current* edge multiset. The selected edges are nevertheless not independent as random variables: the current graph depends on the earlier selections. The chain rule, not a product of unconditional probabilities, is the correct analytical tool.

2.2 The fixed-cut survival calculation

Fix one particular minimum cut C and write $\lambda = |C| = \lambda(G)$. Condition on C having survived until the current graph has r supervertices. By lemma 2.1, C is still a minimum cut of that graph.

Lemma 2.2 (Many edges while a minimum cut survives). *If an r -vertex multigraph has minimum-cut value $\lambda > 0$, then it has at least $r\lambda/2$ edge copies.*

Proof. Every vertex has degree at least λ , since the edges incident to a vertex form a cut separating that vertex. The handshaking identity gives $2m = \sum_v \deg(v) \geq r\lambda$. $\square$

At the r -vertex stage, at most a $\lambda/(r\lambda/2) = 2/r$ fraction of the current edge copies belong to C . Therefore the next contraction avoids C with conditional probability at least $1 - 2/r$.

Theorem 2.3 (Karger's fixed-cut guarantee). *On a connected n -vertex loopless multigraph, RandomContraction returns any fixed minimum cut C with probability at least*

$$\prod_{r=3}^n \left(1 - \frac{2}{r}\right) = \frac{2}{n(n-1)} = \binom{n}{2}^{-1}.$$

Consequently, it returns some minimum cut with at least this probability.

Proof. Let A_r be the event that the contraction made when r supervertices remain does not hit C . Conditioned on all earlier survival events, the preceding argument gives

$$\mathbb{P}(A_r \mid \text{earlier survival}) \geq 1 - \frac{2}{r}.$$

The chain rule in proposition A.1 now gives

$$\mathbb{P}(C \text{ survives}) \geq \prod_{r=3}^n \frac{r-2}{r} = \frac{(n-2)!}{n!/2} = \frac{2}{n(n-1)}.$$

If C survives until two supervertices remain, it is the unique cut of the contracted graph and is returned. Returning C implies returning a minimum cut, which proves the final statement. $\square$

Remark 2.4 (Tightness of the analysis). For the cycle C_n , every pair of edges is a distinct minimum cut, so there are exactly $\binom{n}{2}$ minimum cuts. Fix one such pair C . Conditioned on avoiding C , every intermediate graph is again a cycle, so at the r -supervertex stage exactly two of its r edge copies lie in C . Consequently every inequality in the survival product is an equality and $\mathbb{P}(C \text{ is returned}) = 1/\binom{n}{2}$. Thus the fixed-cut bound is tight.

2.3 Amplification and a counting corollary

Run RandomContraction independently R times and return the smallest cut seen. By [proposition A.3](#) and [theorem 2.3](#),

$$\mathbb{P}(\text{all runs miss every minimum cut}) \leq \exp\left(-\frac{R}{\binom{n}{2}}\right).$$

Taking $R = \lceil c\binom{n}{2} \ln n \rceil$ makes the failure probability at most n^{-c} . This is a Monte Carlo algorithm: it always returns a legitimate cut, and the probability that the cut is nonoptimal can be made polynomially small.

Here is one reproducible input model. Store the m edge copies in a list, randomly permute the list, and scan it with a disjoint-set data structure. Skip an edge whose endpoints are already in the same supervertex; otherwise contract it, stopping at two supervertices. The first eligible edge in the remaining random order is uniform among all current non-loop edge copies, so this exactly simulates RandomContraction. Including a final scan to read the cut, one trial takes $O(m\alpha(n) + n)$ time and $O(m + n)$ space. The probability analysis itself is independent of this representation.

The same random experiment proves a deterministic structural statement.

Corollary 2.5 (Number of minimum cuts). *A connected n -vertex multigraph has at most $\binom{n}{2}$ distinct minimum cuts, viewed as unordered bipartitions (equivalently, distinct cut edge sets). The bound is tight for C_n .*

Proof. In a connected graph, distinct unordered bipartitions have distinct cut-edge sets. Indeed, if $\delta(S) = \delta(T)$, then along every edge uv the two membership bits for $S \triangle T$ agree. Connectivity makes $S \triangle T$ either $\emptyset$ or V , so $T = S$ or $T = \bar{S}$.

For each minimum cut C , let B_C be the event that one run returns C . The events B_C are pairwise disjoint, since one run has only one output, and [theorem 2.3](#) gives $\mathbb{P}(B_C) \geq 1/\binom{n}{2}$. Hence

$$1 \geq \mathbb{P}\left(\bigcup_C B_C\right) = \sum_C \mathbb{P}(B_C) \geq \frac{\#\{\text{minimum cuts}\}}{\binom{n}{2}}.$$

□

Why connectedness matters

The qualification in [corollary 2.5](#) is necessary. A disconnected graph has minimum-cut value zero and can have exponentially many zero cuts, obtained by placing its connected components on either side. The original contraction loop can also run out of edges before only two supervertices remain. Detecting connected components first resolves both issues.

2.4 The Karger-Stein recursive algorithm

The elementary algorithm loses most of its success probability near the end: the factors $1 - 2/r$ become small only when few supervertices remain. Karger and Stein exploit this observation by stopping contraction early and branching.

Let $\text{ContractTo}(G, t)$ repeatedly contract a uniformly random current edge until exactly t supervertices remain. The calculation in [theorem 2.3](#) stops at t and gives, for every fixed minimum cut C ,

$$\mathbb{P}(C \text{ survives from } n \text{ to } t) \geq \prod_{r=t+1}^n \left(1 - \frac{2}{r}\right) = \frac{t(t-1)}{n(n-1)}. \quad (1)$$

FastCut(G) - Karger-Stein

Let $n = |V(G)|$.

1. If $n \leq 6$, enumerate all nontrivial cuts and return a smallest one.
2. Set $t = \lceil n/\sqrt{2} + 1 \rceil$.
3. Independently form $G_1 = \text{ContractTo}(G, t)$ and $G_2 = \text{ContractTo}(G, t)$.
4. Recursively run FastCut on both graphs, lift both returned cuts to G , and return the smaller one.

The rounding is chosen so that

$$q_n := \frac{t(t-1)}{n(n-1)} \geq \frac{1}{2}.$$

Let $p(n)$ be the infimum, over connected n -vertex multigraphs, of the success probability of FastCut. If a fixed minimum cut survives a branch, the contracted graph still has the original minimum-cut value. Therefore any minimum cut returned recursively lifts to a minimum cut of the original graph. The two branches use independent randomness, so

$$p(n) \geq 1 - (1 - q_n p(t))^2 \geq 1 - (1 - \frac{1}{2} p(t))^2 = p(t) - \frac{p(t)^2}{4}. \quad (2)$$

Here is the promised non-asymptotic reason that this recurrence loses only a logarithmic factor. Put $n_0 = n$ and $n_{j+1} = \lceil n_j/\sqrt{2} + 1 \rceil$, and let L be the first index with $n_L \leq 6$. For $n_j \geq 12$,

$$n_{j+1} \leq n_j/\sqrt{2} + 2 \leq 7n_j/8;$$

for $7 \leq n_j < 12$, the size decreases at every step. Hence $L = O(\log n)$ even with rounding.

