Advanced Approximation Algorithms Notes

Instructors: Anupam Gupta and Pravesh K. Kothari
Notes compiled by: Lichen Zhang
Carnegie Mellon University
15-854(B) Advanced Approximation Algorithms

Fall 2021

Contents

1 Introduction and Set Cover **6**
1.1 Optimization problems and approximation ratios . . . 6
1.2 A finer-grained landscape . . . 7
1.3 The set cover problem . . . 8
1.4 The greedy algorithm . . . 8
1.5 Max-k-coverage . . . 9

2 Set Cover and Max-Coverage: LP-Based Techniques **10**
2.1 An algorithmic gap for greedy . . . 10
2.2 Greedy for weighted set cover . . . 11
2.3 Linear programming relaxations . . . 13
2.4 Randomized rounding for set cover . . . 13
2.5 Algorithmic gaps and integrality gaps . . . 14
2.6 Max-k-coverage: derandomizing by pipage rounding . . . 15

3 Hardness of Approximating Max-Coverage **18**
3.1 Hardness of approximation and label cover . . . 18
3.2 The reduction . . . 18
3.3 Soundness . . . 19
3.4 Consequences . . . 22

4 Max-Cut: The Goemans–Williamson Algorithm **24**
4.1 The Max-Cut problem . . . 24
4.2 A 1/2-approximation . . . 24
4.3 Quadratic and linear programs for Max-Cut . . . 25
4.4 A semidefinite relaxation . . . 27
4.5 Semidefinite programs . . . 28
4.6 Rounding: the Goemans–Williamson theorem . . . 29
4.7 Basic facts . . . 30
4.8 Proof of the Goemans–Williamson theorem . . . 31
4.9 Beyond the Goemans–Williamson algorithm . . . 34

5 Max-Cut: Integrality Gaps and Rounding Gaps **35**
5.1 The Goemans–Williamson algorithm, revisited . . . 35
5.2 Integrality gaps and odd cycles . . . 36
5.3 The Feige–Schechtman integrality gap . . . 38
5.4 Rounding gaps and Karloff's theorem . . . 43
5.5 Hypercube versus sphere . . . 45

6 Sparsest Cut: Region Growing and the Leighton–Rao LP **46**
6.1 The sparsest cut problem . . . 46
6.2 The Leighton–Rao linear program . . . 47
6.3 The rounding algorithm . . . 48
6.4 Low-diameter decomposition by region growing . . . 49
6.5 The dense case . . . 51
6.6 The integrality gap: expander graphs . . . 52

7 Sparsest Cut: Metric Embeddings **54**
7.1 Sparsest cut with general demands . . . 54
7.2 The metric relaxation and cut metrics . . . 55
7.3 The cut cone is the set of ℓ_1 metrics . . . 55
7.4 Bourgain's theorem and the rounding . . . 57
7.5 Proof of Bourgain's theorem . . . 57
7.6 Remarks . . . 59

8 Sparsest Cut: Spectral Partitioning and the Cheeger Bounds **60**
8.1 A spectral lower bound . . . 60
8.2 The upper bound: spectral partitioning . . . 61
8.3 Tightness and discussion . . . 64

9 Max-Cut and Eigenvalues of Graphs **66**
9.1 Max-cut and the largest eigenvalue of the Laplacian . . . 66
9.2 The bipartiteness gap and Trevisan's inequality . . . 67
9.3 Generalization to irregular graphs . . . 70
9.4 A spectral algorithm for Max-Cut . . . 73

10 The Arora–Rao–Vazirani Algorithm I **77**
10.1 Sparsest cut as a quadratic program . . . 77
10.2 The semidefinite relaxation . . . 78
10.3 Two rounding attempts . . . 79
10.4 The structure theorem and the proof of the main theorem . . . 81

11 The Arora–Rao–Vazirani Algorithm II **83**
11.1 The structure theorem and vertex separators . . . 83
11.2 A randomized algorithm for vertex separators . . . 84
11.3 The main technical lemma . . . 86
11.4 Chaining . . . 87

12 Local Search Algorithms for Facility Location **90**
12.1 Facility location and k-median . . . 90
12.2 Local search . . . 90
12.3 A lower bound on the locality gap . . . 91
12.4 Local optima are 5-approximations . . . 93
12.5 Polynomial running time . . . 96
12.6 Summary . . . 97

13 LP Rounding Algorithms for Facility Location **98**
13.1 The linear program . . . 98
13.2 LP duality . . . 99
13.3 Clustering and a deterministic 4-approximation . . . 100
13.4 Randomized rounding . . . 102

14 Hardness for Facility Location; Primal-Dual Algorithms **104**
14.1 Hardness of facility location . . . 104
14.2 The primal-dual method . . . 106
14.3 Warm-up: vertex cover . . . 107
14.4 A primal-dual algorithm for facility location . . . 108
14.5 Summary . . . 111

15 Introduction to Hardness of Approximation: Vertex Cover and Sparsest Cut **112**
15.1 Reductions and NP-completeness . . . 112
15.2 Hardness of approximation and the PCP theorem . . . 113
15.3 Independent set and vertex cover . . . 114
15.4 Max-Cut . . . 117
15.5 Non-uniform sparsest cut . . . 117
15.6 What is known . . . 119

16 The Unique Games Conjecture and Boolean Fourier Analysis **120**
16.1 The problem and the conjecture . . . 120
16.2 Optimal hardness for Max-Cut and gadget reductions . . . 122
16.3 Integrality gap and rounding gap instances . . . 122
16.4 Fourier analysis of Boolean functions . . . 124
16.5 Influence of variables . . . 125
16.6 Low-degree influence . . . 127

17 UGC-Based Hardness from Dictator vs. Spread-Out Function Tests I **129**
17.1 Dictator tests . . . 129
17.2 A two-query test . . . 130
17.3 Low-influence functions and the invariance principle . . . 132
17.4 From dictator tests to hardness . . . 133
17.5 The reduction for Max-Cut . . . 134

18 UGC-Based Hardness from Dictator vs. Spread-Out Function Tests II **137**
18.1 The reduction . . . 137
18.2 Completeness . . . 138
18.3 Soundness . . . 138
18.4 Hardness of Max-Cut . . . 140
18.5 Beyond Max-Cut . . . 141

19 Scheduling Problems: LP Structure and Strengthening LPs **143**
19.1 A single machine with release dates . . . 143
19.2 A single machine with precedence constraints . . . 144
19.3 The generalized assignment problem . . . 146
19.4 Lessons . . . 150

20 Iterated Rounding **151**
20.1 Vertices and tight constraints . . . 151
20.2 The Beck–Fiala theorem . . . 151
20.3 Bin packing with few item types . . . 153
20.4 The generalized assignment problem with costs . . . 154

21 Strengthening Convex Relaxations: SDP Hierarchies I **157**
21.1 The knapsack problem . . . 157
21.2 A greedy algorithm and the linear programming relaxation . . . 157
21.3 Strengthening relaxations . . . 159
21.4 Pseudo-distributions . . . 160
21.5 The Cauchy–Schwarz inequality for pseudo-distributions . . . 161
21.6 Pseudo-moments and the moment matrix . . . 162
21.7 Example: Max-Cut . . . 163
21.8 The sum-of-squares hierarchy . . . 163

22 Strengthening Convex Relaxations: SDP Hierarchies II **165**
22.1 Constrained pseudo-distributions . . . 165
22.2 Pseudo-distributions on $\{0,1\}^n$ and low-degree transformations . . . 166
22.3 Local distributions . . . 166
22.4 Two facts about pseudo-distributions for Knapsack . . . 167
22.5 The sum-of-squares relaxation of Knapsack . . . 168
22.6 Reduced polynomials and conditioning on the large items . . . 169

23 Strengthening Convex Relaxations: SDP Hierarchies III **172**
23.1 The setting . . . 172
23.2 Completing the knapsack analysis . . . 173

24 Gaussian Elimination, Linear Systems, and Related Topics **174**
24.1 Computing with real numbers . . . 174
24.2 Rational inputs and the size of solutions . . . 175
24.3 Gaussian elimination . . . 178
24.4 Bit complexity of Gaussian elimination . . . 180

25 The Ellipsoid Algorithm, This Time Without Ellipsoids **184**
25.1 Feasibility and separation oracles . . . 184
25.2 The general approach . . . 184
25.3 Three choices of the region . . . 186
25.4 The simplices method . . . 186
25.5 Solving linear programs . . . 189

26 Eigenvalue Computations **191**
26.1 The problem . . . 191
26.2 Power iteration . . . 191
26.3 Inverse iteration and shifts . . . 192
26.4 Rayleigh quotient iteration . . . 193
26.5 The QR factorization and similar matrices . . . 194
26.6 Orthogonal iteration . . . 195
26.7 The QR algorithm . . . 195

References **197**

1 Introduction and Set Cover

Lecture of August 30, 2021, by Anupam Gupta. Based on the lecturer's handwritten notes.

Many optimization problems are NP-hard, yet they differ sharply in how well they can be approximated in polynomial time. This lecture sets up the framework of the course: optimization problems, approximation ratios, and the c-versus-s search and decision problems that connect algorithms with hardness results. It then turns to the set cover problem. We prove that the greedy algorithm uses at most about $\ln n$ times as many sets as an optimal cover, that greedy covers more than a $1 - 1/e$ fraction of the elements for max-k-coverage whenever k sets can cover everything, and that any guarantee of this kind for max-k-coverage yields a logarithmic approximation for set cover. Lower bounds for greedy, weighted set cover and LP-based algorithms follow in Lecture 2, and the hardness of max-coverage in Lecture 3.

1.1 Optimization problems and approximation ratios

A decision problem asks a yes-or-no question; an optimization problem asks for a best solution. Each classical decision problem comes with natural optimization versions, and their difficulty need not match that of the decision problem:

- 3SAT (NP-complete) against Max 3SAT (maximize the number of satisfied clauses) and Min 3-UNSAT (minimize the number of unsatisfied clauses);
- 2SAT (in P) against Max 2SAT and Min 2-UNSAT;
- 3-coloring (NP-complete) against minimum coloring (use as few colors as possible) and Max 3-cut (color with three colors so as to maximize the number of edges whose endpoints get different colors);
- 2-coloring (in P) against Max 2-cut and Min 2-uncut.

Definition 1.1. An *optimization problem* specifies a set of *instances*. For each instance I it specifies a finite nonempty set $\mathrm{sols}(I)$ of *valid solutions*, such that membership $S \in \mathrm{sols}(I)$ can be checked in polynomial time, and a *value function* $\mathrm{val}_I \colon \mathrm{sols}(I) \to \mathbb{R}_{\geq 0}$, which can be evaluated in polynomial time. The problem is either a *minimization* or a *maximization* problem, and its optimum is

$$\mathrm{OPT}(I) \coloneqq \min_{S \in \mathrm{sols}(I)} \mathrm{val}_I(S) \quad \text{or} \quad \mathrm{OPT}(I) \coloneqq \max_{S \in \mathrm{sols}(I)} \mathrm{val}_I(S),$$

respectively. An *algorithm* for the problem maps each instance I to a solution in $\mathrm{sols}(I)$; we write $\mathrm{ALG}(I)$ both for this solution and for its value. The aim is to make $\mathrm{ALG}(I)$ as small (for minimization) or as large (for maximization) as possible.

Example 1.2 (Max 3SAT)**.** The instances of Max 3SAT are 3CNF formulas with m clauses on n variables, in which every clause is a disjunction of exactly three literals on three distinct variables. The solutions are the truth assignments, that is, the assignments of true or false to each variable. The value of an assignment is the fraction of satisfied clauses,

$$\mathrm{val}(a) = \frac{\#\{\text{clauses satisfied by } a\}}{m} \in [0, 1].$$

A simple algorithm sets each variable to true or false independently with probability $1/2$. It produces an assignment that satisfies a $7/8$ fraction of the clauses in expectation. Indeed, a clause is unsatisfied exactly when its three literals are all false. Since the three variables are distinct, this happens with probability $(1/2)^3 = 1/8$, and linearity of expectation gives an expected value of $7/8$. Whether a polynomial-time algorithm can do better is answered in Fact 1.4.

Definition 1.3. Let $\alpha > 0$. An algorithm for a maximization problem is an *α-approximation*, where $\alpha \leq 1$, if $\mathrm{ALG}(I) \geq \alpha \cdot \mathrm{OPT}(I)$ for every instance I. An algorithm for a minimization problem is an α-approximation, where $\alpha \geq 1$, if $\mathrm{ALG}(I) \leq \alpha \cdot \mathrm{OPT}(I)$ for every instance I. The *approximation ratio* of an algorithm is the worst-case value of $\mathrm{ALG}(I)/\mathrm{OPT}(I)$ over all instances with $\mathrm{OPT}(I) > 0$. For a randomized algorithm, $\mathrm{ALG}(I)$ is replaced by its expectation.

Since every value of Max 3SAT is at most 1, the random assignment of Example 1.2 is a $7/8$-approximation in expectation.

1.2 A finer-grained landscape

NP-hardness does not distinguish between problems that are approximable to very different degrees. The following facts illustrate the range.

Fact 1.4. *The following hold.*

1) *The traveling salesman problem on points in the Euclidean plane is NP-hard, but it has a* polynomial-time approximation scheme*: for every fixed $\varepsilon > 0$ there is a polynomial-time $(1+\varepsilon)$-approximation [Aro98].*

2) *Minimum set cover is NP-hard, but it has a polynomial-time $O(\log n)$-approximation (Theorem 1.9 below).*

3) *Max 3SAT is NP-hard, and the $7/8$-approximation of Example 1.2 is best possible: for every $\varepsilon > 0$, a polynomial-time $(7/8+\varepsilon)$-approximation would imply $\mathrm{P} = \mathrm{NP}$ [Hås01].*

4) *For Max Clique on graphs with n vertices and every $\varepsilon > 0$, a polynomial-time algorithm that always finds a clique of size at least $\mathrm{OPT}/n^{1-\varepsilon}$ would imply $\mathrm{P} = \mathrm{NP}$ [Hås99, Zuc07].*

The course therefore has three kinds of goals: approximation algorithms (positive results), hardness of approximation (negative results), and integrality gaps and algorithmic gaps, which show the limitations of specific algorithms and relaxations.

Hardness results are most naturally stated for problems with a gap between two thresholds. We state the definitions for maximization problems.

Definition 1.5. Let $0 \leq s \leq c$. The *c-vs-s search problem* asks for an algorithm that, on every instance I with $\mathrm{OPT}(I) \geq c$, outputs a solution of value at least s; on other instances it may output any solution. The *c-vs-s decision problem* asks for an algorithm that answers YES on every instance with $\mathrm{OPT}(I) \geq c$ and NO on every instance with $\mathrm{OPT}(I) < s$; on instances with $s \leq \mathrm{OPT}(I) < c$ either answer is acceptable.

Proposition 1.6. *If there is a polynomial-time algorithm for the c-vs-s search problem, then there is one for the c-vs-s decision problem.*

Proof. Run the search algorithm on I, compute the value $\mathrm{ALG}(I)$ of its output, and answer YES if $\mathrm{ALG}(I) \geq s$ and NO otherwise. If the answer is YES, then $\mathrm{OPT}(I) \geq \mathrm{ALG}(I) \geq s$, so YES is acceptable. If the answer is NO, then $\mathrm{ALG}(I) < s$, so $\mathrm{OPT}(I) < c$, because on instances with $\mathrm{OPT}(I) \geq c$ the search algorithm outputs a solution of value at least s; hence NO is acceptable. □

1.3 The set cover problem

Set cover is a general problem that captures many others; Young's survey [You16] of greedy algorithms for it collects many references.

Definition 1.7. An instance of *set cover* is a set system $\mathcal{F} = (U, \{S_1, \ldots, S_m\})$ with $S_i \subseteq U$ for all i, $n = |U|$, and $\bigcup_{i=1}^{m} S_i = U$, together with costs $c_1, \ldots, c_m \geq 0$. A solution is a subcollection $J \subseteq [m]$ with $\bigcup_{i \in J} S_i = U$, called a *cover*. In *weighted* set cover the value of J is its cost $\mathrm{cost}(J) = \sum_{i \in J} c_i$; in *unweighted* set cover it is its cardinality $|J|$, which is the case of unit costs. Both are minimization problems.

1.4 The greedy algorithm

Algorithm 1.8 (Greedy for unweighted set cover)**.** While some element of U is not covered, pick a set S_i that covers the largest number of uncovered elements (ties are broken arbitrarily).

Theorem 1.9 (Johnson [Joh74])**.** *For every instance of unweighted set cover with $n \geq 1$ elements, the greedy algorithm outputs a cover with at most $\lfloor \mathrm{OPT} \cdot \ln n \rfloor + 1$ sets. In particular, greedy is a polynomial-time $(1 + \ln n)$-approximation for unweighted set cover.*

Proof. Let $K = \mathrm{OPT}$, and let n_t be the number of elements that are still uncovered after greedy has picked t sets, so $n_0 = n$.

Claim 1.10. *At any time, some set covers at least a $1/K$ fraction of the currently uncovered elements.*

Proof. The K sets of an optimal cover cover all elements, in particular the n_t uncovered ones. Hence one of these K sets covers at least n_t/K of them. □

Greedy picks a set covering at least as many uncovered elements as any other set, so by Claim 1.10,

$$n_{t+1} \leq n_t\Big(1 - \frac{1}{K}\Big) \quad \text{and hence} \quad n_t \leq n\Big(1 - \frac{1}{K}\Big)^t \leq n\, e^{-t/K}$$

for every t, using $1 - x \leq e^{-x}$. Take $t = \lfloor K \ln n \rfloor + 1$, the least integer exceeding $K \ln n$. Then $n_t \leq n\, e^{-t/K} < n\, e^{-\ln n} = 1$, so $n_t = 0$, and greedy has stopped after at most t steps. Finally, $\lfloor K \ln n \rfloor + 1 \leq K \ln n + K = (1 + \ln n)K$, since $K \geq 1$. □

Remark 1.11. The additive term cannot simply be dropped: for $n = 2$ and the single set $S_1 = U$, every algorithm uses one set, while $\ln 2 < 1$.

1.5 Max-k-coverage

Definition 1.12. An instance of *max-k-coverage* is a set system $(U, \{S_1, \dots, S_m\})$ with $n = |U|$ and an integer $k \geq 1$. A solution is a subcollection of at most k of the sets. Its value is the number of elements it covers; we measure it as the fraction of U covered, so that values lie in $[0, 1]$, and the c-vs-s problems of Definition 1.5 refer to this fraction.

Greedy for max-k-coverage picks k sets, each time one that covers the largest number of uncovered elements.

Proposition 1.13. *Greedy solves the 1-vs-$(1 - 1/e)$ search problem for max-k-coverage: if some k sets cover U, then the k sets picked by greedy cover more than $(1 - 1/e)\, n$ elements.*

Proof. Let n_t be the number of uncovered elements after t greedy steps. Since k sets cover all of U, the argument of Claim 1.10 with $K = k$ gives $n_{t+1} \leq n_t(1 - 1/k)$. After k steps,

$$n_k \leq n\Big(1 - \frac{1}{k}\Big)^k < n\, e^{-1},$$

so the fraction of covered elements is $(n - n_k)/n > 1 - 1/e$. □

Conversely, a good algorithm for max-k-coverage gives a good algorithm for set cover.

Proposition 1.14. *Let $\gamma \in (0, 1)$. If there is a polynomial-time algorithm A for the 1-vs-$(1 - \gamma)$ search problem for max-k-coverage, then there is a polynomial-time algorithm for unweighted set cover that outputs a cover with at most $(\lfloor \log_{1/\gamma} n \rfloor + 1) \cdot \mathrm{OPT} \leq (\log_{1/\gamma} n + 1) \cdot \mathrm{OPT}$ sets.*

Proof. Let $T = \lfloor \log_{1/\gamma} n \rfloor + 1$, so that $\gamma^T n < 1$. For each guess $K \in \{1, \dots, m\}$ of the optimum, run the following procedure. Start with all elements uncovered. In each of T rounds (stopping early once every element is covered), run A with budget K on the set system restricted to the uncovered elements (the sets $S_i \cap U'$, where U' is the set of uncovered elements), add the at most K sets it returns to the solution, and mark their elements as covered. Among the guesses for which the procedure covers U, output the smallest solution.

Consider the guess $K = \mathrm{OPT}$. In every round, the K sets of an optimal cover cover all uncovered elements, so the restricted instance has value 1, and A returns at most K sets covering at least a $1 - \gamma$ fraction of the uncovered elements. Hence after T rounds the number of uncovered elements is at most $\gamma^T n < 1$, that is, zero. The procedure has then produced a cover with at most $TK = T \cdot \mathrm{OPT}$ sets, so the output has at most this many sets. The running time is mT runs of A plus polynomial overhead. □

Corollary 1.15. *Applying Proposition 1.14 to greedy, which solves the 1-vs-$(1 - 1/e)$ search problem by Proposition 1.13, gives a polynomial-time $(\ln n + 1)$-approximation for unweighted set cover.*

Proof. Take $\gamma = 1/e$, so that $\log_{1/\gamma} n = \ln n$. □

Remark 1.16. Read contrapositively, Proposition 1.14 transfers hardness from set cover to max-k-coverage: if approximating set cover within $(1 - \varepsilon) \ln n$ is hard for every $\varepsilon > 0$, then the 1-vs-$(1 - 1/e + \delta)$ problem for max-k-coverage is hard for every $\delta \in (0, 1/e)$. In detail, let $\delta \in (0, 1/e)$ and $\gamma = 1/e - \delta$. Then $\ln(1/\gamma) = 1 + \beta$ with $\beta = \ln(1/(1 - e\delta)) > 0$, so an algorithm for the 1-vs-$(1 - \gamma)$ search problem yields a set cover algorithm with ratio at most $\ln n/(1 + \beta) + 1$. With $\varepsilon = \beta/(2(1 + \beta))$, this is at most $(1 - \varepsilon) \ln n$ whenever $\varepsilon \ln n \geq 1$, that is, for all $n \geq e^{1/\varepsilon}$. The hardness of set cover is a little more complicated to prove [DS14]; a hardness result for max-coverage is proved directly in Lecture 3.

2 Set Cover and Max-Coverage: LP-Based Techniques

Lecture of September 1, 2021, by Anupam Gupta. Based on the lecturer's handwritten notes.

This lecture continues the study of set cover begun in Lecture 1. We first show that the logarithmic guarantee of the greedy algorithm is essentially tight, then extend greedy to weighted set cover and prove its H_n guarantee by a charging argument. The second half introduces the relax-and-round paradigm for linear programs: randomized rounding of the set cover LP gives an $O(\log n)$-approximation, and a hypercube instance shows that the integrality gap of this LP is $\Omega(\log n)$, so no rounding of it can do much better. Finally, for max-k-coverage, pipage rounding turns an optimal LP solution into k sets covering at least a $1 - 1/e$ fraction of the optimum, deterministically. Chapter 1 of [WS11] covers most of these set cover algorithms.

Throughout the lecture, $(U, \{S_1, \ldots, S_m\})$ is a set system with $n = |U| \geq 1$ elements and $\bigcup_i S_i = U$, as in Lecture 1; $c_S \geq 0$ is the cost of a set S, and OPT is the optimal value.

2.1 An algorithmic gap for greedy

The greedy algorithm of Lecture 1 uses at most $\lfloor \mathrm{OPT} \cdot \ln n \rfloor + 1$ sets. It does not do substantially better, even for unweighted set cover. We start with a simple instance on which greedy is off by a factor of about $\log_2 n/2$.

Example 2.1. Let $T \geq 2$ and $W = 2^T - 1$. The universe U is a grid of two rows and W columns, so $n = 2W$. The sets are the two rows R_1, R_2 and T *column blocks* $C_1, \ldots, C_T$: the columns are split into consecutive blocks of widths $2^{T-1}, 2^{T-2}, \ldots, 2, 1$, and C_j consists of all elements in the j-th block, in both rows (see Fig. 1). The two rows form a cover, and no single set covers U, so $\mathrm{OPT} = 2$.

We claim that greedy picks $C_1, C_2, \ldots, C_T$, in this order. Before step j, suppose that exactly the columns of $C_j, \ldots, C_T$ are uncovered, which holds for $j = 1$. Their total width is $2^{T-j+1} - 1$, so each row covers $2^{T-j+1} - 1$ uncovered elements, while C_j covers $2 \cdot 2^{T-j} = 2^{T-j+1}$ of them and every later block covers fewer. Hence C_j covers strictly more uncovered elements than any other set, so greedy picks it whatever its tie-breaking rule, and the assumption holds again before step $j + 1$. Greedy thus uses $T = \log_2(n/2 + 1)$ sets, and its ratio on this instance is

$$\frac{\mathrm{ALG}}{\mathrm{OPT}} = \frac{T}{2} = \frac{\log_2(n+2) - 1}{2} \geq \frac{\log_2 n - 1}{2},$$

which is about $\ln n/(2 \ln 2) \approx 0.72 \ln n$.

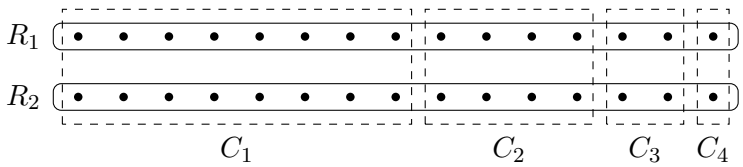


Figure 1: The instance of Example 2.1 for $T = 4$: $n = 30$ elements (dots), two rows R_1, R_2 (solid) forming an optimal cover, and column blocks $C_1, \ldots, C_4$ (dashed) of widths $8, 4, 2, 1$, which greedy picks one after the other.

To approach $\ln n$, use k rows instead of two, so that $\mathrm{OPT} \le k$, and column blocks each of which covers about a $1/k$ fraction of what remains uncovered. Greedy then needs about $\log_{1/(1-1/k)} n$ sets, and its ratio is about

$$\frac{\log_{1/(1-1/k)} n}{k} = \frac{\ln n}{k \ln(1/(1-1/k))} = \Bigl(1 - \Theta\Bigl(\frac{1}{k}\Bigr)\Bigr) \ln n,$$

because $\ln(1+x) = x \pm \Theta(x^2)$ for small x. The following proposition makes this estimate precise.

Proposition 2.2. *For every $\varepsilon \in (0,1)$ and every N, there is an instance of unweighted set cover with $n \ge N$ elements on which greedy uses at least $(1-\varepsilon)(\ln n) \cdot \mathrm{OPT}$ sets.*

Proof. Fix integers $k \ge 2$ and $T \ge 1$. Define widths backward, for $j = T, T-1, \dots, 1$, by

$$w_j = \Bigl\lfloor \frac{r_{j+1}}{k-1} \Bigr\rfloor + 1, \qquad \text{where } r_j \coloneqq w_j + w_{j+1} + \cdots + w_T \text{ and } r_{T+1} \coloneqq 0,$$

so that $w_T = 1$, and let $W = r_1$. The universe is a grid of k rows and W columns, so $n = kW$. The sets are the k rows, which form a cover, so $\mathrm{OPT} \le k$, and column blocks $C_1, \dots, C_T$ of widths $w_1, \dots, w_T$ as in Example 2.1. The widths are nonincreasing, because $r_{j+1} \ge r_{j+2}$, and $(k-1)w_j > r_{j+1}$, that is, $kw_j > r_j$, for every j.

Greedy picks the blocks $C_1, \dots, C_T$ and no row. Indeed, if before some step exactly the columns of $C_j, \dots, C_T$ are uncovered, then each row covers r_j uncovered elements, while C_j covers $kw_j > r_j$ of them, the largest number among the blocks. So greedy picks C_j or, in case of a tie, a later block of the same width; since blocks of equal width are interchangeable, we may relabel so that it picks C_j. Hence greedy uses T sets.

It remains to compare T with $\ln n$. From $r_j = r_{j+1} + w_j \le \frac{k}{k-1} r_{j+1} + 1$ and $1 + k \le \frac{k^2}{k-1}$ we get $r_j + k \le \frac{k}{k-1}(r_{j+1} + k)$, so

$$n = kW \le k(r_1 + k) \le k(k+1)\Bigl(\frac{k}{k-1}\Bigr)^{T-1}.$$

Moreover $k \ln \frac{k}{k-1} = -k \ln(1 - 1/k) \le 1 + 1/k$, because $-\ln(1-x) \le x + x^2$ for $0 \le x \le 1/2$. Taking logarithms,

$$\frac{\mathrm{ALG}}{\mathrm{OPT}} \ge \frac{T}{k} \ge \frac{\ln n - \ln(k(k+1))}{k \ln \frac{k}{k-1}} \ge \frac{\ln n - 2\ln(k+1)}{1 + 1/k}$$

whenever the numerator $\ln n - 2\ln(k+1)$ is nonnegative. Now choose $k = \lceil 2/\varepsilon \rceil$, so that $1/(1+1/k) \ge 1 - \varepsilon/2$, and then T so large that $n \ge N$ and $2\ln(k+1) \le (\varepsilon/2)\ln n$; this is possible because $n \ge W \ge T$. Then $\mathrm{ALG}/\mathrm{OPT} \ge (1-\varepsilon/2)^2 \ln n \ge (1-\varepsilon)\ln n$. □

So greedy is no better than $(1-\varepsilon)\ln n$, and better guarantees require other algorithms. Before turning to them, we extend greedy to weighted set cover.

2.2 Greedy for weighted set cover

In this subsection all costs are positive. Sets of cost zero can be added to every solution for free, after which their elements are removed.

Algorithm 2.3 (Greedy for weighted set cover)**.** While some element is uncovered, let U' be the set of uncovered elements and pick a set S that maximizes $|S \cap U'|/c_S$, the number of newly covered elements per unit cost.

In the analysis, let S_t be the t-th set picked, c_t its cost, $C_t = c_1 + \cdots + c_t$, n_t the number of elements still uncovered after t steps (so $n_0 = n$), and τ the number of steps.

Lemma 2.4. *For every step t,*

$$\frac{n_{t-1} - n_t}{c_t} \geq \frac{n_{t-1}}{\mathrm{OPT}}, \qquad \textit{that is,} \qquad c_t \leq \mathrm{OPT} \cdot \frac{n_{t-1} - n_t}{n_{t-1}}.$$

Proof. The sets of an optimal cover have total cost OPT, and together they cover the n_{t-1} elements that are uncovered before step t. If each of them had fewer than n_{t-1}/OPT newly covered elements per unit cost, they would cover fewer than $\sum_S c_S \cdot n_{t-1}/\mathrm{OPT} = n_{t-1}$ of these elements. Hence one of them has ratio at least n_{t-1}/OPT, and the set chosen by greedy has at least the same ratio. □

Theorem 2.5. *Greedy is a $(1 + \ln n)$-approximation for weighted set cover.*

Proof. By Lemma 2.4, $n_t \leq n_{t-1}(1 - c_t/\mathrm{OPT})$ for every t, and the factor is nonnegative because $n_t \geq 0$. Hence

$$n_t \leq n\Big(1 - \frac{c_1}{\mathrm{OPT}}\Big)\Big(1 - \frac{c_2}{\mathrm{OPT}}\Big) \cdots \Big(1 - \frac{c_t}{\mathrm{OPT}}\Big) \leq n \exp\Big(-\frac{C_t}{\mathrm{OPT}}\Big).$$

Since the last step starts with an uncovered element, $1 \leq n_{\tau-1} \leq n \exp(-C_{\tau-1}/\mathrm{OPT})$, so $C_{\tau-1} \leq \mathrm{OPT} \cdot \ln n$. The last set costs $c_\tau \leq \mathrm{OPT}$ by Lemma 2.4. Hence the total cost is $C_\tau \leq (1 + \ln n) \cdot \mathrm{OPT}$. □

A charging argument gives the sharper bound $H_n \cdot \mathrm{OPT}$, where $H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n} \leq 1 + \ln n$ is the n-th harmonic number.

Theorem 2.6 (Chvátal [Chv79])**.** *The cost of the greedy solution is at most $H_n \cdot \mathrm{OPT}$.*

Proof. By Lemma 2.4, and since each of the $n_{t-1} - n_t$ numbers $1/j$ with $n_t < j \leq n_{t-1}$ is at least $1/n_{t-1}$,

$$c_t \leq \mathrm{OPT} \cdot \frac{n_{t-1} - n_t}{n_{t-1}} \leq \mathrm{OPT} \sum_{j=n_t+1}^{n_{t-1}} \frac{1}{j}.$$

Summing over $t = 1, \ldots, \tau$, the ranges of j partition $\{1, \ldots, n\}$, because $n_0 = n$ and $n_\tau = 0$. Hence

$$\mathrm{ALG} = \sum_{t=1}^{\tau} c_t \leq \mathrm{OPT}\Big(\frac{1}{n} + \frac{1}{n-1} + \cdots + 1\Big) = H_n \cdot \mathrm{OPT}.$$ □

Second proof (charging to elements). When greedy picks S_t, charge its cost c_t equally to the $n_{t-1} - n_t$ elements it newly covers. Every element is charged exactly once, so the total charge is ALG.

Fix a set S^* of an optimal cover, and let $e_1, \ldots, e_s$ be its elements in the order in which greedy covers them (breaking ties arbitrarily). Just before e_j is covered, say at step t, the elements

$e_j, \dots, e_s$ are uncovered, so S^* would cover at least $s - j + 1$ new elements, at ratio at least $(s - j + 1)/c_{S^*}$. Greedy picked a set at least as good, so the charge to e_j is

$$\frac{c_t}{n_{t-1} - n_t} \le \frac{c_{S^*}}{s - j + 1}.$$

Summing over j, the elements of S^* receive a total charge of at most $c_{S^*}(1/s + 1/(s-1) + \cdots + 1) = c_{S^*} H_s \le c_{S^*} H_n$. Every element lies in some set of the optimal cover, so the total charge is at most $\sum_{S^*} c_{S^*} H_n = H_n \cdot \mathrm{OPT}$. □

2.3 Linear programming relaxations

LP-based algorithms follow the *relax-and-round* paradigm:

1) write an integer (linear) program (IP) for the problem;
2) relax it to a linear program (LP) by dropping the integrality constraints;
3) solve the LP, which takes polynomial time (see Lecture 25);
4) round the fractional solution to an integral one.

For a minimization problem the analysis usually has three parts: the IP is exact, $\mathrm{IP}(I) = \mathrm{OPT}(I)$; the relaxation can only lower the optimum, $\mathrm{LP}(I) \le \mathrm{IP}(I)$; and the rounding loses a factor α, $\mathrm{ALG}(I) \le \alpha \cdot \mathrm{LP}(I)$. Together these give $\mathrm{ALG}(I) \le \alpha \cdot \mathrm{OPT}(I)$.

For set cover, take a variable x_S for each set $S \in \{S_1, \dots, S_m\}$. The IP and its LP relaxation are

$$\begin{aligned} \min \quad & \sum_S c_S x_S \\ \text{s.t.} \quad & \sum_{S:\, e \in S} x_S \ge 1 && \text{for all } e \in U, \\ & x_S \in \{0,1\} \text{ (IP)}, \qquad x_S \ge 0 \text{ (LP)} && \text{for all } S. \end{aligned} \tag{2.1}$$

We may assume that an optimal LP solution satisfies $x_S \le 1$ for all S: replacing x_S by $\min(x_S, 1)$ keeps every constraint satisfied and does not increase the cost.

2.4 Randomized rounding for set cover

Think of each $x_S \in [0,1]$ as a probability.

Algorithm 2.7 (Randomized rounding with clean-up)**.** Let x be an optimal solution of the LP (2.1) with $0 \le x_S \le 1$ for all S, and let $T \ge 0$ be an integer.

1) In each of T rounds, select each set S independently with probability x_S.
2) Clean-up: for each element e not covered by a selected set, select a cheapest set containing e.

Output all selected sets.

The clean-up step guarantees that the output is always a cover. For each element e, let $c_{\min}(e) = \min_{S \ni e} c_S$ be the cost of a cheapest set containing e.

Lemma 2.8. *The output of Algorithm 2.7 has expected cost at most*

$$T \cdot \mathrm{LP}(I) + e^{-T} \sum_{e \in U} c_{\min}(e).$$

Proof. The cost of the output is at most the total cost of the sets selected in the rounds plus the cost of the clean-up. By linearity of expectation, the expected cost of the sets selected in one round is $\sum_S c_S \Pr[S \text{ selected}] = \sum_S c_S x_S = \mathrm{LP}(I)$. For an element e, the probability that no set containing e is selected in one round is

$$\prod_{S \ni e} (1 - x_S) \le \exp\Big(-\sum_{S \ni e} x_S\Big) \le e^{-1},$$

by $1 - x \le e^{-x}$ and the LP constraint for e. Since the rounds are independent, e is uncovered after T rounds with probability at most e^{-T}, and only then does the clean-up pay $c_{\min}(e)$ for it. Linearity of expectation again bounds the expected clean-up cost by $e^{-T} \sum_e c_{\min}(e)$. □

Theorem 2.9. *With $T = \lceil \ln n \rceil$, Algorithm 2.7 outputs a cover of expected cost at most $(\lceil \ln n \rceil + 1) \cdot \mathrm{LP}(I) \le (\lceil \ln n \rceil + 1) \cdot \mathrm{OPT}(I)$.*

Proof. For every element e, the LP value is at least the cost of a cheapest set covering e:

$$\mathrm{LP}(I) = \sum_S c_S x_S \ge \sum_{S \ni e} c_S x_S \ge c_{\min}(e) \sum_{S \ni e} x_S \ge c_{\min}(e).$$

Hence $\sum_e c_{\min}(e) \le n \cdot \mathrm{LP}(I)$, and since $e^{-\lceil \ln n \rceil} \le 1/n$, Lemma 2.8 bounds the expected cost by $\lceil \ln n \rceil \cdot \mathrm{LP}(I) + \mathrm{LP}(I)$. □

Homework 2.10. Suppose that every set has at most B elements, where $B \ge 2$. Show that LP rounding gives an $O(\log B)$-approximation, and that greedy does too.

Figure 2 summarizes the chain of inequalities behind Theorem 2.9.

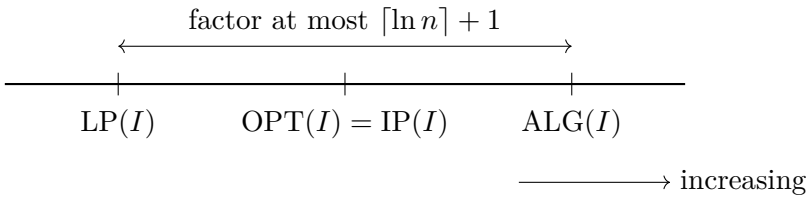


Figure 2: The values for a set cover instance I, in increasing order. The rounding of Theorem 2.9 loses a factor of at most $\lceil \ln n \rceil + 1$ against the LP value, and hence against the optimum.