Define $a_0 = 1$ and $a_{j+1} = a_j - a_j^2/4$. Since $0 < a_j \leq 1$,

$$\frac{1}{a_{j+1}} - \frac{1}{a_j} = \frac{1}{4(1 - a_j/4)} \leq \frac{1}{3}.$$

Thus $a_j \geq 3/(j+3)$. A leaf call succeeds with probability one. Applying [equation \(2\)](#) backward for L levels, using monotonicity of $x - x^2/4$ on $[0, 1]$, gives

$$p(n) \geq a_L = \Omega(1/\log n).$$

For this recurrence analysis, use the standard dense multiplicity-matrix model. When r supervertices remain, sample a first endpoint in proportion to its degree and then an incident edge

copy from its matrix row; the resulting undirected edge copy is uniform. A cumulative-degree structure and a row scan cost $O(r)$ per sample, and merging two rows and columns also costs $O(r)$. Thus a contraction stage beginning with n supervertices costs $O(n^2)$ and

$$T(n) \leq 2T(\lceil n/\sqrt{2} + 1 \rceil) + O(n^2) = O(n^2 \log n).$$

The recursion tree has $O(\log n)$ levels and total $O(n^2)$ contraction work per level, up to constant factors from rounding. More explicitly, if one run succeeds with probability at least $c_0/\log n$, then $R = \lceil (3/c_0) \log^2 n \rceil$ independent runs fail with probability at most $\exp(-c_0 R / \log n) \leq n^{-3}$. Returning the smallest cut seen therefore takes $O(n^2 \log^3 n)$ arithmetic operations and succeeds with high probability.

Theorem 2.6 (Karger-Stein). *On every connected n -vertex multigraph, one execution of FastCut runs in $O(n^2 \log n)$ time in the stated representation and returns a minimum cut with probability $\Omega(1/\log n)$. Independent amplification yields a minimum cut with high probability in $O(n^2 \log^3 n)$ time.*

Remark 2.7 (Historical perspective). The contraction algorithm is valued for its proof idea, not for being the state of the art in worst-case running time for global Min-Cut. Karger later obtained an exact randomized near-linear-time algorithm, and recent work gives deterministic near-linear-time algorithms even for weighted graphs. The models and hidden logarithmic factors differ, so these results should not be read as a single model-free ranking; precise references appear at the end.

Part I preview takeaway

The probability calculation does not analyze all possible outputs at once. It fixes one optimal cut, proves that this witness survives with probability at least $2/[n(n-1)]$, and then uses independent repetition. Karger-Stein improves the algorithm by stopping contraction early and branching, preserving the same fixed-witness analysis while avoiding the weakest late-stage survival factors. Fixed-witness analysis, recursive design, and amplification recur throughout randomized algorithms.

Part II

Approximation Algorithms for Combinatorial Optimization

3 Max-Cut: approximation and derandomization

This second showcase changes the computational goal. Because exact Max-Cut is NP-hard, the central question is no longer whether the optimum can be found, but whether a polynomial-time algorithm can certify a useful fraction of it. The random-cut argument also forms a bridge from Part I to Part II: expectation first proves existence, and conditional expectation turns that proof into a deterministic approximation algorithm. These probability facts are reviewed in appendix A.

For Max-Cut, the goal is reversed:

$$\text{OPT}_{\text{MC}}(G) = \max_{S \subseteq V} |\delta(S)|.$$

The decision problem asking whether a cut of size at least K exists is NP-complete [6], and the optimization problem is NP-hard. An algorithm is an α -approximation for Max-Cut if, on every input G , it runs in polynomial time and returns a cut S_A satisfying

$$|\delta(S_A)| \geq \alpha \text{OPT}_{\text{MC}}(G).$$

This inequality remains meaningful when the optimum is zero.

3.1 A deterministic greedy one-half approximation

Fix any vertex order $v_1, \dots, v_n$. While processing v_i , only $v_1, \dots, v_{i-1}$ have been assigned.

GreedyCut(G)

If $m = 0$, return any nontrivial bipartition. Otherwise initialize $S = T = \emptyset$. For $i = 1, \dots, n$, place v_i on the side that creates more crossing edges between v_i and the already placed vertices. Break ties arbitrarily. Return (S, T) .

Theorem 3.1 (Greedy Max-Cut). *GreedyCut runs in $O(n + m)$ time with adjacency lists and returns a cut of size at least*

$$\frac{m}{2} \geq \frac{1}{2} \text{OPT}_{\text{MC}}(G).$$

Thus it is a deterministic 1/2-approximation.

Proof. Just before v_i is placed, let a_i and b_i be the numbers of its edges to the current sets S and T . Placing v_i in T adds a_i crossing edges; placing it in S adds b_i . The greedy increase is therefore

$$\max\{a_i, b_i\} \geq \frac{a_i + b_i}{2}.$$

Every edge is counted exactly once in $\sum_i (a_i + b_i)$, when its later endpoint is processed. The final cut has size at least $m/2$. Since no cut contains more than all m edges, $\text{OPT}_{\text{MC}}(G) \leq m$.

To implement the algorithm, inspect each edge when its later endpoint is processed; isolated vertices account for the additional $O(n)$ term. When $m > 0$, either both shores are already nonempty or the later endpoint of the first encountered edge is placed opposite its earlier neighbor, so the output is a nontrivial cut. $\square$

This argument extends verbatim to nonnegative edge weights by replacing edge counts with total weights, with the same guard when the total weight is zero. The analysis is tight under permitted tie-breaking: on the two-edge path, process the two leaves first, place them on opposite sides in their initial ties, and then place the center. The algorithm cuts one edge while the optimum cuts both.

3.2 The randomized proof hidden inside the greedy rule

Independently assign each vertex a fair bit $X_v \in \{0, 1\}$ and put it in S or T according to that bit. Let

$$Z = |\delta(S)| = \sum_{uv \in E} \mathbf{1}_{\{X_u \neq X_v\}}.$$

For each edge, $\mathbb{P}(X_u \neq X_v) = 1/2$. By linearity of expectation ([theorem A.2](#)),

$$\mathbb{E}[Z] = \sum_{uv \in E} \mathbb{P}(X_u \neq X_v) = \frac{m}{2} \geq \frac{1}{2} \text{OPT}_{\text{MC}}(G). \quad (3)$$

The averaging principle proves that some cut has value at least $m/2$. However, a single random sample need not achieve its own expectation. The next step turns this existence proof into an always-correct deterministic algorithm.