2.5 Algorithmic gaps and integrality gaps

Two questions measure the limits of this approach.

1) *Algorithmic gap*: whether some instance has $\mathrm{ALG}(I)/\mathrm{OPT}(I) = \Omega(\log n)$ for this LP and rounding. Such an instance shows that this particular algorithm cannot beat $\Theta(\log n)$.

2) *Integrality gap*: whether some instance has $\mathrm{OPT}(I)/\mathrm{LP}(I) = \Omega(\log n)$. The *integrality gap* of the LP is the supremum of $\mathrm{OPT}(I)/\mathrm{LP}(I)$ over all instances with $\mathrm{LP}(I) > 0$. A large integrality gap shows that no rounding of this LP can beat the logarithmic approximation, as long as its analysis compares the output with the LP value, as in $\mathrm{ALG}(I) \le \alpha \cdot \mathrm{LP}(I)$.

Homework 2.11. Show that Algorithm 2.7 with $T = \lceil \ln n \rceil$ has an algorithmic gap of $\Omega(\log n)$: there are instances, with optimal LP solutions x, on which its expected cost is $\Omega(\log n) \cdot \mathrm{OPT}(I)$.

Proposition 2.12. *Let $d \ge 2$ be even, let $U = \{x \in \{0,1\}^d : \|x\|_1 = d/2\}$, and for $i \in [d]$ let $S_i = \{x \in U : x_i = 1\}$, each of cost 1 (the* dictator *sets). Then $n = |U| = \binom{d}{d/2} = \Theta(2^d/\sqrt{d})$, $\mathrm{OPT} = d/2 + 1$ and $\mathrm{LP} \le 2$. Hence the integrality gap of the set cover LP is at least $(d+2)/4 \ge (\log_2 n + 2)/4$, which is $\Omega(\log n)$.*

Proof. If at most $d/2$ sets S_i, $i \in J$, are chosen, then at least $d/2$ coordinates lie outside J, and a vector $x \in U$ whose $d/2$ ones all lie outside J is not covered. Hence $\mathrm{OPT} \ge d/2 + 1$. Conversely, any $d/2 + 1$ of the sets cover U, because every $x \in U$ has only $d/2$ zeros, so $\mathrm{OPT} = d/2 + 1$. For the LP, set $x_{S_i} = 2/d$ for every i. Each $x \in U$ lies in exactly $d/2$ of the sets, so its constraint holds with equality, and the LP value of this solution is $d \cdot 2/d = 2$. Thus $\mathrm{OPT}/\mathrm{LP} \ge (d/2+1)/2$, and $n \le 2^d$ gives $d \ge \log_2 n$. □

Fact 2.13. *A better construction gives an integrality gap of $(1 - o(1)) \ln n$ as $n \to \infty$. We state this without proof.*

2.6 Max-k-coverage: derandomizing by pipage rounding

We now return to max-k-coverage (Lecture 1): choose at most k sets so as to cover as many elements as possible. Its LP relaxation has a variable x_S for each set and a variable z_e for each element:

$$\begin{aligned} \max \quad & \sum_{e \in U} z_e \\ \text{s.t.} \quad & z_e \le \sum_{S \ni e} x_S, \quad z_e \le 1 \quad \text{for all } e \in U, \\ & \sum_S x_S \le k, \\ & x_S, z_e \ge 0 \qquad\qquad \text{for all } S, e. \end{aligned} \tag{2.2}$$

Let (x^*, z^*) be an optimal solution and $\mathrm{LP} = \sum_e z_e^*$ its value. We may assume $x_S^* \le 1$ for all S: replacing x_S^* by $\min(x_S^*, 1)$ keeps every constraint, because $z_e \le 1$, and does not change the objective.

Lemma 2.14. $\mathrm{LP} \ge \mathrm{OPT}$.

Proof. For an optimal choice of k sets, set $x_S = 1$ for the chosen sets, $z_e = 1$ for the covered elements, and all other variables to 0. This is a feasible solution of value OPT. □

As an idealized rounding, pick each set S independently with probability x_S^*. For $x \in [0,1]^m$ define

$$f_e(x) \coloneqq 1 - \prod_{S \ni e} (1 - x_S) \qquad \text{and} \qquad f(x) \coloneqq \sum_{e \in U} f_e(x).$$

Then $f_e(x)$ is the probability that e is covered, and $f(x)$ is the expected number of covered elements.

Lemma 2.15. $f(x^*) \geq (1 - 1/e) \cdot \mathrm{LP}$.

Proof. For every element e,

$$f_e(x^*) = 1 - \prod_{S \ni e} (1 - x_S^*) \geq 1 - \exp\Big(-\sum_{S \ni e} x_S^*\Big) \geq 1 - e^{-z_e^*} \geq \Big(1 - \frac{1}{e}\Big) z_e^*,$$

using $1 - x \leq e^{-x}$, the constraint $z_e^* \leq \sum_{S \ni e} x_S^*$, and the concavity of $z \mapsto 1 - e^{-z}$, which lies above its chord from $(0, 0)$ to $(1, 1 - 1/e)$ on $[0, 1]$. Summing over e proves the lemma. □

The idealized rounding, however, only guarantees that we pick at most k sets in expectation, since the expected number of picked sets is $\sum_S x_S^* \leq k$; the actual number may exceed k. On the other hand, if $x \in \{0, 1\}^m$ is integral with $\sum_S x_S = k$, then x is a choice of k sets, and $f(x)$ is exactly the number of elements they cover. So the idea is to move from x^* through a sequence of points to an integral point, without ever decreasing f:

$$x^* \to x^{(1)} \to x^{(2)} \to \cdots \to x^{\mathrm{int}}, \qquad f(x^*) \leq f(x^{(1)}) \leq f(x^{(2)}) \leq \cdots \leq f(x^{\mathrm{int}}).$$

Since $f(x^*) \geq (1 - 1/e)\mathrm{LP} \geq (1 - 1/e)\mathrm{OPT}$ by Lemmas 2.14 and 2.15, the integral endpoint is a choice of k sets covering at least $(1 - 1/e)\mathrm{OPT}$ elements.

To set this up, assume $k \leq m$ (otherwise all sets together are optimal), and raise some coordinates of x^*, keeping them at most 1, until $\sum_S x_S^* = k$. This does not decrease f, since each f_e is nondecreasing in every coordinate on $[0, 1]^m$.

Lemma 2.16. *Let $x \in [0, 1]^m$, let $S_1 \neq S_2$ be two sets, and for real ε let $x_\varepsilon = x + \varepsilon(\mathbf{e}_{S_1} - \mathbf{e}_{S_2})$, where $\mathbf{e}_S$ denotes the standard basis vector of coordinate S. Then $g(\varepsilon) \coloneqq f(x_\varepsilon)$ is a convex function of ε on the interval where $x_\varepsilon \in [0, 1]^m$.*

Proof. We have $g(\varepsilon) = \sum_e f_e(x_\varepsilon)$, and we split the sum by the position of e.

- If $e \notin S_1 \cup S_2$, then $f_e(x_\varepsilon) = f_e(x)$ is constant.
- If $e \in S_1 \setminus S_2$, then $f_e(x_\varepsilon) = 1 - \big[\prod_{S \ni e,\, S \neq S_1} (1 - x_S)\big](1 - x_{S_1} - \varepsilon)$ is an affine function of ε; so is $f_e(x_\varepsilon)$ for $e \in S_2 \setminus S_1$.
- If $e \in S_1 \cap S_2$, let $P_e = \prod_{S \ni e,\, S \neq S_1, S_2} (1 - x_S)$. Then

  $$f_e(x_\varepsilon) = 1 - P_e(1 - x_{S_1} - \varepsilon)(1 - x_{S_2} + \varepsilon) = c_0 + c_1\varepsilon + c_2\varepsilon^2$$

  with $c_2 = P_e \geq 0$, a convex function of ε.

A sum of convex functions is convex. □

Theorem 2.17 (Ageev–Sviridenko [AS04])**.** *There is a deterministic polynomial-time algorithm that, for every instance of max-k-coverage, outputs at most k sets covering at least $(1 - 1/e) \cdot \mathrm{OPT}$ elements.*

Proof. Start from $x = x^*$, with $x^* \in [0, 1]^m$ and $\sum_S x_S^* = k$ as arranged above. While x is not integral, do the following. Since $\sum_S x_S = k$ is an integer, x has at least two fractional coordinates, say x_{S_1} and x_{S_2}; if only one coordinate were fractional, the sum would not be an integer. Move along the line $\mathbf{e}_{S_1} - \mathbf{e}_{S_2}$, as in Lemma 2.16. Let $\varepsilon_0 = \min(1 - x_{S_1}, x_{S_2}) > 0$, the step at which

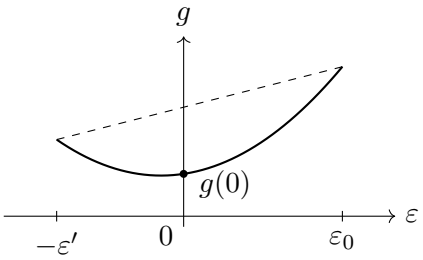


Figure 3: A convex function g on $[-\varepsilon', \varepsilon_0]$ lies below its chord (dashed), so $g(0)$ is at most the larger of $g(-\varepsilon')$ and $g(\varepsilon_0)$.

x_{S_1} reaches 1 or x_{S_2} reaches 0, and $\varepsilon' = \min(x_{S_1}, 1 - x_{S_2}) > 0$, the step at which x_{S_1} reaches 0 or x_{S_2} reaches 1. Since 0 is a convex combination of $-\varepsilon'$ and ε_0, the convexity of g gives

$$g(0) \le \max\bigl(g(-\varepsilon'), g(\varepsilon_0)\bigr)$$

(Fig. 3). Replace x by whichever of $x_{-\varepsilon'}$ and x_{ε_0} has the larger value of f. This keeps $x \in [0,1]^m$ and $\sum_S x_S = k$, does not decrease f, and makes at least one more coordinate integral.

After at most m such steps, x is an integral vector x^{int} with $\sum_S x_S^{\text{int}} = k$, so it is a choice of k sets covering $f(x^{\text{int}}) \ge f(x^*) \ge (1 - 1/e)\text{OPT}$ elements. Each step takes polynomial time, and so does solving the LP. □

3 Hardness of Approximating Max-Coverage

Lecture of September 3, 2021, by Anupam Gupta. Based on the lecturer's handwritten notes.

Greedy solves the 1-vs-$(1-1/e)$ search problem for max-coverage (Lecture 1). This lecture proves a hardness result in the other direction: for every $\varepsilon \in (0, 1/4]$, the 1-vs-$(3/4+\varepsilon)$ decision problem for max-coverage is NP-hard, that is, a polynomial-time algorithm for it would imply P = NP. The proof is a reduction from the label cover problem, whose hardness rests on the PCP theorem and Raz's parallel repetition theorem and is used here as a black box. Completeness of the reduction is direct. For soundness, we decode a good coverage into a labeling by random choices among the labels that the coverage suggests, and we show that the labeling satisfies many edges.

3.1 Hardness of approximation and label cover

Hardness of approximation builds on a chain of results, each of which says that a c-vs-s decision problem (Lecture 1) is NP-hard.

Fact 3.1 (Cook–Levin [Coo71])**.** *The* 1*-vs-*$(1-1/m)$ *decision problem for Max 3SAT on formulas with* m *clauses is NP-hard.*

This follows from the NP-hardness of 3SAT: a satisfiable formula has value 1, and an unsatisfiable one has value at most $1-1/m$. Since an algorithm for the decision problem must answer NO only on values strictly below $1-1/m$, we first repeat every clause twice. The new formula has $2m$ clauses and the same satisfying assignments, and if it is unsatisfiable, its value is at most $1-2/(2m) < 1-1/(2m)$. So such an algorithm, run on the new formula, decides whether the original one is satisfiable.

Fact 3.2 (PCP theorem [AS98, ALM+98, Din07])**.** *There is a constant* $\varepsilon > 0$ *such that the* 1*-vs-*$(1-\varepsilon)$ *decision problem for Max 3SAT is NP-hard.*

The third link concerns label cover.

Definition 3.3. An instance $\mathcal{G}$ of *max label cover* consists of a bipartite graph $G = (U, V, E)$, a finite set K of *keys* for the vertices in U, a finite set L of *labels* for the vertices in V, and for each edge $(u, v) \in E$ a *projection* $\pi_{uv} \colon L \to K$. A *labeling* f assigns a key $f(u) \in K$ to each $u \in U$ and a label $f(v) \in L$ to each $v \in V$. It *satisfies* the edge (u, v) if $f(u) = \pi_{uv}(f(v))$. The value $\mathrm{val}(f)$ is the fraction of edges that f satisfies, and $\mathrm{val}(\mathcal{G}) = \max_f \mathrm{val}(f)$.

Theorem 3.4 (Raz [Raz98])**.** *For every* $\eta > 0$ *there is* $q = \mathrm{poly}(1/\eta)$ *such that the* 1*-vs-*η *decision problem for max label cover is NP-hard on instances with* $|K|, |L| \le q$*. Moreover, these instances satisfy* $|U| = |V|$*, and their bipartite graph is regular.*

The theorem follows from the PCP theorem and Raz's parallel repetition theorem; we state it without proof and start from it.

3.2 The reduction

Fix an instance $\mathcal{G}$ of max label cover as in Theorem 3.4: the graph $G = (U, V, E)$ is d-regular with $d \ge 1$ and $|U| = |V|$. Let $k = |K|$ and $\ell = |L|$. We build an instance $\mathcal{H}$ of max-coverage, in which at most T sets may be chosen and the value is the largest fraction of elements covered.

1) *Elements.* For each edge $(u, v) \in E$, take a cube $Q_{uv} = \{0, 1\}^K$, a separate copy for each edge, whose coordinates x_a are indexed by the keys $a \in K$. There are $|E| \cdot 2^k = d|U| \cdot 2^k$ elements.

2) *Sets.* For each $u \in U$ and each key $a \in K$, let

$$S_{u,a} = \bigcup_{v\colon (u,v)\in E} \{x \in Q_{uv} : x_a = 0\},$$

the half of each cube at u on one side of the a-th dimension; it has $d \cdot 2^{k-1}$ elements. For each $v \in V$ and each label $\alpha \in L$, let

$$S_{v,\alpha} = \bigcup_{u\colon (u,v)\in E} \{x \in Q_{uv} : x_{\pi_{uv}(\alpha)} = 1\}.$$

There are $k|U| + \ell|V| \le q(|U| + |V|)$ sets.

3) *Budget.* $T = |U| + |V| = 2|U|$.

For a fixed bound q on k and ℓ, the instance $\mathcal{H}$ has polynomial size and is computed in polynomial time.

Two features of the construction are used throughout. First, the cube Q_{uv} meets only the sets of its endpoints: $S_{u,a} \cap Q_{uv} = \{x : x_a = 0\}$, $S_{v,\alpha} \cap Q_{uv} = \{x : x_{\pi_{uv}(\alpha)} = 1\}$, and all other sets miss Q_{uv}. Second, all cubes have the same size, so the fraction of elements covered by a collection of sets is the average, over the edges, of the fraction of Q_{uv} that it covers.

The reduction is to have two properties, for a suitable $\varepsilon = \varepsilon(\eta)$:

- *completeness*: if $\mathrm{val}(\mathcal{G}) = 1$, then $\mathrm{val}(\mathcal{H}) = 1$;
- *soundness*: if $\mathrm{val}(\mathcal{G}) < \eta$, then $\mathrm{val}(\mathcal{H}) < 3/4 + \varepsilon$.

Lemma 3.5 (Completeness)**.** *If* $\mathrm{val}(\mathcal{G}) = 1$*, then* $\mathcal{H}$ *has* T *sets that cover every element exactly once; in particular,* $\mathrm{val}(\mathcal{H}) = 1$*.*

Proof. Let f be a labeling that satisfies all edges, and choose the sets $S_{u,f(u)}$ for $u \in U$ and $S_{v,f(v)}$ for $v \in V$, one per vertex, T in all. For each edge (u, v) we have $f(u) = \pi_{uv}(f(v)) =: a$. So $S_{u,f(u)}$ covers the points $x \in Q_{uv}$ with $x_a = 0$, and $S_{v,f(v)}$ covers the other points of Q_{uv}, those with $x_a = 1$. No other chosen set meets Q_{uv}, so every element is covered exactly once. □

3.3 Soundness

Theorem 3.6 (Soundness)**.** *Let* $\varepsilon \in (0, 1/4]$*, and let* $\mathcal{G}$ *and* $\mathcal{H}$ *be as above. If* $\mathrm{val}(\mathcal{H}) \ge 3/4 + \varepsilon$*, then*

$$\mathrm{val}(\mathcal{G}) \ge \frac{\varepsilon^3}{2000} =: \eta(\varepsilon).$$

This is the contrapositive of the soundness property with $\eta = \varepsilon^3/2000$. Throughout its proof, $\mathcal{S}^*$ is a collection of at most T sets that covers at least a $3/4 + \varepsilon$ fraction of the elements.

As a warm-up, suppose that $\mathcal{S}^*$ contains exactly one set per vertex. This is a big assumption, which the rest of the proof removes.

Proposition 3.7. *Suppose that for every vertex $w \in U \cup V$, $\mathcal{S}^*$ contains exactly one of the sets $S_{w,\cdot}$, namely $S_{u,f(u)}$ for $u \in U$ and $S_{v,f(v)}$ for $v \in V$. Then the labeling f satisfies at least a 4ε fraction of the edges.*

Proof. Let (u, v) be an edge that f does not satisfy, and let $a = f(u)$ and $b = \pi_{uv}(f(v))$, so $a \neq b$. In Q_{uv}, the set $S_{u,f(u)}$ covers the points with $x_a = 0$ and $S_{v,f(v)}$ those with $x_b = 1$; together they cover all points except those with $x_a = 1$ and $x_b = 0$, which is $3/4$ of the cube. A satisfied edge has its whole cube covered, by the proof of Lemma 3.5. If p is the fraction of satisfied edges, the covered fraction is therefore

$$p \cdot 1 + (1 - p) \cdot \frac{3}{4} = \frac{3}{4} + \frac{p}{4} \geq \frac{3}{4} + \varepsilon,$$

so $p \geq 4\varepsilon$. □

In general, $\mathcal{S}^*$ may choose several sets for a vertex, or none at all, and it is not clear which labeling to extract. Two observations give hope. First, $\mathcal{S}^*$ chooses at most one set per vertex on average, so the situation may not be much worse than in the warm-up. Second, choosing many sets that meet the same cube, but that are not complementary halves, gives diminishing returns. We now make these ideas precise.

Definition 3.8. For $u \in U$ and $v \in V$, let

$$\mathrm{Suggest}(u) = \{a \in K : S_{u,a} \in \mathcal{S}^*\}, \qquad \mathrm{Suggest}(v) = \{\alpha \in L : S_{v,\alpha} \in \mathcal{S}^*\},$$

and for an edge (u, v) let $\mathrm{Suggest}(u, v)$ be the disjoint union of $\mathrm{Suggest}(u)$ and $\mathrm{Suggest}(v)$, so that $|\mathrm{Suggest}(u, v)| = |\mathrm{Suggest}(u)| + |\mathrm{Suggest}(v)|$.

Below, (u, v) denotes a uniformly random edge of E.

Lemma 3.9. $\mathbb{E}_{(u,v)}\big[|\mathrm{Suggest}(u, v)|\big] \leq 2$.

Proof. Since G is regular and $|U| = |V|$, the endpoint u of a uniformly random edge is uniform on U, and v is uniform on V. Hence

$$\begin{aligned}\mathbb{E}_{(u,v)}\big[|\mathrm{Suggest}(u, v)|\big] &= \mathbb{E}_u\big[|\mathrm{Suggest}(u)|\big] + \mathbb{E}_v\big[|\mathrm{Suggest}(v)|\big] \\ &= \frac{1}{|U|}\Big[\sum_{u \in U} |\mathrm{Suggest}(u)| + \sum_{v \in V} |\mathrm{Suggest}(v)|\Big],\end{aligned}$$

using $|U| = |V|$ in the second step. The bracket counts the sets in $\mathcal{S}^*$, which number at most $T = |U| + |V|$, so the expectation is at most $(|U| + |V|)/|U| = 2$. □

Definition 3.10. The *random decoding* of $\mathcal{S}^*$ is the random labeling f that picks $f(w)$ uniformly at random from $\mathrm{Suggest}(w)$, for each vertex w independently, and arbitrarily if $\mathrm{Suggest}(w) = \emptyset$. An edge (u, v) is

- *consistent* if there are $a \in \mathrm{Suggest}(u)$ and $\alpha \in \mathrm{Suggest}(v)$ with $\pi_{uv}(\alpha) = a$;
- *sparse* if both u and v are sparse, where a vertex w is *sparse* if $|\mathrm{Suggest}(w)| \leq 10/\varepsilon$;
- *good* if it is sparse and consistent.

Lemma 3.11 (Claim 1)**.** *The random decoding satisfies* $\mathbb{E}_f[\mathrm{val}(f)] \geq \varepsilon^3/2000$*. Hence some labeling of* $\mathcal{G}$ *has value at least* $\varepsilon^3/2000$*.*

Lemma 3.12 (Claim 2)**.** *At least an* $\varepsilon/20$ *fraction of the edges are good.*

Proof of Lemma 3.11, assuming Lemma 3.12. Let (u, v) be a good edge, and let $a \in \mathrm{Suggest}(u)$ and $\alpha \in \mathrm{Suggest}(v)$ with $\pi_{uv}(\alpha) = a$. By sparsity, $f(u) = a$ with probability $1/|\mathrm{Suggest}(u)| \geq \varepsilon/10$, and independently $f(v) = \alpha$ with probability at least $\varepsilon/10$. So f satisfies (u, v) with probability at least $(\varepsilon/10)^2$. By Lemma 3.12 and linearity of expectation, the expected fraction of satisfied edges is at least

$$\frac{\varepsilon}{20} \cdot \left(\frac{\varepsilon}{10}\right)^2 = \frac{\varepsilon^3}{2000}.$$

Some labeling attains at least the expectation. □

Two facts about cubes prepare the proof of Lemma 3.12.

Lemma 3.13 (Fact 1)**.** *Let* (u, v) *be an edge that is not consistent, with* $|\mathrm{Suggest}(u, v)| = t$*. Then* $\mathcal{S}^*$ *covers at most a* $1 - 2^{-t}$ *fraction of* Q_{uv}*.*

Proof. The sets of $\mathcal{S}^*$ that meet Q_{uv} are the $S_{u,a}$ with $a \in \mathrm{Suggest}(u)$ and the $S_{v,\alpha}$ with $\alpha \in \mathrm{Suggest}(v)$. A point $x \in Q_{uv}$ is therefore uncovered exactly when $x_a = 1$ for all $a \in A \coloneqq \mathrm{Suggest}(u)$ and $x_b = 0$ for all $b \in B \coloneqq \pi_{uv}(\mathrm{Suggest}(v))$. Since the edge is not consistent, $A \cap B = \emptyset$, so these conditions fix $|A| + |B| \leq t$ distinct coordinates, and the uncovered points form a $2^{-(|A|+|B|)} \geq 2^{-t}$ fraction of the cube. Equality holds when π_{uv} is injective on $\mathrm{Suggest}(v)$, but not in general: if $\mathrm{Suggest}(u) = \emptyset$ and $\mathrm{Suggest}(v)$ consists of two labels with the same key, then $t = 2$ while half of the cube is covered. □

Lemma 3.14 (Fact 2, concavity)**.** *Let* $t_1, \ldots, t_r \geq 0$ *be reals with average* $\bar{t} = (t_1 + \cdots + t_r)/r$*. Then*

$$\sum_{i=1}^{r} (1 - 2^{-t_i}) \leq r(1 - 2^{-\bar{t}}),$$

with equality when $t_1 = \cdots = t_r = \bar{t}$*.*

Proof. By the inequality of arithmetic and geometric means,

$$\frac{1}{r} \sum_{i=1}^{r} 2^{-t_i} \geq \left(\prod_{i=1}^{r} 2^{-t_i}\right)^{1/r} = 2^{-\sum_i t_i/r} = 2^{-\bar{t}}.$$ □

Proof of Lemma 3.12. Suppose, for contradiction, that fewer than an $\varepsilon/20$ fraction of the edges are good. A consistent edge that is not good has an endpoint that is not sparse, so

$$\Pr_{(u,v)}[(u, v) \text{ consistent}] \leq \Pr[(u, v) \text{ good}] + \Pr[u \text{ not sparse}] + \Pr[v \text{ not sparse}].$$

The first term is less than $\varepsilon/20$ by assumption. By Markov's inequality, $\Pr[u \text{ not sparse}] \leq (\varepsilon/10)\mathbb{E}_u[|\mathrm{Suggest}(u)|]$, and similarly for v; by Lemma 3.9 the two expectations sum to at most 2, so the last two terms sum to at most $\varepsilon/5$. Hence fewer than an $\varepsilon/20 + \varepsilon/5 = \varepsilon/4$ fraction of the edges are consistent.

Let C be the set of consistent edges and $I = E \setminus C$ the set of inconsistent ones, so $c \coloneqq |C|/|E| < \varepsilon/4$ and $|I| > (1 - \varepsilon/4)|E|$. For an edge e, let cov_e be the fraction of its cube covered by $\mathcal{S}^*$, and let $t_e = |\mathrm{Suggest}(e)|$.

The inconsistent edges get high coverage. Each consistent edge contributes at most a full cube, so $\sum_{e \in I} \text{cov}_e \geq (3/4 + \varepsilon)|E| - |C|$, and the average coverage over I is

$$\frac{1}{|I|} \sum_{e \in I} \text{cov}_e \geq \left(\frac{3}{4} + \varepsilon - c\right) \frac{|E|}{|I|} \geq \frac{3}{4} + \varepsilon - c > \frac{3}{4} + \frac{3\varepsilon}{4}.$$

The inconsistent edges have few suggestions. By Lemma 3.9, $\sum_{e \in E} t_e \leq 2|E|$, so the average of t_e over I is

$$\bar{t} \leq \frac{2|E|}{|I|} < \frac{2}{1 - \varepsilon/4} \leq 2 + \varepsilon,$$

where the last step holds because $\varepsilon \leq 2$.

These two facts are incompatible. By Lemmas 3.13 and 3.14 and since $1 - 2^{-t}$ increases with t,

$$\frac{1}{|I|} \sum_{e \in I} \text{cov}_e \leq \frac{1}{|I|} \sum_{e \in I} (1 - 2^{-t_e}) \leq 1 - 2^{-\bar{t}} \leq 1 - 2^{-(2+\varepsilon)} = 1 - \frac{1}{4} \cdot 2^{-\varepsilon}.$$

Since $2^{-\varepsilon} = e^{-\varepsilon \ln 2} \geq 1 - \varepsilon \ln 2$, the right-hand side is at most

$$1 - \frac{1}{4}(1 - \varepsilon \ln 2) = \frac{3}{4} + \frac{\varepsilon \ln 2}{4} < \frac{3}{4} + \frac{3\varepsilon}{4},$$

which contradicts the lower bound on the average coverage over I. □

Proof of Theorem 3.6. Combine Lemmas 3.11 and 3.12. □

3.4 Consequences

To recap, Theorem 3.4 says that it is NP-hard to distinguish label cover instances with $\text{val}(\mathcal{G}) = 1$ from those with $\text{val}(\mathcal{G}) < \eta$. The reduction says that if we could distinguish, in polynomial time, max-coverage instances with $\text{val}(\mathcal{H}) = 1$ from those with $\text{val}(\mathcal{H}) < 3/4 + \varepsilon$, then we could distinguish label cover instances with $\text{val}(\mathcal{G}) = 1$ from those with $\text{val}(\mathcal{G}) < \varepsilon^3/2000$.

Theorem 3.15. *For every $\varepsilon \in (0, 1/4]$, the 1-vs-$(3/4 + \varepsilon)$ decision problem for max-coverage is NP-hard.*

Proof. Let $\eta = \varepsilon^3/2000$, and let $q = \text{poly}(1/\eta)$ be as in Theorem 3.4. Given a label cover instance $\mathcal{G}$ as in that theorem, build $\mathcal{H}$ in polynomial time; this is possible since q is a constant. If $\text{val}(\mathcal{G}) = 1$, then $\text{val}(\mathcal{H}) = 1$ by Lemma 3.5. If $\text{val}(\mathcal{G}) < \eta$, then $\text{val}(\mathcal{H}) < 3/4 + \varepsilon$ by Theorem 3.6. Hence a polynomial-time algorithm for the 1-vs-$(3/4 + \varepsilon)$ decision problem for max-coverage would solve the 1-vs-η decision problem for max label cover, which is NP-hard. □

Corollary 3.16. *Let $\varepsilon \in (0, 1/4]$. If $\text{P} \neq \text{NP}$, then no polynomial-time algorithm approximates max-coverage within a factor of $3/4 + \varepsilon$.*

Proof. A $(3/4 + \varepsilon)$-approximation outputs a solution of value at least $3/4 + \varepsilon$ whenever the optimum is 1, so it solves the 1-vs-$(3/4 + \varepsilon)$ search problem. A polynomial-time algorithm for the search problem gives one for the decision problem (Lecture 1), which is NP-hard by Theorem 3.15. □

Remark 3.17. A more elaborate reduction, from a k-partite version of label cover, shows that max-coverage cannot be approximated within any factor better than $1 - 1/e \approx 0.632$ unless P = NP [Fei98]; pipage rounding (Lecture 2) achieves this factor. A different extension proves the $\Omega(\log n)$ hardness of set cover. The first such result is due to Lund and Yannakakis [LY94]; Feige [Fei98] obtained the tight factor $(1 - \varepsilon) \ln n$ under an assumption slightly stronger than P $\neq$ NP; and Dinur and Steurer [DS14] proved it assuming only P $\neq$ NP.

4 Max-Cut: The Goemans–Williamson Algorithm

Lecture of September 8, 2021, by Pravesh K. Kothari. Based on the lecturer's handwritten notes.

The Max-Cut problem asks for a partition of the vertices of a graph that cuts as many edges as possible. It is NP-hard, and a uniformly random cut already cuts half of the edges in expectation, so the question is whether an efficient algorithm can beat the ratio $1/2$. We first see that linear programming relaxations do not help: the natural linear relaxation, even strengthened by constraints for triangles, has value 1 on every triangle-free graph, and much larger linear programs are known to fail as well. We then write Max-Cut as a quadratic program, relax it to a semidefinite program, and round its solution with a random hyperplane. This is the algorithm of Goemans and Williamson [GW95]: it finds a cut of expected value at least 0.878 times the optimum, and a cut of expected value at least $1 - 2\sqrt{\varepsilon}$ on graphs whose max-cut is at least $1 - \varepsilon$. The analysis rests on the rotation invariance of the Gaussian distribution and on Sheppard's formula for the probability that a random hyperplane separates two unit vectors.

4.1 The Max-Cut problem

Throughout this lecture, $G = (V, E)$ is an undirected graph without loops or parallel edges, with $n = |V|$ vertices and $m = |E| \geq 1$ edges. For $S \subseteq V$ we write $\bar{S} = V \setminus S$, and $E(S, \bar{S})$ for the set of edges with exactly one endpoint in S.

Definition 4.1. The *cut* $(S, \bar{S})$ *cuts* the edges in $E(S, \bar{S})$, and its *value* is $|E(S, \bar{S})|/m$. The *max-cut* of G is

$$\operatorname{maxcut}(G) := \max_{S \subseteq V} \frac{|E(S, \bar{S})|}{m} \in [0, 1].$$

The *Max-Cut problem* is, given G, to find a set $S \subseteq V$ that maximizes $|E(S, \bar{S})|$.

Strictly speaking, $\operatorname{maxcut}(G)$ is the normalized max-cut; since we never use the unnormalized maximum, we simply call it the max-cut of G.

Fact 4.2. *Computing* $\operatorname{maxcut}(G)$ *is NP-hard.*

The proof is a reduction from the maximum independent set problem, which we omit; the NP-completeness of the decision version of Max-Cut is proved in [GJS76].

Remark 4.3. In contrast, the minimum of $|E(S, \bar{S})|$ over all sets $\emptyset \neq S \subsetneq V$, the min-cut of G, can be computed in polynomial time. Replacing the minimum by a maximum changes the complexity of the problem.

4.2 A $1/2$-approximation

We measure the quality of an algorithm for Max-Cut by its approximation ratio, a number at most 1 since Max-Cut is a maximization problem.

Definition 4.4. Let $\alpha \in [0, 1]$. A (possibly randomized) algorithm for Max-Cut is an α-*approximation algorithm*, or has *approximation ratio* α, if for every graph G with at least one edge it outputs a set $S \subseteq V$ with

$$\mathbb{E}\left[\frac{|E(S, \bar{S})|}{m}\right] \geq \alpha \cdot \operatorname{maxcut}(G),$$

where the expectation is over the random choices of the algorithm.

A ratio of $1/2$ is easy to achieve.

Lemma 4.5. *Let S be a uniformly random subset of V; equivalently, put each vertex $v \in V$ into S with probability $1/2$, independently of the other vertices. Then*

$$\mathbb{E}_S\left[\frac{|E(S,\bar{S})|}{m}\right] = \frac{1}{2}.$$

In particular, $\operatorname{maxcut}(G) \geq 1/2$. *Since* $\operatorname{maxcut}(G) \leq 1$, *outputting a uniformly random set S is a $1/2$-approximation algorithm for Max-Cut.*

Proof. Fix an edge $\{u,v\} \in E$. It is cut when exactly one of its endpoints lies in S, so by independence

$$\Pr[\{u,v\} \text{ is cut}] = \Pr[u \in S,\ v \notin S] + \Pr[u \notin S,\ v \in S] = \tfrac{1}{4} + \tfrac{1}{4} = \tfrac{1}{2}.$$

By linearity of expectation, $\mathbb{E}|E(S,\bar{S})| = m/2$. Some set S attains at least the expected value, which gives $\operatorname{maxcut}(G) \geq 1/2$. Finally, $1/2 \geq \frac{1}{2}\operatorname{maxcut}(G)$. □

The analysis never looks at the optimal cut: it compares the output with the trivial upper bound $\operatorname{maxcut}(G) \leq 1$, that is, with all m edges, which serves as a surrogate for the unknown optimum. To do better, we need better upper bounds on $\operatorname{maxcut}(G)$; relaxations will provide them.

Randomness is not needed for the ratio $1/2$. The following local search algorithm is deterministic.

Algorithm 4.6 (Local search for Max-Cut)**.**

1) Start from an arbitrary set $S \subseteq V$.
2) While there is a vertex v such that moving v to the other side of the cut (from S to $\bar{S}$, or from $\bar{S}$ to S) increases $|E(S,\bar{S})|$, move such a vertex.
3) When there is no such vertex, return S.

Homework 4.7. Show that Algorithm 4.6 stops after at most m moves and returns a set S with $|E(S,\bar{S})| \geq m/2$. Hence it is a deterministic polynomial-time $1/2$-approximation algorithm for Max-Cut.

4.3 Quadratic and linear programs for Max-Cut

We now ask whether some polynomial-time algorithm for Max-Cut has approximation ratio larger than $1/2$. For Set Cover, linear programming relaxations helped (Lecture 2), so we first try them here. We start from an exact formulation of Max-Cut as a quadratic program:

$$\begin{aligned} \text{maximize} \quad & \frac{1}{4m}\sum_{\{u,v\}\in E}(x_u - x_v)^2 \\ \text{subject to} \quad & x_u \in \{-1,1\} \quad \text{for all } u \in V. \end{aligned} \tag{4.1}$$

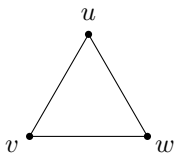
Every $x = (x_u)_{u\in V} \in \{\pm 1\}^V$ is the ± 1 indicator vector of the set $S_x = \{u \in V : x_u = 1\}$. The term $(x_u - x_v)^2$ equals 4 if the edge $\{u,v\}$ is cut by $(S_x, \bar{S}_x)$ and 0 otherwise, so the objective of (4.1) is the value of the cut $(S_x, \bar{S}_x)$, and the optimum of (4.1) is $\operatorname{maxcut}(G)$. In particular, solving quadratic programs is NP-hard in general (Fact 4.2). The approach of the course is to relax such a program to one we can solve, and then round the solution.

Linear programs. A linear program must have a linear objective, so we linearize: we introduce a variable y_{uv} for each edge $\{u, v\} \in E$, standing for $(x_u - x_v)^2/4$, and consider

$$\mathrm{LP}(G) \coloneqq \max\Big\{\frac{1}{m} \sum_{\{u,v\}\in E} y_{uv} \;:\; y_{uv} \in [0,1] \text{ for all } \{u,v\} \in E\Big\}. \tag{4.2}$$

Every cut $(S, \bar{S})$ gives a feasible solution, with $y_{uv} = 1$ for the cut edges and $y_{uv} = 0$ for the others, whose objective value is the value of the cut. Hence $\mathrm{LP}(G) \geq \mathrm{maxcut}(G)$. But the solution $y \equiv 1$ is feasible for every graph, so $\mathrm{LP}(G) = 1$ always, and this relaxation is useless.

We can add constraints that every cut satisfies. A cut cuts either none or exactly two of the three edges of a triangle:



So, for every triangle $\{u, v, w\}$ in G, we may add the constraint

$$y_{uv} + y_{vw} + y_{uw} \leq 2. \tag{4.3}$$

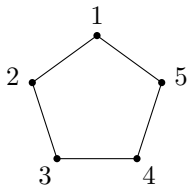
More generally, we may add any linear constraint that is satisfied by the solutions y coming from all cuts, and the program remains a relaxation.

The triangle constraints do not help: if G has no triangles, there are no constraints (4.3), the solution $y \equiv 1$ is still feasible, and the relaxation still has value 1. To measure how far a relaxation can be from the truth, we use the following notion.

Definition 4.8. Let $\mathrm{REL}(G)$ be the optimal value of a relaxation of Max-Cut, so that $\mathrm{REL}(G) \geq \mathrm{maxcut}(G)$ for every graph G. The *integrality gap* of the relaxation on G is $\mathrm{maxcut}(G)/\mathrm{REL}(G) \in (0, 1]$, and the integrality gap of the relaxation is the infimum of this ratio over all graphs with at least one edge.

An algorithm whose analysis compares its output with $\mathrm{REL}(G)$, as the analyses in this lecture do, cannot prove an approximation ratio larger than the integrality gap.

Example 4.9 (The 5-cycle)**.** Let $G = C_5$ be the cycle on the vertices $1, 2, 3, 4, 5$, with edges $\{1,2\}, \{2,3\}, \{3,4\}, \{4,5\}, \{5,1\}$.