For this calculation we harmlessly allow an assignment with an empty shore and give it value zero. If $m > 0$, every assignment of value at least $m/2$ is automatically nontrivial. If $m = 0$, any nontrivial bipartition is optimal.

Suppose $X_{v_1}, \dots, X_{v_{i-1}}$ have already been fixed; this partial assignment is an event H of positive probability. The next unassigned bit is still fair after conditioning on H , so the current conditional expectation is the average of its two children:

$$\mathbb{E}[Z \mid X_{v_1} = x_1, \dots, X_{v_{i-1}} = x_{i-1}] = \frac{1}{2} \sum_{b \in \{0,1\}} \mathbb{E}[Z \mid X_{v_1} = x_1, \dots, X_{v_i} = b].$$

Choose the value b giving the larger child expectation. The conditional expectation never decreases, and after all bits are fixed it equals the value of the resulting cut. Therefore the final value is at least the initial $m/2$.

Why is this abstract rule exactly GreedyCut? At step i , every edge not joining v_i to an already assigned vertex has the same conditional expected contribution under both choices for X_{v_i} . An edge from v_i to an unassigned vertex still crosses with probability $1/2$, and an edge not incident to v_i is unaffected. The only difference is whether the a_i or b_i edges to previously placed vertices cross. Maximizing conditional expectation is precisely the greedy choice.

Method of conditional expectations

Start from a randomized construction whose expected objective value is good. Expose its random choices one at a time. At each step select a value whose conditional expectation is at least the current one. If those expectations can be compared efficiently, the final deterministic object is at least as good as the original expectation.

3.3 Optional bridge: pairwise independence is enough

The calculation in [equation \(3\)](#) only inspects pairs (X_u, X_v) along edges. Mutual independence of all n bits is stronger than necessary. It is enough that every X_v is fair and every pair is independent.

Let

$$\ell = \lceil \log_2(n+1) \rceil.$$

Write $\mathbb{F}_2 = \{0,1\}$ for the field whose addition is xor, and define $\langle a, r \rangle = \sum_{j=1}^{\ell} a_j r_j \pmod{2}$ for vectors in $\mathbb{F}_2^{\ell}$. A linear map is *surjective* when every output has a preimage; its rank is the dimension of its image. There are at least n nonzero vectors in $\mathbb{F}_2^{\ell}$, so assign each vertex a distinct nonzero label $a_v \in \mathbb{F}_2^{\ell}$. Choose a uniform seed $R \in \mathbb{F}_2^{\ell}$ and define

$$Y_v = \langle a_v, R \rangle \pmod{2}.$$

Theorem 3.2 (Small pairwise-independent sample space). *The bits $(Y_v)_{v \in V}$ are uniform and pairwise independent.*

Proof. For nonzero a_v , the linear functional $R \mapsto \langle a_v, R \rangle$ has rank one, so exactly half of the seeds give each bit value. If $u \neq v$, then a_u and a_v are distinct nonzero vectors. Over $\mathbb{F}_2$, they are linearly independent: the only nonzero scalar is 1. Hence

$$R \mapsto (\langle a_u, R \rangle, \langle a_v, R \rangle)$$

has rank two. Each of the four output pairs has exactly $2^{\ell-2}$ preimages, so its probability is $1/4$. $\square$

The full family indexed by all nonzero labels is not three-wise independent. Whenever $a, b, a + b$ are among the selected labels, with a, b distinct and nonzero,

$$Y_a \oplus Y_b = Y_{a+b},$$

and $\mathbb{P}(Y_a = Y_b = Y_{a+b} = 0) = 1/4 \neq 1/8$. The construction *generates* many correlated bits from an ℓ -bit seed; it does not create additional entropy.

Since pairwise independence makes $\mathbb{P}(Y_u \neq Y_v) = 1/2$ for every edge, the resulting cut distribution still has expected value $m/2$. It is supported on only $2^\ell = O(n)$ cuts rather than all 2^n assignments. Enumerating every seed and returning the best cut therefore derandomizes the construction.

PairwiseSeedSearch(G)

Choose distinct nonzero labels $a_v \in \mathbb{F}_2^\ell$. For each of the 2^ℓ seeds $r \in \mathbb{F}_2^\ell$, place v according to $\langle a_v, r \rangle$, evaluate the cut, and return a largest candidate.

There are $2^\ell \leq 2(n+1)$ seeds. Straight enumeration takes

$$O(2^\ell(m + n\ell)) = O(nm + n^2 \log n)$$

time in the bit-operation model: for each seed we compute n length- ℓ inner products and score m edges. This honest bound includes the scoring step. It is slower than GreedyCut and is included because the reduction from n random bits to $O(\log n)$ random bits is a fundamental pseudorandomness idea. On a word RAM with word size $\Omega(\log n)$ bits, a Walsh–Hadamard transform can score all seeds simultaneously in $O(m + n \log n)$ word operations; this refinement is left as an advanced exercise.

The zero seed assigns every vertex to one side. If $m > 0$, it cannot be the best seed because the average candidate value is $m/2 > 0$. If $m = 0$, return any nontrivial bipartition instead.

3.4 Beyond one half

The celebrated algorithm of Goemans and Williamson relaxes each sign $x_v \in \{-1, 1\}$ to a unit vector x_v and maximizes $\frac{1}{2} \sum_{uv \in E} (1 - \langle x_u, x_v \rangle)$. A random hyperplane then separates vectors with inner product ρ with probability $\arccos(\rho)/\pi$. Comparing this probability with the relaxed edge contribution proves that the returned cut has expected value at least $\alpha_{\text{GW}} \text{OPT}_{\text{MC}}(G)$, where

$$\alpha_{\text{GW}} = \min_{-1 \leq \rho < 1} \frac{2 \arccos(\rho)}{\pi(1 - \rho)} \approx 0.878567.$$

The Unique Games Conjecture is an unproved hardness conjecture introduced by Khot [8]. Assuming it, Khot, Kindler, Mossel, and O'Donnell show that, for every $\varepsilon > 0$, achieving ratio $\alpha_{\text{GW}} + \varepsilon$ is NP-hard [9]. We will revisit semidefinite relaxation and randomized hyperplane rounding later in the course.

Part II preview takeaway

Approximation replaces exact optimization by a certified performance ratio. For Max-Cut, a random assignment has expected value $m/2$; the method of conditional expectations preserves that value while fixing the choices, giving a deterministic $1/2$ -approximation.

Part III**Spectral Algorithms and Spectral Graph Theory****4 Spectral partitioning and conductance**

This third showcase replaces a discrete search over vertex sets by a continuous optimization over vectors, and then rounds the vector back to a cut with a proof certificate. This relaxation-and-rounding pattern is central to the spectral part of the course. Generic facts about symmetric matrices, Rayleigh–Ritz, and Cauchy–Schwarz are collected in [appendix B](#); the graph-specific theory is developed here.

The spectral route

cut objective $\rightarrow$ Rayleigh-quotient relaxation $\rightarrow$ μ_2 -eigenvector $\rightarrow$ sweep cut.

The eigenvalue certifies the relaxation; the sweep lemma certifies the rounding.

Min-Cut favors tiny shores: separating a single low-degree vertex may be optimal. To search for a coherent cluster, normalize the boundary by the size of the smaller side. For a d -regular graph with $d \geq 1$, define

$$\phi(S) = \frac{|\delta(S)|}{d \min\{|S|, |\bar{S}|\}}, \quad \phi(G) = \min_{\emptyset \neq S \subsetneq V} \phi(S).$$

The normalization makes $0 \leq \phi(S) \leq 1$: it asks what fraction of the edge incidences on the smaller shore leave that shore.

Terminology

The quantity $h(G) = \min_{|S| \leq n/2} |\delta(S)|/|S|$ is usually called *edge expansion* or the quotient-cut objective, and $\phi(G) = h(G)/d$ on a d -regular graph. Uniform Sparsest Cut instead minimizes

$$\frac{n|\delta(S)|}{|S||\bar{S}|}.$$

After choosing the smaller shore, this is within a factor two of $h(S)$. Nonuniform Sparsest Cut introduces a separate demand graph and is a different problem.

4.1 Adjacency, Laplacian, and the Rayleigh quotient

Let $A \in \mathbb{R}^{n \times n}$ be the adjacency matrix, where A_{uv} is the number of edge copies between u and v . Since A is real and symmetric, it has an orthonormal eigenbasis and real eigenvalues

$$\theta_1 \geq \theta_2 \geq \dots \geq \theta_n.$$

Proposition 4.1 (Elementary spectrum of a regular graph). *For a d -regular multigraph:*

- (i) $|\theta_i| \leq d$ for every i ;
- (ii) $A\mathbf{1} = d\mathbf{1}$, so $\theta_1 = d$;
- (iii) the multiplicity of eigenvalue d equals the number of connected components. In particular, G is connected if and only if $\theta_2 < d$.

Proof. If $Ax = \theta x$ and $|x_i| = \max_j |x_j| > 0$, then

$$|\theta||x_i| = \left| \sum_j A_{ij}x_j \right| \leq \sum_j A_{ij}|x_j| \leq d|x_i|,$$

which proves (i). Every row sum is d , proving (ii).

For a connected graph, suppose $Ax = dx$. After replacing x by $-x$ if needed, choose i with $x_i = \max_j x_j > 0$. Then

$$dx_i = \sum_j A_{ij}x_j \leq \sum_j A_{ij}x_i = dx_i.$$

Equality forces every neighbor of i to have value x_i ; connectivity propagates this value to every vertex. Thus the d -eigenspace is spanned by $\mathbf{1}$. For a disconnected graph, the indicator vectors of its components are linearly independent d -eigenvectors, and the same argument on each component shows that they span the full eigenspace. $\square$

It is cleaner to work with the normalized Laplacian

$$\mathcal{L} = I - \frac{1}{d}A.$$

Its eigenvalues are

$$0 = \mu_1 \leq \mu_2 \leq \dots \leq \mu_n \leq 2, \quad \mu_i = 1 - \frac{\theta_i}{d}.$$

For every vector $f \in \mathbb{R}^V$,

$$f^\top \mathcal{L} f = \frac{1}{d} \sum_{uv \in E} (f_u - f_v)^2. \quad (4)$$

Indeed, $f^\top A f = 2 \sum_{uv \in E} f_u f_v$ and $d \sum_v f_v^2 = \sum_{uv \in E} (f_u^2 + f_v^2)$, so expanding the squares gives the identity. If G is connected, choose the first normalized eigenvector as $\mathbf{1}/\sqrt{n}$. Expanding any $f \perp \mathbf{1}$ in the remaining orthonormal eigenvectors gives the variational formula

$$\mu_2 = \min_{\substack{f \neq 0 \\ f \perp \mathbf{1}}} \mathcal{R}(f), \quad \mathcal{R}(f) := \frac{\sum_{uv \in E} (f_u - f_v)^2}{d \sum_{v \in V} f_v^2}. \quad (5)$$

Indeed, the quotient is a weighted average of $\mu_2, \dots, \mu_n$ in that eigenbasis.

The discrete optimization over sets has now been relaxed to a continuous optimization over vectors. The remaining task is rounding: turn a good vector into a good cut.

4.2 Sweep rounding

Let f be an eigenvector for μ_2 , equivalently an adjacency eigenvector for θ_2 , chosen orthogonal to $\mathbf{1}$. Sort the vertices so that

$$f(u_1) \geq f(u_2) \geq \dots \geq f(u_n),$$

and let $P_i = \{u_1, \dots, u_i\}$ for $1 \leq i < n$.

SpectralSweep(G) for a d -regular graph

1. Compute a nonzero μ_2 -eigenvector $f \in \mathbb{R}^n$ with $f \perp \mathbf{1}$.
2. Sort the vertices by nonincreasing f -coordinate.
3. Evaluate $\phi(P_i) = |\delta(P_i)| / [d \min\{i, n - i\}]$ for every $1 \leq i < n$.
4. Return a prefix P_{i^*} of minimum conductance.

The range is $i < n$: $P_n = V$ would have an empty complement and undefined conductance. Once the ordering is known, all prefix boundaries can be evaluated in $O(m + n)$ time by updating the boundary as one vertex is added; sorting costs $O(n \log n)$.

The display uses an exact-real eigenvector model to keep the main proof transparent. The rounding statement is robust: for any nonzero $f \perp \mathbf{1}$ with $\mathcal{R}(f) \leq \eta$, the same median-and-sweep proof below returns a prefix of conductance at most $\sqrt{2\eta}$. Thus a numerical eigensolver may return an approximate Rayleigh vector; subtracting its mean enforces orthogonality. The cost of producing such a vector depends on the requested accuracy and the spectral gap, while ties in sorting may be broken arbitrarily.

4.3 Cheeger's inequality, with the rounding proof

Theorem 4.2 (Cheeger's inequality for regular graphs). *For every d -regular graph,*

$$\frac{\mu_2}{2} \leq \phi(G) \leq \sqrt{2\mu_2}.$$

Moreover, *SpectralSweep* returns a set S satisfying

$$\phi(S) \leq \sqrt{2\mu_2} \leq 2\sqrt{\phi(G)}.$$

Equivalently, with adjacency eigenvalues, $\mu_2 = 1 - \theta_2/d$.