The graph has no triangles, so $y_{12} = y_{23} = y_{34} = y_{45} = y_{51} = 1$ is a feasible solution of the linear program with the triangle constraints, and its value is 1. On the other hand, $\mathrm{maxcut}(C_5) = 4/5$: every cut crosses a cycle an even number of times, so it cuts at most 4 of the 5 edges, and $S = \{1, 3\}$ cuts exactly the 4 edges other than $\{4, 5\}$. Hence the integrality gap of the relaxation on C_5 is $4/5$.

In fact, much more is true: no linear program of reasonable size beats the ratio $1/2$.

Fact 4.10 (Kothari, Meka, Raghavendra [KMR17])**.** *Beating the ratio $1/2$ for Max-Cut requires linear programming relaxations, even in the sense of extended formulations (which may use auxiliary variables), of size more than 2^{n^c} for some constant $c > 0$. Precisely, for every $\varepsilon > 0$ there is a constant $c > 0$ such that, for every sufficiently large n, every linear programming relaxation of Max-Cut on n-vertex graphs with at most 2^{n^c} constraints has integrality gap less than $1/2 + \varepsilon$ on some n-vertex graph.*

We state this without proof; see [KMR17] for the precise model of linear programming relaxations.

Remark 4.11 (Intuition). For arbitrarily large degrees d, there are d-regular graphs G_1 and G_2 such that G_1 is bipartite, so that $\operatorname{maxcut}(G_1) = 1$, while $\operatorname{maxcut}(G_2)$ is close to $1/2$, which is the smallest possible value by Lemma 4.5. Yet the neighborhood of every vertex looks exactly the same in both graphs: if both have girth larger than $2r+1$ (that is, no cycles of length at most $2r+1$), then in either graph the vertices at distance at most r from any vertex induce a tree isomorphic to the ball of radius r around a vertex of the infinite d-regular tree. In this tree the center has d neighbors and every other vertex at distance less than r from the center has one parent and $d-1$ children. Such pairs exist: for G_2 one can take a non-bipartite Ramanujan graph of degree d and large girth [LPS88], whose adjacency eigenvalues other than d lie in $[-2\sqrt{d-1}, 2\sqrt{d-1}]$. If A is the adjacency matrix of G_2 and $x \in \{\pm 1\}^n$, the cut $(S_x, \bar{S}_x)$ has value $\frac{1}{2} - \frac{x^\top A x}{2nd} \le \frac{1}{2} - \frac{\lambda_{\min}(A)}{2d}$, so $\operatorname{maxcut}(G_2) \le \frac{1}{2} + \frac{\sqrt{d-1}}{d}$. For G_1 one can take the bipartite double cover of G_2, which is d-regular, bipartite, and has girth at least that of G_2. Since G_2 has no triangles, the linear program (4.2) with the triangle constraints (4.3) has value 1 on G_2, so its integrality gap is at most $\frac{1}{2} + \frac{\sqrt{d-1}}{d}$ for each such d, and hence at most $1/2$. Linear programs, and local algorithms in general, seem unable to distinguish such pairs of graphs.

4.4 A semidefinite relaxation

Since linear constraints on the variables y_{uv} do not suffice, we look for nonlinear constraints that give more power. The starting point is to rewrite the objective of (4.1). For $x \in \{\pm 1\}^V$ we have $x_u^2 = x_v^2 = 1$, so $(x_u - x_v)^2 = 2 - 2x_u x_v$, and

$$\frac{1}{4m} \sum_{\{u,v\} \in E} (x_u - x_v)^2 = \frac{1}{4m} \sum_{\{u,v\} \in E} (2 - 2x_u x_v) = \frac{1}{2} - \frac{1}{2m} \sum_{\{u,v\} \in E} x_u x_v. \tag{4.4}$$

The right-hand side is a linear function of the products $x_u x_v$. Identify V with $[n]$, and for $x \in \{\pm 1\}^n$ consider the $n \times n$ matrix $M = xx^\top$, with entries $M_{uv} = x_u x_v$. It has diagonal entries $M_{uu} = x_u^2 = 1$, and it is positive semidefinite, since $w^\top M w = \langle x, w \rangle^2 \ge 0$ for every $w \in \mathbb{R}^n$. This suggests a new relaxation, in which a matrix X takes the place of $xx^\top$:

$$\begin{aligned} \text{maximize} \quad & \frac{1}{2} - \frac{1}{2m} \sum_{\{u,v\} \in E} X_{uv} \\ \text{subject to} \quad & X_{uu} = 1 \quad \text{for all } u \in V \quad \text{(linear constraints)}, \\ & X \succeq 0 \qquad\qquad\qquad\quad \text{(a nonlinear constraint)}. \end{aligned}$$

It is a relaxation because we have dropped the constraint that X has rank one. Indeed, a positive semidefinite matrix of rank one has the form $X = ww^\top$ with $w \in \mathbb{R}^n$, and then $X_{uu} = w_u^2 = 1$ forces $w \in \{\pm 1\}^n$. So the feasible matrices of rank one are exactly the matrices $xx^\top$ with $x \in \{\pm 1\}^n$, on which the objective is (4.4).

4.5 Semidefinite programs

Semidefinite programs form a class of convex programs.

Definition 4.12. Let $K \subseteq \mathbb{R}^N$ be a convex set and $c \in \mathbb{R}^N$. The *convex program* with linear objective c and feasible region K is

$$\text{maximize } \sum_{i=1}^{N} c_i X_i \quad \text{subject to } X \in K.$$

Convexity alone does not make an optimization problem easy: convex programming in general is NP-hard, and how the set K is described matters. For example, let $K_n \subseteq \mathbb{R}^{n \times n}$ be the convex hull of the matrices $xx^\top$ with $x \in \{\pm 1\}^n$. A linear function attains its maximum over this polytope at one of its vertices, which are among the matrices $xx^\top$, so by (4.4) the maximum of $\frac{1}{2} - \frac{1}{2m}\sum_{\{u,v\}\in E} X_{uv}$ over $X \in K_n$ equals $\operatorname{maxcut}(G)$. Linear optimization over K_n is therefore as hard as Max-Cut. Some convex programs, however, can be solved in polynomial time. Linear programs are an example: there K is an intersection of half-spaces,

$$K = \{X \in \mathbb{R}^N : \langle a_i, X\rangle \le b_i \text{ for } 1 \le i \le k\}$$

for some $a_1, \dots, a_k \in \mathbb{R}^N$ and $b \in \mathbb{R}^k$, and linear programs with rational data can be solved in polynomial time.

Definition 4.13. A symmetric matrix $X \in \mathbb{R}^{n\times n}$ is *positive semidefinite*, written $X \succeq 0$, if $w^\top X w \ge 0$ for every $w \in \mathbb{R}^n$. The *Frobenius inner product* of $A, X \in \mathbb{R}^{n\times n}$ is

$$\langle A, X\rangle_F \coloneqq \sum_{j,k=1}^{n} A(j,k)\, X(j,k).$$

A *semidefinite program* (SDP) in the $n \times n$ matrix variable X is a convex program, with a linear objective $\langle C, X\rangle_F$, whose feasible region is

$$K = \{X \in \mathbb{R}^{n\times n} \text{ symmetric} : X \succeq 0 \text{ and } \langle A_i, X\rangle_F \le b_i \text{ for } 1 \le i \le k\}$$

for some matrices $C, A_1, \dots, A_k \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^k$.

The set K is convex: for each w, the condition $w^\top X w \ge 0$ is a linear inequality in X, so K is an intersection of half-spaces with the subspace of symmetric matrices. An equality constraint $\langle A_i, X\rangle_F = b_i$, such as $X_{uu} = 1$, is the pair of inequalities $\langle A_i, X\rangle_F \le b_i$ and $\langle -A_i, X\rangle_F \le -b_i$.

Semidefinite programs can be solved approximately, for instance by the ellipsoid method, which is the subject of Lecture 25.

The semidefinite program for Max-Cut. For a graph $G = (V, E)$ with $V = [n]$ and $m \ge 1$ edges, define

$$\operatorname{SDP}(G) \coloneqq \max\Big\{\frac{1}{2} - \frac{1}{2m}\sum_{\{u,v\}\in E} X_{uv} \;:\; X \in \mathbb{R}^{n\times n} \text{ symmetric},\ X_{uu} = 1 \text{ for all } u \in V,\ X \succeq 0\Big\}. \tag{4.5}$$

The objective can also be written as $\frac{1}{m}\sum_{\{u,v\}\in E} \frac{1 - X_{uv}}{2}$. The maximum is attained, because the feasible region contains the identity matrix, is closed, and is bounded, since every feasible X has $|X_{uv}| \le 1$ for all u, v (Lemma 4.24 below). We use the solvability of semidefinite programs only for (4.5), in the following form.

Fact 4.14 ([WS11, Chapter 6]). *For every graph G with $m \geq 1$ edges and every $\delta > 0$, a feasible solution X of (4.5) whose objective value is at least $\mathrm{SDP}(G) - \delta$ can be computed in time polynomial in n and $\log(1/\delta)$.*

4.6 Rounding: the Goemans–Williamson theorem

We measure the rounding not only by its approximation ratio, but also by what it guarantees on graphs whose max-cut is large.

Definition 4.15. Let $c \in [0,1]$ and let $s \leq c$ be a real number. A (possibly randomized) algorithm for Max-Cut is a (c,s)-*approximation algorithm* if on every graph G with at least one edge and $\mathrm{maxcut}(G) \geq c$, it outputs a set $S \subseteq V$ with $\mathbb{E}[|E(S,\bar{S})|/m] \geq s$.

For $x \in \{\pm 1\}^n$ we write $\mathrm{cut}_G(x) \coloneqq \frac{1}{m}\sum_{\{u,v\}\in E} \frac{1-x_u x_v}{2}$ for the value of the cut $(S_x, \bar{S}_x)$. The analysis below uses the constant

$$\alpha_{\mathrm{GW}} \coloneqq \min_{0<\theta\leq\pi} \frac{2\theta}{\pi(1-\cos\theta)}. \tag{4.6}$$

The minimum exists, because the function is continuous on $(0,\pi]$ and tends to $+\infty$ as $\theta \to 0$. Its value is found by calculus or a computer: the minimizer is $\theta^* \approx 2.3311$, at which $\cos\theta^* \approx -0.69$, and $\alpha_{\mathrm{GW}} \approx 0.87856$. In particular, $\alpha_{\mathrm{GW}} > 0.8785$.

Theorem 4.16 (Goemans–Williamson [GW95]). *There is a randomized polynomial-time algorithm that takes as input a symmetric positive semidefinite matrix $X \in \mathbb{R}^{n\times n}$, $n \geq 2$, with $X_{ii} = 1$ for all i, and outputs a vector $x \in \{\pm 1\}^n$ such that for every pair $u, v \in [n]$ the following hold.*

1) *For every $\varepsilon \geq 0$: if $\frac{1}{2} - \frac{1}{2}X_{uv} \geq 1-\varepsilon$, then $\mathbb{E}_x[\frac{1}{2} - \frac{1}{2}x_u x_v] \geq 1 - 2\sqrt{\varepsilon}$. In fact, $\mathbb{E}_x[\frac{1}{2} - \frac{1}{2}x_u x_v] \geq 1 - \sqrt{\varepsilon}$.*

2) *$\mathbb{E}_x[\frac{1}{2} - \frac{1}{2}x_u x_v] \geq \alpha_{\mathrm{GW}}(\frac{1}{2} - \frac{1}{2}X_{uv})$, where α_{GW} is the constant of (4.6).*

The algorithm is Algorithm 4.23 below, and the proof of the theorem follows it. As is customary, we analyze it assuming exact real arithmetic and exact samples from the Gaussian distribution; we do not treat numerical precision.

Corollary 4.17. *For every $\varepsilon \in (0,1]$, there is a randomized polynomial-time $(1-\varepsilon, 1-2\sqrt{\varepsilon})$-approximation algorithm for Max-Cut. Moreover, the approximation ratio of this algorithm is at least 0.878.*

Proof. Let G be a graph with $V = [n]$ and $m \geq 1$ edges, so that $n \geq 2$. The algorithm computes a feasible solution X of (4.5) whose objective value is at least $\mathrm{SDP}(G) - \delta$, where $\delta = \min\{\varepsilon, 10^{-4}\}$, using Fact 4.14. It then applies the algorithm of Theorem 4.16 to X and outputs the resulting x, that is, the set S_x.

First, $\mathrm{SDP}(G) \geq \mathrm{maxcut}(G)$, because the semidefinite program is a relaxation: if $x^* \in \{\pm 1\}^n$ is the indicator vector of a maximum cut, then $X = x^*(x^*)^\top$ is feasible and, by (4.4), its objective value is $\mathrm{cut}_G(x^*) = \mathrm{maxcut}(G)$.

Now suppose that $\mathrm{maxcut}(G) \geq 1 - \varepsilon$. By linearity of expectation,

$$\mathbb{E}_x[\mathrm{cut}_G(x)] = \frac{1}{m}\sum_{\{u,v\}\in E} \mathbb{E}_x\Big[\frac{1}{2} - \frac{1}{2}x_u x_v\Big],$$

and the theorem bounds each term. The hypothesis of the theorem holds only on average over the edges, not for each edge, so we apply it to each edge with its own parameter. For $e = \{u, v\} \in E$ let $\varepsilon_e := 1 - (\frac{1}{2} - \frac{1}{2}X_{uv}) \in [0, 1]$. The bound $\mathbb{E}_x[\frac{1}{2} - \frac{1}{2}x_u x_v] \geq 1 - \sqrt{\varepsilon_e}$ of Theorem 4.16(1), applied with $\varepsilon = \varepsilon_e$, and the concavity of the square root give

$$\begin{aligned}\mathbb{E}_x[\mathrm{cut}_G(x)] &\geq 1 - \frac{1}{m}\sum_{e\in E}\sqrt{\varepsilon_e} \geq 1 - \sqrt{\frac{1}{m}\sum_{e\in E}\varepsilon_e} \\ &\geq 1 - \sqrt{1 - (\mathrm{SDP}(G) - \delta)} \geq 1 - \sqrt{\varepsilon + \delta} \geq 1 - \sqrt{2\varepsilon} \geq 1 - 2\sqrt{\varepsilon}.\end{aligned}$$

Here we used that $\frac{1}{m}\sum_e \varepsilon_e$ is 1 minus the objective value of X, and that $\mathrm{SDP}(G) \geq \mathrm{maxcut}(G) \geq 1 - \varepsilon$.

A similar argument gives the approximation ratio. By Theorem 4.16(2), averaged over the edges, then by $\mathrm{SDP}(G) \geq \mathrm{maxcut}(G)$ and by $\mathrm{maxcut}(G) \geq 1/2$ (Lemma 4.5),

$$\begin{aligned}\mathbb{E}_x[\mathrm{cut}_G(x)] &\geq \alpha_{\mathrm{GW}}(\mathrm{SDP}(G) - \delta) \geq \alpha_{\mathrm{GW}}(\mathrm{maxcut}(G) - 10^{-4}) \\ &\geq \alpha_{\mathrm{GW}}(1 - 2\cdot 10^{-4})\,\mathrm{maxcut}(G) \geq 0.878\,\mathrm{maxcut}(G),\end{aligned}$$

since $\alpha_{\mathrm{GW}}(1 - 2\cdot 10^{-4}) > 0.8783$. Both steps run in polynomial time. □

4.7 Basic facts

The proof of Theorem 4.16 uses a few basic facts about positive semidefinite matrices and Gaussian random vectors.

Lemma 4.18 (Cholesky factorization)**.** *For every positive semidefinite $X \in \mathbb{R}^{n\times n}$ there is a matrix $Z \in \mathbb{R}^{n\times n}$ with $X = ZZ^\top$.*

Proof. Since X is symmetric, it has an eigenvalue decomposition $X = U\Lambda U^\top$, where U is orthogonal and Λ is diagonal. Since $X \succeq 0$, the diagonal entries of Λ, the eigenvalues of X, are nonnegative. Let $\Lambda^{1/2}$ be the diagonal matrix of their square roots, and set $Z = U\Lambda^{1/2}$. Then $ZZ^\top = U\Lambda^{1/2}\Lambda^{1/2}U^\top = U\Lambda U^\top = X$. □

The factor Z constructed in the proof is in general not triangular. The Cholesky factorization in the strict sense, with Z lower triangular, also exists for every positive semidefinite X, but any factor serves below. If $z_1, \ldots, z_n \in \mathbb{R}^n$ are the rows of Z, then $X_{uv} = \langle z_u, z_v\rangle$ for all u, v. In particular, if $X_{uu} = 1$ for all u, then the rows z_u are unit vectors.

Definition 4.19. The *standard Gaussian distribution* on $\mathbb{R}$ is the distribution with probability density function $t \mapsto \frac{1}{\sqrt{2\pi}}e^{-t^2/2}$. A *standard Gaussian vector* in $\mathbb{R}^n$ is a random vector $g = (g_1, \ldots, g_n)$ whose coordinates are independent standard Gaussians.

By independence, the density of a standard Gaussian vector is

$$p(x) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}}e^{-x_i^2/2} = \frac{1}{(\sqrt{2\pi})^n}e^{-\sum_i x_i^2/2} = \frac{1}{(\sqrt{2\pi})^n}e^{-\|x\|_2^2/2}, \qquad x \in \mathbb{R}^n.$$

Proposition 4.20 (Rotation invariance)**.** *Let $H \in \mathbb{R}^{n\times n}$ be orthogonal, that is, $H^\top H = HH^\top = I$, and let g be a standard Gaussian vector in $\mathbb{R}^n$. Then Hg has the same distribution as g.*

Proof. The map $x \mapsto Hx$ is linear and invertible with $|\det H| = 1$ and inverse $H^\top$, so by the change of variables formula, Hg has the density $y \mapsto p(H^\top y)$. The ℓ_2 norm is invariant under orthogonal maps, $\|H^\top y\|_2 = \|y\|_2$, so $p(H^\top y) = p(y)$ for all y, and Hg has the same density as g. □

Corollary 4.21. *Let $g = (g_1, g_2)$ be a standard Gaussian vector in $\mathbb{R}^2$. Then the point $g/\|g\|_2$ is uniformly distributed on the unit circle. Equivalently, writing $(g_1, g_2) = R(\cos\varphi, \sin\varphi)$ with $R > 0$ and $\varphi \in [0, 2\pi)$, the angle φ is uniformly distributed on $[0, 2\pi)$.*

Proof. The vector g is nonzero with probability 1, so $P := g/\|g\|_2$ is defined with probability 1. For any two unit vectors $a, b \in \mathbb{R}^2$ there is an orthogonal H with $Ha = b$, namely the rotation taking a to b. By Proposition 4.20, Hg has the same distribution as g, so the distribution of P is the same near a as near b. To make this precise, let H be a rotation. Then $H^\top g$ has the same distribution as g and $\|H^\top g\|_2 = \|g\|_2$, so for every arc A of the unit circle,

$$\Pr[P \in HA] = \Pr[H^\top P \in A] = \Pr[P \in A].$$

Since a rotation maps an arc onto any other arc of the same length, the probability that P lies in an arc depends only on the length of the arc. A single point has probability 0, because $P = a$ means that g lies on a ray, a set of Lebesgue measure 0. For every integer $k \geq 1$, the circle is the disjoint union of k half-open arcs of length $2\pi/k$, so each of them has probability $1/k$, and an arc of length $2\pi j/k$ has probability j/k. By monotonicity, every arc of length ℓ has probability $\ell/(2\pi)$; that is, P is uniform on the circle. □

Corollary 4.22. *Let $n \geq 2$, let $z_1, z_2 \in \mathbb{R}^n$ be unit vectors with $\beta = \langle z_1, z_2\rangle$, and let $g = (g_1, \ldots, g_n)$ be a standard Gaussian vector in $\mathbb{R}^n$. Then $|\beta| \leq 1$, and*

$$(\langle z_1, g\rangle, \langle z_2, g\rangle) \quad \textit{has the same distribution as} \quad (g_1,\ \beta g_1 + \sqrt{1-\beta^2}\, g_2).$$

Proof. By the Cauchy–Schwarz inequality, $|\beta| \leq \|z_1\|_2 \|z_2\|_2 = 1$. There is an orthogonal matrix H with

$$Hz_1 = e_1 \quad \text{and} \quad Hz_2 = \beta e_1 + \sqrt{1-\beta^2}\, e_2.$$

Indeed, if $|\beta| < 1$, extend the orthonormal pair z_1, $w = (z_2 - \beta z_1)/\sqrt{1-\beta^2}$ to an orthonormal basis $z_1, w, b_3, \ldots, b_n$ of $\mathbb{R}^n$ and let H have these vectors as its rows; if $|\beta| = 1$, then $z_2 = \beta z_1$, and any orthonormal basis starting with z_1 works. Now we use rotation invariance. Since $H^\top H = I$, we have $\langle z_i, g\rangle = \langle Hz_i, Hg\rangle$, and Hg has the same distribution as g by Proposition 4.20. Hence $(\langle z_1, g\rangle, \langle z_2, g\rangle)$ has the same distribution as $(\langle Hz_1, g\rangle, \langle Hz_2, g\rangle) = (g_1, \beta g_1 + \sqrt{1-\beta^2}\, g_2)$. □

4.8 Proof of the Goemans–Williamson theorem

The algorithm of Theorem 4.16 rounds the vectors z_u by a random hyperplane through the origin, with a Gaussian normal vector g. We use the convention $\operatorname{sign}(t) = 1$ for $t \geq 0$ and $\operatorname{sign}(t) = -1$ for $t < 0$; the value at 0 does not matter, because $\langle g, z_u\rangle = 0$ has probability 0 for a unit vector z_u.

Algorithm 4.23 (Random hyperplane rounding)**.** Input: a symmetric positive semidefinite $X \in \mathbb{R}^{n\times n}$ with $X_{uu} = 1$ for all $u \in [n]$.

1) Compute $Z \in \mathbb{R}^{n\times n}$ with $X = ZZ^\top$ (Lemma 4.18), and let $z_1, \ldots, z_n$ be its rows.

2) Sample a standard Gaussian vector $g = (g_1, \dots, g_n)$.

3) For each $u \in [n]$, set $x_u = \operatorname{sign}(\langle g, z_u \rangle)$.

Output: $x = (x_1, \dots, x_n) \in \{\pm 1\}^n$.

Analysis of the rounding. Fix $u, v \in [n]$. Then $(x_u, x_v) = (\operatorname{sign}\langle g, z_u\rangle, \operatorname{sign}\langle g, z_v\rangle)$, and we care about the random variable

$$\frac{1}{2} - \frac{1}{2} x_u x_v = \begin{cases} 0 & \text{if } x_u = x_v, \\ 1 & \text{if } x_u \neq x_v, \end{cases}$$

whose expectation is $\Pr[x_u \neq x_v]$. By Corollary 4.22, the pair $(\langle z_u, g\rangle, \langle z_v, g\rangle)$ is distributed as a centered Gaussian vector in $\mathbb{R}^2$ with covariance matrix $\left(\begin{smallmatrix} 1 & X_{uv} \\ X_{uv} & 1 \end{smallmatrix}\right)$. So it remains to answer an elementary question, the probability that two such correlated Gaussians have different signs; Sheppard's lemma answers it. First we check that the correlation X_{uv} lies in $[-1, 1]$.

Lemma 4.24. *If $X \in \mathbb{R}^{n \times n}$ is positive semidefinite and $X_{uu} = X_{vv} = 1$, then $|X_{uv}| \leq 1$.*

Proof. Apply $w^\top X w \geq 0$ to $w = e_u + e_v$ and to $w = e_u - e_v$, and use $X_{vu} = X_{uv}$: this gives $2 + 2X_{uv} \geq 0$ and $2 - 2X_{uv} \geq 0$. Rearranging, $-1 \leq X_{uv} \leq 1$. □

Lemma 4.25 (Sheppard's lemma [She99])**.** *Let $n \geq 2$, let $z_u, z_v \in \mathbb{R}^n$ be unit vectors, and write $\langle z_u, z_v\rangle = X_{uv} = -1 + \eta$, where $\eta \in [0, 2]$. Let g be a standard Gaussian vector in $\mathbb{R}^n$, and let $x_u = \operatorname{sign}\langle g, z_u\rangle$ and $x_v = \operatorname{sign}\langle g, z_v\rangle$. Then*

$$\Pr[x_u \neq x_v] = \frac{\arccos(X_{uv})}{\pi}, \tag{4.7}$$

and

$$\Pr[x_u \neq x_v] = 1 - \frac{\sqrt{2\eta}}{\pi} + O(\eta^{3/2}) \quad \text{as } \eta \to 0.$$

Moreover, $\Pr[x_u \neq x_v] \geq 1 - \sqrt{\eta/2} \geq 1 - 2\sqrt{\eta}$ for every $\eta \in [0, 2]$.

Proof. We reduce to two-dimensional geometry. Let $\theta_{uv} = \arccos(X_{uv}) \in [0, \pi]$ be the angle between z_u and z_v, so that $X_{uv} = \cos\theta_{uv}$ and $\sqrt{1 - X_{uv}^2} = \sin\theta_{uv}$. By Corollary 4.22, the pair $(\langle z_u, g\rangle, \langle z_v, g\rangle)$ has the same distribution as

$$(g_1,\ (-1+\eta) g_1 + \sqrt{2\eta - \eta^2}\, g_2) = (g_1,\ \cos\theta_{uv}\, g_1 + \sin\theta_{uv}\, g_2),$$

since $1 - (1-\eta)^2 = 2\eta - \eta^2$. Hence

$$\Pr[x_u \neq x_v] = \Pr\bigl[\operatorname{sign}(g_1) \neq \operatorname{sign}\bigl((-1+\eta) g_1 + \sqrt{2\eta - \eta^2}\, g_2\bigr)\bigr].$$

Write $(g_1, g_2) = R(\cos\varphi, \sin\varphi)$ with $R > 0$; by Corollary 4.21, φ is uniform on $[0, 2\pi)$. Then $g_1 = R\cos\varphi$ and $\cos\theta_{uv}\, g_1 + \sin\theta_{uv}\, g_2 = R\cos(\varphi - \theta_{uv})$, so the two signs are those of $\cos\varphi$ and $\cos(\varphi - \theta_{uv})$. In the plane, these are the signs of the inner products of the direction $(\cos\varphi, \sin\varphi)$ with e_1 and with $(\cos\theta_{uv}, \sin\theta_{uv})$, the images of z_u and z_v; see Fig. 4. Apart from the finitely many angles at which one of the cosines vanishes, which have probability 0, the signs differ exactly when φ lies in one of the two arcs $(\frac{\pi}{2}, \frac{\pi}{2} + \theta_{uv})$ and $(\frac{3\pi}{2}, \frac{3\pi}{2} + \theta_{uv})$, taken modulo 2π.

Indeed, on the first arc $\cos\varphi < 0 < \cos(\varphi - \theta_{uv})$, on the second $\cos(\varphi - \theta_{uv}) < 0 < \cos\varphi$, and on the two remaining arcs both cosines have the same sign. The two arcs have total length $2\theta_{uv}$, so

$$\Pr[x_u \neq x_v] = \frac{2\theta_{uv}}{2\pi} = \frac{\theta_{uv}}{\pi} = \frac{\arccos(X_{uv})}{\pi}.$$

For the expansion, write $X_{uv} = -(1-\eta)$. Then $\arccos(-(1-\eta)) = \pi - \arccos(1-\eta)$, and

$$\arccos(1-\eta) = \sqrt{2\eta} + \frac{(2\eta)^{3/2}}{24} + O(\eta^2).$$

To see this, let $\psi = \arcsin\sqrt{\eta/2} \in [0, \pi/2]$. Then $\cos(2\psi) = 1 - 2\sin^2\psi = 1 - \eta$ and $2\psi \in [0, \pi]$, so

$$\arccos(1-\eta) = 2\arcsin\sqrt{\eta/2} \qquad \text{for } \eta \in [0,2], \tag{4.8}$$

and the expansion follows from $\arcsin t = t + t^3/6 + O(t^5)$ with $t = \sqrt{\eta/2}$. Plugging in,

$$\Pr[x_u \neq x_v] = \frac{\pi - \arccos(1-\eta)}{\pi} = 1 - \frac{\sqrt{2\eta}}{\pi} + O(\eta^{3/2}).$$

For the bound valid for every $\eta \in [0,2]$, note that $\arcsin$ is convex on $[0,1]$ (its second derivative $t/(1-t^2)^{3/2}$ is nonnegative on $[0,1)$), with $\arcsin 0 = 0$ and $\arcsin 1 = \pi/2$, so $\arcsin t \leq \frac{\pi}{2}t$ for $t \in [0,1]$. By (4.8), $\arccos(1-\eta) \leq \pi\sqrt{\eta/2}$, and therefore

$$\Pr[x_u \neq x_v] = 1 - \frac{\arccos(1-\eta)}{\pi} \geq 1 - \sqrt{\eta/2} \geq 1 - 2\sqrt{\eta}. \qquad \square$$

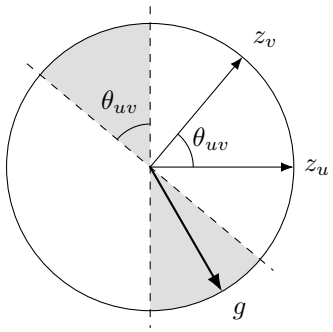


Figure 4: The plane spanned by z_u and z_v, at angle θ_{uv}. The dashed lines are the lines orthogonal to z_u and to z_v. A direction g in one of the two shaded sectors, each of angle θ_{uv}, has inner products of opposite signs with z_u and z_v, so $x_u \neq x_v$; for a uniformly random direction this happens with probability $2\theta_{uv}/(2\pi)$.

Proof of Theorem 4.16. The algorithm is Algorithm 4.23. Its first step takes polynomial time, and so do the others. Fix $u, v \in [n]$. The rows z_u and z_v are unit vectors with $\langle z_u, z_v\rangle = X_{uv}$, so Lemma 4.25 applies with $\eta = 1 + X_{uv} \in [0,2]$ (Lemma 4.24), and $\mathbb{E}_x[\frac{1}{2} - \frac{1}{2}x_u x_v] = \Pr[x_u \neq x_v]$.

(1) The hypothesis $\frac{1}{2} - \frac{1}{2}X_{uv} \geq 1 - \varepsilon$ means $\eta = 1 + X_{uv} \leq 2\varepsilon$. By Lemma 4.25,

$$\Pr[x_u \neq x_v] \geq 1 - \sqrt{\eta/2} \geq 1 - \sqrt{\varepsilon} \geq 1 - 2\sqrt{\varepsilon}.$$

(2) By (4.7), $\Pr[x_u \neq x_v] = \theta_{uv}/\pi$, where $\theta_{uv} = \arccos(X_{uv}) \in [0, \pi]$ and $\frac{1}{2} - \frac{1}{2}X_{uv} = \frac{1-\cos\theta_{uv}}{2}$. If $\theta_{uv} = 0$, both sides of the claimed inequality are 0. Otherwise

$$\frac{\Pr[x_u \neq x_v]}{\frac{1}{2} - \frac{1}{2}X_{uv}} = \frac{2\theta_{uv}}{\pi(1 - \cos\theta_{uv})} \geq \alpha_{\mathrm{GW}}$$

by the definition (4.6) of α_{GW}. The worst case is an edge with $X_{uv} = \cos\theta^* \approx -0.69$. □

4.9 Beyond the Goemans–Williamson algorithm

The Goemans–Williamson algorithm can be improved in some settings.

1) For graphs of bounded degree one can beat 0.878: for every integer Δ there is a constant $\varepsilon_\Delta > 0$ and a polynomial-time algorithm with approximation ratio at least $\alpha_{\mathrm{GW}} + \varepsilon_\Delta$ on graphs of maximum degree at most Δ [FKL02].

2) There is a rounding of the semidefinite program (4.5) that does better than the random hyperplane in some regimes of c, which gives a better (c, s)-approximation curve [OW08].

The limitations of the Goemans–Williamson algorithm are the subject of Lecture 5. The Unique Games Conjecture is introduced in Lecture 16; assuming it, no polynomial-time algorithm has approximation ratio $\alpha_{\mathrm{GW}} + \varepsilon$ for any constant $\varepsilon > 0$, unless P = NP [KKMO07, MOO10]. In this sense the Goemans–Williamson algorithm is optimal.

5 Max-Cut: Integrality Gaps and Rounding Gaps

Lecture of September 10, 2021, by Pravesh K. Kothari. Based on the lecturer's handwritten notes.

The Goemans–Williamson algorithm of Lecture 4 cuts, in expectation, at least $\alpha_{\mathrm{GW}} \approx 0.878$ times the value of a semidefinite relaxation of Max-Cut; this lecture asks whether a better rounding of the same relaxation could do better. Integrality gaps, graphs whose max-cut is much smaller than the value of the relaxation, limit what any rounding can achieve. Odd cycles show that the guarantee $1 - O(\sqrt{\varepsilon})$ on graphs of value $1 - \varepsilon$ has the right form, and the construction of Feige and Schechtman [FS02] on the high-dimensional sphere shows that no rounding of this relaxation beats the ratio α_{GW}. Rounding gaps are graphs on which the relaxation is exact but the random hyperplane rounding of an optimal solution still loses the factor α_{GW}. We prove Karloff's theorem [Kar99] that such graphs exist, using a construction on the hypercube and a basic fact from Fourier analysis on the hypercube (Lecture 16).

5.1 The Goemans–Williamson algorithm, revisited

Until weighted graphs are introduced in Definition 5.6, $G = (V, E)$ is an undirected graph without loops or parallel edges, with $n = |V|$ vertices and $m = |E| \geq 1$ edges. We write $\{u, v\} \sim E$ for an edge chosen uniformly at random from E. Identifying a finite set with the uniform distribution on it turns averages over the set into expectations, and we use this throughout.

As in Lecture 4, Max-Cut is the quadratic program

$$\mathrm{maxcut}(G) = \max_{x \in \{\pm 1\}^V} \frac{1}{4m} \sum_{\{u,v\} \in E} (x_u - x_v)^2 = \max_{x \in \{\pm 1\}^V} \Big(\frac{1}{2} - \frac{1}{2} \mathbb{E}_{\{u,v\} \sim E} x_u x_v \Big), \tag{5.1}$$

where we used $(x_u - x_v)^2 = 2 - 2x_u x_v$ for $x_u, x_v \in \{\pm 1\}$. Replacing the rank-one matrix $xx^\top$ by an arbitrary positive semidefinite matrix with unit diagonal gives the semidefinite relaxation

$$\mathrm{SDP}(G) \coloneqq \max \Big\{ \frac{1}{2} - \frac{1}{2} \mathbb{E}_{\{u,v\} \sim E} X_{uv} \;:\; X \in \mathbb{R}^{V \times V} \text{ symmetric},\ X \succeq 0,\ X_{uu} = 1 \text{ for all } u \in V \Big\}. \tag{5.2}$$

A symmetric matrix X is positive semidefinite with unit diagonal if and only if $X_{uv} = \langle z_u, z_v \rangle$ for some unit vectors $z_u \in \mathbb{R}^n$: one direction is the Cholesky factorization $X = ZZ^\top$ of Lecture 4, and conversely $w^\top X w = \| \sum_u w_u z_u \|_2^2 \geq 0$. Hence the semidefinite program assigns a unit vector to each vertex, and

$$\mathrm{SDP}(G) = \max \Big\{ \frac{1}{2} - \frac{1}{2} \mathbb{E}_{\{u,v\} \sim E} \langle z_u, z_v \rangle \;:\; z_u \in \mathbb{S}^{n-1} \text{ for all } u \in V \Big\}, \tag{5.3}$$

where $\mathbb{S}^{d-1} = \{ z \in \mathbb{R}^d : \|z\|_2 = 1 \}$ denotes the unit sphere of $\mathbb{R}^d$. We call such an assignment $u \mapsto z_u$ a *solution* or an *embedding*. Since $X = xx^\top$ is feasible for every $x \in \{\pm 1\}^V$, we have $\mathrm{SDP}(G) \geq \mathrm{maxcut}(G)$.

The Goemans–Williamson algorithm [GW95] rounds a solution as follows.

Algorithm 5.1 (Random hyperplane rounding)**.** Input: a feasible solution X of (5.2).

1) Compute $Z \in \mathbb{R}^{V \times n}$ with $X = ZZ^\top$ (Cholesky factorization), and let z_u be the row of Z indexed by u, so that $X_{uv} = \langle z_u, z_v \rangle$ and $\|z_u\|_2 = 1$.

2) Choose a standard Gaussian vector $g = (g_1, \ldots, g_n)$.

3) Output $x \in \{\pm 1\}^V$ with $x_u = \operatorname{sign}(\langle g, z_u \rangle)$ for all $u \in V$.

By Sheppard's formula (Lecture 4), for every pair $u, v \in V$,

$$\mathbb{E}_x\Big[\frac{1}{2} - \frac{1}{2}x_u x_v\Big] = \Pr[x_u \neq x_v] = \frac{\arccos(X_{uv})}{\pi}. \tag{5.4}$$

In particular, if $\frac{1}{2} - \frac{1}{2}X_{uv} = 1 - \eta$, then $\mathbb{E}_x[\frac{1}{2} - \frac{1}{2}x_u x_v] \geq 1 - 2\sqrt{\eta}$. Thus an edge whose endpoints have inner product $\rho \in [-1, 1]$ contributes $\frac{1}{2} - \frac{1}{2}\rho$ to the objective of the semidefinite program and $\arccos(\rho)/\pi$ to the expected value of the rounded cut. The worst ratio of the two defines the constant of the algorithm.

Definition 5.2. For $\rho \in [-1, 1)$ let

$$\gamma(\rho) \coloneqq \frac{\arccos(\rho)/\pi}{\frac{1}{2} - \frac{1}{2}\rho}.$$

Let $\alpha_{\mathrm{GW}} \coloneqq \min_{-1 \leq \rho < 1} \gamma(\rho)$, and let ρ_* be a point at which the minimum is attained.

The minimum exists, because γ is continuous on $[-1, 1)$ and tends to $+\infty$ as $\rho \to 1$. With $\theta = \arccos \rho \in (0, \pi]$ we have $\gamma(\rho) = 2\theta/(\pi(1 - \cos\theta))$, so α_{GW} is the constant of Lecture 4. The minimum is attained at exactly one point. Indeed, the derivative of $2\theta/(\pi(1 - \cos\theta))$ in θ has the sign of $\varphi(\theta) = 1 - \cos\theta - \theta \sin\theta$. Since $\varphi(0) = 0$ and $\varphi'(\theta) = -\theta\cos\theta$, the function φ is negative on $(0, \pi/2]$ and strictly increasing on $[\pi/2, \pi]$, with $\varphi(\pi) = 2$. So φ has exactly one zero, at which γ, as a function of θ, changes from decreasing to increasing. Numerically, $\rho_* \approx -0.689$ and $\alpha_{\mathrm{GW}} \approx 0.878$. Averaging (5.4) over the edges shows that the rounding of an optimal solution outputs a cut of expected value $\mathbb{E}_x[\frac{1}{2} - \frac{1}{2}\mathbb{E}_{\{u,v\}\sim E} x_u x_v] \geq \alpha_{\mathrm{GW}} \operatorname{SDP}(G)$.