If G is disconnected, then $\mu_2 = 0$ by [proposition 4.1](#), while a connected component has zero boundary, so $\phi(G) = 0$. Moreover, a μ_2 -eigenvector orthogonal to $\mathbf{1}$ is constant on each component and takes at least two distinct values; a sweep prefix at a change of value is a union of components and has conductance zero. Thus the theorem, including the algorithmic claim, is immediate in this case. The two proofs below may therefore assume that G is connected.

Proof: a sparse cut gives a low-energy vector. Fix S with $s = |S| \leq n/2$ and define

$$x = \mathbf{1}_S - \frac{s}{n} \mathbf{1}.$$

Then $x \perp \mathbf{1}$. Only crossing edges contribute to the numerator of $\mathcal{R}(x)$, each by 1, while

$$\sum_v x_v^2 = \frac{s(n-s)}{n}.$$

By equation (5),

$$\mu_2 \leq \mathcal{R}(x) = \frac{n|\delta(S)|}{ds(n-s)} \leq \frac{2|\delta(S)|}{ds} = 2\phi(S).$$

Minimizing over S proves $\mu_2/2 \leq \phi(G)$. $\square$

The reverse direction is the algorithmic heart. We isolate the rounding calculation.

Lemma 4.3 (Sweep lemma). *Let $g : V \rightarrow \mathbb{R}_{\geq 0}$ be nonzero and supported on at most $n/2$ vertices. Among the nonempty threshold sets $S_t = \{v : g_v^2 \geq t\}$ with $t > 0$, there is one satisfying*

$$\phi(S_t) \leq \sqrt{2\mathcal{R}(g)}.$$

Proof. The layer-cake, or discrete coarea, identities are

$$\begin{aligned} \int_0^\infty |\delta(S_t)| dt &= \sum_{uv \in E} |g_u^2 - g_v^2|, \\ \int_0^\infty d|S_t| dt &= d \sum_v g_v^2. \end{aligned}$$

For example, an edge uv lies in $\delta(S_t)$ precisely for t between g_u^2 and g_v^2 . For every $t > 0$, S_t is contained in the support of g , so $|S_t| \leq n/2$; the single threshold $t = 0$ has measure zero and does not affect either integral. Averaging the two integrals therefore shows that some nonempty S_t has

$$\phi(S_t) \leq \frac{\sum_{uv \in E} |g_u^2 - g_v^2|}{d \sum_v g_v^2}.$$

Now apply Cauchy-Schwarz and regularity:

$$\begin{aligned} \sum_{uv \in E} |g_u^2 - g_v^2| &\leq \left(\sum_{uv \in E} (g_u - g_v)^2 \right)^{1/2} \left(\sum_{uv \in E} (g_u + g_v)^2 \right)^{1/2} \\ &\leq \left(\sum_{uv \in E} (g_u - g_v)^2 \right)^{1/2} \left(2d \sum_v g_v^2 \right)^{1/2}. \end{aligned}$$

Dividing by $d \sum_v g_v^2$ gives $\sqrt{2\mathcal{R}(g)}$. $\square$

Proof of the sweep direction in theorem 4.2. Let $f \perp \mathbf{1}$ be a μ_2 -eigenvector. Choose a median m of its coordinates, so at most $n/2$ coordinates are strictly above m and at most $n/2$ are strictly below it. Put $h = f - m\mathbf{1}$. Adding a constant changes no edge difference, while

$$\sum_v h_v^2 = \sum_v f_v^2 + nm^2 \geq \sum_v f_v^2,$$

because $f \perp \mathbf{1}$. Hence $\mathcal{R}(h) \leq \mathcal{R}(f) = \mu_2$.

Write $h = h_+ - h_-$, where $h_+ = \max\{h, 0\}$ and $h_- = \max\{-h, 0\}$. Each support has size at most $n/2$. Edge by edge,

$$(h_u - h_v)^2 \geq (h_{+,u} - h_{+,v})^2 + (h_{-,u} - h_{-,v})^2,$$

and $\|h\|_2^2 = \|h_+\|_2^2 + \|h_-\|_2^2$. Therefore at least one nonzero $g \in \{h_+, h_-\}$ satisfies $\mathcal{R}(g) \leq \mathcal{R}(h) \leq \mu_2$.

By lemma 4.3, some threshold set of g has conductance at most $\sqrt{2\mu_2}$. A threshold set of h_+ is a prefix in the order of f ; a threshold set of h_- is a suffix, whose complement is such a prefix and has the same conductance. Thus SpectralSweep considers a cut at least this good. Finally, the already proved inequality $\mu_2 \leq 2\phi(G)$ gives $\sqrt{2\mu_2} \leq 2\sqrt{\phi(G)}$. $\square$

4.4 Irregular graphs

The volume normalization is the natural version for irregular and weighted graphs. Let symmetric weights $w_{uv} = w_{vu} \geq 0$ define $A_{uv} = w_{uv}$, put $d_v = \sum_u w_{uv} > 0$, and let $D = \text{diag}(d_v)$, $L = D - A$, and

$$\mathcal{L} = I - D^{-1/2}AD^{-1/2}.$$

Define

$$\begin{aligned} \text{vol}(S) &= \sum_{v \in S} d_v, & w(\delta(S)) &= \sum_{u \in S, v \notin S} w_{uv}, \\ \Phi(S) &= \frac{w(\delta(S))}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}}, & \Phi(G) &= \min_{\emptyset \neq S \subsetneq V} \Phi(S). \end{aligned}$$

Zero-degree vertices are removed before these quantities are formed. If $z = D^{1/2}f$, then $z \perp D^{1/2}\mathbf{1}$ exactly when $\sum_v d_v f_v = 0$, and Rayleigh–Ritz gives

$$\mu_2 = \min_{\substack{f \neq 0 \\ \sum_v d_v f_v = 0}} \mathcal{R}_D(f), \quad \mathcal{R}_D(f) = \frac{\sum_{uv} w_{uv} (f_u - f_v)^2}{\sum_v d_v f_v^2},$$

where the numerator sums over unordered weighted edges once. Thus, if z is a second eigenvector of $\mathcal{L}$, the algorithm must sweep $f_v = z_v / \sqrt{d_v}$.

Theorem 4.4 (Cheeger's inequality for weighted graphs). *For the graph above,*

$$\frac{\mu_2}{2} \leq \Phi(G) \leq \sqrt{2\mu_2}.$$

A sweep of $f = D^{-1/2}z$ finds a set attaining the upper bound.

Proof. If the graph is disconnected, a positive-volume component has zero boundary, $\mu_2 = \Phi(G) = 0$, and a threshold between two distinct component values of a zero-eigenvector gives such a sweep cut. Assume henceforth that the graph is connected.

For the easy direction, choose S with $\text{vol}(S) \leq \text{vol}(V)/2$ and set $x = \mathbf{1}_S - [\text{vol}(S)/\text{vol}(V)]\mathbf{1}$. Then $\sum_v d_v x_v = 0$, its energy is $w(\delta(S))$, and

$$\sum_v d_v x_v^2 = \frac{\text{vol}(S) \text{vol}(\bar{S})}{\text{vol}(V)}.$$

Consequently $\mu_2 \leq \mathcal{R}_D(x) \leq 2\Phi(S)$; minimizing over S proves the lower bound on $\Phi(G)$.

For the sweep direction, take a minimizing f and choose a *weighted median* m : both $\{v : f_v > m\}$ and $\{v : f_v < m\}$ have volume at most $\text{vol}(V)/2$. Put $h = f - m\mathbf{1}$. Its energy is unchanged and, because $\sum_v d_v f_v = 0$,

$$\sum_v d_v h_v^2 = \sum_v d_v f_v^2 + m^2 \text{vol}(V) \geq \sum_v d_v f_v^2.$$

Split $h = h_+ - h_-$ into positive and negative parts. The edgewise inequality used in the regular proof shows that one nonzero $g \in \{h_+, h_-\}$ has $\mathcal{R}_D(g) \leq \mu_2$; its support has volume at most $\text{vol}(V)/2$.

For $S_t = \{v : g_v^2 \geq t\}$, weighted coarea gives

$$\int_0^\infty w(\delta(S_t)) dt = \sum_{uv} w_{uv} |g_u^2 - g_v^2|, \quad \int_0^\infty \text{vol}(S_t) dt = \sum_v d_v g_v^2.$$

Since every S_t lies in the smaller-volume support, averaging and then Cauchy–Schwarz yield a nonempty threshold set with

$$\begin{aligned} \Phi(S_t) &\leq \frac{\sum_{uv} w_{uv} |g_u^2 - g_v^2|}{\sum_v d_v g_v^2} \\ &\leq \frac{\sqrt{\sum_{uv} w_{uv} (g_u - g_v)^2} \sqrt{\sum_{uv} w_{uv} (g_u + g_v)^2}}{\sum_v d_v g_v^2} \\ &\leq \sqrt{2\mathcal{R}_D(g)} \leq \sqrt{2\mu_2}, \end{aligned}$$

where the last inequality uses $\sum_{uv} w_{uv} (g_u + g_v)^2 \leq 2 \sum_v d_v g_v^2$. Thresholds of h_+ are prefixes in the order of f ; thresholds of h_- are suffixes, whose complements are prefixes with the same conductance. Hence the sweep finds the asserted set. $\square$

For a d -regular unweighted graph, $\text{vol}(S) = d|S|$ and $\Phi = \phi$, so this theorem specializes to [theorem 4.2](#).

4.5 Two applications of spectral coordinates

Image segmentation. Represent pixels or small regions as vertices and connect visually similar nearby pairs with large edge weights. Shi and Malik’s normalized-cut objective is [\[14\]](#)

$$\text{Ncut}(S, \bar{S}) = \frac{w(\delta(S))}{\text{vol}(S)} + \frac{w(\delta(\bar{S}))}{\text{vol}(\bar{S})}.$$

This is not exactly conductance. If $\text{vol}(S) \leq \text{vol}(\bar{S})$, then $\Phi(S) \leq \text{Ncut}(S, \bar{S}) \leq 2\Phi(S)$. Both objectives lead to normalized-Laplacian relaxations: an eigenvector supplies a continuous segmentation coordinate, and thresholding turns it into a discrete region.

Graph visualization. Map a vertex v to two nontrivial Laplacian coordinates, for example $(f_2(v), f_3(v))$. Vertices that interact similarly tend to appear nearby. For the cycle C_n with vertices $j = 0, \dots, n-1$, the two coordinates

$$f_2(j) = \cos(2\pi j/n), \quad f_3(j) = \sin(2\pi j/n)$$

span the first nontrivial Laplacian eigenspace, and the spectral embedding places the vertices exactly on a circle.

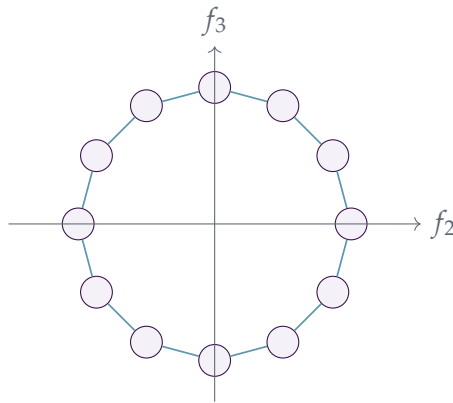


Figure 3: The exact two-dimensional spectral embedding of a cycle.

Part III preview takeaway

The spectral algorithm has a square-root guarantee, not a constant-factor guarantee: it returns S with $\phi(S) \leq 2\sqrt{\phi(G)}$. Its power is that the vector relaxation is computable and the rounding proof is constructive. When additional spectral gaps or cluster structure are present, substantially sharper guarantees may be possible.

5 Exercises

Problems marked $\star$ require an idea beyond substituting into a displayed formula. Problems marked $\star\star$ are suitable as short projects.

Part I: Randomized algorithms and randomization techniques

1. **Amplification.** Starting only from $p \geq 2/\lceil n(n-1) \rceil$, derive the exact number of independent Karger trials sufficient for failure probability at most an arbitrary $\delta \in (0, 1)$. Specialize to $\delta = n^{-3}$.
2. **Contraction invariant.** Formalize a contracted graph as a partition of the original vertex set. Prove directly that every cut of the contracted graph lifts to a cut of equal size in the original graph.
3. **Tight example.** Prove that C_n has exactly $\binom{n}{2}$ distinct minimum cuts as unordered bipartitions, and that RandomContraction returns each one with probability exactly $1/\binom{n}{2}$.
4. $\star$ **Weighted contraction.** Extend [theorem 2.3](#) to a graph with positive edge weights, where $w(\delta(S)) = \sum_{e \in \delta(S)} w_e$, by choosing an edge with probability proportional to its weight. State carefully what changes in the runtime model.
5. $\star$ **Karger-Stein constants.** Using [section 2.4](#), give an explicit upper bound on the recursion depth and an explicit constant $c > 0$ for which $p(n) \geq c/\log_2 n$ for all sufficiently large n .