This raises two questions: whether some algorithm for Max-Cut does better, and, more modestly, whether some other rounding of the same semidefinite program does better. We explore them through *gaps*: graphs on which the relaxation, or its rounding, provably loses.

5.2 Integrality gaps and odd cycles

The first question is how far $\operatorname{SDP}(G)$ can be from $\operatorname{maxcut}(G)$.

Definition 5.3. The *integrality gap* of the semidefinite program (5.2) on a graph G is the ratio $\operatorname{maxcut}(G)/\operatorname{SDP}(G)$, a number in $(0, 1]$. The integrality gap of the semidefinite program is the infimum of this ratio over all graphs with at least one edge.

No rounding of a solution can produce a cut better than $\operatorname{maxcut}(G)$, so on a graph with integrality gap β no rounding guarantees more than $\beta \cdot \operatorname{SDP}(G)$. In the other direction, the Goemans–Williamson analysis shows that the integrality gap is at least α_{GW}.

Lemma 5.4 (Qualitative integrality gap)**.** *For an integer $k \geq 1$, let C_{2k+1} be the cycle on the $2k+1$ vertices $0, 1, \ldots, 2k$, with the edges $\{j, j+1\}$ for $0 \leq j < 2k$ and the edge $\{2k, 0\}$. Then $\operatorname{SDP}(C_{2k+1}) \geq 1 - O(1/k^2)$; more precisely,*

$$\operatorname{SDP}(C_{2k+1}) \geq \frac{1}{2} + \frac{1}{2}\cos\frac{\pi}{2k+1} \geq 1 - \frac{\pi^2}{4(2k+1)^2}.$$

On the other hand, $\operatorname{maxcut}(C_{2k+1}) = 1 - \frac{1}{2k+1}$.

Proof. Every cut crosses a closed walk an even number of times, so no cut cuts all $2k+1$ edges of the odd cycle, and $\operatorname{maxcut}(C_{2k+1}) \le \frac{2k}{2k+1}$. Putting the even vertices on one side and the odd vertices on the other cuts every edge except $\{2k, 0\}$, so equality holds.

To bound $\operatorname{SDP}(C_{2k+1})$ from below, we must exhibit a solution, that is, unit vectors $z_0, \dots, z_{2k}$, with $\mathbb{E}_{\{u,v\}\sim E}\langle z_u, z_v\rangle \le -(1 - O(1/k^2))$. So the two endpoints of every edge must be placed at an obtuse angle as close to π as possible. We divide the unit circle of $\mathbb{R}^2$ into $2k+1$ equal sectors and set, with $\phi = 2\pi/(2k+1)$,

$$z_j := (\cos(jk\phi), \sin(jk\phi)) \in \mathbb{S}^1 \qquad (j \in \mathbb{Z}),$$

padded with zeros to vectors of $\mathbb{R}^{2k+1}$. Thus vertex j sits at the $(jk \bmod (2k+1))$-th of the $2k+1$ equally spaced points of the circle. Since $\gcd(k, 2k+1) = 1$, the map $j \mapsto jk \bmod (2k+1)$ is a bijection of $\{0, 1, \dots, 2k\}$, so every point carries exactly one vertex (Fig. 5). Since $(2k+1)k\phi = 2\pi k$, the vector z_j depends only on $j \bmod (2k+1)$, and for every j the vector z_{j+1} is z_j rotated by the angle

$$k\phi = \frac{2\pi k}{2k+1} = \pi - \frac{\pi}{2k+1}.$$

This covers every edge of C_{2k+1}, including $\{2k, 0\}$. Hence the endpoints of every edge are at the angle $\pi - \frac{\pi}{2k+1}$, and

$$\mathbb{E}_{\{u,v\}\sim E}\langle z_u, z_v\rangle = \cos\Big(\pi - \frac{\pi}{2k+1}\Big) = -\cos\Big(\frac{\pi}{2k+1}\Big) \le -1 + \frac{\pi^2}{2(2k+1)^2},$$

where we used $\cos x \ge 1 - x^2/2$. By (5.3), $\operatorname{SDP}(C_{2k+1}) \ge \frac{1}{2} + \frac{1}{2}\cos\frac{\pi}{2k+1} \ge 1 - \frac{\pi^2}{4(2k+1)^2}$. □

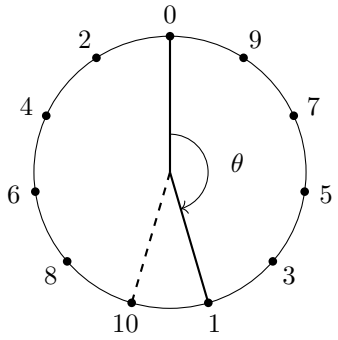


Figure 5: The embedding of C_{11} ($k = 5$) in the proof of Lemma 5.4: vertex j sits at the point $(5j \bmod 11) \cdot 2\pi/11$ of the circle, measured clockwise from the top. Each vertex is at the angle $\theta = 10\pi/11 = \pi - \pi/11$ from both of its neighbors. The solid radii show the edge $\{0, 1\}$, and the dashed radius points to vertex 10, the other neighbor of 0.

Remark 5.5. In terms of a parameter ε, take k of order $1/\sqrt{\varepsilon}$. Then the odd cycle has $\operatorname{SDP}(C_{2k+1}) \ge 1 - O(\varepsilon)$, while $\operatorname{maxcut}(C_{2k+1}) = 1 - \Theta(\sqrt{\varepsilon})$. Precisely, with $\varepsilon_k = \pi^2/(4(2k+1)^2)$ we have $\operatorname{SDP}(C_{2k+1}) \ge 1 - \varepsilon_k$ and $\operatorname{maxcut}(C_{2k+1}) = 1 - (2/\pi)\sqrt{\varepsilon_k}$. The Goemans–Williamson analysis turns a solution of value $1 - \varepsilon$ into a cut of expected value at least $1 - 2\sqrt{\varepsilon}$ (Lecture 4). So the odd cycles give an integrality gap that matches this approximation curve qualitatively: no rounding of the semidefinite program improves the loss $O(\sqrt{\varepsilon})$ to $o(\sqrt{\varepsilon})$.

5.3 The Feige–Schechtman integrality gap

The odd cycles are embedded on the circle, the unit sphere of $\mathbb{R}^2$. A higher-dimensional generalization of this construction, due to Feige and Schechtman [FS02], gives integrality gap at most $\alpha_{\mathrm{GW}} + \varepsilon$ for every $\varepsilon > 0$, which matches the Goemans–Williamson analysis exactly. The construction is a weighted graph, and we first extend the definitions to such graphs.

Definition 5.6. A *weighted graph* $G = (V, \mathcal{E})$ consists of a finite set V and a probability distribution $\mathcal{E}$ on $V \times V$ that is symmetric, $\mathcal{E}(u, v) = \mathcal{E}(v, u)$ for all u, v, and satisfies $\mathcal{E}(u, v) > 0$ for some $u \neq v$. We write $(u, v) \sim \mathcal{E}$ for a random edge. Define

$$\begin{aligned}\operatorname{maxcut}(G) &\coloneqq \max_{x \in \{\pm 1\}^V} \mathbb{E}_{(u,v)\sim\mathcal{E}}\Big[\frac{1}{2} - \frac{1}{2}x_u x_v\Big],\\ \operatorname{SDP}(G) &\coloneqq \max\Big\{\mathbb{E}_{(u,v)\sim\mathcal{E}}\Big[\frac{1}{2} - \frac{1}{2}X_{uv}\Big] \;:\; X \in \mathbb{R}^{V\times V} \text{ symmetric},\ X \succeq 0,\\ &\qquad\qquad X_{uu} = 1 \text{ for all } u \in V\Big\}.\end{aligned}$$

For a feasible X, let $\operatorname{alg}_{\mathrm{GW}}(G; X)$ be the expected value $\mathbb{E}_x \mathbb{E}_{(u,v)\sim\mathcal{E}}[\frac{1}{2} - \frac{1}{2}x_u x_v]$ of the cut output by Algorithm 5.1 on input X. For unit vectors z_u, $u \in V$, we write $\operatorname{alg}_{\mathrm{GW}}(G; z)$ for $\operatorname{alg}_{\mathrm{GW}}(G; X)$ with the Gram matrix $X_{uv} = \langle z_u, z_v\rangle$.

A graph $G = (V, E)$ is the weighted graph in which $(u, v) \sim \mathcal{E}$ is a uniformly random edge $\{u, v\} \sim E$ listed in a uniformly random order; for it, the definitions agree with (5.1) and (5.2). Pairs (u, u), which are loops, are allowed, but they contribute nothing to either objective, since $x_u x_u = X_{uu} = 1$. As before, X may be replaced by the Gram matrix of unit vectors z_u, and $\operatorname{SDP}(G) \geq \operatorname{maxcut}(G)$. The analysis of the rounding is edge by edge, so it applies verbatim: by (5.4) and Definition 5.2,

$$\operatorname{alg}_{\mathrm{GW}}(G; X) = \mathbb{E}_{(u,v)\sim\mathcal{E}} \frac{\arccos(X_{uv})}{\pi} \geq \alpha_{\mathrm{GW}} \mathbb{E}_{(u,v)\sim\mathcal{E}}\Big[\frac{1}{2} - \frac{1}{2}X_{uv}\Big],$$

because $\arccos(X_{uv})/\pi = \gamma(X_{uv})(\frac{1}{2} - \frac{1}{2}X_{uv})$ when $X_{uv} < 1$, and both sides vanish when $X_{uv} = 1$. For an optimal X this gives $\operatorname{maxcut}(G) \geq \operatorname{alg}_{\mathrm{GW}}(G; X) \geq \alpha_{\mathrm{GW}} \operatorname{SDP}(G)$. The integrality gap of a weighted graph is defined as in Definition 5.3; it is at least α_{GW}.

Theorem 5.7 (Feige–Schechtman [FS02])**.** *For every $\varepsilon > 0$ there is a weighted graph G such that*

$$\operatorname{maxcut}(G) \leq (\alpha_{\mathrm{GW}} + \varepsilon) \operatorname{SDP}(G).$$

Consequently, the integrality gap of the semidefinite program over weighted graphs equals α_{GW}.

The proof is at the end of this subsection.

Preliminary considerations. Suppose that we construct a weighted graph G together with an embedding $u \mapsto z_u \in \mathbb{S}^{d-1}$ such that every edge (u, v) in the support of $\mathcal{E}$ has $\langle z_u, z_v\rangle = \rho$, for one fixed $\rho \in [-1, 1)$. Then $\operatorname{SDP}(G) \geq \frac{1}{2} - \frac{1}{2}\rho$. For the integrality gap to be close to α_{GW} when this embedding is optimal, we need $\operatorname{maxcut}(G)$ to be close to $\alpha_{\mathrm{GW}}(\frac{1}{2} - \frac{1}{2}\rho)$. But the rounding of the embedding cuts each edge with probability $\arccos(\rho)/\pi$, so

$$\operatorname{maxcut}(G) \geq \operatorname{alg}_{\mathrm{GW}}(G; z) = \frac{\arccos(\rho)}{\pi} = \gamma(\rho)\Big(\frac{1}{2} - \frac{1}{2}\rho\Big).$$

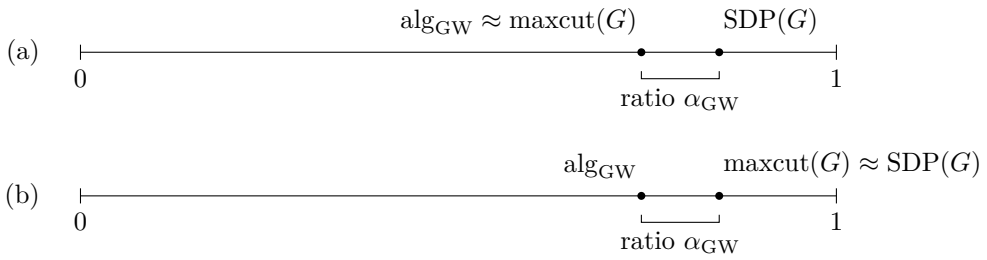


Figure 6: Where the factor α_{GW} is lost. (a) An integrality gap instance: the rounding is nearly optimal, and the relaxation overestimates the max-cut. (b) A rounding gap instance: the relaxation is nearly exact, and the rounding of an optimal solution loses. In both constructions of this lecture the two values approach $\arccos(\rho_*)/\pi \approx 0.742$ and $\frac{1}{2} - \frac{1}{2}\rho_* \approx 0.845$.

So we must have $\gamma(\rho)$ close to its minimum, that is, ρ close to ρ_*, and the rounding must be nearly optimal: $\mathrm{alg}_{\mathrm{GW}} \approx \mathrm{maxcut}(G)$. The whole gap then lies between $\mathrm{maxcut}(G)$ and $\mathrm{SDP}(G)$, as in Fig. 6(a).

We define the graph directly through an embedding into a sphere.

Definition 5.8. An *embedded graph* is a weighted graph $G = (V, \mathcal{E})$ together with a map $V \to \mathbb{S}^{d-1}$, $u \mapsto z_u$, for some $d \geq 1$. Its *objective value* is

$$\mathrm{Obj}(G) \coloneqq \mathbb{E}_{(u,v)\sim\mathcal{E}}\Big[\frac{1}{2} - \frac{1}{2}\langle z_u, z_v\rangle\Big].$$

Proposition 5.9. *For every embedded graph G, $\mathrm{Obj}(G) \leq \mathrm{SDP}(G)$.*

Proof. The Gram matrix $X_{uv} = \langle z_u, z_v\rangle$ is symmetric and positive semidefinite with unit diagonal, so it is feasible, and its objective value is $\mathrm{Obj}(G)$. □

To make $\mathrm{Obj}(G)$ close to $\frac{1}{2} - \frac{1}{2}\rho_*$, we must make $\langle z_u, z_v\rangle$ close to ρ_* for almost every edge.

The sphere graph. We first construct an infinite geometric graph and then discretize it. Its vertices are all the points of the unit sphere $\mathbb{S}^{d-1}$, and its edges form a symmetric probability distribution over pairs of unit vectors. Let σ be the uniform probability measure on $\mathbb{S}^{d-1}$, so that $\sigma(A)$ is the fractional surface area of A.

Definition 5.10. Let $d \geq 2$ and $\rho \in (-1, 0]$. The *sphere graph* $G^{\mathrm{sph}}_{d,\rho}$ has vertex set $\mathbb{S}^{d-1}$, embedded by the identity map, and edge distribution $\mathcal{E}^{\mathrm{sph}}_{d,\rho}$: the distribution of a pair (u, v) of independent σ-distributed points conditioned on $\langle u, v\rangle \leq \rho$. A cut is a measurable set $A \subseteq \mathbb{S}^{d-1}$, or equivalently its ± 1 indicator $f\colon \mathbb{S}^{d-1} \to \{\pm 1\}$, and its value is the fraction of edges crossing it,

$$\mu_\rho(A) \coloneqq \Pr_{(u,v)\sim\mathcal{E}^{\mathrm{sph}}_{d,\rho}}\big[\mathbf{1}_A(u) \neq \mathbf{1}_A(v)\big].$$

Let $\mathrm{maxcut}(G^{\mathrm{sph}}_{d,\rho}) \coloneqq \sup_A \mu_\rho(A)$ over measurable A, and $\mathrm{Obj}(G^{\mathrm{sph}}_{d,\rho}) \coloneqq \mathbb{E}[\frac{1}{2} - \frac{1}{2}\langle u, v\rangle]$ for $(u, v) \sim \mathcal{E}^{\mathrm{sph}}_{d,\rho}$.

The conditioning is well defined, since $\Pr[\langle u,v\rangle \le \rho] > 0$ for $\rho > -1$, and the distribution is symmetric, since (u,v) and (v,u) have the same law. The supremum defining the max-cut is attained, by Corollary 5.15 below. The edges of the sphere graph have inner product at most ρ, and in high dimension most of them have inner product close to ρ. To prove this we use the following standard fact, which we state without proof.

Fact 5.11. *Let $d \ge 2$, and let u, v be independent and σ-distributed on $\mathbb{S}^{d-1}$. Then $\langle u,v\rangle$ has the density $c_d(1-t^2)^{(d-3)/2}$ on $(-1,1)$, where $c_d > 0$ is a normalizing constant.*

Lemma 5.12. *Let $d \ge 4$ and $\rho \in (-1,0)$, and let u, v be independent and σ-distributed on $\mathbb{S}^{d-1}$. Then*

$$\rho \ge \mathbb{E}[\langle u,v\rangle \mid \langle u,v\rangle \le \rho] \ge \rho - O\Big(\frac{1}{\sqrt{d}}\Big);$$

more precisely, the conditional mean is at least $\rho - \frac{1}{(d-3)|\rho|}$.

Proof. The first inequality holds because the conditional distribution is supported on $[-1,\rho]$. For the second, let $a = |\rho| \in (0,1)$ and $\kappa = (d-3)/2 > 0$. By Fact 5.11, conditioned on $\langle u,v\rangle \le \rho$, the variable $s = \rho - \langle u,v\rangle$ has on $[0, 1-a]$ a density proportional to $q(s) = (1-(a+s)^2)^\kappa$. Let $0 \le \eta \le 1-a$ and $0 \le s \le 1-a-\eta$. Then

$$1-(a+s+\eta)^2 = 1-(a+s)^2 - 2(a+s)\eta - \eta^2 \le (1-(a+s)^2)(1-2a\eta),$$

because $2(a+s)\eta \ge 2a\eta \ge 2a\eta(1-(a+s)^2)$. Since $0 \le 2a\eta \le 2a(1-a) \le \frac{1}{2}$, this gives $q(s+\eta) \le (1-2a\eta)^\kappa q(s) \le e^{-(d-3)a\eta} q(s)$. Integrating over s,

$$\int_\eta^{1-a} q(s)\,ds = \int_0^{1-a-\eta} q(s+\eta)\,ds \le e^{-(d-3)a\eta} \int_0^{1-a} q(s)\,ds,$$

that is, $\Pr[s \ge \eta \mid \langle u,v\rangle \le \rho] \le e^{-(d-3)a\eta}$ for every $\eta \ge 0$ (for $\eta > 1-a$ the probability is 0). Hence

$$\mathbb{E}[s \mid \langle u,v\rangle \le \rho] = \int_0^\infty \Pr[s \ge \eta \mid \langle u,v\rangle \le \rho]\,d\eta \le \int_0^\infty e^{-(d-3)a\eta}\,d\eta = \frac{1}{(d-3)a}. \qquad \square$$

The deviation is thus $O(1/d)$, and in particular $O(1/\sqrt{d})$. This is a phenomenon of high-dimensional geometry: conditioned on being at most ρ, the inner product of two random unit vectors is very likely just below ρ. For the objective value, the lemma is not needed: every edge of $G^{\mathrm{sph}}_{d,\rho}$ has $\langle u,v\rangle \le \rho$, so $\mathrm{Obj}(G^{\mathrm{sph}}_{d,\rho}) \ge \frac{1}{2} - \frac{1}{2}\rho$. The lemma gives the reverse bound $\mathrm{Obj}(G^{\mathrm{sph}}_{d,\rho}) \le \frac{1}{2} - \frac{1}{2}\rho + \frac{1}{2(d-3)|\rho|}$, and it is used below to bound the max-cut. Thus the objective value is $\frac{1}{2} - \frac{1}{2}\rho + O(1/d)$.

The max-cut of the sphere graph. Let $(A, \bar{A})$ be a cut, with fractional surface area $a = \sigma(A)$. The hardest part of the proof is the following isoperimetric theorem. A *cap* is a set $\{u \in \mathbb{S}^{d-1} : \langle u,h\rangle \ge \tau\}$ for a unit vector h and $\tau \in [-1,1]$; a cap with $\tau = 0$ is a *hemisphere*, and $\sigma(\text{cap})$ decreases continuously from 1 to 0 as τ increases from -1 to 1.

Theorem 5.13 (Feige–Schechtman [FS02])**.** *Let $d \ge 2$, $-1 < \rho \le 0$ and $a \in [0,1]$. Among all measurable sets $A \subseteq \mathbb{S}^{d-1}$ with $\sigma(A) = a$, the value $\mu_\rho(A)$ is maximized by a cap of fractional surface area a.*

We state this theorem without proof; see [FS02]. Feige and Schechtman argue further that μ_ρ is maximized at hemispheres. The idea is shown in Fig. 7: if C is a cap inside a hemisphere H, then every point $p \in H \setminus C$ has at least as many edges going out of H as going into C, so adding p to C does not decrease the cut.

Lemma 5.14. *Let $d \geq 2$ and $\rho \in (-1, 0]$, let $H = \{u : \langle u, h\rangle \geq 0\}$ be a hemisphere, and let $C \subseteq H$ be measurable. Then $\mu_\rho(C) \leq \mu_\rho(H)$.*

Proof. For measurable $X, Y \subseteq \mathbb{S}^{d-1}$ let $e(X, Y) = \Pr[u \in X, v \in Y]$ for $(u, v) \sim \mathcal{E}^{\text{sph}}_{d,\rho}$. By symmetry $e(X, Y) = e(Y, X)$, so $\mu_\rho(A) = 2e(A, \bar{A})$. For $p \in \mathbb{S}^{d-1}$, let $N(p) = \{v : \langle v, p\rangle \leq \rho\}$ be its set of neighbors, and let $P = \Pr_{u,v\sim\sigma}[\langle u, v\rangle \leq \rho]$; then $e(X, Y) = \frac{1}{P}\int_X \sigma(N(u) \cap Y)\, d\sigma(u)$.

First we show that every $p \in H$ satisfies $\sigma(N(p) \cap H) \leq \sigma(N(p) \setminus H)$. Let $R(v) = v - 2\langle v, h\rangle h$ be the reflection in the hyperplane bounding H; it is orthogonal, so it preserves σ. If $v \in N(p) \cap H$, then $\langle Rv, h\rangle = -\langle v, h\rangle \leq 0$ and, since $\langle v, h\rangle \geq 0$ and $\langle h, p\rangle \geq 0$,

$$\langle Rv, p\rangle = \langle v, p\rangle - 2\langle v, h\rangle\langle h, p\rangle \leq \langle v, p\rangle \leq \rho.$$

So R maps $N(p) \cap H$ into $(N(p) \setminus H) \cup \{v : \langle v, h\rangle = 0\}$, and the great sphere $\{v : \langle v, h\rangle = 0\}$ has measure 0. This proves the claim.

Now let $D = H \setminus C$, so that $\bar{C}$ is the disjoint union of D and $\bar{H}$. Then

$$\begin{aligned} e(H, \bar{H}) - e(C, \bar{C}) &= \big(e(C, \bar{H}) + e(D, \bar{H})\big) - \big(e(C, D) + e(C, \bar{H})\big) = e(D, \bar{H}) - e(D, C) \\ &\geq e(D, \bar{H}) - e(D, H) = \frac{1}{P}\int_D \big(\sigma(N(u) \setminus H) - \sigma(N(u) \cap H)\big)\, d\sigma(u) \geq 0, \end{aligned}$$

using $e(C, D) = e(D, C)$, then $C \subseteq H$, then the claim for every $u \in D \subseteq H$. □

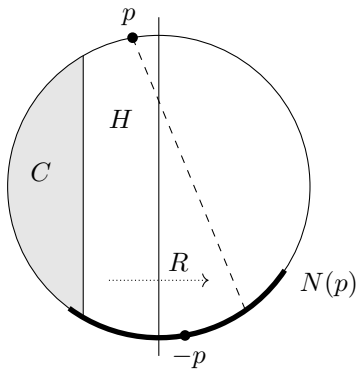


Figure 7: The proof of Lemma 5.14, in a plane through h. The hemisphere H is the left half, bounded by the vertical diameter, and the cap $C \subseteq H$ is shaded. The neighbors of a point $p \in H \setminus C$ form the thick arc $N(p)$ around $-p$; one edge from p is dashed. The reflection R in the boundary of H maps the part of $N(p)$ inside H into the part outside H.

Corollary 5.15. *Let $d \geq 2$ and $\rho \in (-1, 0]$. Every hemisphere H is a maximum cut of $G^{\text{sph}}_{d,\rho}$, all hemispheres have the same value, and*

$$\operatorname{maxcut}(G^{\text{sph}}_{d,\rho}) = \mu_\rho(H) = \mathbb{E}_{(u,v)\sim\mathcal{E}^{\text{sph}}_{d,\rho}}\Big[\frac{\arccos\langle u, v\rangle}{\pi}\Big].$$

If moreover $d \geq 4$ and $\rho < 0$, then

$$\operatorname{maxcut}(G^{\mathrm{sph}}_{d,\rho}) \leq \frac{\arccos \rho}{\pi} + \frac{1 - \arccos(\rho)/\pi}{(1+\rho)(d-3)|\rho|}.$$

Proof. Every orthogonal map Q preserves σ and inner products, so (Qu, Qv) has the same distribution as (u, v), and $\mu_\rho(QA) = \mu_\rho(A)$. Since any two hemispheres are related by an orthogonal map, all hemispheres have the same value. Let A be measurable with $\sigma(A) = a$. Since $\mu_\rho(A) = \mu_\rho(\bar{A})$, we may assume $a \leq \frac{1}{2}$. By Theorem 5.13, $\mu_\rho(A) \leq \mu_\rho(C)$ for a cap $C = \{u : \langle u, h\rangle \geq \tau\}$ with $\sigma(C) = a$. A cap with $\tau < 0$ contains the hemisphere $\{u : \langle u, h\rangle \geq 0\}$ together with a set of positive measure, so it has measure larger than $\frac{1}{2}$; hence $\tau \geq 0$, and C lies in the hemisphere $H = \{u : \langle u, h\rangle \geq 0\}$. By Lemma 5.14, $\mu_\rho(A) \leq \mu_\rho(C) \leq \mu_\rho(H)$.

To compute $\mu_\rho(H)$, let g be a standard Gaussian vector in $\mathbb{R}^d$ and $H_g = \{u : \langle u, g\rangle \geq 0\}$; almost surely $g \neq 0$ and H_g is a hemisphere, so $\mu_\rho(H) = \mathbb{E}_g[\mu_\rho(H_g)]$. Exchanging the order of expectation and using Sheppard's formula (5.4) for the unit vectors u, v,

$$\mu_\rho(H) = \mathbb{E}_{(u,v)} \Pr_g[\mathbf{1}_{H_g}(u) \neq \mathbf{1}_{H_g}(v)] = \mathbb{E}_{(u,v)}\Big[\frac{\arccos\langle u, v\rangle}{\pi}\Big].$$

For the last bound, note that arccos is convex on $[-1, 0]$, since its second derivative $-t(1-t^2)^{-3/2}$ is nonnegative there. So on $[-1, \rho]$ it lies below its chord: $\arccos t \leq \arccos \rho + (\pi - \arccos \rho)(\rho - t)/(1+\rho)$. Taking the expectation over the edges and applying Lemma 5.12 proves the bound. □

The cut output by the Goemans–Williamson rounding of the sphere graph is a hemisphere: with the identity embedding, it is $\{u : \langle g, u\rangle \geq 0\}$ for a standard Gaussian vector g. So the rounding outputs optimal cuts of the sphere graph, and by Corollary 5.15 their value is close to $\arccos(\rho)/\pi$. To obtain a finite graph, the final construction takes as vertices small regions of the sphere.

Proof of Theorem 5.7. Let $\varepsilon > 0$ and $\rho = \rho_*$, and write $c = \frac{1}{2} - \frac{1}{2}\rho_*$, so that $\arccos(\rho_*)/\pi = \gamma(\rho_*)\, c = \alpha_{\mathrm{GW}} c$. Let $K = (1 - \arccos(\rho_*)/\pi)/((1+\rho_*)|\rho_*|)$. Choose $d \geq 4$ and $\delta \in (0, c)$ such that

$$\frac{\alpha_{\mathrm{GW}} c + K/(d-3)}{c - \delta} \leq \alpha_{\mathrm{GW}} + \varepsilon,$$

which is possible because the left side tends to α_{GW} as $d \to \infty$ and $\delta \to 0$.

Partition $\mathbb{S}^{d-1}$ into finitely many nonempty measurable pieces $P_1, \dots, P_N$ of diameter at most δ, for instance the nonempty intersections of the sphere with the cubes $\prod_{i=1}^d [k_i\delta/\sqrt{d}, (k_i+1)\delta/\sqrt{d})$, $k \in \mathbb{Z}^d$. Let G be the weighted graph on $[N]$ with $\mathcal{E}(i, j) = \Pr[u \in P_i, v \in P_j]$ for $(u, v) \sim \mathcal{E}^{\mathrm{sph}}_{d,\rho}$. This distribution is symmetric, and it has no loops, because the endpoints of an edge are at distance $\|u - v\|_2 \geq \sqrt{2} > \delta$.

Every cut of G is a cut of the sphere graph with the same value: for $f\colon [N] \to \{\pm 1\}$, the set $A = \bigcup_{f(i)=1} P_i$ satisfies $\Pr_{(i,j)\sim\mathcal{E}}[f(i) \neq f(j)] = \mu_\rho(A)$. By Corollary 5.15, $\operatorname{maxcut}(G) \leq \alpha_{\mathrm{GW}} c + K/(d-3)$.

For the semidefinite program, pick $z_i \in P_i$ for each i. If $u \in P_i$ and $v \in P_j$, then by the Cauchy–Schwarz inequality

$$|\langle z_i, z_j\rangle - \langle u, v\rangle| \leq |\langle z_i - u, z_j\rangle| + |\langle u, z_j - v\rangle| \leq 2\delta.$$

By Proposition 5.9 and the bound $\mathrm{Obj}(G^{\mathrm{sph}}_{d,\rho}) \geq \frac{1}{2} - \frac{1}{2}\rho$,

$$\mathrm{SDP}(G) \geq \mathbb{E}_{(i,j)\sim\mathcal{E}}\Big[\frac{1}{2} - \frac{1}{2}\langle z_i, z_j\rangle\Big] \geq \mathrm{Obj}(G^{\mathrm{sph}}_{d,\rho}) - \delta \geq c - \delta.$$

Hence $\mathrm{maxcut}(G)/\mathrm{SDP}(G) \leq \alpha_{\mathrm{GW}} + \varepsilon$. Together with the bound $\mathrm{maxcut}(G) \geq \alpha_{\mathrm{GW}}\,\mathrm{SDP}(G)$ for every weighted graph, this shows that the integrality gap over weighted graphs is α_{GW}. □

This yields an integrality gap of $\alpha_{\mathrm{GW}} + \varepsilon \approx 0.878 + \varepsilon$. On the sphere graph, which achieves the integrality gap, the Goemans–Williamson rounding outputs optimal cuts. So the algorithm does as well as it possibly could there, and all of the loss is in the relaxation.

5.4 Rounding gaps and Karloff's theorem

A different question concerns the rounding rather than the relaxation: we look for weighted graphs G on which Algorithm 5.1, applied to an optimal solution X, gives $\mathrm{alg}_{\mathrm{GW}}(G; X) \leq (\alpha_{\mathrm{GW}} + \varepsilon)\,\mathrm{maxcut}(G)$, for arbitrarily small $\varepsilon > 0$. On such a graph the relaxation is nearly exact: since $\alpha_{\mathrm{GW}}\,\mathrm{SDP}(G) \leq \mathrm{alg}_{\mathrm{GW}}(G; X)$, we get $\mathrm{SDP}(G) \leq (1 + \varepsilon/\alpha_{\mathrm{GW}})\,\mathrm{maxcut}(G)$, as in Fig. 6(b).

The value $\mathrm{alg}_{\mathrm{GW}}(G; X)$ is a function of the solution X, and the semidefinite program may have several optimal solutions. In particular, if $\mathrm{SDP}(G) = \mathrm{maxcut}(G)$, then the integral solution $X = xx^{\top}$ of a maximum cut x is optimal, and the rounding of X outputs x or $-x$, both maximum cuts. So the rounding cannot lose the factor α_{GW} when the solution is an integral optimum, and the question is whether it loses it on some optimal solution. We call such a graph, together with such an optimal solution, a *rounding gap* instance.

Theorem 5.16 (Karloff [Kar99])**.** *For every $\varepsilon > 0$ there are a weighted graph G and an optimal solution $u \mapsto z_u$ of its semidefinite program such that*

$$\mathrm{SDP}(G) \leq \mathrm{maxcut}(G) + \varepsilon \qquad \textit{and} \qquad \mathrm{alg}_{\mathrm{GW}}(G; z) \leq (\alpha_{\mathrm{GW}} + \varepsilon)\,\mathrm{maxcut}(G),$$

with $\mathrm{alg}_{\mathrm{GW}}$ *as in Definition 5.6.*

We again define the graph together with an embedding, which we then show to be optimal. The embedding is the hypercube $\{\pm 1/\sqrt{d}\}^d$, whose points are the vectors of $\{\pm 1\}^d$ normalized to length 1. As before, we put edges between pairs whose inner product is close to ρ_*.

Definition 5.17. Let $d \geq 1$ and $\rho \in [-1, 1)$. A *ρ-correlated pair* (x, y) is obtained by choosing $x \in \{\pm 1\}^d$ uniformly at random and then negating each coordinate of x independently with probability $\frac{1}{2} - \frac{1}{2}\rho$ to get y. The weighted graph $G^{\mathrm{cube}}_{d,\rho}$ has vertex set $\{\pm 1\}^d$, edge distribution the law of a ρ-correlated pair, and the embedding $z_x = x/\sqrt{d} \in \mathbb{S}^{d-1}$.

The edge distribution is symmetric: a pair (x, y) at Hamming distance k has probability $2^{-d}p^k(1-p)^{d-k}$ with $p = \frac{1}{2} - \frac{1}{2}\rho$, which is symmetric in x and y. Since $p > 0$, it puts positive mass on pairs with $x \neq y$.

Lemma 5.18. *Let (x, y) be a ρ-correlated pair and $T = \langle z_x, z_y\rangle = \frac{1}{d}\sum_{i=1}^{d} x_i y_i$. Then $\mathbb{E}[T] = \rho$, so $\mathrm{Obj}(G^{\mathrm{cube}}_{d,\rho}) = \frac{1}{2} - \frac{1}{2}\rho$. Moreover, for every $t \geq 0$,*

$$\Pr\Big[|T - \rho| \geq \frac{t}{\sqrt{d}}\Big] \leq 2e^{-t^2/2}.$$

Proof. The product $x_i y_i$ equals -1 if the i-th coordinate was negated and $+1$ otherwise. These events are independent over i, and $\mathbb{E}[x_i y_i] = (\frac{1}{2} + \frac{1}{2}\rho) - (\frac{1}{2} - \frac{1}{2}\rho) = \rho$. By linearity of expectation, $\mathbb{E}[T] = \rho$. The tail bound is the Chernoff bound, in the form of Hoeffding's inequality [Hoe63]: for the average T of d independent random variables with values in $[-1, 1]$ and mean ρ, $\Pr[|T - \rho| \geq s] \leq 2e^{-ds^2/2}$; take $s = t/\sqrt{d}$. □

So most edges have inner product close to ρ, and the rounding behaves as if every edge had inner product exactly ρ.

Lemma 5.19. *Let $\rho \in [-1, 1)$ and $\eta > 0$. For all sufficiently large d,*

$$\mathrm{alg}_{\mathrm{GW}}(G^{\mathrm{cube}}_{d,\rho}; z) \leq \frac{\arccos \rho}{\pi} + \eta.$$

Proof. By (5.4), $\mathrm{alg}_{\mathrm{GW}}(G^{\mathrm{cube}}_{d,\rho}; z) = \mathbb{E}[\arccos(T)]/\pi$ with T as in Lemma 5.18. The function arccos is continuous on the compact interval $[-1, 1]$, hence uniformly continuous, so there is $s > 0$ with $|\arccos t - \arccos \rho| \leq \pi\eta/2$ whenever $t \in [-1, 1]$ and $|t - \rho| \leq s$. Since $0 \leq \arccos \leq \pi$, splitting the expectation according to whether $|T - \rho| \leq s$ and applying Lemma 5.18 with $t = s\sqrt{d}$ gives

$$\frac{\mathbb{E}[\arccos T]}{\pi} \leq \frac{\arccos \rho}{\pi} + \frac{\eta}{2} + \Pr\big[|T - \rho| > s\big] \leq \frac{\arccos \rho}{\pi} + \frac{\eta}{2} + 2e^{-ds^2/2},$$

and the last term is at most $\eta/2$ once $d \geq 2\ln(4/\eta)/s^2$. □

So the rounding outputs a cut of value $\arccos(\rho)/\pi + o(1)$ as $d \to \infty$. There are, however, cuts of value $\frac{1}{2} - \frac{1}{2}\rho$: the *dictator cuts* $D_i = \{x \in \{\pm 1\}^d : x_i = 1\}$, for $i \in [d]$, which in the embedding consist of the points whose i-th coordinate is $+1/\sqrt{d}$. Indeed,

$$\Pr_{(x,y)}\big[\mathbf{1}_{D_i}(x) \neq \mathbf{1}_{D_i}(y)\big] = \Pr[\text{the } i\text{-th coordinate of } x \text{ is negated}] = \frac{1}{2} - \frac{1}{2}\rho.$$

So far we have shown that $\mathrm{Obj}(G^{\mathrm{cube}}_{d,\rho}) = \frac{1}{2} - \frac{1}{2}\rho$, that $\mathrm{alg}_{\mathrm{GW}} \leq \arccos(\rho)/\pi + o(1)$, and that $\mathrm{maxcut}(G^{\mathrm{cube}}_{d,\rho}) \geq \frac{1}{2} - \frac{1}{2}\rho$. It remains to show that the embedding is optimal, that is, that no other solution of the semidefinite program has a larger objective value. This needs a basic fact from Fourier analysis on the hypercube, the subject of Lecture 16. For $f\colon \{\pm 1\}^d \to \mathbb{R}$ and $\rho \in [-1, 1)$, the *noise stability* of f is

$$S_\rho(f) \coloneqq \mathbb{E}_{(x,y)}\big[f(x) f(y)\big],$$

where (x, y) is a ρ-correlated pair.