Part II: Approximation algorithms for combinatorial optimization

6. **Conditional expectation formula.** For a partial Max-Cut assignment with placed sets S, T and unassigned set U , let $E(U)$ denote the edges with both endpoints in U . Verify

$$\mathbb{E}[Z \mid S, T] = |E(S, T)| + \frac{1}{2}|E(S \cup T, U)| + \frac{1}{2}|E(U)|.$$

Use it to rederive the greedy choice.

7. **Weighted Max-Cut.** For nonnegative weights, prove that the random-cut and greedy arguments return a cut of weight at least half the total edge weight and hence a $1/2$ -approximation. Give an example showing why the proof fails if negative weights are allowed.
8. **Pairwise but not mutual.** Let A, B be independent fair bits and $C = A \oplus B$. Prove that every pair among A, B, C is independent, but the triple is not. What cut-value distribution does this produce on a triangle whose three vertices receive A, B, C ?
9. **★ Fast seed scoring.** For labels $a_v \in \mathbb{F}_2^\ell$, let c_z count edges uv with $a_u + a_v = z$. Prove that the cut produced by seed r has value

$$\frac{1}{2} \left(m - \sum_{z \in \mathbb{F}_2^\ell} c_z (-1)^{\langle z, r \rangle} \right).$$

Explain how a Walsh–Hadamard transform evaluates all seeds in $O(m + n \log n)$ operations on a word RAM with word size $\Omega(\log(m + n))$ bits.

Part III: Spectral algorithms and spectral graph theory

10. **Connected components and eigenvalue multiplicity.** Complete the component-wise argument in [proposition 4.1](#) to prove that the multiplicity of eigenvalue d is exactly the number of connected components.
11. **Test Cheeger's constants.** Compute μ_2 and $\phi(G)$ for K_n , C_n , and $K_{r,r}$ for $n \geq 3$ and $r \geq 2$. Include any parity cases in the exact conductance values. Compare both sides of [theorem 4.2](#) and identify where the inequalities are loose.
12. **★ Linear-time sweep evaluation.** Once the vertices are sorted, design an $O(m + n)$ procedure that computes every $|\delta(P_i)|$. Hint: when u_i enters the prefix, classify its incident edges by whether the other endpoint has already entered.
13. **★ Irregular Cheeger sweep.** With $L = D - A$ and $D = \text{diag}(d_v)$, start from the generalized eigenproblem $Lf = \mu Df$. Reproduce the easy direction $\mu_2 \leq 2\Phi(G)$ and identify exactly where vertex counts become volumes.

A Probability and randomization toolkit

Every random experiment in the main text has a finite set Ω of possible outcomes. A probability distribution assigns numbers $p(\omega) \geq 0$ with $\sum_{\omega \in \Omega} p(\omega) = 1$. An event is a set $A \subseteq \Omega$, and

$$\mathbb{P}(A) = \sum_{\omega \in A} p(\omega).$$

This immediately implies $\mathbb{P}(\bar{A}) = 1 - \mathbb{P}(A)$ and monotonicity: $A \subseteq B$ implies $\mathbb{P}(A) \leq \mathbb{P}(B)$.

A.1 Conditional probability and adaptive sequences

If $\mathbb{P}(B) > 0$, the conditional probability of A given B is

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}.$$

Conditioning is essential in adaptive randomized algorithms: later random choices are analyzed after the history of earlier choices has been exposed.

Proposition A.1 (Chain rule). *Let $A_1, \dots, A_r$ be events and put $H_i = \bigcap_{j=1}^i A_j$, with $H_0 = \Omega$. If $\mathbb{P}(H_i) > 0$ for $0 \leq i < r$, then*

$$\mathbb{P}\left(\bigcap_{i=1}^r A_i\right) = \prod_{i=1}^r \mathbb{P}(A_i \mid H_{i-1}).$$

Proof. The definition gives $\mathbb{P}(H_i) = \mathbb{P}(H_{i-1})\mathbb{P}(A_i \mid H_{i-1})$. Multiplying these identities telescopes. The positivity assumption is not cosmetic: ordinary conditional probability is undefined when its conditioning event has probability zero. $\square$

A.2 Expectation and conditional expectation

A random variable is a function $X : \Omega \rightarrow \mathbb{R}$. Its expectation is

$$\mathbb{E}[X] = \sum_{\omega \in \Omega} p(\omega)X(\omega).$$

For an event A , the indicator $\mathbf{1}_A$ equals 1 on A and 0 outside A , so $\mathbb{E}[\mathbf{1}_A] = \mathbb{P}(A)$.

For an event H with $\mathbb{P}(H) > 0$, the expectation conditioned on H is

$$\mathbb{E}[Z \mid H] = \sum_{\omega \in H} \mathbb{P}(\omega \mid H)Z(\omega) = \frac{1}{\mathbb{P}(H)} \sum_{\omega \in H} p(\omega)Z(\omega).$$

Theorem A.2 (Linearity of expectation). *For arbitrary random variables $X_1, \dots, X_r$ and constants $a_1, \dots, a_r$,*

$$\mathbb{E}\left[\sum_{i=1}^r a_i X_i\right] = \sum_{i=1}^r a_i \mathbb{E}[X_i].$$

No independence assumption is needed.

Proof. Interchange two finite sums:

$$\sum_{\omega} p(\omega) \sum_i a_i X_i(\omega) = \sum_i a_i \sum_{\omega} p(\omega) X_i(\omega).$$

$\square$

Averaging principle

If Z is a real random variable on a finite space, then some outcome ω satisfies $Z(\omega) \geq \mathbb{E}[Z]$. Otherwise every value would be strictly below its weighted average. This is the basic probabilistic method: positive average performance certifies the existence of a good deterministic object.

Suppose H records the choices exposed so far and a fresh random bit B is uniform and independent of H . Partitioning H into $H \cap \{B = 0\}$ and $H \cap \{B = 1\}$ gives the two-child identity

$$\mathbb{E}[Z \mid H] = \frac{1}{2}\mathbb{E}[Z \mid H, B = 0] + \frac{1}{2}\mathbb{E}[Z \mid H, B = 1].$$

Consequently, at least one choice of B preserves or increases the conditional expectation. Repeating this argument one bit at a time yields the *method of conditional expectations*, provided the two conditional expectations can be computed efficiently.

A.3 Independence and amplification

Events $A_1, \dots, A_r$ are mutually independent if for every $I \subseteq \{1, \dots, r\}$,

$$\mathbb{P}\left(\bigcap_{i \in I} A_i\right) = \prod_{i \in I} \mathbb{P}(A_i).$$

Discrete random variables are mutually independent when the analogous identity holds for all choices of values. Pairwise independence asks for this only for pairs. Pairwise independence is strictly weaker, but it is already enough for the random-cut calculation in section 3.