Lemma 5.20 (Dictator is stablest)**.** *Let $d \geq 1$, $-1 \leq \rho \leq 0$, and $f\colon \{\pm 1\}^d \to \mathbb{R}$. Then*

$$S_\rho(f) \geq \rho \mathbb{E}_x\big[f(x)^2\big].$$

Proof. For $S \subseteq [d]$ let $\chi_S(x) = \prod_{i \in S} x_i$, with $\chi_\emptyset = 1$. Let (x, y) be a ρ'-correlated pair, for any $\rho' \in [-1, 1]$ (for $\rho' = 1$ this means $y = x$). For $S, S' \subseteq [d]$,

$$\chi_S(x)\chi_{S'}(y) = \prod_{i \in S \cap S'} x_i y_i \prod_{i \in S \setminus S'} x_i \prod_{i \in S' \setminus S} y_i.$$

The pairs (x_i, y_i) are independent over i, and $\mathbb{E}[x_i] = \mathbb{E}[y_i] = 0$ (both x and y are uniform) while $\mathbb{E}[x_i y_i] = \rho'$. Hence $\mathbb{E}[\chi_S(x)\chi_{S'}(y)] = \rho'^{|S|}$ if $S = S'$, and 0 otherwise. For $\rho' = 1$ this says that the 2^d functions χ_S are orthonormal for the inner product $\mathbb{E}_x[f(x)g(x)]$; so they form a basis of the 2^d-dimensional space of functions on $\{\pm1\}^d$, and $f = \sum_S \hat{f}(S)\chi_S$ for some coefficients $\hat{f}(S)$. With $\rho' = \rho$ we get

$$S_\rho(f) = \sum_{S,S'} \hat{f}(S)\hat{f}(S')\mathbb{E}[\chi_S(x)\chi_{S'}(y)] = \sum_S \rho^{|S|}\hat{f}(S)^2, \qquad \mathbb{E}_x[f(x)^2] = \sum_S \hat{f}(S)^2.$$

For every S we have $\rho^{|S|} \geq \rho$: for $S = \emptyset$ because $1 \geq \rho$, and for $|S| \geq 1$ because $\rho^{|S|} - \rho = \rho(\rho^{|S|-1} - 1) \geq 0$, as $\rho \leq 0$ and $\rho^{|S|-1} \leq 1$. Hence $S_\rho(f) \geq \rho \sum_S \hat{f}(S)^2 = \rho\mathbb{E}_x[f(x)^2]$. □

Equality holds for the dictator functions $f(x) = x_i$, whose noise stability is $\mathbb{E}[x_i y_i] = \rho$.

Lemma 5.21. *Let $d \geq 1$ and $-1 \leq \rho \leq 0$. Then*

$$\mathrm{SDP}(G^{\mathrm{cube}}_{d,\rho}) = \mathrm{maxcut}(G^{\mathrm{cube}}_{d,\rho}) = \frac{1}{2} - \frac{1}{2}\rho,$$

and the embedding $z_x = x/\sqrt{d}$ is an optimal solution.

Proof. Let $Z\colon \{\pm1\}^d \to \mathbb{S}^{N-1}$ be any solution of the semidefinite program; every feasible X is the Gram matrix of such vectors, with $N = 2^d$. For $j \in [N]$ let $F_j(x)$ be the j-th coordinate of $Z(x)$, so that $\sum_{j=1}^N F_j(x)^2 = 1$ for every x. By Lemma 5.20 applied to each F_j,

$$\mathbb{E}_{(x,y)}\langle Z(x), Z(y)\rangle = \sum_{j=1}^N \mathbb{E}_{(x,y)}\big[F_j(x)F_j(y)\big] \geq \rho \sum_{j=1}^N \mathbb{E}_x\big[F_j(x)^2\big] = \rho\mathbb{E}_x\Big[\sum_{j=1}^N F_j(x)^2\Big] = \rho.$$

So every solution has objective value at most $\frac{1}{2} - \frac{1}{2}\rho$, that is, $\mathrm{SDP}(G^{\mathrm{cube}}_{d,\rho}) \leq \frac{1}{2} - \frac{1}{2}\rho$. The embedding z attains this value (Lemma 5.18), and so does the dictator cut D_1, so $\frac{1}{2} - \frac{1}{2}\rho \leq \mathrm{maxcut}(G^{\mathrm{cube}}_{d,\rho}) \leq \mathrm{SDP}(G^{\mathrm{cube}}_{d,\rho}) \leq \frac{1}{2} - \frac{1}{2}\rho$. □

Proof of Theorem 5.16. Let $\varepsilon > 0$, and let $G = G^{\mathrm{cube}}_{d,\rho_*}$ with the embedding z. Since $\rho_* < 0$, Lemma 5.21 shows that z is an optimal solution and $\mathrm{SDP}(G) = \mathrm{maxcut}(G) = \frac{1}{2} - \frac{1}{2}\rho_*$. By Lemma 5.19 with $\eta = \varepsilon(\frac{1}{2} - \frac{1}{2}\rho_*)$, for all sufficiently large d,

$$\mathrm{alg}_{\mathrm{GW}}(G; z) \leq \frac{\arccos\rho_*}{\pi} + \eta = (\alpha_{\mathrm{GW}} + \varepsilon)\Big(\frac{1}{2} - \frac{1}{2}\rho_*\Big) = (\alpha_{\mathrm{GW}} + \varepsilon)\,\mathrm{maxcut}(G),$$

because $\arccos(\rho_*)/\pi = \gamma(\rho_*)(\frac{1}{2} - \frac{1}{2}\rho_*) = \alpha_{\mathrm{GW}}(\frac{1}{2} - \frac{1}{2}\rho_*)$. □

5.5 Hypercube versus sphere

Remark 5.22. Seen from the rounding, the two constructions behave the same: in both, almost every edge has inner product close to ρ_*, and the random hyperplane cuts about $\arccos(\rho_*)/\pi$ of the edges. They differ in their integral cuts. The hypercube has the special large dictator cuts, of value $\frac{1}{2} - \frac{1}{2}\rho_*$, which make the relaxation exact. The sphere has none: no cut of the sphere graph beats a hemisphere (Corollary 5.15).

Remark 5.23. Stronger relaxations rule out the particular vector solution used above. For ±1 variables, the degree-4 sum-of-squares relaxation (Lecture 21) implies the ℓ_2^2 triangle inequalities $\|v_i - v_j\|^2 \leq \|v_i - v_k\|^2 + \|v_k - v_j\|^2$, and the vectors of the sphere graph violate them: for unit vectors $v_i \perp v_j$ and $v_k = (v_i + v_j)/\sqrt{2}$, the left side is 2 and the right side is $2(2 - \sqrt{2}) < 2$. Whether such stronger relaxations improve on the Goemans–Williamson ratio α_{GW} for Max-Cut is not known.

6 Sparsest Cut: Region Growing and the Leighton–Rao LP

Lecture of September 13, 2021, by Anupam Gupta. Based on the lecturer's handwritten notes.

After set cover and max-coverage (Lectures 1 to 3) and Max-Cut (Lectures 4 and 5), we turn to a third problem, the sparsest cut problem: find a cut of a graph that separates little capacity relative to the sizes of its two sides. It is NP-hard, and this lecture and Lectures 7 and 8 approach it with linear programs, metric embeddings and eigenvalues. Here we introduce the linear programming relaxation of Leighton and Rao [LR99], whose variables form a metric on the vertices, and round it to a cut within a factor $O(\log n)$ of the optimum. The rounding rests on region growing, a technique for cutting a metric into pieces of small diameter while cutting little capacity, which Chapter 8 of [WS11] treats in detail. Finally, constant degree expander graphs show that the integrality gap of this relaxation is $\Omega(\log n)$, so the rounding is optimal up to the constant.

Throughout the lecture, $G = (V, E)$ is an undirected graph with $n = |V| \geq 2$ vertices, and each edge $e = \{i, j\} \in E$ has a capacity $c_e = c_{ij} \geq 0$. For $S \subseteq V$ we write $\bar{S} = V \setminus S$ and $E(S, \bar{S})$ for the set of edges with exactly one endpoint in S, and we set

$$\mathrm{cap}(S, \bar{S}) \coloneqq \sum_{e \in E(S,\bar{S})} c_e.$$

6.1 The sparsest cut problem

A minimum cut of G may separate a single vertex of small degree from the rest of the graph and so tell us little about the structure of G. To find a bottleneck of the graph, we instead normalize the capacity of a cut by the sizes of its sides, as in Fig. 8.

Definition 6.1. Let $\emptyset \neq S \subsetneq V$. The *edge expansion* of the cut $(S, \bar{S})$ is $\mathrm{cap}(S, \bar{S}) / \min\{|S|, |\bar{S}|\}$, and its *sparsity* is

$$\Phi(S) \coloneqq \frac{\mathrm{cap}(S, \bar{S})}{|S| \cdot |\bar{S}|}.$$

The *sparsity* of G is $\Phi(G) \coloneqq \min_{\emptyset \neq S \subsetneq V} \Phi(S)$, and the *(uniform) sparsest cut problem* asks for a cut S with $\Phi(S) = \Phi(G)$.

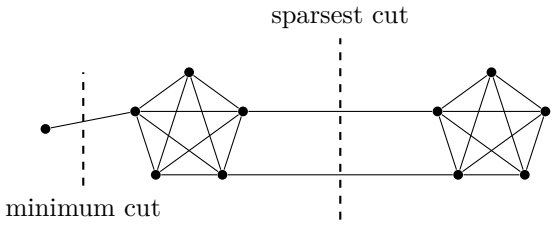


Figure 8: A graph with unit capacities. The minimum cut separates the pendant vertex, and the sparsest cut separates the two dense clusters. Here the minimum cut has one edge and sparsity $1/10$; the sparsest cut has two edges, sides of sizes 6 and 5, and sparsity $2/30$.

The two measures in Definition 6.1 agree up to a factor of two once the edge expansion is scaled by $1/n$, so the sparsity of G captures its edge expansion to within a constant factor.

Proposition 6.2. *For every $\emptyset \neq S \subsetneq V$,*

$$\frac{1}{n} \cdot (\textit{edge expansion of } S) \leq \Phi(S) \leq \frac{2}{n} \cdot (\textit{edge expansion of } S).$$

Proof. By symmetry we may assume $|S| \leq |\bar{S}|$. Then $|S| \cdot |\bar{S}| = |S| \cdot (n - |S|)$, and $n/2 \leq n - |S| \leq n$. Dividing $\operatorname{cap}(S, \bar{S})$ by $|S| \cdot (n - |S|)$ instead of by $|S|$ therefore multiplies it by a factor between $1/n$ and $2/n$. □

Computing $\Phi(G)$ is NP-hard [MS90], so we look for approximation algorithms. Leighton and Rao [LR99] gave an $O(\log n)$-approximation based on a linear program, and showed that the integrality gap of their linear program is $\Omega(\log n)$; both results are proved in this lecture. Arora, Rao and Vazirani [ARV09] improved the approximation ratio to $O(\sqrt{\log n})$ using a semidefinite program (Lectures 10 and 11). Naor and Young [NY18] showed that for the generalization of the problem with demands (Lecture 7), the corresponding semidefinite relaxation has integrality gap $\Omega(\sqrt{\log n})$.

6.2 The Leighton–Rao linear program

As for Max-Cut, we give each pair of vertices a length that should record whether the pair is separated. In the ideal setting $y_{ij} \in \{0, 1\}$, with $y_{ij} = 1$ exactly when i and j lie on different sides of the cut. These lengths satisfy the triangle inequality: if $y_{ij} = 1$, then $y_{ik} = 1$ or $y_{kj} = 1$ for every k. The relaxation keeps the triangle inequality and drops integrality.

Definition 6.3. A *metric* on V is a function $y \colon V \times V \to \mathbb{R}_{\geq 0}$ with $y_{ii} = 0$, $y_{ij} = y_{ji}$ and $y_{ij} \leq y_{ik} + y_{kj}$ for all $i, j, k \in V$. Distinct points may be at distance 0 (such functions are also called semimetrics). We identify a metric with the vector $(y_{ij})_{i<j}$ of its values on unordered pairs. For $S \subseteq V$, the *cut metric* δ_S is the metric with $\delta_S(i, j) = 1$ if exactly one of i, j lies in S, and $\delta_S(i, j) = 0$ otherwise.

For a metric y that is not identically zero, consider the ratio

$$\frac{\sum_{\{i,j\} \in E} c_{ij} y_{ij}}{\sum_{i<j} y_{ij}}. \tag{6.1}$$

For the cut metric $y = \delta_S$ with $\emptyset \neq S \subsetneq V$, the numerator is $\operatorname{cap}(S, \bar{S})$ and the denominator is $|S| \cdot |\bar{S}|$, so (6.1) equals $\Phi(S)$.

Lemma 6.4. *The minimum of (6.1) over all metrics y with values in $\{0, 1\}$ that are not identically zero equals $\Phi(G)$.*

Proof. The cut metrics show that the minimum is at most $\Phi(G)$. Conversely, let y be a $\{0, 1\}$-valued metric that is not identically zero. Relate i to j when $y_{ij} = 0$. This is an equivalence relation on V: it is reflexive and symmetric, and it is transitive because $y_{ij} \leq y_{ik} + y_{kj} = 0$ whenever $y_{ik} = y_{kj} = 0$. Let $P_1, \ldots, P_k$ be its classes; then $k \geq 2$, and $y_{ij} = 1$ exactly when i and j lie in different classes. A pair in different classes P_a and P_b is separated by exactly two of the cuts $(P_1, \bar{P}_1), \ldots, (P_k, \bar{P}_k)$, namely by the a-th and the b-th, and a pair inside one class is separated by none of them. Hence $y = \frac{1}{2} \sum_{a=1}^{k} \delta_{P_a}$, and

$$\frac{\sum_{\{i,j\} \in E} c_{ij} y_{ij}}{\sum_{i<j} y_{ij}} = \frac{\sum_{a=1}^{k} \operatorname{cap}(P_a, \bar{P}_a)}{\sum_{a=1}^{k} |P_a| \cdot |\bar{P}_a|} \geq \min_{1 \leq a \leq k} \Phi(P_a) \geq \Phi(G),$$

because a ratio of two sums of nonnegative terms, with positive denominators, is at least the smallest of the termwise ratios. □

Minimizing (6.1) over all metrics is therefore a relaxation of the sparsest cut problem. Its objective is a ratio, but such objectives are easy to handle over cones. A set $K \subseteq \mathbb{R}^N$ is a *cone* if $\lambda x \in K$ for all $x \in K$ and $\lambda \geq 0$. The metrics on V form a cone.

Lemma 6.5. *Let $K \subseteq \mathbb{R}^N$ be a cone and $a, b \in \mathbb{R}^N$. Then*

$$\inf\Big\{\frac{a^\top x}{b^\top x} : x \in K,\ b^\top x > 0\Big\} = \inf\{a^\top x : x \in K,\ b^\top x = 1\},$$

and $x \mapsto x/b^\top x$ maps feasible points of the left side to feasible points of the right side with the same objective value.

Proof. If $x \in K$ and $b^\top x > 0$, then $x' = x/b^\top x$ lies in K, satisfies $b^\top x' = 1$, and has $a^\top x' = a^\top x / b^\top x$. Conversely, every $x \in K$ with $b^\top x = 1$ is feasible on the left with the same value. □

In our setting $b = (1, \dots, 1) \geq 0$ and metrics are nonnegative, so the only metric excluded on the left is $y = 0$. By Lemma 6.5, the relaxation is the linear program

$$\begin{aligned} \mathrm{LP}(G) = \min \quad & \sum_{\{i,j\} \in E} c_{ij} y_{ij} \\ \text{s.t.} \quad & \sum_{i<j} y_{ij} = 1, \\ & y_{ij} \leq y_{ik} + y_{kj} \quad \text{for all distinct } i, j, k \in V, \\ & y_{ij} \geq 0 \quad \text{for all } i < j, \end{aligned} \tag{6.2}$$

called the *metric relaxation* or the *Leighton–Rao linear program*. It has $\binom{n}{2}$ variables and $O(n^3)$ constraints, so it can be solved in polynomial time, and by Lemmas 6.4 and 6.5 we have $\mathrm{LP}(G) \leq \Phi(G)$. For a feasible solution y of (6.2), the average of y_{ij} over the $\binom{n}{2}$ pairs is $1/\binom{n}{2}$.

The two main results of the lecture show that this relaxation is off by a factor $\Theta(\log n)$ in the worst case.

Theorem 6.6 (Leighton and Rao [LR99])**.** *Given a feasible solution y of (6.2) with value $W = \sum_{\{i,j\} \in E} c_{ij} y_{ij}$, Algorithm 6.8 below finds, in polynomial time, a cut $\emptyset \neq S \subsetneq V$ with*

$$\Phi(S) \leq 72 \lceil \log_2(n+1) \rceil \cdot W = O(\log n) \cdot W.$$

Applied to an optimal y, it is an $O(\log n)$-approximation algorithm for the sparsest cut problem.

Theorem 6.7 (Leighton and Rao [LR99])**.** *There are an absolute constant $c > 0$ and, for infinitely many n, graphs G on n vertices with unit capacities such that $\Phi(G) \geq c \log n \cdot \mathrm{LP}(G)$.*

6.3 The rounding algorithm

Fix a feasible solution y of (6.2) with value W. The algorithm first cuts V into pieces of small diameter, chosen smaller than the average distance $1/\binom{n}{2}$. If all pieces are small, they can be grouped into a balanced cut. If one piece is large, then many vertices are close together, and a cut by distance from that piece is good.

Algorithm 6.8 (Rounding the metric relaxation)**.** Let $R = 1/(4n^2)$.

1) Compute the low-diameter decomposition of Lemma 6.10 with radius R: a partition of V into pieces of diameter at most $2R$, such that the edges between different pieces have total capacity at most $4kW/R$, where $k = \lceil \log_2(n+1) \rceil$.

2) If every piece has at most $n/3$ vertices, let $L = \emptyset$ and add whole pieces to L, one at a time, until $|L| \geq n/3$. Output L.

3) Otherwise, let D be a piece with more than $n/3$ vertices, the *dense* piece. For $v \in V$ let $d_v = \min_{u \in D} y_{uv}$ be the distance from v to D, order the vertices as $v_1, \ldots, v_n$ with $d_{v_1} \leq \cdots \leq d_{v_n}$, and let $S_i = \{v_1, \ldots, v_i\}$. Output the set S_i, $1 \leq i \leq n-1$, of smallest sparsity.

Two facts remain to be shown: that every metric admits the decomposition of step 1 (Lemma 6.10), and that the dense case of step 3 is easy (Lemma 6.12). We first derive the theorem from them.

Proof of Theorem 6.6. Let $k = \lceil \log_2(n+1) \rceil$.

Case 1: every piece has at most $n/3$ vertices. Before the last piece was added, L had fewer than $n/3$ vertices, and the last piece has at most $n/3$, so $n/3 \leq |L| < 2n/3$. The function $t \mapsto t(n-t)$ is concave, so on $[n/3, 2n/3]$ it is smallest at the endpoints, and $|L| \cdot |\bar{L}| \geq (n/3)(2n/3) = 2n^2/9$. Every edge of $E(L, \bar{L})$ joins two different pieces, so $\operatorname{cap}(L, \bar{L}) \leq 4kW/R = 16kn^2W$. Hence

$$\Phi(L) \leq \frac{16kn^2W}{2n^2/9} = 72kW.$$

Case 2: some piece D has more than $n/3$ vertices. The piece has diameter at most $2R$, so Lemma 6.12 applies and the output has sparsity at most $12W \leq 72kW$.

Both steps run in polynomial time by Lemma 6.10 and because step 3 compares $n-1$ cuts. □

6.4 Low-diameter decomposition by region growing

There are many ways to cut a metric into pieces of small diameter. We use a region growing argument. Throughout this subsection, y is any metric on V (not necessarily normalized), $W = \sum_{\{i,j\} \in E} c_{ij} y_{ij}$, and $R > 0$.

Think of each edge $\{i, j\}$ as a pipe with cross-section c_{ij} and length y_{ij}, so that its volume is $c_{ij} y_{ij}$ and the total volume of the graph is W. We grow a ball around a vertex v and stop at a radius where the capacity crossing the boundary of the ball is small compared to the volume inside the ball. The cut capacity is then charged to the volume removed with the ball.

The algorithm repeatedly removes a ball from the set $U \subseteq V$ of vertices that are not yet in a piece. Let $E(U)$ be the set of edges with both endpoints in U. For $v \in U$ and $r \geq 0$, let $B_U(v, r) = \{u \in U : y_{vu} \leq r\}$, let $\operatorname{cap}_U(v, r)$ be the total capacity of the edges of $E(U)$ with exactly one endpoint in $B_U(v, r)$, and define the *volume*

$$\operatorname{vol}_U(v, r) := \frac{W}{n} + \sum_{\substack{\{i,j\} \in E(U) \\ i,j \in B_U(v,r)}} c_{ij} y_{ij} + \sum_{\substack{\{i,j\} \in E(U) \\ i \in B_U(v,r),\ j \notin B_U(v,r)}} c_{ij}(r - y_{vi}).$$

The first term is an imaginary volume placed at v, which makes the volume positive from the start. The second counts the edges inside the ball, and the third counts the part of each boundary edge that lies within distance r of v. Figure 9 shows the layers used below.

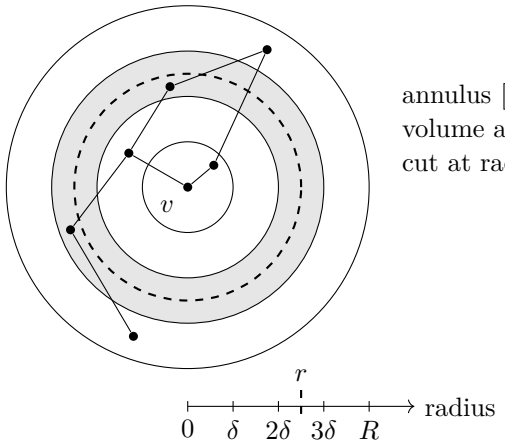


Figure 9: Region growing around v with $k = 4$ layers of width $\delta = R/k$. If the volume within radius 3δ is at most twice the volume within radius 2δ, the algorithm cuts at the cheapest radius $r \in [2\delta, 3\delta]$.

Lemma 6.9. *Fix $U \subseteq V$ and $v \in U$. The function $r \mapsto \mathrm{vol}_U(v, r)$ is nondecreasing, and for all $0 \le a < b$,*

$$\int_a^b \mathrm{cap}_U(v, r)\, dr \le \mathrm{vol}_U(v, b) - \mathrm{vol}_U(v, a).$$

Proof. Consider one edge $e = \{i, j\} \in E(U)$, with $y_{vi} \le y_{vj}$. Its contribution to $\mathrm{vol}_U(v, r)$ is 0 for $r < y_{vi}$, it is $c_e(r - y_{vi})$ for $y_{vi} \le r < y_{vj}$, and it is $c_e y_{ij}$ for $r \ge y_{vj}$. By the triangle inequality, $y_{vj} - y_{vi} \le y_{ij}$, so at $r = y_{vj}$ the contribution jumps up from $c_e(y_{vj} - y_{vi})$ to $c_e y_{ij}$, or stays the same; with a general metric it may jump, because the edge need not lie on a shortest path from v. Hence each contribution is nondecreasing in r, and so is the volume. The edge e is counted in $\mathrm{cap}_U(v, r)$ exactly when $y_{vi} \le r < y_{vj}$, and on that interval its contribution grows at rate c_e. Therefore the increase of its contribution between a and b is at least c_e times the length of $[a, b] \cap [y_{vi}, y_{vj})$, which is the integral of its term in $\mathrm{cap}_U(v, r)$ over $[a, b]$. Summing over the edges proves the inequality. □

Lemma 6.10 (Low-diameter decomposition)**.** *Let $k = \lceil \log_2(n + 1) \rceil$. There is a polynomial-time algorithm that partitions V into pieces $P_1, \ldots, P_s$ with centers $v_t \in P_t$ such that $y_{v_t u} \le R$ for all $u \in P_t$, so that all distances inside a piece are at most $2R$, and such that the edges between different pieces have total capacity at most $4kW/R$.*

Proof. Let $\delta = R/k$. Start with $U = V$. While $U \ne \emptyset$, pick any $v \in U$ and write $V_i = \mathrm{vol}_U(v, i\delta)$ for $0 \le i \le k$.

There is a layer $i \in \{0, \ldots, k-1\}$ with $V_{i+1} \le 2V_i$. Otherwise the volume more than doubles k times, so $V_k > 2^k V_0 \ge 2^k W/n$. Since $2^k \ge n + 1$, this gives $V_k > W + W/n$. But V_k is at most the imaginary volume W/n plus the total volume $\sum_{e \in E(U)} c_e y_e \le W$, a contradiction.

Fix such an i, let $a = i\delta$ and $b = (i + 1)\delta$, and choose $r \in [a, b]$ that minimizes $\mathrm{cap}_U(v, r)$. The function $r \mapsto \mathrm{cap}_U(v, r)$ changes only at the values y_{vu}, $u \in U$, so the minimum is attained at a or at one of these values in $(a, b]$. The cheapest level cut costs no more than a level cut at a uniformly random radius in $[a, b]$, so by Lemma 6.9

$$\mathrm{cap}_U(v, r) \le \frac{1}{\delta} \int_a^b \mathrm{cap}_U(v, \rho)\, d\rho \le \frac{V_{i+1} - V_i}{R/k} \le \frac{V_{i+1}}{R}\, k \le \frac{2V_i}{R}\, k \le \frac{2k}{R}\, \mathrm{vol}_U(v, r),$$

where the last step uses $r \geq a$ and the monotonicity of the volume. Make $P = B_U(v, r)$ a piece with center v, and remove it from U. Every $u \in P$ has $y_{vu} \leq r \leq R$.

Charging. When P is removed, we pay $\mathrm{cap}_U(v, r)$, and we charge this cost to the volume $\mathrm{vol}_U(v, r)$ cut away with P, at a rate of at most $2k/R$ per unit of volume. Apart from the imaginary W/n, the volume $\mathrm{vol}_U(v, r)$ consists of the full or partial volumes of edges of $E(U)$ with at least one endpoint in P, and these edges are not in $E(U \setminus P)$. So the real volumes charged at different steps come from disjoint sets of edges, and their total is at most W. Each step removes at least its center, so there are at most n steps, and the imaginary volumes total at most $n \cdot W/n = W$.

An edge between two different pieces has both endpoints in U at the step that removes the first of them, and it is counted in $\mathrm{cap}_U(v, r)$ at that step. Hence the edges between pieces have total capacity at most

$$\frac{2k}{R} \cdot (\text{total volume cut away}) \leq \frac{2k}{R}\Big(W + n \cdot \frac{W}{n}\Big) = \frac{4kW}{R}. \qquad \square$$

Remark 6.11. Randomized versions of this decomposition are due to Bartal [Bar96], Călinescu, Karloff and Rabani [CKR05], and Miller, Peng and Xu [MPX13]. The deterministic version above is *local*: it cuts, and charges the cut to the volume removed. This technique, called *region growing*, is used widely in approximation algorithms and in distributed computing, where low-diameter decompositions go back to Awerbuch and Peleg [AP90] and to Linial and Saks [LS93].

6.5 The dense case

It remains to handle step 3 of Algorithm 6.8, in which many vertices are close together.

Lemma 6.12. *Let y be a feasible solution of (6.2) with value W, let $R = 1/(4n^2)$, and let $D \subseteq V$ satisfy $|D| \geq n/3$ and $y_{uw} \leq 2R$ for all $u, w \in D$. With d_v and $S_1, \dots, S_{n-1}$ as in step 3 of Algorithm 6.8,*

$$\mathrm{ALG} := \min_{1 \leq i \leq n-1} \Phi(S_i) \leq 12W.$$

Proof. The vertices are far from D on average. For $i \in V$ let u_i be a vertex of D with $y_{iu_i} = d_i$. By the triangle inequality and the small diameter of D, $y_{ij} \leq y_{iu_i} + y_{u_iu_j} + y_{u_jj} \leq d_i + d_j + 2R$ for all $i, j \in V$. Hence

$$1 = \sum_{i<j} y_{ij} \leq \sum_{i,j \in V} (d_i + d_j + 2R) = 2n \sum_{i \in V} d_i + 2n^2 R = 2n \sum_{i \in V} d_i + \frac{1}{2},$$

by the choice of R, and therefore $\sum_{i \in V} d_i \geq 1/(4n)$.

Level cuts. For $\alpha \geq 0$, let $B_\alpha = \{v \in V : d_v \leq \alpha\}$ be the ball of radius α around D (see Fig. 10). It contains D, and it is one of the sets S_i whenever $B_\alpha \neq V$, whatever the order of vertices with equal distances. So for every $\alpha \geq 0$,

$$\mathrm{cap}(B_\alpha, \bar{B}_\alpha) \geq \mathrm{ALG} \cdot |B_\alpha| \cdot |\bar{B}_\alpha| \geq \mathrm{ALG} \cdot \frac{n}{3} \cdot |\bar{B}_\alpha|,$$

where both sides are 0 if $B_\alpha = V$. An edge $\{i, j\}$ is cut by B_α exactly for α between d_i and d_j, and $|d_i - d_j| \leq y_{ij}$ by the triangle inequality. Integrating over α gives

$$W = \sum_{\{i,j\} \in E} c_{ij} y_{ij} \geq \sum_{\{i,j\} \in E} c_{ij} |d_i - d_j| = \int_0^\infty \mathrm{cap}(B_\alpha, \bar{B}_\alpha)\, d\alpha \geq \mathrm{ALG} \cdot \frac{n}{3} \int_0^\infty |\bar{B}_\alpha|\, d\alpha.$$

Finally, $\int_0^\infty |\bar{B}_\alpha|\, d\alpha = \sum_{v \in V} d_v \geq 1/(4n)$, because $v \in \bar{B}_\alpha$ exactly for $\alpha < d_v$. Hence $W \geq \mathrm{ALG} \cdot \frac{n}{3} \cdot \frac{1}{4n} = \mathrm{ALG}/12$. □

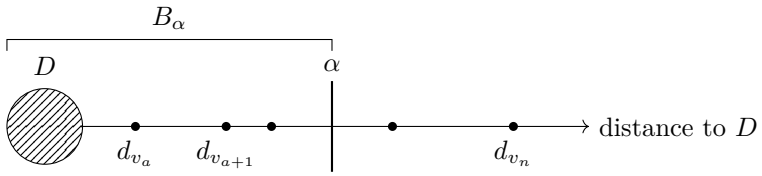


Figure 10: The dense case: the vertices of D are at distance 0 from D ($a = |D| + 1$), and the others are laid out by their distance to D. The ball B_α is a prefix of this order.

6.6 The integrality gap: expander graphs

The rounding of Theorem 6.6 loses a factor $O(\log n)$, and this is unavoidable for the metric relaxation. The bad instances come from the other important class of counterexamples in this course, constant degree expander graphs. An expander is a graph in which every set that is not too large has many edges leaving it.

Definition 6.13. Let $d \geq 1$ and $\alpha > 0$. A d-regular graph $G = (V, E)$ is an *α-edge expander* if $|E(S, \bar{S})| \geq \alpha \cdot d \cdot |S|$ for every $S \subseteq V$ with $|S| \leq |V|/2$.

Statements about a single graph are not meaningful here, so we consider infinite families of graphs. When a family is called a family of expanders without naming α, it is meant that α is a positive absolute constant.

Fact 6.14. *There are $\alpha > 0$ and an infinite family of 3-regular α-edge expanders; see [HLW06].*

For example, the union of a constant number of random perfect matchings on n vertices is an expander with high probability. Deterministic constructions are also known: the graph on $\mathbb{Z}_p$, for a prime p, in which each x is joined to $x+1$, $x-1$ and x^{-1} (with $0^{-1} = 0$); graphs defined by groups, such as the graphs on $\mathbb{Z}_m \times \mathbb{Z}_m$ of Margulis and of Gabber and Galil [GG81]; and the zig-zag product of Reingold, Vadhan and Wigderson [RVW02].

Proof of Theorem 6.7. Let G be a 3-regular α-edge expander on $n \geq 36$ vertices, as in Fact 6.14, with unit capacities. It is connected, because every nonempty S with $|S| \leq n/2$ has an edge leaving it.

All cuts are expensive. Let $\emptyset \neq S \subsetneq V$, and by symmetry let $|S| \leq n/2$. Then $\Phi(S) = |E(S, \bar{S})|/(|S| \cdot |\bar{S}|) \geq 3\alpha|S|/(|S| \cdot n) = 3\alpha/n$, so $\Phi(G) \geq 3\alpha/n$, in accordance with Proposition 6.2.

Most vertices are far apart. Let $h(i, j)$ be the hop distance (number of edges on a shortest path) between i and j. A vertex has at most $3 \cdot 2^{t-1} \leq 3^t$ vertices at hop distance exactly $t \geq 1$, because the first step has 3 choices and every later step at most 2 new ones. Hence at most $1 + \sum_{t=1}^{s} 3 \cdot 2^{t-1} = 3 \cdot 2^s - 2$ vertices lie within hop distance s. For $s = \lfloor \log_2(n/6) \rfloor$ this is less than $n/2$, so more than $n/2$ vertices are at hop distance at least $s + 1 > \log_2(n/6)$ from any given vertex. Consequently

$$H := \sum_{i<j} h(i, j) = \frac{1}{2} \sum_{i \in V} \sum_{j \in V} h(i, j) \geq \frac{1}{2} \cdot n \cdot \frac{n}{2} \log_2 \frac{n}{6} = \frac{n^2}{4} \log_2 \frac{n}{6}.$$

A cheap LP solution. Give every edge the length $1/H$ and let $y_{ij} = h(i,j)/H$ be the resulting shortest-path distance. This is a metric with $\sum_{i<j} y_{ij} = 1$, so it is feasible for (6.2), and since $|E| = 3n/2$,

$$\mathrm{LP}(G) \le \sum_{\{i,j\}\in E} y_{ij} = \frac{3n/2}{H} \le \frac{6}{n\log_2(n/6)} = O\Big(\frac{1}{n\log n}\Big).$$

Hence $\Phi(G)/\mathrm{LP}(G) \ge \frac{\alpha}{2}\log_2(n/6) \ge \frac{\alpha}{4}\log_2 n$, since $\log_2 6 \le \frac{1}{2}\log_2 n$ for $n \ge 36$. So the theorem holds with $c = \alpha/4$, for logarithms to base 2. □

In this instance the linear program pretends that the sparsity is about $1/(n\log n)$, while every cut has sparsity of order $1/n$: on an expander, a typical pair of vertices is at distance $\Omega(\log n)$, and the metric relaxation spreads its unit of total length so thinly over the edges that each edge has length only $O(1/(n^2\log n))$. Lecture 7 connects the relaxation to embeddings of metrics into ℓ_1, and Lecture 8 turns to eigenvalues and eigenvectors.

7 Sparsest Cut: Metric Embeddings

Lecture of September 15, 2021, by Anupam Gupta. Based on the lecturer's handwritten notes.

This lecture generalizes the sparsest cut problem of Lecture 6 to arbitrary demands between pairs of vertices, and gives an $O(\log k)$-approximation for it, where k is the number of pairs with nonzero demand. For uniform demands this is another proof of the $O(\log n)$ guarantee of Lecture 6. The approach goes through the geometry of metrics: the cuts of a graph generate exactly the metrics that embed isometrically into ℓ_1, so the metric relaxation is good when its solution is close to an ℓ_1 metric. Bourgain's theorem says that every n-point metric embeds into ℓ_1 with distortion $O(\log n)$, and Linial, London and Rabinovich [LLR95] used it to round the relaxation. We prove Bourgain's theorem with a random construction close to his original one [Bou85]; see Chapter 4 of Matoušek's lecture notes [Mat13] and Chapter 15 of [WS11] for more on metric embeddings and cut problems.

7.1 Sparsest cut with general demands

Throughout the lecture, $G = (V, E)$ is an undirected graph on $n \geq 2$ vertices with capacities $c_e \geq 0$, and $\text{cap}(S, \bar{S})$ is the total capacity of the edges with exactly one endpoint in S, as in Lecture 6. In Lecture 6 we minimized the sparsity $\text{cap}(S, \bar{S})/(|S| \cdot |\bar{S}|)$, which counts one unit of demand between every pair of vertices. We now allow arbitrary demands.

Definition 7.1. Let $D_{ij} = D_{ji} \geq 0$ be a *demand* for each pair of distinct vertices i, j, not all zero. For $S \subseteq V$, let $\text{dem}(S, \bar{S}) = \sum_{i \in S,\, j \in \bar{S}} D_{ij}$. The *generalized sparsity* of a cut with $\text{dem}(S, \bar{S}) > 0$ is

$$\Phi(S) := \frac{\text{cap}(S, \bar{S})}{\text{dem}(S, \bar{S})} = \frac{\sum_{e \in E(S, \bar{S})} c_e}{\sum_{i \in S,\, j \in \bar{S}} D_{ij}},$$

the ratio of the capacity of the edges cut to the demand separated. The *generalized sparsest cut problem* asks for the minimum Φ^* of $\Phi(S)$ over all S with $\text{dem}(S, \bar{S}) > 0$. We write k for the number of pairs $\{i, j\}$ with $D_{ij} > 0$.

With $D_{ij} = 1$ for all pairs we recover the uniform sparsest cut problem of Lecture 6. The other extreme is a nonzero demand $D_{st} = 1$ between a single pair s, t, with all other demands 0. Then $\Phi(S)$ is defined only for cuts separating s from t, where it equals $\text{cap}(S, \bar{S})$, so the problem is the minimum s-t cut problem, which is solvable in polynomial time by maximum flow algorithms [FF56, EK72]. With nonzero demands between two pairs s_1, t_1 and s_2, t_2, the problem is still solvable in polynomial time [Hu63, RW66, Sey80]. For more general demand patterns the problem is NP-hard.

Homework 7.2. Show that the generalized sparsest cut problem is NP-hard, by a reduction from Max-Cut.

The main result of this lecture is an $O(\log k)$-approximation algorithm for the generalized sparsest cut problem (Corollary 7.9 and Remark 7.15). For uniform demands $k = \binom{n}{2}$ and $\log k = O(\log n)$, which gives another proof of the bound of Theorem 6.6, up to the constant and with a randomized algorithm.

7.2 The metric relaxation and cut metrics

We use metrics and cut metrics δ_S as in Definition 6.3, and we think of c, D and a metric y as vectors in $\mathbb{R}^{\binom{n}{2}}$, indexed by unordered pairs, with $c_{ij} = 0$ when $\{i,j\} \notin E$. The *metric relaxation* of the generalized sparsest cut problem is

$$\begin{aligned} \mathrm{LP} = \min \quad & \sum_{\{i,j\}\in E} c_{ij} y_{ij} \\ \text{s.t.} \quad & \sum_{i<j} D_{ij} y_{ij} = 1, \\ & y \text{ is a metric: } y_{ij} \geq 0 \text{ and } y_{ij} \leq y_{ik} + y_{kj} \text{ for all } i,j,k. \end{aligned} \tag{7.1}$$

The ideal formulation would require y to be the indicator of a cut, that is, $y = \delta_S$ for some $S \subseteq V$. For a cut metric, $c^\top \delta_S = \mathrm{cap}(S, \bar{S})$ and $D^\top \delta_S = \mathrm{dem}(S, \bar{S})$, so

$$\Phi^* = \min_{\substack{y \text{ a cut metric} \\ D^\top y > 0}} \frac{c^\top y}{D^\top y}, \qquad \mathrm{LP} = \min_{\substack{y \text{ a metric} \\ D^\top y > 0}} \frac{c^\top y}{D^\top y},$$

where the second equality is Lemma 6.5. The relaxation simply drops the requirement that the metric be a cut metric, and $\mathrm{LP} \leq \Phi^*$. To compare the two minima, we first replace the discrete set of cut metrics by a convex set.