Proposition A.3 (Independent repetition). *If each of R independent trials succeeds with probability at least p , then all trials fail with probability at most*

$$(1 - p)^R \leq e^{-pR}.$$

Thus $R \geq p^{-1} \ln(1/\delta)$ makes the failure probability at most δ .

Proof. Independence multiplies the failure probabilities. The inequality $1 - x \leq e^{-x}$ for $x \geq 0$ gives the bound. $\square$

Throughout these notes, *with high probability* means probability $1 - o(1/n)$. Whenever amplification is invoked, the constant can, for example, be chosen to make the failure probability at most n^{-3} .

B Linear-algebra toolkit

This appendix records the generic facts used in the spectral showcase. The graph-specific objects - adjacency matrices, Laplacians, conductance, and sweep rounding - are developed in the main text.

For vectors $x, y \in \mathbb{R}^n$, write $\langle x, y \rangle = x^T y$ and $\|x\|_2 = \sqrt{\langle x, x \rangle}$. The notation $x \perp y$ means $\langle x, y \rangle = 0$.

Proposition B.1 (Cauchy–Schwarz). *For all $x, y \in \mathbb{R}^n$,*

$$|\langle x, y \rangle| \leq \|x\|_2 \|y\|_2.$$

Proof. The claim is immediate if $y = 0$. Otherwise, $0 \leq \|x - ty\|_2^2$ for every real t . Substituting $t = \langle x, y \rangle / \|y\|_2^2$ and rearranging gives $\langle x, y \rangle^2 \leq \|x\|_2^2 \|y\|_2^2$. $\square$

Theorem B.2 (Spectral theorem). *Every real symmetric matrix $M \in \mathbb{R}^{n \times n}$ has real eigenvalues and an orthonormal basis of real eigenvectors. Equivalently, there are an orthogonal matrix Q and a real diagonal matrix Λ such that $M = Q\Lambda Q^T$.*

Proposition B.3 (Rayleigh–Ritz). *Let M be real and symmetric, with eigenvalues $\theta_1 \leq \theta_2 \leq \dots \leq \theta_n$ and corresponding orthonormal eigenvectors $q_1, \dots, q_n$. Then*

$$\min_{x \neq 0} \frac{x^T M x}{x^T x} = \theta_1, \quad \min_{\substack{x \neq 0 \\ x \perp q_1}} \frac{x^T M x}{x^T x} = \theta_2.$$

Proof. Write $x = \sum_i a_i q_i$. Then

$$\frac{x^T M x}{x^T x} = \frac{\sum_i \theta_i a_i^2}{\sum_i a_i^2},$$

a weighted average of the eigenvalues. Its minimum is θ_1 , attained at q_1 . Under $x \perp q_1$, one has $a_1 = 0$, so the minimum becomes θ_2 , attained at q_2 . $\square$

Bibliographic notes

Part I: randomized algorithms and randomization techniques. Karger’s contraction algorithm first appeared in his 1993 SODA paper [1]; the recursive improvement is due to Karger and Stein [2]. Karger’s later near-linear randomized algorithm is developed in [3]. Stoer and Wagner give a classical deterministic algorithm [4]; a modern deterministic near-linear result for weighted graphs is due to Henzinger, Li, Rao, and Wang [5]. Motwani and Raghavan [15] provide broader background on randomized algorithms.

Part II: approximation algorithms for combinatorial optimization. Classical NP-completeness results for Max-Cut appear in [6]. The semidefinite approximation is due to Goemans and Williamson [7]. Khot introduced the Unique Games Conjecture [8]; the conditional optimality of the Goemans–Williamson ratio is established in [9]. Williamson and Shmoys [16] give a broad treatment of approximation algorithms.

Part III: spectral algorithms and spectral graph theory. Chung’s monograph [10] is a standard reference for spectral graph theory. Discrete graph analogues of Cheeger’s geometric inequality were developed by Dodziuk and by Alon–Milman [11, 12]; a weighted treatment appears in [13]. The image-segmentation connection is classically developed by Shi and Malik [14].

References

- [1] David R. Karger. Global min-cuts in RNC, and other ramifications of a simple min-cut algorithm. In *Proceedings of SODA*, pages 21-30, 1993. <https://doi.org/10.5555/313559.313605>.
- [2] David R. Karger and Clifford Stein. A new approach to the minimum cut problem. *Journal of the ACM*, 43(4):601-640, 1996. <https://doi.org/10.1145/234533.234534>.
- [3] David R. Karger. Minimum cuts in near-linear time. *Journal of the ACM*, 47(1):46-76, 2000. <https://doi.org/10.1145/331605.331608>.
- [4] Mechthild Stoer and Frank Wagner. A simple min-cut algorithm. *Journal of the ACM*, 44(4):585-591, 1997. <https://doi.org/10.1145/263867.263872>.
- [5] Monika Henzinger, Jason Li, Satish Rao, and Di Wang. Deterministic near-linear time minimum cut in weighted graphs. In *Proceedings of SODA*, pages 3089-3139, 2024. <https://doi.org/10.1137/1.9781611977912.111>.
- [6] Michael R. Garey, David S. Johnson, and Larry Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237-267, 1976. [https://doi.org/10.1016/0304-3975\(76\)90059-1](https://doi.org/10.1016/0304-3975(76)90059-1).
- [7] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115-1145, 1995. <https://doi.org/10.1145/227683.227684>.
- [8] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of STOC*, pages 767-775, 2002. <https://doi.org/10.1145/509907.510017>.
- [9] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for Max-Cut and other 2-variable CSPs. *SIAM Journal on Computing*, 37(1):319-357, 2007. <https://doi.org/10.1137/S0097539705447372>.
- [10] Fan R. K. Chung. *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics 92, AMS, 1997. <https://doi.org/10.1090/cbms/092>.
- [11] Jozef Dodziuk. Difference equations, isoperimetric inequality and transience of certain random walks. *Transactions of the American Mathematical Society*, 284(2):787-794, 1984. <https://doi.org/10.1090/S0002-9947-1984-0743744-X>.
- [12] Noga Alon and Vitali D. Milman. λ_1 , isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory, Series B*, 38(1):73-88, 1985. [https://doi.org/10.1016/0095-8956\(85\)90092-9](https://doi.org/10.1016/0095-8956(85)90092-9).
- [13] Fan R. K. Chung and Kevin Oden. Weighted graph Laplacians and isoperimetric inequalities. *Pacific Journal of Mathematics*, 192(2):257-273, 2000. <https://doi.org/10.2140/pjm.2000.192.257>.
- [14] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888-905, 2000. <https://doi.org/10.1109/34.868688>.
- [15] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- [16] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.