Definition 7.3. The *cut cone* on $V = \{1, \dots, n\}$ is

$$K_n \coloneqq \Big\{ \sum_{S \subseteq V} \alpha_S \delta_S : \alpha_S \geq 0 \text{ for all } S \Big\}.$$

The *metric cone* M_n is the set of all metrics on V.

The metric cone is closed under nonnegative scaling and under addition, and every cut metric is a metric. Since K_n is generated by cut metrics, it follows that $K_n \subseteq M_n$.

7.3 The cut cone is the set of ℓ_1 metrics

A metric d on V is *ℓ_1-embeddable* if there are an integer $m \geq 1$ and a map $\varphi \colon V \to \mathbb{R}^m$ with $d(i,j) = \|\varphi(i) - \varphi(j)\|_1 = \sum_{t=1}^m |\varphi_t(i) - \varphi_t(j)|$ for all $i, j \in V$. Here ℓ_1 is the Manhattan metric.

Theorem 7.4. *A metric d on V lies in the cut cone K_n if and only if it is ℓ_1-embeddable.*

Proof. Cut cone to ℓ_1. Let $d = \sum_S \alpha_S \delta_S$ with $\alpha_S \geq 0$, and let $S_1, \dots, S_m$ be the sets with $\alpha_S > 0$ (if there are none, then $d = 0$ and a constant map to $\mathbb{R}$ works). Use one coordinate for each of them, and map

$$\varphi(i) = (\alpha_{S_1} \mathbf{1}(i \in S_1), \dots, \alpha_{S_m} \mathbf{1}(i \in S_m)).$$

Then

$$\|\varphi(i) - \varphi(j)\|_1 = \sum_{t=1}^m \alpha_{S_t} |\mathbf{1}(i \in S_t) - \mathbf{1}(j \in S_t)| = \sum_{t=1}^m \alpha_{S_t} \delta_{S_t}(i,j) = d(i,j).$$

ℓ_1 to cut cone. Since $\|\varphi(i) - \varphi(j)\|_1$ is the sum over the coordinates of $|\varphi_t(i) - \varphi_t(j)|$, and K_n is closed under addition, it suffices to treat one coordinate, that is, a metric $d(i,j) = |x_i - x_j|$ given

by values $x_i \in \mathbb{R}$. Order the vertices as $u_1, \dots, u_n$ with $x_{u_1} \le \cdots \le x_{u_n}$, and let $S_t = \{u_1, \dots, u_t\}$. For $a < b$, the pair u_a, u_b is separated by S_t exactly when $a \le t < b$, so

$$\sum_{t=1}^{n-1} (x_{u_{t+1}} - x_{u_t})\, \delta_{S_t}(u_a, u_b) = \sum_{t=a}^{b-1} (x_{u_{t+1}} - x_{u_t}) = x_{u_b} - x_{u_a} = d(u_a, u_b).$$

This writes d as a nonnegative combination of the cut metrics δ_{S_t}. □

Example 7.5. Take four points i, j, k, l on the line, with values $x_i = 0.3$, $x_j = x_k = 0.7$ and $x_l = 1.4$, so that, for example, $d(i,j) = |0.3 - 0.7| = 0.4$ and $d(l,j) = 0.7$. With $S_1 = \{i\}$ and $S_2 = \{i, j, k\}$ (Fig. 11),

$$d = (0.7 - 0.3)\, \delta_{S_1} + (1.4 - 0.7)\, \delta_{S_2};$$

the cut $\{i, j\}$ between j and k gets coefficient $0.7 - 0.7 = 0$. For instance, $d(i,l) = 0.4 + 0.7 = 1.1$, since both cuts separate i from l.

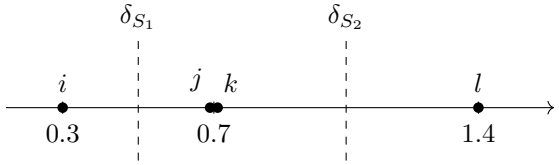


Figure 11: The metric of Example 7.5: points i, j, k, l at 0.3, 0.7, 0.7, 1.4, and the two cuts $S_1 = \{i\}$ (weight 0.4) and $S_2 = \{i, j, k\}$ (weight 0.7).

Remark 7.6. The proof is algorithmic. Given an embedding $\varphi\colon V \to \mathbb{R}^m$, sorting each coordinate writes $\|\varphi(i) - \varphi(j)\|_1 = \sum_S \alpha_S \delta_S(i,j)$ with at most $m(n-1)$ cuts, in $O(mn \log n)$ time.

Corollary 7.7. *The optimum of the generalized sparsest cut problem is*

$$\Phi^* = \min\Big\{ \frac{c^\top d}{D^\top d} : d \text{ is } \ell_1\text{-embeddable},\ D^\top d > 0 \Big\}.$$

Moreover, if $d = \sum_S \alpha_S \delta_S$ with $\alpha_S \ge 0$ and $D^\top d > 0$, then some S with $\alpha_S > 0$ and $D^\top \delta_S > 0$ has $\Phi(S) \le c^\top d / D^\top d$.

Proof. Cut metrics are ℓ_1-embeddable, so the minimum is at most Φ^*. Conversely, let d be ℓ_1-embeddable with $D^\top d > 0$. By Theorem 7.4, $d = \sum_S \alpha_S \delta_S$ with $\alpha_S \ge 0$. By linearity,

$$\frac{c^\top d}{D^\top d} = \frac{\sum_S \alpha_S\, c^\top \delta_S}{\sum_S \alpha_S\, D^\top \delta_S} \ge \min_{S:\ \alpha_S > 0,\ D^\top \delta_S > 0} \frac{c^\top \delta_S}{D^\top \delta_S},$$

because the terms with $D^\top \delta_S = 0$ only add nonnegative amounts to the numerator, and a ratio of sums of nonnegative terms is at least the smallest termwise ratio among the terms with positive denominators. The right side is $\Phi(S)$ for the minimizing S, which proves both claims. □

So the sparsest cut is a minimum over ℓ_1-embeddable metrics, and the metric relaxation is the same minimum over all metrics. The integrality gap measures how far metrics are from ℓ_1.

7.4 Bourgain's theorem and the rounding

A map $\varphi\colon V \to \mathbb{R}^m$ is an embedding of a metric y into ℓ_1 with *distortion* $\beta \geq 1$ if $y_{ij}/\beta \leq \|\varphi(i) - \varphi(j)\|_1 \leq y_{ij}$ for all $i, j \in V$. Bourgain's theorem says that every finite metric is close to an ℓ_1 metric.

Theorem 7.8 (Bourgain [Bou85])**.** *There is an absolute constant C such that every metric y on $n \geq 2$ points embeds into ℓ_1 with distortion at most $C \log n$, in dimension $m = O(\log^2 n)$. Such an embedding can be computed by a randomized polynomial-time algorithm [LLR95].*

We prove the theorem in Section 7.5, with $C \log n = 88\lfloor \log_2 n \rfloor$. First we use it to round the metric relaxation.

Corollary 7.9 (Linial, London and Rabinovich [LLR95])**.** *There is a randomized polynomial-time $O(\log n)$-approximation algorithm for the generalized sparsest cut problem.*

Proof. Solve (7.1) and let y^* be an optimal solution, so $\mathrm{LP} = c^\top y^* / D^\top y^*$ with $D^\top y^* = 1$. Let φ be the embedding of y^* given by Theorem 7.8, with distortion $\beta = O(\log n)$, and let d be the ℓ_1 metric $d(i,j) = \|\varphi(i) - \varphi(j)\|_1$. Since $d \leq y^*$ and $d \geq y^*/\beta$ entrywise, and $c, D \geq 0$, we have $c^\top d \leq c^\top y^*$ and $D^\top d \geq D^\top y^*/\beta > 0$. Hence

$$\mathrm{LP} = \frac{c^\top y^*}{D^\top y^*} \geq \frac{c^\top d}{D^\top d} \cdot \frac{1}{\beta}.$$

Decompose $d = \sum_S \alpha_S \delta_S$ as in the proof of Theorem 7.4 (Remark 7.6). By Corollary 7.7, the best of these cuts S^* satisfies

$$\Phi(S^*) \leq \frac{c^\top d}{D^\top d} \leq \beta \cdot \mathrm{LP} \leq \beta \cdot \Phi^* = O(\log n) \cdot \Phi^*.$$

The random embedding of Section 7.5 has the required distortion with probability at least $1 - 1/n$; the distortion can be computed, and the construction is repeated until it succeeds. □

In short, Bourgain's theorem says that for every metric y there is an ℓ_1 metric d that is multiplicatively close to y, and the sparsest cut problem is rounded by this embedding of y into d.

7.5 Proof of Bourgain's theorem

There are several proofs; we give Bourgain's original one, almost. Fix a metric y on V, $n \geq 2$. For $x \in V$ and a nonempty set $A \subseteq V$, write $y(x, A) = \min_{u \in A} y_{xu}$ for the distance from x to A. The embedding has $\lfloor \log_2 n \rfloor$ groups of $\Theta(\log n)$ coordinates each.

Algorithm 7.10 (Bourgain's embedding)**.** Let $p_0 = 1/11$, $L = \lfloor \log_2 n \rfloor$ and $T = \lceil 32 \ln n / p_0 \rceil$. For each scale $g = 1, \ldots, L$ and each repetition $\rho = 1, \ldots, T$, pick a random set $R_{g,\rho} \subseteq V$ that contains each point independently with probability 2^{-g}, so that $\mathbb{E}|R_{g,\rho}| = n/2^g$, and set

$$\varphi_{g,\rho}(i) = y(i, R_{g,\rho}) \quad \text{for all } i \in V,$$

with $\varphi_{g,\rho} \equiv 0$ if $R_{g,\rho} = \emptyset$. Output $\varphi = (\varphi_{g,\rho})_{g,\rho}\colon V \to \mathbb{R}^{LT}$.

Claim 7.11. *For all $i, j \in V$:*

(**) with probability 1, $\|\varphi(i) - \varphi(j)\|_1 \le LT \cdot y_{ij}$, *where* $LT = \Theta(\log^2 n)$;

(*) with probability at least $1 - 1/n$, *simultaneously for all pairs,* $\|\varphi(i) - \varphi(j)\|_1 \ge (p_0 T/8) \cdot y_{ij} = \Omega(\log n) \cdot y_{ij}$.

Given the claim, $\varphi/(LT)$ has distortion at most $LT/(p_0 T/8) = 8L/p_0 = 88\lfloor \log_2 n \rfloor$ with probability at least $1 - 1/n$, which proves Theorem 7.8.

Proof of (**). Let $A \subseteq V$ be nonempty, and let $u \in A$ attain $y(j, A)$. Then $y(i, A) \le y_{iu} \le y_{ij} + y(j, A)$, and symmetrically, so $|y(i, A) - y(j, A)| \le y_{ij}$. Every one of the LT coordinates therefore contributes at most y_{ij}. □

It remains to show that the embedding gives a reasonable contribution to every distance. Fix $i \ne j$ with $y_{ij} > 0$ (pairs at distance 0 satisfy (*) trivially). Write $B(x, r) = \{u : y_{xu} \le r\}$ for the closed ball and $B^\circ(x, r) = \{u : y_{xu} < r\}$ for the open ball. For each $t \ge 0$ with $2^t \le n$, let

$$r_t = \min\{r \ge 0 : |B(i, r)| \ge 2^t \text{ and } |B(j, r)| \ge 2^t\}$$

be the smallest radius at which both balls have at least 2^t points. Then $r_0 = 0 \le r_1 \le r_2 \le \dots$. While $r_t < y_{ij}/4$, the balls $B(i, r_t)$ and $B(j, r_t)$ are disjoint, since a common point u would give $y_{ij} \le y_{iu} + y_{uj} < y_{ij}/2$, so they contain at least 2^{t+1} points and $t + 1 \le \log_2 n$. Hence $r_L \ge y_{ij}/4$, and $f = \min\{t \ge 1 : r_t \ge y_{ij}/4\}$ satisfies $1 \le f \le L$. We truncate the last radius: let $\hat{r}_t = r_t$ for $0 \le t < f$ and $\hat{r}_f = y_{ij}/4$. Then

$$0 = \hat{r}_0 \le \hat{r}_1 \le \cdots \le \hat{r}_{f-1} < \hat{r}_f = \frac{y_{ij}}{4}.$$

Claim 7.12. *Let* $1 \le t \le f$ *and let* R *contain each point independently with probability* 2^{-t}. *With probability at least* p_0 *(a* good event*),* $|y(i, R) - y(j, R)| \ge \hat{r}_t - \hat{r}_{t-1}$, *with* $R \ne \emptyset$.

Proof. One open ball is small. One of $B^\circ(i, \hat{r}_t)$ and $B^\circ(j, \hat{r}_t)$ has fewer than 2^t points: the radius $\hat{r}_t$ is determined by one of the two vertices. Indeed, if $\hat{r}_t < r_t$, then by the minimality of r_t one of the closed balls of radius $\hat{r}_t$ already has fewer than 2^t points. If $\hat{r}_t = r_t$ and both open balls had at least 2^t points, then, since an open ball is the increasing union of the closed balls of smaller radius and V is finite, both closed balls of some radius $r < r_t$ would have at least 2^t points, contradicting the minimality of r_t. The roles of i and j are symmetric, so we may assume that $|B^\circ(i, \hat{r}_t)| < 2^t$.

The other ball is large and disjoint. We have $|B(j, \hat{r}_{t-1})| \ge 2^{t-1}$ by the definition of r_{t-1}. A point u in both $B^\circ(i, \hat{r}_t)$ and $B(j, \hat{r}_{t-1})$ would give $y_{ij} \le y_{iu} + y_{uj} < \hat{r}_t + \hat{r}_{t-1} < y_{ij}/2$, so the two balls are disjoint (Fig. 12).

The good event. Let $\mathcal{G}$ be the event that R misses $B^\circ(i, \hat{r}_t)$ and hits $B(j, \hat{r}_{t-1})$. The two balls are disjoint and R picks points independently, so

$$\Pr[\mathcal{G}] \ge \left(1 - \frac{1}{2^t}\right)^{2^t} \left(1 - \left(1 - \frac{1}{2^t}\right)^{2^{t-1}}\right) \ge \frac{1}{4}(1 - e^{-1/2}) \ge \frac{1}{11} = p_0,$$

using $(1 - x)^{1/x} \ge 1/4$ for $0 < x \le 1/2$ and $(1 - x)^{1/(2x)} \le e^{-1/2}$. On $\mathcal{G}$, the set R is nonempty, every point of R is at distance at least $\hat{r}_t$ from i, and some point of R is within $\hat{r}_{t-1}$ of j. So $y(i, R) \ge \hat{r}_t$ and $y(j, R) \le \hat{r}_{t-1}$, and the coordinate given by R satisfies $|y(i, R) - y(j, R)| \ge \hat{r}_t - \hat{r}_{t-1}$. □

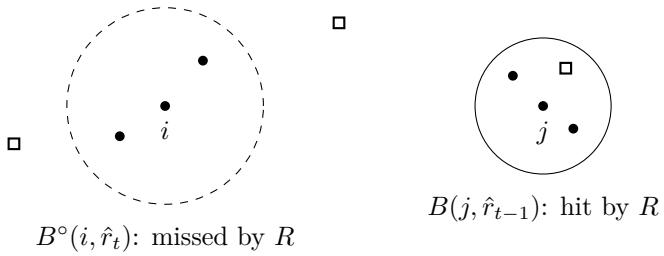


Figure 12: The good event of Claim 7.12. The open ball around i has fewer than 2^t points, the closed ball around j has at least 2^{t-1} points, and the two balls are disjoint. Squares are points of R.

Proof of (*). Fix a pair i, j with $y_{ij} > 0$ and a scale $t \in \{1, \dots, f\}$. The T coordinates of scale t are independent, and by Claim 7.12 each is good with probability at least p_0, so a constant fraction of them are good with high probability. By the Chernoff bound, fewer than $p_0 T/2$ of them are good with probability at most $\exp(-p_0 T/8) \le n^{-4}$. A union bound over fewer than $n^2/2$ pairs and at most $\log_2 n$ scales bounds the probability that this happens for some pair and some scale by $(\log_2 n)/(2n^2) \le 1/n$. Outside this event, every pair i, j gets a contribution of at least $(p_0 T/2)(\hat{r}_t - \hat{r}_{t-1})$ from scale t, for every $t \le f$. Summing over all sampling scales, the sum telescopes:

$$\|\varphi(i) - \varphi(j)\|_1 \ge \frac{p_0 T}{2}\Big[(\hat{r}_1 - \hat{r}_0) + (\hat{r}_2 - \hat{r}_1) + \cdots + \Big(\frac{y_{ij}}{4} - \hat{r}_{f-1}\Big)\Big] = \frac{p_0 T}{8}\, y_{ij} = \Omega(y_{ij} \log n).$$

This proves (*), and Bourgain's theorem follows by scaling down by $LT = \Theta(\log^2 n)$. □

7.6 Remarks

Remark 7.13. A similar argument shows that the same map φ, suitably scaled, is an embedding with distortion $O(\log n)$ into ℓ_p for every $p \ge 1$. Each coordinate is at most y_{ij}, so $\|\varphi(i) - \varphi(j)\|_p \le (LT)^{1/p} y_{ij}$; by Hölder's inequality, $\|\varphi(i) - \varphi(j)\|_p \ge (LT)^{1/p-1} \|\varphi(i) - \varphi(j)\|_1 \ge (LT)^{1/p-1}(p_0 T/8)\, y_{ij}$. The ratio of the two bounds is again $8L/p_0$.

Remark 7.14. A different proof of an $O(\log n)$-distortion embedding into ℓ_1 follows from approximating metrics by random distributions over trees. This technique is due to Bartal [Bar96], and the distortion $O(\log n)$ to Fakcharoenphol, Rao and Talwar [FRT04].

Remark 7.15. The proof of Corollary 7.9 needs the lower bound $\|\varphi(i) - \varphi(j)\|_1 \ge y_{ij}/O(\log k)$ only for the pairs with $D_{ij} > 0$, while the upper bound must hold for all pairs. This reduces $\log n$ to $\log k$. Let $\mathcal{T}$ be the set of at most $2k$ vertices that occur in a pair with $D_{ij} > 0$, and run Algorithm 7.10 with the sets $R_{g,\rho}$ sampled from $\mathcal{T}$ only, with $|\mathcal{T}|$ in place of n in L and T. The proof of (**) is unchanged, and the proof of (*), with balls restricted to $\mathcal{T}$, applies to pairs $i, j \in \mathcal{T}$. The distortion on the demand pairs is then $O(\log|\mathcal{T}|) = O(\log k)$, and the proof of Corollary 7.9 gives an $O(\log k)$-approximation.

Remark 7.16. The integrality gap of the metric relaxation is $\Omega(\log n)$ already for uniform demands, by Theorem 6.7. The uniform problem is a special case of the generalized one, so the lower bound on the gap holds here too, and the rounding above is optimal up to the constant for this relaxation. Better relaxations use semidefinite programs, which capture spectral information; eigenvalue methods are the subject of Lecture 8, and semidefinite programs for sparsest cut that of Lectures 10 and 11.

8 Sparsest Cut: Spectral Partitioning and the Cheeger Bounds

Lecture of September 17, 2021, by Anupam Gupta. Based on the lecturer's handwritten notes.

This lecture approaches the sparsest cut problem of Lecture 6 through eigenvalues; Trevisan's lecture notes [Tre16] cover this material in more detail. For a d-regular graph with unit capacities, relaxing the sparsest cut problem from ± 1 vectors to real vectors gives the lower bound λ_2/n on the sparsity, where λ_2 is the second smallest eigenvalue of the Laplacian. The Cheeger bounds complement it with an upper bound $O(\sqrt{d\lambda_2}/n)$, and the proof of the upper bound is an algorithm, spectral partitioning: it sorts the vertices by their entries in an eigenvector and returns the best prefix cut. The hypercube and the cycle show that both bounds can be tight. We also discuss the conductance and the normalized Laplacian, which play the same roles for irregular graphs, and what the Cheeger bounds say about expanders.

Throughout the lecture, $G = (V, E)$ is a d-regular graph without loops or parallel edges, with $n = |V| \geq 2$ vertices, $d \geq 1$, and unit capacities. As in Lecture 6, the sparsity of a cut $\emptyset \neq S \subsetneq V$ is $\phi(S) = |E(S, \bar{S})|/(|S| \cdot |\bar{S}|)$, here written with a lowercase ϕ, and $\phi(G) = \min_{\emptyset \neq S \subsetneq V} \phi(S)$. The Laplacian of G is $L_G = dI - A_G$, where A_G is the adjacency matrix, and its eigenvalues are $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. For every $x \in \mathbb{R}^V$,

$$x^\top L_G x = \sum_{\{i,j\} \in E} (x_i - x_j)^2, \tag{8.1}$$

so L_G is positive semidefinite and $L_G \mathbf{1} = 0$. Hence the all-ones vector $\mathbf{1}$ is an eigenvector for the smallest eigenvalue $\lambda_1 = 0$.

8.1 A spectral lower bound

For $S \subseteq V$, let $x \in \{-1, 1\}^V$ be 1 on S and -1 on $\bar{S}$. An edge or a pair contributes $(x_i - x_j)^2 = 4$ if it is separated by the cut and 0 otherwise, so

$$\phi(S) = \frac{\sum_{\{i,j\} \in E} (x_i - x_j)^2}{\sum_{i<j} (x_i - x_j)^2} = \frac{2 \sum_{\{i,j\} \in E} (x_i - x_j)^2}{\sum_{(i,j) \in V \times V} (x_i - x_j)^2},$$

where the first denominator sums over unordered pairs and the second over ordered pairs. This is very similar to the formulation of Max-Cut in Lecture 4. Hence $\phi(G)$ is the minimum of the right side over all nonconstant $x \in \{-1, 1\}^V$. If we relax this to all nonconstant $x \in \mathbb{R}^V$, the minimum can only fall. Two observations simplify the relaxation.

1) Both the numerator and the denominator depend only on differences $x_i - x_j$, so shifting x by a multiple of $\mathbf{1}$ changes neither. We may therefore assume that $x \perp \mathbf{1}$, that is, $\sum_i x_i = 0$.

2) For every $x \in \mathbb{R}^V$,

$$\sum_{(i,j) \in V \times V} (x_i - x_j)^2 = 2n \sum_{i \in V} x_i^2 - 2\Big(\sum_{i \in V} x_i\Big)^2,$$

and for $x \perp \mathbf{1}$ the last term vanishes.

With (8.1), the relaxation is therefore

$$\min_{x \perp \mathbf{1},\ x \neq 0} \frac{2 \sum_{\{i,j\} \in E} (x_i - x_j)^2}{2n \sum_{i \in V} x_i^2} = \frac{1}{n} \min_{x \perp \mathbf{1},\ x \neq 0} \frac{x^\top L_G x}{x^\top x} = \frac{\lambda_2(L_G)}{n} \leq \phi(G).$$

The last equality is the Courant–Fischer theorem: $\mathbf{1}$ is an eigenvector for λ_1, so the minimum of the Rayleigh quotient over the vectors orthogonal to $\mathbf{1}$ is the second eigenvalue. We have proved the following.

Theorem 8.1. *For every $\emptyset \neq S \subsetneq V$, we have $\phi(S) \geq \lambda_2(L_G)/n$.*

The Cheeger inequality bounds $\phi(G)$ from above by a function of λ_2 as well. This gives a good characterization of $\phi(G)$ for some families of graphs, especially expanders, although it can be poor for other families, such as the cycle (Example 8.9).

Remark 8.2. There are many variants of sparsity. Besides the sparsity, the edge expansion $|E(S,\bar{S})|/\min\{|S|,|\bar{S}|\}$ is between $n/2$ and n times the sparsity (Proposition 6.2), and the *conductance* is $|E(S,\bar{S})|/\min\{\sum_{v\in S}\deg(v), \sum_{v\in\bar{S}}\deg(v)\}$. For irregular graphs one may need the normalized Laplacian $I - D^{-1/2}AD^{-1/2}$, where D is the diagonal matrix of degrees (Section 8.3). These quantities are related, but a version of Cheeger's inequality found in the literature should be checked to relate exactly the quantities at hand; otherwise a translation is needed.

The main result of the lecture is the following pair of bounds.

Theorem 8.3 (Cheeger bounds [Che70, AM85])**.** *The sparsity of the d-regular graph G satisfies*

$$\frac{\lambda_2(L_G)}{n} \leq \phi(G) \leq \frac{8\sqrt{d\,\lambda_2(L_G)}}{n}.$$

Moreover, the cut found by Algorithm 8.4 has sparsity at most $8\sqrt{d\,\lambda_2(L_G)}/n$.

The lower bound is Theorem 8.1. There are graphs on which each of the bounds is tight up to a constant factor (Examples 8.8 and 8.9).

8.2 The upper bound: spectral partitioning

The proof of the upper bound is algorithmic.

Algorithm 8.4 (Spectral partitioning)**.** Find an eigenvector $x \in \mathbb{R}^V$ of L_G for λ_2 with $x \perp \mathbf{1}$. Order the vertices as $v_1, \ldots, v_n$ so that $x_{v_1} \leq x_{v_2} \leq \cdots \leq x_{v_n}$, and let $S_i = \{v_1, \ldots, v_i\}$. Output the set S_i, $1 \leq i \leq n-1$, with the smallest sparsity $\phi(S_i)$.

Such an eigenvector exists by the spectral theorem, since L_G has an orthonormal eigenbasis containing $\mathbf{1}/\sqrt{n}$. The sets S_i are the *level cuts* (or threshold cuts) of x: every nonempty set $\{v : x_v \leq \theta\}$ other than V, and the complement of every nonempty set $\{v : x_v \geq \theta\}$ other than V, is one of the S_i, whatever the order of equal entries. For $x \neq 0$ we write

$$R(x) = \frac{\sum_{\{i,j\}\in E}(x_i - x_j)^2}{\sum_{i\in V} x_i^2} = \frac{x^\top L_G x}{x^\top x}$$

for the *Rayleigh quotient* of x; for the eigenvector x of Algorithm 8.4, $R(x) = \lambda_2$. The proof has two lemmas. The first breaks x into (almost) its positive and negative parts.

Lemma 8.5. *Let $x \in \mathbb{R}^V$ with $x \perp \mathbf{1}$ and $x \neq 0$. There are nonzero vectors $y, z \in \mathbb{R}^V_{\geq 0}$ with disjoint supports such that*

$$R(y) \leq 8R(x) \quad \text{and} \quad R(z) \leq 8R(x),$$

and such that every level set $\{i : y_i^2 \geq \alpha\}$ and $\{i : z_i^2 \geq \alpha\}$ with $\alpha > 0$ is of the form $\{i : x_i \geq \theta\}$ or $\{i : x_i \leq \theta\}$ for some $\theta \in \mathbb{R}$.

Proof. Scaling does not change Rayleigh quotients or level sets, so let $\|x\|_2 = 1$; then $R(x) = \sum_{\{i,j\}\in E}(x_i - x_j)^2$.

First try. Let $y = \max(x, 0)$ and $z = -\min(x, 0)$, componentwise. For example, $x = (-0.7, -0.3, 0, 0.5, 1.2)$ gives $y = (0, 0, 0, 0.5, 1.2)$ and $z = (0.7, 0.3, 0, 0, 0)$. The maps $t \mapsto \max(t, 0)$ and $t \mapsto -\min(t, 0)$ do not increase distances, so

$$\sum_{\{i,j\}\in E}(y_i - y_j)^2 \le \sum_{\{i,j\}\in E}(x_i - x_j)^2 \quad \text{and} \quad \sum_{\{i,j\}\in E}(z_i - z_j)^2 \le \sum_{\{i,j\}\in E}(x_i - x_j)^2. \tag{8.2}$$

If $\|y\|_2^2 \ge 1/4$ and $\|z\|_2^2 \ge 1/4$, then $R(y), R(z) \le 4R(x)$, and we are done. But one of the parts may be small even though $x \perp \mathbf{1}$: for $x = (1, -\frac{1}{n-1}, \dots, -\frac{1}{n-1})$ we have $\|z\|_2^2 = \frac{1}{n-1}$, while $\|x\|_2^2 = \frac{n}{n-1}$.

Second try: shift x. Replacing x by $-x$ exchanges y and z, so we may assume that $\|z\|_2^2 < 1/4$, and hence $\|y\|_2^2 = 1 - \|z\|_2^2 > 3/4$. Let

$$x' = x - \frac{1}{2\sqrt{n}}\mathbf{1}, \qquad y' = \max(x', 0), \qquad z' = -\min(x', 0).$$

The shift moves mass to help z. It does not change the differences $x_i - x_j$, so (8.2) still holds for y' and z'. Since $x \perp \mathbf{1}$, Pythagoras gives $\|x'\|_2^2 = \|x\|_2^2 + \|\frac{1}{2\sqrt{n}}\mathbf{1}\|_2^2 = 5/4$.

- Because the shift is negative, $0 \le y'_i \le y_i$ for all i, so $\|y'\|_2^2 \le \|y\|_2^2 \le \|x\|_2^2 = 1$. As y' and z' have disjoint supports, $\|z'\|_2^2 = \|x'\|_2^2 - \|y'\|_2^2 \ge 5/4 - 1 = 1/4$.
- Since $y_i \le y'_i + \frac{1}{2\sqrt{n}}$ and $(a+b)^2 \le 2(a^2+b^2)$,

$$\frac{3}{4} \le \|y\|_2^2 = \sum_i y_i^2 \le \sum_i \Big(y'_i + \frac{1}{2\sqrt{n}}\Big)^2 \le 2\sum_i \Big((y'_i)^2 + \frac{1}{4n}\Big) = 2\|y'\|_2^2 + \frac{1}{2},$$

  so $\|y'\|_2^2 \ge 1/8$.

Hence y' and z' are nonzero, have disjoint supports, and satisfy $R(y') \le 8R(x)$ and $R(z') \le 4R(x)$ by (8.2); take $y = y'$ and $z = z'$.

Level sets. In both tries, $y = \max(x - c\mathbf{1}, 0)$ and $z = -\min(x - c\mathbf{1}, 0)$ for some $c \in \mathbb{R}$, possibly after replacing x by $-x$. For $\alpha > 0$, $\{i : y_i^2 \ge \alpha\} = \{i : x_i \ge c + \sqrt{\alpha}\}$ and $\{i : z_i^2 \ge \alpha\} = \{i : x_i \le c - \sqrt{\alpha}\}$: all we have done is shift and truncate x, so level sets of y and z are level sets of x. □

Since y and z have disjoint supports, at least one of them has support of size at most $n/2$. The second lemma finds a good level cut inside such a support.

Lemma 8.6. *Let $y \in \mathbb{R}^V_{\ge 0}$, $y \ne 0$, with $|\{i : y_i > 0\}| \le n/2$. There is a level set $S = \{i : y_i^2 \ge \alpha\}$ with $\alpha > 0$, contained in the support of y, such that*

$$\frac{|E(S, \bar{S})|}{|S|} \le \sqrt{2dR(y)} \qquad \textit{and hence} \qquad \phi(S) \le \frac{\sqrt{8dR(y)}}{n}.$$

Proof. Scaling does not change $R(y)$, so we may assume that $\max_i y_i = 1$, and hence $y_i \in [0, 1]$ for all i. Choose α uniformly at random in $(0, 1]$, and let $S_\alpha = \{i : y_i^2 \ge \alpha\}$, a random set. It is

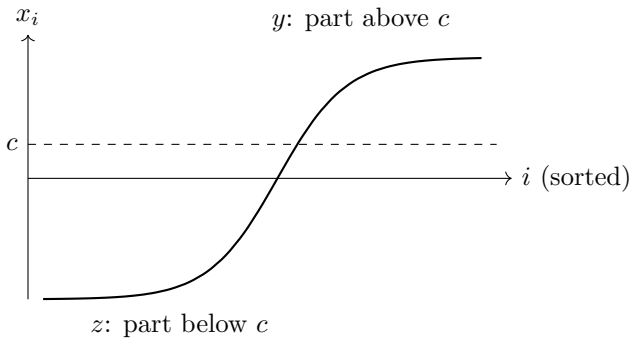


Figure 13: The vectors of Lemma 8.5: the entries of x in sorted order, the threshold c ($c = 0$ in the first try, $c = 1/(2\sqrt{n})$ in the second), and the parts y above and z below it.

nonempty, because it contains a vertex with $y_i = 1$, and it is contained in the support of y. An edge $\{i, j\}$ is cut by S_α exactly when one of y_i^2, y_j^2 is less than α and the other is at least α, so

$$\begin{aligned}
\mathbb{E}\big[|E(S_\alpha, \bar{S}_\alpha)|\big] &= \sum_{\{i,j\}\in E} \Pr_\alpha\big[\min(y_i^2, y_j^2) < \alpha \le \max(y_i^2, y_j^2)\big] = \sum_{\{i,j\}\in E} |y_i^2 - y_j^2| \\
&= \sum_{\{i,j\}\in E} |y_i - y_j|(y_i + y_j) \\
&\le \sqrt{\sum_{\{i,j\}\in E} (y_i - y_j)^2} \cdot \sqrt{\sum_{\{i,j\}\in E} (y_i + y_j)^2} \\
&\le \sqrt{\sum_{\{i,j\}\in E} (y_i - y_j)^2} \cdot \sqrt{2d \sum_{i\in V} y_i^2} = \sqrt{2dR(y)} \sum_{i\in V} y_i^2.
\end{aligned}$$

The first inequality is Cauchy–Schwarz. The second uses $(y_i + y_j)^2 \le 2(y_i^2 + y_j^2)$ and the fact that each vertex lies on exactly d edges, and the last equality uses the definition of $R(y)$. On the other hand, $\mathbb{E}|S_\alpha| = \sum_i \Pr_\alpha[y_i^2 \ge \alpha] = \sum_i y_i^2$. Therefore

$$\mathbb{E}\Big[|E(S_\alpha, \bar{S}_\alpha)| - \sqrt{2dR(y)}\,|S_\alpha|\Big] \le 0,$$

and by the probabilistic method there is $\alpha \in (0, 1]$ with $|E(S_\alpha, \bar{S}_\alpha)|/|S_\alpha| \le \sqrt{2dR(y)}$. Since $|S_\alpha| \le n/2$, we have $|\bar{S}_\alpha| \ge n/2$, and

$$\phi(S_\alpha) = \frac{|E(S_\alpha, \bar{S}_\alpha)|}{|S_\alpha| \cdot |\bar{S}_\alpha|} \le \frac{|E(S_\alpha, \bar{S}_\alpha)|}{|S_\alpha| \cdot n/2} \le \frac{2}{n}\sqrt{2dR(y)} = \frac{\sqrt{8dR(y)}}{n}.$$

So the best level set has at most this sparsity. □

Proof of the upper bound in Theorem 8.3. Let x be the eigenvector of Algorithm 8.4, so that $R(x) = \lambda_2$. Lemma 8.5 gives y and z, and one of them, say w, has support of size at most $n/2$ and $R(w) \le 8\lambda_2$. By Lemma 8.6, a level set S of w has $\phi(S) \le \sqrt{8d \cdot 8\lambda_2}/n = 8\sqrt{d\lambda_2}/n$. By Lemma 8.5, S is a set $\{i : x_i \ge \theta\}$ or $\{i : x_i \le \theta\}$, and $S \ne V$ because $|S| \le n/2 < n$. Either S or $\bar{S}$ is one of the sets S_i of Algorithm 8.4, and $\phi(S) = \phi(\bar{S})$. So the output of the algorithm has sparsity at most $8\sqrt{d\lambda_2}/n$, and so does the best cut of G. □

8.3 Tightness and discussion

To recap, the Cheeger bounds say that

$$\Omega\Big(\frac{\lambda_2}{n}\Big) \le \phi(G) \le O\Big(\frac{\sqrt{\lambda_2 d}}{n}\Big).$$

When d is a constant and $\lambda_2 = \Omega(1)$, they determine $\phi(G)$ up to constant factors. In particular, computing λ_2 gives a way to certify that a constant degree graph is an expander, even though computing $\phi(G)$ is NP-hard. In general the ratio of the two bounds is $O(\sqrt{d/\lambda_2})$, and λ_2 ranges as follows.

Proposition 8.7. *$0 \le \lambda_2(L_G) \le \frac{n}{n-1}\, d \le 2d$, and $\lambda_n(L_G) \le 2d$.*

Proof. By (8.1), L_G is positive semidefinite, so $\lambda_2 \ge 0$, and $x^\top L_G x \le \sum_{\{i,j\}\in E} 2(x_i^2 + x_j^2) = 2d\|x\|_2^2$ gives $\lambda_n \le 2d$. The trace of L_G is nd, and $\lambda_1 = 0$, so $\lambda_2 \le \frac{1}{n-1}\sum_{i=2}^n \lambda_i = \frac{nd}{n-1}$. □

The upper bound is attained by the complete graph K_n, for which $d = n-1$ and $\lambda_2 = n > d$. When $\lambda_2 = \Omega(d)$, the Cheeger bounds are tight up to constant factors; this is the spectral definition of an expander, discussed below.

Example 8.8 (Hypercube)**.** Let G be the hypercube on $\{0,1\}^d$, with an edge between two strings that differ in one coordinate, so that G is d-regular with $n = 2^d$. The cut $S = \{x : x_1 = 0\}$ has 2^{d-1} edges and $\phi(S) = 2^{d-1}/(2^{d-1}\cdot 2^{d-1}) = 2/n$. Moreover $\lambda_2(L_G) = 2$: for $T \subseteq \{1,\dots,d\}$, the vector $\chi_T(x) = (-1)^{\sum_{i\in T} x_i}$ is an eigenvector of L_G with eigenvalue $2|T|$, and these 2^d vectors form an orthogonal basis. By Theorem 8.1, $\phi(G) \ge 2/n$, so $\phi(G) = 2/n$ and the lower Cheeger bound is tight here.

Example 8.9 (Cycle and path)**.** Let $G = C_n$ be the cycle, with $d = 2$ and $n \ge 3$. Every cut $\emptyset \ne S \subsetneq V$ cuts at least two edges and has $|S|\cdot|\bar S| \le \lfloor n/2\rfloor\lceil n/2\rceil$, and the cut into two arcs of $\lfloor n/2\rfloor$ and $\lceil n/2\rceil$ vertices cuts exactly two edges. Hence $\phi(C_n) = 2/(\lfloor n/2\rfloor\lceil n/2\rceil) = \Theta(1/n^2)$. A balanced cut that is not of this form can be much worse: for even n, the set of every second vertex cuts all n edges and has sparsity $4/n$. The Laplacian eigenvalues of C_n are $2 - 2\cos(2\pi j/n)$, $0 \le j < n$, with eigenvectors $v \mapsto \cos(2\pi jv/n)$ and $v \mapsto \sin(2\pi jv/n)$, so $\lambda_2 = 2 - 2\cos(2\pi/n)$. Thus $\lambda_2 = \Theta(1/n^2)$ and $\sqrt{\lambda_2 d}/n = \Theta(1/n^2)$. The upper Cheeger bound is tight up to a constant factor here. The path P_n behaves in the same way: $\phi(P_n) = 1/(\lfloor n/2\rfloor\lceil n/2\rceil)$ and $\lambda_2(L_{P_n}) = 2 - 2\cos(\pi/n)$, both $\Theta(1/n^2)$, although P_n is not regular.

Irregular graphs and conductance. The bounds generalize to graphs that are not regular, but there it is convenient to use the conductance and the normalized Laplacian. Let G be a graph without isolated vertices, with $m = |E|$ edges and degree matrix D. For $S \subseteq V$ let $\mathrm{vol}(S) = \sum_{i\in S}\deg(i)$, so that $\mathrm{vol}(S)/\mathrm{vol}(V) = \mathrm{vol}(S)/(2m)$ is the fraction of edge endpoints that lie in S. The *conductance* of G is

$$h(G) = \min_{\emptyset\ne S,\ \mathrm{vol}(S)\le \mathrm{vol}(V)/2} \frac{|E(S,\bar S)|}{\mathrm{vol}(S)},$$

and its *normalized Laplacian* is $\mathcal{L}_G = I - D^{-1/2}A_G D^{-1/2}$. In this setting, Cheeger's inequality says that

$$\Theta(\lambda_2(\mathcal{L}_G)) \le h(G) \le \Theta\Big(\sqrt{\lambda_2(\mathcal{L}_G)}\Big); \tag{8.3}$$

more precisely, $\lambda_2(\mathcal{L}_G)/2 \le h(G) \le \sqrt{2\lambda_2(\mathcal{L}_G)}$ (for regular graphs, see [Tre16, Theorem 4.1]).

As a sanity check, let G be d-regular again. For $|S| \le n/2$, the conductance of S is $|E(S,\bar{S})|/\operatorname{vol}(S) = |E(S,\bar{S})|/(d|S|) = \frac{|\bar{S}|}{d}\,\phi(S)$, which lies between $\frac{n}{2d}\phi(S)$ and $\frac{n}{d}\phi(S)$. Also $\mathcal{L}_G = \frac{1}{d}L_G$, so $\lambda_2(\mathcal{L}_G) = \lambda_2(L_G)/d$, and (8.3) says

$$\Theta\Big(\frac{\lambda_2(L_G)}{d}\Big) \le \frac{n}{d}\,\phi(G) \le \Theta\Big(\sqrt{\frac{\lambda_2(L_G)}{d}}\Big),$$

which is what we proved in Theorem 8.3.

The second eigenvalue as a measure of connectivity. If G has k connected components, then $\lambda_1 = \lambda_2 = \cdots = \lambda_k = 0$, and $\lambda_{k+1} > 0$, so the multiplicity of the eigenvalue 0 is exactly the number of components. So $\lambda_2 = 0$ exactly when G is disconnected. The Cheeger bounds give a robust version of this statement.

- If λ_2 is close to 0, then G has a sparse cut, by the upper bound.
- If λ_2 is far from 0, then G has no sparse cut, by the lower bound.

A family of d-regular graphs is a family of *spectral expanders* if $\lambda_2(L_G) \ge \varepsilon d$ for a constant $\varepsilon > 0$. There is then a large spectral gap between $\lambda_1 = 0$ and $\lambda_2 = \Omega(d)$, while $\lambda_n \le 2d$ by Proposition 8.7.

Algorithmic aspects. The proof of the upper bound is algorithmic, and it shows more: for every $x \perp \mathbf{1}$ with $x \neq 0$, the best level cut of x has sparsity at most $8\sqrt{d\,R(x)}/n$, by Lemmas 8.5 and 8.6. So we do not need to compute an exact eigenvector for the second eigenvalue; any vector $x \perp \mathbf{1}$ with a small Rayleigh quotient is good enough. Spectral partitioning is commonly used in vision applications [SM00], and the eigenvector for λ_2 is called the *Fiedler vector* [Fie73]. Kwok, Lau, Lee, Oveis Gharan and Trevisan [KLL+13] showed that spectral partitioning performs well even when λ_2 is small, provided that a later eigenvalue λ_k is large: for d-regular G and every $k \ge 2$ with $\lambda_k > 0$, its output S satisfies

$$\phi(S) \le O\Big(\frac{k\,\lambda_2\sqrt{d}}{\sqrt{\lambda_k}}\Big) \cdot \frac{1}{n}.$$

Spectral bounds and the linear program. Cheeger's inequality shows that $\lambda_2(L_G)$ approximates $\phi(G)$ well when the expansion of G is large, while the integrality gaps of the Leighton–Rao linear program arise exactly on such expander-like graphs (Theorem 6.7). Semidefinite programs, which capture spectral ideas as well as the constraints of linear programs, combine the two techniques; this leads to the algorithm of Arora, Rao and Vazirani in Lectures 10 and 11. Using the second eigenvector to find a sparse cut also has a counterpart for Max-Cut, where eigenvector-based methods work too (Lecture 9).

9 Max-Cut and Eigenvalues of Graphs

Lecture of September 20, 2021, by Pravesh K. Kothari. Based on the lecturer's handwritten notes.

The Goemans–Williamson algorithm of Lecture 4 approximates Max-Cut by rounding a semidefinite program, while linear programs do no better than the trivial ratio $1/2$. Just as Lecture 8 related the sparsest cut to the second-smallest eigenvalue of the Laplacian, this lecture relates the max-cut to the largest eigenvalue λ_n, through the bipartiteness gap $\beta(G)$. The main result is Trevisan's inequality $\frac{1}{2}\beta_n \le \beta(G) \le \sqrt{2\beta_n}$ with $\beta_n = 2 - \lambda_n/d$, an analog of Cheeger's inequality proved by a rounding with two thresholds. Applied recursively, it yields an algorithm that needs only eigenvectors and cuts a $1 - 4\sqrt{\varepsilon}$ fraction of the edges whenever the max-cut is $1 - \varepsilon$, which qualitatively matches the Goemans–Williamson guarantee. Combined with a greedy cut, this algorithm has approximation ratio at least $32/63$, so a spectral method also beats $1/2$.

9.1 Max-cut and the largest eigenvalue of the Laplacian

Throughout this lecture, $G = (V, E)$ is an undirected graph without loops or parallel edges, with $V = [n]$ and $m = |E| \ge 1$ edges. For $S \subseteq V$ we write $\bar{S} = V \setminus S$, $E(S, \bar{S})$ for the set of edges with exactly one endpoint in S, and $E(S)$ for the set of edges with both endpoints in S. For disjoint $A, B \subseteq V$, $E(A, B)$ is the set of edges with one endpoint in A and the other in B. As in Lecture 4, the max-cut of G is

$$\operatorname{maxcut}(G) = \max_{S \subseteq V} \frac{|E(S, \bar{S})|}{m},$$

and we write $\operatorname{cut}(S) = |E(S, \bar{S})|/m$ for the value of the cut $(S, \bar{S})$.

The Goemans–Williamson algorithm [GW95] of Lecture 4 takes a graph G and outputs a random set $S \subseteq V$ with the following two guarantees: if $\operatorname{maxcut}(G) \ge 1 - \varepsilon$, then $\mathbb{E}[\operatorname{cut}(S)] \ge 1 - O(\sqrt{\varepsilon})$; and in every case $\mathbb{E}[\operatorname{cut}(S)] \ge \alpha_{\mathrm{GW}} \operatorname{maxcut}(G)$, where $\alpha_{\mathrm{GW}} \approx 0.878$. It is based on semidefinite programming, and Lecture 4 also explains why linear programs cannot approximate Max-Cut better than $1/2$. In Lecture 8, spectral methods related the sparsest cut to the second-smallest eigenvalue λ_2 of the Laplacian. Here we relate Max-Cut to the largest eigenvalue of the Laplacian. To keep the exposition simple, until the generalization to irregular graphs we assume that G is d-regular with $d \ge 1$, so that $m = nd/2$.

Recall that the Laplacian of G is $L_G = D_G - A_G$, which for a d-regular graph is $dI - A_G$, and that

$$x^\top L_G x = \sum_{\{i,j\} \in E} (x_i - x_j)^2 \qquad \text{for every } x \in \mathbb{R}^n. \tag{9.1}$$

Let $x \in \{\pm 1\}^n$ indicate a set S, that is, $x_i = 1$ if and only if $i \in S$. Then $(x_i - x_j)^2$ is 4 for an edge of $E(S, \bar{S})$ and 0 for every other edge, so $x^\top L_G x = 4|E(S, \bar{S})|$. Since $\|x\|_2^2 = n$ and $m = nd/2$, it follows that

$$\operatorname{maxcut}(G) = \max_{x \in \{\pm 1\}^n} \frac{x^\top L_G x}{2nd} = \max_{x \in \{\pm 1\}^n} \frac{1}{2d} \cdot \frac{x^\top L_G x}{\|x\|_2^2}. \tag{9.2}$$

Definition 9.1. For a graph G on n vertices, let

$$\lambda_n \coloneqq \max_{x \in \mathbb{R}^n,\ x \neq 0} \frac{x^\top L_G x}{\|x\|_2^2}.$$

By the spectral theorem, λ_n is the largest eigenvalue of L_G, and the maximum is attained exactly at its eigenvectors.

Up to the factor $1/(2d)$, the quantity λ_n differs from the right side of (9.2) only in dropping the Boolean constraint $x \in \{\pm 1\}^n$. In particular, $\mathrm{maxcut}(G) \le \lambda_n/(2d)$. As with λ_2 and the sparsest cut, we would like a converse: if λ_n is large, then the max-cut is large.

There is a subtlety. Unlike λ_2, the eigenvalue λ_n is not robust: it can be determined by a very small part of the graph and ignore all the rest. We first bound λ_n.

Proposition 9.2. *If G is d-regular, then $\lambda_n \le 2d$.*

Proof. Let ℓ_i denote the i-th row of L_G. We have $\lambda_n \le \max_i \|\ell_i\|_1$. Indeed, let $x \ne 0$ satisfy $L_G x = \lambda_n x$, and choose i with $|x_i| = \|x\|_\infty > 0$. Then $|\lambda_n||x_i| = |\sum_j (L_G)_{ij} x_j| \le \sum_j |(L_G)_{ij}||x_j| \le \|\ell_i\|_1 |x_i|$. The row ℓ_i has the diagonal entry d and d entries equal to -1, so $\|\ell_i\|_1 = 2d$. □

Proposition 9.3. *If G is d-regular and has a connected component that is bipartite, then $\lambda_n = 2d$.*

Proof. Let $S \subseteq V$ be the vertex set of a bipartite connected component, and let $S = L \cup R$ be its bipartition into a left vertex set L and a right vertex set R. Define $x \in \mathbb{R}^n$ by $x_i = 1$ for $i \in L$, $x_i = -1$ for $i \in R$, and $x_i = 0$ for $i \notin S$. An edge with no endpoint in S contributes 0 to (9.1), and no edge joins S to $\bar{S}$ because S is a connected component. Every edge of $E(S)$ joins L to R and contributes $(\pm 2)^2 = 4$. Finally, all d neighbors of a vertex of S lie in S, so $|E(S)| = d|S|/2$. Hence

$$x^\top L_G x = \sum_{\{i,j\} \in E(S)} (x_i - x_j)^2 = 4 \cdot \frac{d|S|}{2} = 2d|S|, \qquad \frac{x^\top L_G x}{\|x\|_2^2} = \frac{2d|S|}{|S|} = 2d.$$

Thus $\lambda_n \ge 2d$, and Proposition 9.2 gives equality. □

So λ_n is governed entirely by a bipartite component, however small, and more generally it can be influenced by small, almost bipartite parts of the graph. We therefore cannot hope to relate $\mathrm{maxcut}(G)$ directly to λ_n. We can still hope to prove a robust converse of Proposition 9.3. If $\lambda_n \ge 2d(1-\varepsilon)$, then there should be a set S of vertices whose induced subgraph is almost bipartite and almost disconnected from the rest of the graph. That is, S should split into L and R so that all but an $O(\sqrt{\varepsilon})$ fraction of the edges incident to S join L to R. The next definition makes this precise.

9.2 The bipartiteness gap and Trevisan's inequality

Definition 9.4. Let G be d-regular with $d \ge 1$. For disjoint sets $L, R \subseteq V$ with $S = L \cup R \ne \emptyset$, let

$$\beta(L,R) := \frac{2|E(L)| + 2|E(R)| + |E(S,\bar{S})|}{d\,|S|}.$$

The *bipartiteness gap* of G is $\beta(G) := \min \beta(L,R)$, the minimum over all such pairs.

A pair (L,R) corresponds to the vector $y \in \{\pm 1, 0\}^n$ with $y_i = 1$ on L, $y_i = -1$ on R, and $y_i = 0$ on $\bar{S}$, and $y \ne 0$ because $S \ne \emptyset$. For an edge $\{i,j\}$, the value $|y_i + y_j|$ is 2 on $E(L)$ and on $E(R)$, is 1 on $E(S,\bar{S})$, and is 0 on $E(L,R)$ and on $E(\bar{S})$. Since $\sum_i |y_i| = |S|$,

$$\beta(G) = \min_{y \in \{\pm 1, 0\}^n,\ y \ne 0} \frac{\sum_{\{i,j\} \in E} |y_i + y_j|}{d \sum_i |y_i|}. \tag{9.3}$$

The denominator $d|S|$ counts the endpoints in S of edges, an edge of $E(L)$ or $E(R)$ has two endpoints in S, and an edge of $E(S, \bar{S})$ has one. So $\beta(L, R)$ is the fraction of the edges incident to S that are not cut by (L, R), where each edge is counted once for each of its endpoints in S; see Fig. 14.

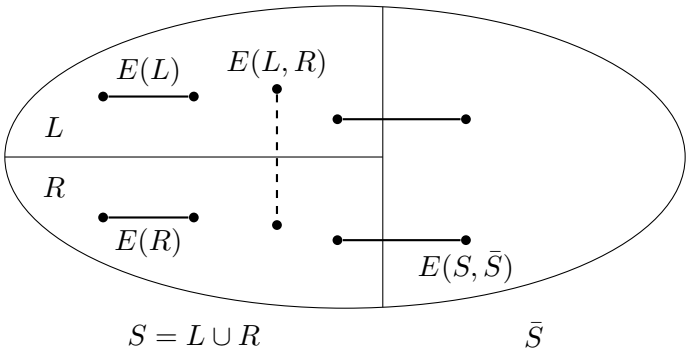


Figure 14: The quantities in the bipartiteness gap $\beta(L, R)$. Edges of $E(L)$ and of $E(R)$ count twice in the numerator, edges of $E(S, \bar{S})$ (here one from L and one from R) count once, and the dashed edges of $E(L, R)$, which the pair (L, R) cuts, do not count.

If G has a bipartite connected component with sides L and R, then $E(L) = E(R) = E(S, \bar{S}) = \emptyset$, so $\beta(G) = 0$; and $\beta(G) \geq 0$ always. Given Proposition 9.3, we may hope to relate $\beta(G)$ to λ_n, after the following normalization.

Definition 9.5. For a d-regular graph G with $d \geq 1$, let $\beta_n := 2 - \lambda_n/d$.

By Proposition 9.2, $\beta_n \geq 0$, and $\beta_n = 0$ when G has a bipartite connected component. The proof of the main theorem rounds a random vector and then needs a fixed outcome that is at least as good as the average; the following averaging fact provides it.

Lemma 9.6. *Let $u_1, \ldots, u_k \geq 0$, $v_1, \ldots, v_k > 0$ and $w_1, \ldots, w_k \geq 0$ be real numbers with $\sum_i w_i = 1$. Then*

$$\min_i \frac{u_i}{v_i} \leq \frac{\sum_i w_i u_i}{\sum_i w_i v_i}.$$

Proof. Let a denote the right side; its denominator is positive, since some $w_i > 0$ and every $v_i > 0$. Suppose, for a contradiction, that $u_i > a v_i$ for every i. Multiplying by $w_i \geq 0$ and summing, with strict inequality for an index with $w_i > 0$, gives $\sum_i w_i u_i > a \sum_i w_i v_i$, that is, $\frac{\sum_i w_i u_i}{\sum_i w_i v_i} > a$, a contradiction. □

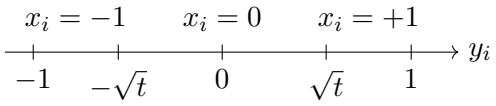
Both the proof of the main theorem below and the algorithm built on it round a real vector y to a vector in $\{\pm 1, 0\}^n$, using two thresholds.

Algorithm 9.7 (Two-threshold rounding)**.** Input: a vector $y \in \mathbb{R}^n$ with $\max_i y_i^2 = 1$.

1) Choose t uniformly at random from $(0, 1]$.

2) For each $i \in [n]$, set

$$x_i = \begin{cases} +1 & \text{if } y_i \geq \sqrt{t}, \\ -1 & \text{if } y_i \leq -\sqrt{t}, \\ 0 & \text{otherwise.} \end{cases}$$



Since $t > 0$, the coordinate x_i is nonzero if and only if $y_i^2 \geq t$, and then x_i has the sign of y_i. Because $y_i^2 \in [0,1]$, this gives

$$\Pr[x_i \neq 0] = \Pr[t \leq y_i^2] = y_i^2 \qquad \text{for every } i. \tag{9.4}$$

Unlike the threshold rounding behind Cheeger's inequality in Lecture 8, which splits all vertices by a single threshold, this rounding uses two thresholds $\pm\sqrt{t}$ and leaves the vertices with $y_i^2 < t$ unassigned, with $x_i = 0$.

Claim 9.8. *Let $y \in \mathbb{R}^n$ with $\max_i y_i^2 = 1$, and let x be the output of Algorithm 9.7. For every pair $i, j \in [n]$,*

$$\mathbb{E}|x_i + x_j| \leq |y_i + y_j|(|y_i| + |y_j|).$$

Proof. By symmetry we may assume that $y_i^2 \geq y_j^2$. Then $x_j \neq 0$ implies $x_i \neq 0$.

Case 1: y_i and y_j have the same sign, that is, $y_i y_j \geq 0$ (either may be 0). Then $y_i \geq y_j \geq 0$ or $y_i \leq y_j \leq 0$, and the possible outcomes (x_i, x_j) are $(0,0)$, $(\pm 1, 0)$ and $(\pm 1, \pm 1)$, with equal signs. Hence $|x_i + x_j| = 2$ if $t \leq y_j^2$, $|x_i + x_j| = 1$ if $y_j^2 < t \leq y_i^2$, and $|x_i + x_j| = 0$ if $t > y_i^2$. Thus

$$\mathbb{E}|x_i + x_j| = 1 \cdot \Pr[y_j^2 < t \leq y_i^2] + 2\Pr[t \leq y_j^2] = (y_i^2 - y_j^2) + 2y_j^2 = y_i^2 + y_j^2 \leq \big(|y_i| + |y_j|\big)^2,$$

and $\big(|y_i| + |y_j|\big)^2 = |y_i + y_j|(|y_i| + |y_j|)$ because y_i and y_j have the same sign.

Case 2: y_i and y_j have opposite signs, that is, $y_i y_j < 0$. When $t \leq y_j^2$, both x_i and x_j are nonzero with opposite signs, so $x_i + x_j = 0$. When $y_j^2 < t \leq y_i^2$, only x_i is nonzero and $|x_i + x_j| = 1$. When $t > y_i^2$, both vanish. Thus

$$\mathbb{E}|x_i + x_j| = \Pr[y_j^2 < t \leq y_i^2] = y_i^2 - y_j^2 = \big(|y_i| - |y_j|\big)\big(|y_i| + |y_j|\big) = |y_i + y_j|(|y_i| + |y_j|),$$

where the last step uses $|y_i + y_j| = |y_i| - |y_j|$ for numbers of opposite signs with $|y_i| \geq |y_j|$. □

Theorem 9.9 (Trevisan's inequality [Tre12]; see also Soto [Sot15])**.** *For every d-regular graph G with $d \geq 1$,*

$$\frac{1}{2}\beta_n \leq \beta(G) \leq \sqrt{2\beta_n}.$$

Proof. Let $Q_G = dI + A_G$, so that $x^\top Q_G x = \sum_{\{i,j\} \in E} (x_i + x_j)^2$ for every $x \in \mathbb{R}^n$. For every x, the identity $(x_i + x_j)^2 + (x_i - x_j)^2 = 2(x_i^2 + x_j^2)$, summed over the edges, gives

$$\sum_{\{i,j\} \in E} \big[(x_i + x_j)^2 + (x_i - x_j)^2\big] = \sum_{\{i,j\} \in E} 2(x_i^2 + x_j^2) = 2d\|x\|_2^2,$$

because each vertex lies on exactly d edges. Hence $x^\top L_G x + x^\top Q_G x = 2d\|x\|_2^2$, and since λ_n is the maximum of $x^\top L_G x / \|x\|_2^2$,

$$\min_{x \in \mathbb{R}^n,\ x \neq 0} \frac{x^\top Q_G x}{\|x\|_2^2} = 2d - \lambda_n = d\beta_n, \tag{9.5}$$

with the minimum attained at the eigenvectors of L_G for λ_n.

The first inequality. Let (L, R) attain the minimum in Definition 9.4, and let $y \in \{\pm 1, 0\}^n$ be the corresponding vector, so that $y \neq 0$. For $y \in \{\pm 1, 0\}^n$ we have $|y_i + y_j| \leq 2$, hence $(y_i + y_j)^2 \leq 2|y_i + y_j|$, and $\|y\|_2^2 = \sum_i |y_i|$. By (9.5) and (9.3),

$$2d - \lambda_n \leq \frac{\sum_{\{i,j\}\in E}(y_i + y_j)^2}{\|y\|_2^2} \leq \frac{\sum_{\{i,j\}\in E} 2|y_i + y_j|}{\sum_i |y_i|} = 2d\beta(L, R) = 2d\beta(G).$$

Dividing by d gives $\beta_n \leq 2\beta(G)$.

The second inequality. Let $y \in \mathbb{R}^n$ be an eigenvector of L_G for λ_n, so that $y^\top Q_G y = (2d - \lambda_n)\|y\|_2^2 = d\beta_n\|y\|_2^2$. Rescaling, we may assume that $\max_i y_i^2 = 1$. Let x be the output of Algorithm 9.7 on y. Summing Claim 9.8 over the edges and applying the Cauchy–Schwarz inequality,

$$\begin{aligned}
\mathbb{E} \sum_{\{i,j\}\in E} |x_i + x_j| &\leq \sum_{\{i,j\}\in E} |y_i + y_j|(|y_i| + |y_j|) \\
&\leq \sqrt{\sum_{\{i,j\}\in E} (y_i + y_j)^2} \sqrt{\sum_{\{i,j\}\in E} (|y_i| + |y_j|)^2} \\
&\overset{\text{(i)}}{\leq} \sqrt{d\beta_n\|y\|_2^2} \sqrt{\sum_{\{i,j\}\in E} 2(y_i^2 + y_j^2)} \\
&\overset{\text{(ii)}}{=} \sqrt{d\beta_n\|y\|_2^2}\sqrt{2d\|y\|_2^2} = \sqrt{2\beta_n}\, d\|y\|_2^2.
\end{aligned}$$

Step (i) uses $y^\top Q_G y = d\beta_n\|y\|_2^2$ and $(a + b)^2 \leq 2(a^2 + b^2)$, and step (ii) uses d-regularity. By (9.4), $\mathbb{E}\sum_i |x_i| = \sum_i \Pr[t \leq y_i^2] = \sum_i y_i^2 = \|y\|_2^2$. Together,

$$\mathbb{E} \sum_{\{i,j\}\in E} |x_i + x_j| \leq \sqrt{2\beta_n}\, d\, \mathbb{E} \sum_i |x_i|. \tag{9.6}$$

The rounded vector x depends on t only through the set $\{i : y_i^2 \geq t\}$, so it takes at most n distinct values $x^{(1)}, \ldots, x^{(k)}$, with probabilities $w_1, \ldots, w_k$ summing to 1. Each $x^{(l)}$ is nonzero, because some i has $y_i^2 = 1 \geq t$. Let $u_l = \sum_{\{i,j\}\in E} |x_i^{(l)} + x_j^{(l)}|$ and $v_l = d\sum_i |x_i^{(l)}| > 0$. Then (9.6) says $\sum_l w_l u_l \leq \sqrt{2\beta_n}\sum_l w_l v_l$, so by Lemma 9.6 some l satisfies $u_l/v_l \leq \sqrt{2\beta_n}$. By (9.3), $\beta(G) \leq u_l/v_l \leq \sqrt{2\beta_n}$. □

Remark 9.10. The proof is constructive, as for Cheeger's inequality: computing an eigenvector y for λ_n and trying the at most n threshold vectors (the outputs for the positive values t among $y_1^2, \ldots, y_n^2$) finds disjoint L, R with $L \cup R \neq \emptyset$ and $\beta(L, R) \leq \sqrt{2\beta_n}$.

9.3 Generalization to irregular graphs

For irregular graphs the key change is to measure the sizes of sets and the lengths of vectors differently, using the degrees, and to work with a normalized Laplacian. In this subsection, G has no isolated vertices. Let $d_i \geq 1$ be the degree of vertex i, let $D = \operatorname{diag}(d_1, \ldots, d_n)$, and let $D^{-1/2} = \operatorname{diag}(1/\sqrt{d_1}, \ldots, 1/\sqrt{d_n})$. For $S \subseteq V$ and $y \in \mathbb{R}^n$ we write

$$|S|_D \coloneqq \sum_{i\in S} d_i, \qquad \|y\|_D^2 \coloneqq y^\top D y = \sum_i d_i y_i^2.$$

Definition 9.11. The *normalized Laplacian* of G is $\mathcal{L}_G \coloneqq D^{-1/2} L_G D^{-1/2}$, and

$$\lambda_n(\mathcal{L}_G) \coloneqq \max_{x \in \mathbb{R}^n,\ x \neq 0} \frac{x^\top \mathcal{L}_G x}{\|x\|_2^2}$$

is its largest eigenvalue.

Substituting $y = D^{-1/2}x$, which is a bijection of $\mathbb{R}^n \setminus \{0\}$, gives

$$\lambda_n(\mathcal{L}_G) = \max_{y \in \mathbb{R}^n,\ y \neq 0} \frac{y^\top L_G y}{y^\top D y} = \max_{y \neq 0} \frac{\sum_{\{i,j\} \in E} (y_i - y_j)^2}{\sum_i d_i y_i^2}, \tag{9.7}$$

and the maximum is attained at $y = D^{-1/2}x$ for the eigenvectors x of $\mathcal{L}_G$ for $\lambda_n(\mathcal{L}_G)$. So moving to the normalized Laplacian simply replaces the ℓ_2 length of the vector by the weighted ℓ_2 length $\|y\|_D$.

Definition 9.12. For disjoint sets $L, R \subseteq V$ with $S = L \cup R \neq \emptyset$, let

$$\beta(L, R) \coloneqq \frac{2|E(L)| + 2|E(R)| + |E(S, \bar{S})|}{|S|_D},$$

and let the *bipartiteness gap* of G be $\beta(G) \coloneqq \min \beta(L, R)$ over all such pairs. As in (9.3),

$$\beta(G) = \min_{y \in \{\pm 1, 0\}^n,\ y \neq 0} \frac{\sum_{\{i,j\} \in E} |y_i + y_j|}{\sum_i d_i |y_i|}. \tag{9.8}$$

For a d-regular graph, $|S|_D = d|S|$, so this agrees with Definition 9.4.

Proposition 9.13. *If G has a bipartite connected component, then $\lambda_n(\mathcal{L}_G) = 2$.*

Proof. Let S be the vertex set of a bipartite connected component, with sides L and R, and let $y_i = 1$ for $i \in L$, $y_i = -1$ for $i \in R$, and $y_i = 0$ otherwise. Every edge incident to S joins L to R, so $|S|_D = 2|E(L, R)|$, and

$$\frac{y^\top L_G y}{y^\top D y} = \frac{\sum_{\{i,j\} \in E} (y_i - y_j)^2}{\sum_i d_i y_i^2} = \frac{4|E(L, R)|}{|S|_D} = 2.$$

By (9.7), $\lambda_n(\mathcal{L}_G) \geq 2$. Conversely, for every $z \in \mathbb{R}^n$ and every edge $\{i, j\}$ we have $(z_i - z_j)^2 \leq 2(z_i^2 + z_j^2)$, and summing over the edges gives $z^\top L_G z \leq 2 \sum_i d_i z_i^2$, so $\lambda_n(\mathcal{L}_G) \leq 2$. □

Definition 9.14. Let $\beta_n \coloneqq 2 - \lambda_n(\mathcal{L}_G)$.

For a d-regular graph, $\mathcal{L}_G = L_G/d$, so this agrees with Definition 9.5.

Proposition 9.15.

$$\beta_n = \min_{y \in \mathbb{R}^n,\ y \neq 0} \frac{\sum_{\{i,j\} \in E} (y_i + y_j)^2}{\sum_i d_i y_i^2},$$

and the minimum is attained at $y = D^{-1/2}x$ for every eigenvector x of $\mathcal{L}_G$ for $\lambda_n(\mathcal{L}_G)$.

Proof. For every $y \in \mathbb{R}^n$,

$$\sum_{\{i,j\}\in E} [(y_i + y_j)^2 + (y_i - y_j)^2] = 2 \sum_{\{i,j\}\in E} (y_i^2 + y_j^2) = 2 \sum_i d_i y_i^2. \tag{9.9}$$

By (9.7), $\sum_{\{i,j\}\in E}(y_i - y_j)^2 \le \lambda_n(\mathcal{L}_G) \sum_i d_i y_i^2$ for every y, with equality at $y = D^{-1/2}x$ for the eigenvectors x. Subtracting this from (9.9) gives $\sum_{\{i,j\}\in E}(y_i + y_j)^2 \ge (2 - \lambda_n(\mathcal{L}_G)) \sum_i d_i y_i^2$, with equality at the same vectors. □

Theorem 9.16 (Trevisan [Tre12])**.** *For every graph G without isolated vertices,*

$$\frac{1}{2}\beta_n \le \beta(G) \le \sqrt{2\beta_n}.$$

Proof. *The first inequality.* For $y \in \{\pm 1, 0\}^n$ we have $(y_i + y_j)^2 \le 2|y_i + y_j|$ and $\sum_i d_i y_i^2 = \sum_i d_i |y_i|$. By Proposition 9.15 and (9.8),

$$\begin{aligned} \beta_n &= \min_{y\in\mathbb{R}^n,\ y\neq 0} \frac{\sum_{\{i,j\}\in E}(y_i + y_j)^2}{\sum_i d_i y_i^2} \le \min_{y\in\{\pm1,0\}^n,\ y\neq 0} \frac{\sum_{\{i,j\}\in E}(y_i + y_j)^2}{\sum_i d_i y_i^2} \\ &\le \min_{y\in\{\pm1,0\}^n,\ y\neq 0} \frac{\sum_{\{i,j\}\in E} 2|y_i + y_j|}{\sum_i d_i y_i^2} = 2\beta(G). \end{aligned}$$

The second inequality. Let φ be an eigenvector of $\mathcal{L}_G$ for $\lambda_n(\mathcal{L}_G)$ and let $y = D^{-1/2}\varphi$, so that $\sum_{\{i,j\}\in E}(y_i + y_j)^2 = \beta_n \sum_i d_i y_i^2$ by Proposition 9.15. Rescale y so that $\max_i y_i^2 = 1$, and let $x \in \{\pm 1, 0\}^n$ be the output of Algorithm 9.7 on y. Claim 9.8 holds as before, since its proof does not use regularity. Summing it over the edges and applying the Cauchy–Schwarz inequality as in the regular case,

$$\begin{aligned} \mathbb{E} \sum_{\{i,j\}\in E} |x_i + x_j| &\le \sum_{\{i,j\}\in E} |y_i + y_j|(|y_i| + |y_j|) \\ &\le \sqrt{\sum_{\{i,j\}\in E} (y_i + y_j)^2} \sqrt{\sum_{\{i,j\}\in E} (|y_i| + |y_j|)^2} \\ &\le \sqrt{\beta_n \sum_i d_i y_i^2} \sqrt{\sum_{\{i,j\}\in E} 2(y_i^2 + y_j^2)} = \sqrt{2\beta_n} \sum_i d_i y_i^2. \end{aligned}$$

By (9.4), $\mathbb{E}\sum_i d_i |x_i| = \sum_i d_i \Pr[t \le y_i^2] = \sum_i d_i y_i^2$. Hence

$$\mathbb{E} \sum_{\{i,j\}\in E} |x_i + x_j| \le \sqrt{2\beta_n}\, \mathbb{E} \sum_i d_i |x_i|.$$

As in the regular case, x takes at most n distinct values, all nonzero, so by Lemma 9.6 some x in the support satisfies $\sum_{\{i,j\}\in E} |x_i + x_j| \le \sqrt{2\beta_n} \sum_i d_i |x_i|$. By (9.8), $\beta(G) \le \sqrt{2\beta_n}$. □

As in Remark 9.10, such an x can be found by searching over the at most n threshold vectors. We record this procedure.

Corollary 9.17 (Trevisan's rounding)**.** *Let G be a graph without isolated vertices. Compute an eigenvector φ of $\mathcal{L}_G$ for $\lambda_n(\mathcal{L}_G)$, set $y = D^{-1/2}\varphi$ scaled so that $\max_i y_i^2 = 1$, and among the outputs of Algorithm 9.7 for the positive values t among $y_1^2, \dots, y_n^2$, take one, z, minimizing $\sum_{\{i,j\}\in E} |z_i + z_j| / \sum_i d_i |z_i|$. Then $L = \{i : z_i = 1\}$ and $R = \{i : z_i = -1\}$ are disjoint, $L \cup R \neq \emptyset$, and $\beta(L, R) \le \sqrt{2\beta_n}$.*

Proof. For t in an interval $(a, b]$ between consecutive values in $\{0\} \cup \{y_1^2, \ldots, y_n^2\}$, the output of Algorithm 9.7 on y equals its output for $t = b$, so every outcome is among the vectors tried. The proof of Theorem 9.16 shows that some outcome $z \neq 0$ satisfies the bound. □

9.4 A spectral algorithm for Max-Cut

Trevisan's rounding is a *partial* rounding: it finds a set $S = L \cup R$ such that the pair (L, R) cuts almost all edges incident to S, but it says nothing about the edges of $E(\bar{S})$. The idea is to beat $1/2$ on the edges of $E(L, R)$, to do acceptably on the edges of $E(S, \bar{S})$ by cutting half of them, and to recurse on the edges of $E(\bar{S})$. For $U \subseteq V$, let $G[U] = (U, E(U))$ be the subgraph induced by U. As in Lecture 4, we assume exact arithmetic, here for the eigenvector computations.

Algorithm 9.18 (Recursive spectral cut)**.** Input: a graph $G = (V, E)$. Output: a partition of V into two sets.

1) If $E = \emptyset$, return $(V, \emptyset)$.

2) Apply Trevisan's rounding (Corollary 9.17) to G with its isolated vertices removed, obtaining disjoint L, R with $S = L \cup R \neq \emptyset$.

3) Apply the algorithm recursively to $G[\bar{S}]$, obtaining a partition (L', R') of $\bar{S}$.

4) Return the better of the two cuts $(L \cup L', R \cup R')$ and $(L \cup R', R \cup L')$, that is, the one that cuts more edges of G.

The recursion terminates because $G[\bar{S}]$ has fewer vertices than G. See Fig. 15. For a graph G with at least one edge, let $\mathrm{ALG}(G)$ be the fraction of the edges of G cut by the output. It may depend on the eigenvectors chosen, and the bounds below hold for every choice.

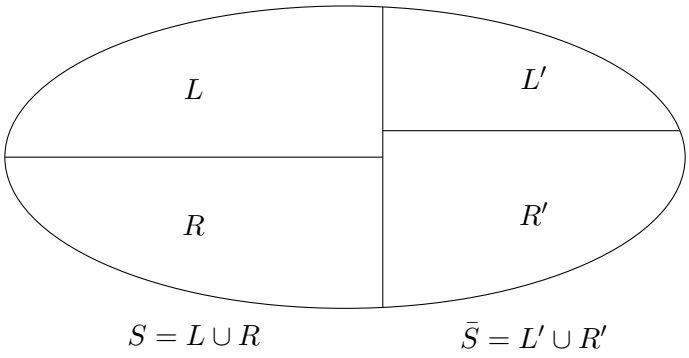


Figure 15: One step of Algorithm 9.18. Trevisan's rounding finds $S = L \cup R$, the recursion splits $\bar{S}$ into L' and R', and the output is the better of $(L \cup L', R \cup R')$ and $(L \cup R', R \cup L')$.

Theorem 9.19. *For every graph G with at least one edge, if* $\mathrm{maxcut}(G) = 1 - \varepsilon$*, then* $\mathrm{ALG}(G) \geq 1 - 4\sqrt{\varepsilon}$.

Corollary 9.20. *The algorithm that outputs the better of the greedy cut and the cut of Algorithm 9.18 cuts at least* $\frac{32}{63}\,\mathrm{maxcut}(G) \cdot m$ *edges of every graph G with $m \geq 1$ edges. Thus its approximation ratio is at least* $32/63 > 0.507$.

The rest of this subsection proves these two results. The first step bounds the bipartiteness gap of a graph with a large max-cut.

Claim 9.21. *Let G have no isolated vertices and* $\operatorname{maxcut}(G) = 1-\varepsilon$. *Then $\beta(G) \le \varepsilon$, and hence $\beta_n \le 2\varepsilon$.*

Proof. Let S_* be an optimal cut of G, and take $S = V$, $L = S_*$ and $R = \bar{S}_*$, so that $E(S, \bar{S}) = \emptyset$. Since $|E(S_*)|+|E(\bar{S}_*)|+|E(S_*, \bar{S}_*)| = m$ and $|E(S_*, \bar{S}_*)| = (1-\varepsilon)m$, we have $|E(S_*)|+|E(\bar{S}_*)| = \varepsilon m$. Since $|V|_D = 2m$,

$$\beta(G) \le \beta(S_*, \bar{S}_*) = \frac{2|E(S_*)| + 2|E(\bar{S}_*)|}{|V|_D} = \frac{2\varepsilon m}{2m} = \varepsilon.$$

By Theorem 9.16, $\frac{1}{2}\beta_n \le \beta(G) \le \varepsilon$. □

By Corollary 9.17, Trevisan's rounding applied to such a graph finds disjoint L, R with $S = L \cup R \neq \emptyset$ and

$$\beta(L, R) = \frac{2|E(L)| + 2|E(R)| + |E(S, \bar{S})|}{|S|_D} \le \sqrt{2 \cdot 2\varepsilon} = 2\sqrt{\varepsilon}. \tag{9.10}$$

The analysis uses Theorem 9.16 and Corollary 9.17 for general graphs, not only for regular ones, because the recursion runs on $G[\bar{S}]$, which is in general not regular even when G is.

Next we bound the number of edges cut by the better of $(L \cup L', R \cup R')$ and $(L \cup R', R \cup L')$; merging the two partial cuts is easy. Since $S = L \cup R$ and $\bar{S} = L' \cup R'$,

$$|E(S, \bar{S})| = |E(L, L')| + |E(L, R')| + |E(R, L')| + |E(R, R')|.$$

The cut $(L \cup L', R \cup R')$ cuts the edges of $E(L, R)$, $E(L', R')$, $E(L, R')$ and $E(R, L')$; the cut $(L \cup R', R \cup L')$ cuts those of $E(L, R)$, $E(L', R')$, $E(L, L')$ and $E(R, R')$. Since

$$\max\{|E(L, L')| + |E(R, R')|,\ |E(L, R')| + |E(R, L')|\} \ge \frac{1}{2}|E(S, \bar{S})|,$$

the better of the two cuts cuts at least half of the edges of $E(S, \bar{S})$. The recursion cuts $|E(L', R')|$ edges of $G[\bar{S}]$, which is $|E(\bar{S})|\mathrm{ALG}(G[\bar{S}])$ when $E(\bar{S}) \neq \emptyset$ and 0 otherwise. Hence

$$m\mathrm{ALG}(G) \ge |E(L, R)| + \frac{1}{2}|E(S, \bar{S})| + |E(\bar{S})|\mathrm{ALG}(G[\bar{S}]), \tag{9.11}$$

where the last term is read as 0 when $E(\bar{S}) = \emptyset$.

Lemma 9.22. *Let $L, R \subseteq V$ be disjoint, let $S = L \cup R$ with $|S|_D > 0$, and let $\varepsilon \ge 0$. If $\beta(L, R) \le 2\sqrt{\varepsilon}$, then*

$$\frac{|E(L, R)| + \frac{1}{2}|E(S, \bar{S})|}{|E(L)| + |E(R)| + |E(L, R)| + |E(S, \bar{S})|} \ge 1 - 2\sqrt{\varepsilon}.$$

The denominator is $|E(S)| + |E(S, \bar{S})|$, the number of edges incident to S.

Proof. Write $a = |E(L)|+|E(R)|$, $b = |E(L,R)|$ and $c = |E(S,\bar{S})|$. Every edge of $E(S) = E(L)\cup E(R)\cup E(L,R)$ has two endpoints in S, and every edge of $E(S,\bar{S})$ has one, so $|S|_D = 2a+2b+c$. Hence $\beta(L,R) \le 2\sqrt{\varepsilon}$ means

$$\frac{2a+c}{2a+2b+c} = 1 - \frac{2b}{2a+2b+c} \le 2\sqrt{\varepsilon}, \qquad \text{or equivalently} \qquad \frac{b}{a+b+\frac{1}{2}c} \ge 1-2\sqrt{\varepsilon}.$$

The ratio $b/(a+b+\frac{1}{2}c)$ is at most 1, and adding the same nonnegative number $\frac{1}{2}c$ to the numerator and the denominator of a ratio at most 1 does not decrease it. Hence $(b+\frac{1}{2}c)/(a+b+c) \ge 1-2\sqrt{\varepsilon}$, which is the claim. □

Combining (9.11), (9.10) and Lemma 9.22,

$$m\mathrm{ALG}(G) \ge (1-2\sqrt{\varepsilon})(|E(S)| + |E(S,\bar{S})|) + |E(\bar{S})|\mathrm{ALG}(G[\bar{S}]). \tag{9.12}$$

It remains to see how large the max-cut of $G[\bar{S}]$ can be.

Proposition 9.23. *Let* $\mathrm{maxcut}(G) = 1-\varepsilon$*, let* $S \subseteq V$*, and assume that* $E(\bar{S}) \neq \emptyset$*. Then*

$$\mathrm{maxcut}(G[\bar{S}]) \ge 1 - \varepsilon\frac{m}{|E(\bar{S})|}.$$

Proof. An optimal cut of G cuts at most all of the edges of $E(S)$ and of $E(S,\bar{S})$, and, being a cut of $G[\bar{S}]$ when restricted to $\bar{S}$, at most $\mathrm{maxcut}(G[\bar{S}])\,|E(\bar{S})|$ edges of $E(\bar{S})$. So

$$(1-\varepsilon)m = \mathrm{maxcut}(G)\,m \le |E(S)| + |E(S,\bar{S})| + \mathrm{maxcut}(G[\bar{S}])\,|E(\bar{S})|.$$

Using $m = |E(S)| + |E(S,\bar{S})| + |E(\bar{S})|$,

$$\begin{aligned}
\mathrm{maxcut}(G[\bar{S}]) &\ge \frac{1}{|E(\bar{S})|}\Big[(1-\varepsilon)(|E(S)| + |E(S,\bar{S})| + |E(\bar{S})|) - |E(S)| - |E(S,\bar{S})|\Big] \\
&= \frac{1}{|E(\bar{S})|}\Big[(1-\varepsilon)|E(\bar{S})| - \varepsilon(|E(S)| + |E(S,\bar{S})|)\Big] \\
&= 1 - \varepsilon - \varepsilon\,\frac{|E(S)| + |E(S,\bar{S})|}{|E(\bar{S})|} = 1 - \varepsilon\frac{m}{|E(\bar{S})|}.
\end{aligned}$$

□

Proof of Theorem 9.19. For $n \ge 2$ and $\varepsilon \in [0,1]$, let $V_n(\varepsilon)$ be the largest value of $1-\mathrm{ALG}(G)$ over all graphs G with at most n vertices, at least one edge and $\mathrm{maxcut}(G) \ge 1-\varepsilon$, and over all choices of eigenvectors in the algorithm. The largest value exists, because $1-\mathrm{ALG}(G)$ is a fraction whose denominator is at most n^2. The theorem states that $V_n(\varepsilon) \le 4\sqrt{\varepsilon}$ for all n and ε. We prove this by induction on n; the argument for n uses the bound only for graphs with fewer vertices.

Let G have n vertices, at least one edge and $\mathrm{maxcut}(G) = 1-\varepsilon'$ with $\varepsilon' \le \varepsilon$, and consider a run of the algorithm on G. Removing isolated vertices changes neither the edges nor the max-cut, so Claim 9.21 and (9.12) apply with ε'. Let $\delta = |E(\bar{S})|/m \in [0,1]$, so that $|E(S)| + |E(S,\bar{S})| = (1-\delta)m$.

If $\delta = 0$, then (9.12) gives $\mathrm{ALG}(G) \ge 1 - 2\sqrt{\varepsilon'} \ge 1 - 4\sqrt{\varepsilon}$.

If $\delta > 0$, then $G[\bar{S}]$ has at least one edge and fewer vertices than G, because $S \neq \emptyset$, and by Proposition 9.23, $\mathrm{maxcut}(G[\bar{S}]) \ge 1 - \varepsilon'/\delta \ge 1 - \varepsilon/\delta$. If $\varepsilon/\delta \le 1$, the induction hypothesis gives

$1-\mathrm{ALG}(G[\bar{S}]) \le V_{n-1}(\varepsilon/\delta) \le 4\sqrt{\varepsilon/\delta}$; if $\varepsilon/\delta > 1$, the bound $1-\mathrm{ALG}(G[\bar{S}]) \le 1 < 4\sqrt{\varepsilon/\delta}$ holds trivially. Dividing (9.12) by m,

$$\begin{aligned} 1-\mathrm{ALG}(G) &\le 1-(1-2\sqrt{\varepsilon})(1-\delta)-\delta\mathrm{ALG}(G[\bar{S}]) \\ &= 2\sqrt{\varepsilon}(1-\delta)+\delta(1-\mathrm{ALG}(G[\bar{S}])) \\ &\le 2\sqrt{\varepsilon}(1-\delta)+\delta\cdot 4\sqrt{\frac{\varepsilon}{\delta}} = 2\sqrt{\varepsilon}(1-\delta)+4\sqrt{\delta}\sqrt{\varepsilon}, \end{aligned}$$

where the first line uses $\varepsilon' \le \varepsilon$. For the run at hand, and when $\varepsilon/\delta \le 1$, this is the recursion $1-\mathrm{ALG}(G) \le 2\sqrt{\varepsilon}(1-\delta)+\delta V_{n-1}(\varepsilon/\delta)$. Now

$$2\sqrt{\varepsilon}(1-\delta) = 2\sqrt{\varepsilon}(1+\sqrt{\delta})(1-\sqrt{\delta}) \le 4\sqrt{\varepsilon}(1-\sqrt{\delta}),$$

because $1+\sqrt{\delta} \le 2$. Therefore $1-\mathrm{ALG}(G) \le 4\sqrt{\varepsilon}(1-\sqrt{\delta})+4\sqrt{\delta}\sqrt{\varepsilon} = 4\sqrt{\varepsilon}$. Since G and the run were arbitrary, $V_n(\varepsilon) \le 4\sqrt{\varepsilon}$. □

Proof of Corollary 9.20. The greedy algorithm processes the vertices in any order and places each vertex on the side opposite to the majority of its already placed neighbors. Each edge is decided when its later endpoint is placed, and at least half of the edges decided at each step are cut, so it cuts at least $m/2$ edges. Let $\varepsilon = 1-\mathrm{maxcut}(G) < 1$. By Theorem 9.19, the better of the two cuts has value at least $\max\{\frac{1}{2}, 1-4\sqrt{\varepsilon}\}$, so its ratio to $\mathrm{maxcut}(G) = 1-\varepsilon$ is at least

$$\max\Big\{\frac{1}{2(1-\varepsilon)},\ \frac{1-4\sqrt{\varepsilon}}{1-\varepsilon}\Big\}.$$

The two terms are equal at $\varepsilon = 1/64$, where each is $32/63$. If $\varepsilon \ge 1/64$, the first term is at least $\frac{1}{2\cdot 63/64} = \frac{32}{63}$. If $\varepsilon \le 1/64$, put $s = \sqrt{\varepsilon} \le 1/8$; then $63(1-4s)-32(1-s^2) = 32s^2-252s+31 = (1-8s)(31-4s) \ge 0$, so the second term is at least $32/63$. □

Remark 9.24. Finer analyses give better ratios for variants of this algorithm: 0.531 in Trevisan's paper [Tre12], and 0.614 by Soto [Sot15].

10 The Arora–Rao–Vazirani Algorithm I

Lecture of September 22, 2021, by Pravesh K. Kothari. Based on the lecturer's handwritten notes.

The Leighton–Rao linear program approximates the sparsest cut within a factor $O(\log n)$, and Lecture 6 showed that this relaxation cannot do better. This lecture and the next present the algorithm of Arora, Rao and Vazirani [ARV09], which achieves $O(\sqrt{\log n})$ by rounding a semidefinite relaxation whose solutions define a metric that is also a squared Euclidean distance. After writing this relaxation, we examine two natural roundings: hyperplane rounding as in Lecture 4, which cuts short edges too often, and region growing as in Lecture 6, which may return a badly unbalanced cut. The remedy is the ARV structure theorem: the vectors contain two large sets that are far apart, and growing a region around one of them gives an $O(\sqrt{\log n})$-approximation when the sparsest cut is balanced. The structure theorem is proved in Lecture 11.

10.1 Sparsest cut as a quadratic program

Throughout this lecture, $G = (V, E)$ is an undirected graph with $V = [n]$, $n \geq 2$, and $m = |E|$ edges, all of unit capacity. For $S \subseteq V$ we write $\bar{S} = V \setminus S$ and $E(S, \bar{S})$ for the set of edges with exactly one endpoint in S. As in Lecture 6, the *sparsity* of a cut and of the graph are

$$\Phi(S) = \frac{|E(S,\bar{S})|}{|S| \cdot |\bar{S}|} \quad (\emptyset \neq S \subsetneq V), \qquad \Phi(G) = \min_{\emptyset \neq S \subsetneq V} \Phi(S). \tag{10.1}$$

A set S^* with $\Phi(S^*) = \Phi(G)$ is a *sparsest cut*.

We encode a set $S \subseteq V$ by the vector $x \in \{-1, 1\}^n$ with $x_i = 1$ exactly when $i \in S$. Then $\frac{1}{4}(x_i - x_j)^2$ equals 1 if i and j lie on different sides of the cut and 0 otherwise. Hence

$$\frac{1}{4} \sum_{\{i,j\} \in E} (x_i - x_j)^2 = |E(S, \bar{S})|, \qquad \frac{1}{4} \sum_{i,j \in [n]} (x_i - x_j)^2 = 2|S|(n - |S|),$$

where the second sum runs over all ordered pairs $(i, j) \in [n]^2$, so that each separated pair is counted twice. Throughout the lecture, sums over $i, j \in [n]$ are over ordered pairs. For $\beta > 0$, consider the quadratic program

$$\begin{aligned} \min \ & \frac{1}{4} \sum_{\{i,j\} \in E} (x_i - x_j)^2 \\ \text{s.t.} \ & \frac{1}{4} \sum_{i,j \in [n]} (x_i - x_j)^2 = \beta, \\ & x_i \in \{-1, 1\} \quad \text{for all } i \in [n]. \end{aligned} \tag{10.2}$$

It minimizes $|E(S, \bar{S})|$ over the sets S with $2|S|(n - |S|) = \beta$. Fix a sparsest cut S^* and let $\beta^* = 2|S^*|(n - |S^*|)$. Every feasible S for $\beta = \beta^*$ has $|E(S, \bar{S})| = \Phi(S)\beta^*/2 \geq \Phi(G)\beta^*/2$, with equality for $S = S^*$, so the optimum of (10.2) with $\beta = \beta^*$ is $\Phi(G)\beta^*/2$. We do not know β^*, but it is one of the numbers $2s(n - s)$ with $1 \leq s \leq \lfloor n/2 \rfloor$, so an algorithm can guess it by trying each of them.

For illustration we treat the *balanced case*, in which

$$\beta^* \geq \frac{n^2}{10}.$$

This says that the sparsest cut is roughly balanced, that is, $|S^*| = \Omega(n)$ and $n - |S^*| = \Omega(n)$. Precisely, for $s \le n/2$ we have $n/2 \le n - s \le n$, so $sn \le 2s(n-s) \le 2sn$; hence $\min\{|S^*|, n-|S^*|\} \ge n/10$ implies $\beta^* \ge n^2/10$, which in turn implies $\min\{|S^*|, n-|S^*|\} \ge n/20$. The balanced case is also the hardest one, and the general case can be reduced to it; we do not discuss the reduction here (see [ARV09]).

10.2 The semidefinite relaxation

We relax (10.2) by replacing each x_i with a vector $v_i \in \mathbb{R}^n$:

$$\begin{aligned} \min\ & \frac{1}{4} \sum_{\{i,j\} \in E} \|v_i - v_j\|_2^2 \\ \text{s.t.}\ & \|v_i\|_2^2 = 1 \quad \text{for all } i \in [n], \\ & \frac{1}{4} \sum_{i,j \in [n]} \|v_i - v_j\|_2^2 = \beta, \\ & \|v_i - v_k\|_2^2 \le \|v_i - v_j\|_2^2 + \|v_j - v_k\|_2^2 \quad \text{for all } i,j,k \in [n]. \end{aligned} \tag{10.3}$$

The last family of constraints is the *squared triangle inequality*. In terms of the Gram matrix $X = (\langle v_i, v_j \rangle)_{i,j}$, the objective and all constraints are linear in X, and vectors with a given Gram matrix exist exactly when $X \succeq 0$. So (10.3) is a semidefinite program, which can be solved in polynomial time up to an arbitrarily small additive error, as in Lecture 4; we ignore this error. We write $\mathrm{SDP}(\beta)$ for its optimum value.

The squared triangle inequality is a genuine restriction. Arbitrary vectors v_i, v_j, v_k satisfy the triangle inequality $\|v_i - v_k\|_2 \le \|v_i - v_j\|_2 + \|v_j - v_k\|_2$, but this does not imply the squared version, even for unit vectors. For instance, $v_1 = e_1$, $v_2 = (e_1 + e_2)/\sqrt{2}$ and $v_3 = e_2$ have $\|v_1 - v_3\|_2^2 = 2$, while $\|v_1 - v_2\|_2^2 + \|v_2 - v_3\|_2^2 = 2(2 - \sqrt{2}) < 2$.

Proposition 10.1. *Let S^* be a sparsest cut and $\beta^* = 2|S^*|(n - |S^*|)$. Let $v_i = e_1$ for $i \in S^*$ and $v_i = -e_1$ for $i \notin S^*$. Then $v_1, \dots, v_n$ is a feasible solution of (10.3) with $\beta = \beta^*$, and its objective value is $|E(S^*, \bar{S}^*)| = \Phi(G)\beta^*/2$. In particular, $\mathrm{SDP}(\beta^*) \le \Phi(G)\beta^*/2$.*

Proof. Let $x \in \{-1,1\}^n$ encode S^*. Then $v_i = x_i e_1$, so $\|v_i\|_2^2 = 1$ and $\|v_i - v_j\|_2^2 = (x_i - x_j)^2$ for all i, j. By the two identities before (10.2), the second constraint holds with $\beta = \beta^*$, and the objective value is $|E(S^*, \bar{S}^*)| = \Phi(S^*)|S^*||\bar{S}^*| = \Phi(G)\beta^*/2$.

It remains to check the squared triangle inequality, that is, $(a-b)^2 \le (a-c)^2 + (c-b)^2$ for all $a, b, c \in \{-1, 1\}$. This is trivial if $a = b$. Otherwise c differs from one of a and b, so the right side is at least $4 = (a-b)^2$. □

The following theorem is the goal of this lecture and the next. The constraint $\beta \ge n^2/10$ is the balanced case.

Theorem 10.2 (Arora, Rao and Vazirani [ARV09])**.** *There is an absolute constant $K > 0$ with the following property. Let $\beta \ge n^2/10$, and let $v_1, \dots, v_n \in \mathbb{R}^n$ be a feasible solution of (10.3) with objective value*

$$C = \frac{1}{4} \sum_{\{i,j\} \in E} \|v_i - v_j\|_2^2.$$

Then there is a randomized rounding algorithm that, given G and $v_1, \dots, v_n$, outputs in expected polynomial time a set $\emptyset \neq S \subsetneq V$ with

$$\frac{|E(S, \bar{S})|}{|S|(n - |S|)} \leq K\sqrt{\ln n} \cdot \frac{C}{\beta}.$$

Corollary 10.3. *Suppose that the sparsest cut is balanced, $\beta^* \geq n^2/10$. If the algorithm of Theorem 10.2 is applied to an optimal solution of (10.3) with $\beta = \beta^*$, it outputs a set S with $\Phi(S) \leq \frac{K}{2}\sqrt{\ln n} \cdot \Phi(G)$. Running it for every $\beta = 2s(n-s) \geq n^2/10$ with $1 \leq s \leq \lfloor n/2 \rfloor$ for which (10.3) is feasible, and returning the sparsest of the cuts found, gives an $O(\sqrt{\log n})$-approximation for the balanced case.*

Proof. By Proposition 10.1, $C = \mathrm{SDP}(\beta^*) \leq \Phi(G)\beta^*/2$, so Theorem 10.2 gives $\Phi(S) \leq K\sqrt{\ln n} \cdot C/\beta^* \leq \frac{K}{2}\sqrt{\ln n} \cdot \Phi(G)$. The value β^* is among those tried. □

The metric defined by the vectors. For a feasible solution of (10.3), define

$$d(i, j) = \frac{1}{4}\|v_i - v_j\|_2^2 \qquad (i, j \in [n]).$$

The squared triangle inequality says that $d(i, j) \leq d(i, k) + d(k, j)$ for all i, j, k. Thus d is a metric on $[n]$ in the sense of Lecture 6: distinct points may be at distance 0. Since the v_i are unit vectors, $0 \leq d(i, j) \leq 1$. In terms of d, the objective of (10.3) is $\sum_{\{i,j\} \in E} d(i, j)$ and the second constraint is $\sum_{i,j \in [n]} d(i, j) = \beta$. The metric d is of a special kind.

Definition 10.4. A metric d on $[n]$ is *squared Euclidean*, or *of negative type*, if there are vectors $u_1, \dots, u_n$ in some $\mathbb{R}^D$ with $d(i, j) = \|u_i - u_j\|_2^2$ for all $i, j \in [n]$.

The metric d above is of negative type, with $u_i = v_i/2$.

Remark 10.5. The semidefinite program refines the Leighton–Rao linear program [LR99] of Lecture 6, which corresponds to d being an arbitrary metric. Indeed, for a feasible solution of (10.3) we have $\sum_{i<j} d(i, j) = \beta/2$, so $y_{ij} = 2d(i, j)/\beta$ is a metric with $\sum_{i<j} y_{ij} = 1$, a feasible solution of the Leighton–Rao linear program with objective value $2C/\beta$. The semidefinite program asks in addition that d be of negative type and come from unit vectors.

10.3 Two rounding attempts

Throughout this subsection, $v_1, \dots, v_n$ is a feasible solution of (10.3) with objective value C, and d is its metric.

Hyperplane rounding. A first idea is to round as in the Goemans–Williamson algorithm for Max-Cut [GW95] (Lecture 4). Let $g \sim N(0, I_n)$ be a standard Gaussian vector, and set $x_i = \mathrm{sign}(\langle g, v_i \rangle)$ for every i. (With probability 1 no $\langle g, v_i \rangle$ is 0.)

Lemma 10.6 (Gaussian rounding)**.** *Let $u, w \in \mathbb{R}^n$ be unit vectors, let $\delta = \frac{1}{4}\|u - w\|_2^2$, and let $g \sim N(0, I_n)$. Then*

$$\Pr[\mathrm{sign}\langle g, u \rangle \neq \mathrm{sign}\langle g, w \rangle] = \frac{\arccos(1 - 2\delta)}{\pi} = \Theta(\|u - w\|_2) = \Theta(\sqrt{\delta}).$$

More precisely, the probability equals $\frac{2}{\pi}\arcsin\sqrt{\delta}$ and lies between $\frac{2}{\pi}\sqrt{\delta}$ and $\sqrt{\delta}$.

Proof. Since $\|u - w\|_2^2 = 2 - 2\langle u, w\rangle$, we have $\frac{1}{2} - \frac{1}{2}\langle u, w\rangle = \delta$, that is, $\langle u, w\rangle = 1 - 2\delta$. By the two-dimensional analysis of Lecture 4 (Fig. 16), the signs differ with probability θ/π, where $\theta = \arccos(1 - 2\delta) \in [0, \pi]$ is the angle between u and w. Since $\cos\theta = 1 - 2\sin^2(\theta/2)$, we get $\sin^2(\theta/2) = \delta$, so $\theta = 2\arcsin\sqrt{\delta}$. For $t \in [0, 1]$ we have $t \le \arcsin t \le \frac{\pi}{2}t$: the first inequality holds because $\sin s \le s$ for $s \ge 0$, and the second because arcsin is convex on $[0, 1]$ with $\arcsin 0 = 0$ and $\arcsin 1 = \pi/2$. Hence $\frac{2}{\pi}\sqrt{\delta} \le \theta/\pi \le \sqrt{\delta}$, and $\theta/\pi = \frac{2}{\pi}\sqrt{\delta}\,(1 + o(1))$ as $\delta \to 0$. Finally, $\sqrt{\delta} = \frac{1}{2}\|u - w\|_2$. □

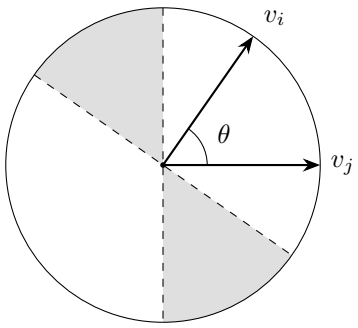


Figure 16: The plane spanned by v_i and v_j, which meet at angle θ. The projection of g onto this plane points in a uniformly random direction. The signs of $\langle g, v_i\rangle$ and $\langle g, v_j\rangle$ differ exactly when this direction lies in one of the two shaded wedges, which are bounded by the dashed lines perpendicular to v_i and v_j and have angle θ each.

Remark 10.7. Hyperplane rounding cuts short edges with too large a probability. The distance $d(i, j)$ is a number between 0 and 1, and it can be as small as, say, $(\ln n)^{-10}$. An edge $\{i, j\}$ contributes $d(i, j)$ to the objective C, but by Lemma 10.6 it is cut with probability at least $\frac{2}{\pi}\sqrt{d(i, j)}$. For $d(i, j) = (\ln n)^{-10}$ this is $\frac{2}{\pi}(\ln n)^{-5}$, while an analysis that loses only a factor $O(\sqrt{\log n})$ against C can afford a probability of about $\sqrt{\ln n} \cdot (\ln n)^{-10}$ for this edge, smaller by a factor of order $(\ln n)^{4.5}$.

Region growing. The second idea is region growing, as in Lecture 6. For a nonempty set $A \subseteq V$ and $i \in V$, let

$$d(i, A) = \min_{j \in A} d(i, j)$$

be the distance from i to the nearest point of A. The triangle inequality for d implies that

$$d(i, A) \le d(i, j) + d(j, A) \qquad \text{for all } i, j \in V, \tag{10.4}$$

since $d(i, A) \le d(i, k) \le d(i, j) + d(j, k)$ for a point $k \in A$ with $d(j, k) = d(j, A)$.

Lemma 10.8 (Region growing)**.** *Let $A \subseteq V$ be nonempty. Choose $r \in [0, 1]$ uniformly at random, and set*

$$S = \{k \in V : d(k, A) \le r\}.$$

Let $x \in \{0, 1\}^n$ be the indicator vector of S. Then $\Pr[x_i \ne x_j] \le d(i, j)$ for all $i, j \in V$. Consequently,

$$\mathbb{E}\,|E(S, \bar{S})| \le \sum_{\{i,j\} \in E} d(i, j) = C.$$

Proof. By symmetry we may assume $d(i,A) \geq d(j,A)$. By (10.4), $d(i,A) - d(j,A) \leq d(i,j)$. Both distances lie in $[0,1]$, so

$$\Pr[x_i \neq x_j] = \Pr\big[d(j,A) \leq r < d(i,A)\big] = d(i,A) - d(j,A) \leq d(i,j).$$

Summing over the edges, $\mathbb{E}\,|E(S,\bar{S})| = \sum_{\{i,j\}\in E} \Pr[x_i \neq x_j] \leq \sum_{\{i,j\}\in E} d(i,j)$, which is the objective value C. □

So region growing cuts few edges in expectation, at most the value of the semidefinite program. This does not yet give a sparse cut, because the cut may be very unbalanced: for all we know, $S = V$. To compete with $\Phi(G)$ we need $2|S|(n-|S|) \geq \beta/O(\sqrt{\log n})$. The set S is too large when most vertices i end up near the set A, that is, when $d(i,A)$ is small, and are therefore put into S. To avoid this, a significant fraction of the vertices must be far from the set A from which the region grows. The key lemma of Arora, Rao and Vazirani says that such a set A exists.

10.4 The structure theorem and the proof of the main theorem

Theorem 10.9 (ARV structure theorem; Arora, Rao and Vazirani [ARV09], Lee)**.** *There are absolute constants $a, b > 0$ with the following property. Let $n \geq 2$, and let $v_1, \ldots, v_n \in \mathbb{R}^n$ be unit vectors such that $d(i,j) = \frac{1}{4}\|v_i - v_j\|_2^2$ satisfies the triangle inequality $d(i,j) \leq d(i,k) + d(k,j)$ for all $i,j,k \in [n]$. Suppose that*

$$\sum_{i,j\in[n]} d(i,j) \geq \frac{n^2}{10} \qquad \textit{(the balanced case).}$$

Then there exist sets $A, B \subseteq [n]$ such that

1) *$|A| \geq an$ and $|B| \geq an$, and*

2) *$d(i,j) \geq \Delta$ for all $i \in A$ and $j \in B$, where $\Delta = b/\sqrt{\ln n}$.*

Moreover, such sets can be found by a randomized algorithm in expected polynomial time.

The theorem is proved in Lecture 11. It immediately gives the main theorem: we grow a region around A.

Proof of Theorem 10.2. Let $d(i,j) = \frac{1}{4}\|v_i - v_j\|_2^2$. By the constraints of (10.3), the v_i are unit vectors, d satisfies the triangle inequality, and $\sum_{i,j\in[n]} d(i,j) = \beta \geq n^2/10$. Hence Theorem 10.9 applies, and the algorithm first finds sets A and B as in that theorem. Since A and B are nonempty and $d \leq 1$, we have $\Delta \leq 1$. Moreover, $\beta \leq n^2/2$, because

$$\frac{1}{4}\sum_{i,j\in[n]} \|v_i - v_j\|_2^2 = \frac{1}{4}\sum_{i,j\in[n]} (2 - 2\langle v_i, v_j\rangle) = \frac{n^2}{2} - \frac{1}{2}\Big\|\sum_{i=1}^n v_i\Big\|_2^2.$$

Choose $r \in [0,1]$ uniformly at random and let $S_r = \{k \in V : d(k,A) \leq r\}$. By Lemma 10.8, $\mathbb{E}\,|E(S_r, \bar{S}_r)| \leq C$. If $r < \Delta$, then $A \subseteq S_r$, because $d(i,A) = 0$ for $i \in A$, and $B \cap S_r = \emptyset$, because $d(j,A) \geq \Delta > r$ for $j \in B$. In that case $|S_r|(n - |S_r|) \geq |A||B| \geq a^2n^2$. Since $\Pr[r < \Delta] = \Delta$, using $\beta \leq n^2/2$ in the last step we get

$$\mathbb{E}\big[|S_r|(n-|S_r|)\big] \geq \Delta a^2 n^2 = \frac{a^2 b\, n^2}{\sqrt{\ln n}} \geq \frac{2a^2 b\, \beta}{\sqrt{\ln n}},$$

which is $\beta/O(\sqrt{\log n})$. Hence

$$\frac{\mathbb{E}\,|E(S_r, \bar{S}_r)|}{\mathbb{E}\big[|S_r|(n - |S_r|)\big]} \leq t := \frac{\sqrt{\ln n}}{2a^2 b} \cdot \frac{C}{\beta}.$$

It remains to pass from this ratio of expectations to a single cut. Let $X_r = |E(S_r, \bar{S}_r)| - t|S_r|(n - |S_r|)$, so that $\mathbb{E}X_r \leq 0$. If $S_r = \emptyset$ or $S_r = V$, then $X_r = 0$. Suppose that $X_r > 0$ whenever $\emptyset \neq S_r \subsetneq V$. For $r < \Delta$ the set S_r contains A and misses B, so this case has probability at least $\Delta > 0$, and then $\mathbb{E}X_r > 0$, a contradiction. Hence some r gives $\emptyset \neq S_r \subsetneq V$ with $X_r \leq 0$, that is, $\Phi(S_r) \leq t$. The set S_r changes only at the n values $r = d(k, A)$, so the algorithm computes the sets $S_{d(k,A)}$ for $k \in V$ and outputs the one of smallest sparsity among those that are nonempty proper subsets of V. This proves the theorem with $K = 1/(2a^2 b)$. The running time is that of finding A and B plus a polynomial amount of work. □

The proof shows that the cut found satisfies $\Phi(S) \leq O(\sqrt{\log n}) \cdot C/\beta$, which is at most $O(\sqrt{\log n}) \cdot \Phi(G)$ when the semidefinite program is solved with $\beta = \beta^*$ (Corollary 10.3). Everything therefore rests on Theorem 10.9.

11 The Arora–Rao–Vazirani Algorithm II

Lecture of September 24, 2021, by Pravesh K. Kothari. Based on the lecturer's handwritten notes.

Lecture 10 reduced the $O(\sqrt{\log n})$-approximation for the balanced sparsest cut problem to the ARV structure theorem, which this lecture proves. The theorem has nothing to do with the graph: it asserts that a squared Euclidean metric on unit vectors with large average distance contains two sets of linear size at distance $\Omega(1/\sqrt{\log n})$ from each other. We find such sets by projecting the vectors onto a random Gaussian direction, keeping the points with very negative and very positive projections, and removing a greedily chosen matching of close pairs; either the remaining sets are large, or the matching is large. A chaining argument then shows that the matching cannot be large: it links matching edges into long paths, so that expected maxima over neighborhoods grow, and compares this growth with the $O(\sqrt{\log n})$ bound for the maximum of Gaussian variables. This argument, based on expected maxima over k-neighborhoods, is the one in Rothvoss's lecture notes [Rot16].

11.1 The structure theorem and vertex separators

Throughout this lecture, $n \geq 2$, and $v_1, \ldots, v_n \in \mathbb{R}^n$ are unit vectors such that

$$d(i,j) = \frac{1}{4}\|v_i - v_j\|_2^2 \qquad (i,j \in [n])$$

satisfies the triangle inequality $d(i,j) \leq d(i,k) + d(k,j)$ for all $i,j,k \in [n]$. Then d is a metric of negative type on $[n]$ with values in $[0,1]$ (Lecture 10). Sums over $i,j \in [n]$ are over ordered pairs. We restate the theorem to be proved.

Theorem 11.1 (ARV structure theorem; Arora, Rao and Vazirani [ARV09], Lee)**.** *There are absolute constants $a, b > 0$ with the following property. Let $n \geq 2$, and let $v_1, \ldots, v_n \in \mathbb{R}^n$ be unit vectors such that $d(i,j) = \frac{1}{4}\|v_i - v_j\|_2^2$ satisfies the triangle inequality. Suppose that $\sum_{i,j\in[n]} d(i,j) \geq n^2/10$. Then there exist sets $A, B \subseteq [n]$ with $|A| \geq an$, $|B| \geq an$, and $d(i,j) \geq b/\sqrt{\ln n}$ for all $i \in A$ and $j \in B$. Moreover, such sets can be found by a randomized algorithm in expected polynomial time.*

The theorem is a statement about well-separated sets in a squared Euclidean metric space in which the average distance is large. We rephrase it in terms of a graph on the points.

Definition 11.2. For $\Delta > 0$, let $H = H_\Delta$ be the directed graph with vertex set $[n]$ and edge set

$$E(H) = \{(i,j) \in [n]^2 : d(i,j) \leq \Delta\}.$$

Since d is symmetric and $d(i,i) = 0$, the edge set $E(H)$ is symmetric and contains every loop (i,i).

Definition 11.3. Sets $A, B \subseteq [n]$ form a *vertex separator* (A,B) in H if no edge of H goes from A to B, that is, $E(H) \cap (A \times B) = \emptyset$. For a constant $a > 0$, a vertex separator is *a-good* if $|A| \cdot |B| \geq an^2$.

A pair (A,B) is a vertex separator in H_Δ exactly when $d(i,j) > \Delta$ for all $i \in A$ and $j \in B$. If it is a-good, then $|A| \geq an$ and $|B| \geq an$, because both sets have at most n elements. So Theorem 11.1 says that H_Δ has a good vertex separator for some $\Delta = \Theta(1/\sqrt{\log n})$.

11.2 A randomized algorithm for vertex separators

Fix $\Delta > 0$ and let $H = H_\Delta$.

Algorithm 11.4 (Vertex separator)**.** Input: the vectors $v_1, \dots, v_n$ and $\Delta > 0$.

1) Sample $g \sim N(0, I_n)$ and set $y_i = \langle g, v_i \rangle$ for every $i \in [n]$.

2) Let $A^0 = \{i : y_i \le -1\}$ and $B^0 = \{j : y_j \ge 1\}$.

3) Find a maximal matching M in the set of edges $E(H) \cap (A^0 \times B^0)$ greedily: start with $M = \emptyset$, process the pairs $(i, j) \in [n] \times [n]$ in lexicographic order, and add (i, j) to M if it lies in $E(H) \cap (A^0 \times B^0)$ and neither i nor j is an endpoint of an edge already in M.

4) Let $V(M)$ be the set of endpoints of edges of M. Output $A = A^0 \backslash V(M)$ and $B = B^0 \backslash V(M)$.

Each y_i is a standard Gaussian variable, so $\mathbb{E} y_i^2 = 1$, and $y_i - y_j = \langle g, v_i - v_j \rangle$, so

$$\mathbb{E}(y_i - y_j)^2 = \|v_i - v_j\|_2^2 = 4\, d(i, j).$$

The sets A^0, B^0, A, B and the matching M are random variables, since they are functions of the Gaussian vector g. The sets A^0 and B^0 are disjoint, and the edges of M are directed from A^0 to B^0. In particular,

$$(i, j) \in M \quad \text{implies} \quad y_j - y_i \ge 2. \tag{11.1}$$

We call i a *tail* of M if i has an outgoing edge in M, and a *head* of M if it has an incoming edge in M. Since M is a matching and $A^0 \cap B^0 = \emptyset$, no vertex is both.

Proposition 11.5. *For every outcome of g, the pair (A, B) is a vertex separator in H.*

Proof. Suppose that $(i, j) \in E(H) \cap (A \times B)$. Then (i, j) lies in $E(H) \cap (A^0 \times B^0)$, and neither i nor j is an endpoint of an edge of M. We could add (i, j) to M, so M would not be maximal, a contradiction. □

The matching has a useful symmetry. Write $M(g)$ for the matching computed from g, and $\mathrm{rev}(M)$ for the set of reversed edges $\{(j, i) : (i, j) \in M\}$.

Lemma 11.6. *For every g, $M(-g) = \mathrm{rev}(M(g))$. Consequently, for every matching μ we have $\Pr[M = \mu] = \Pr[M = \mathrm{rev}(\mu)]$, and for every $i \in [n]$,*

$$\Pr[i \text{ is a head of } M] = \Pr[i \text{ is a tail of } M].$$

Proof. Replacing g by $-g$ replaces y by $-y$, which exchanges A^0 and B^0. Since $E(H)$ is symmetric, the edge set $F = E(H) \cap (A^0 \times B^0)$ is replaced by $\mathrm{rev}(F)$. The processing order of the greedy algorithm is fixed in advance and does not depend on y. However, the lexicographic order on $\mathrm{rev}(F)$ corresponds, on F, to the order that compares heads first and tails second. So we must show that the greedy algorithm on F gives the same matching in the tail-first order (pairs (i, j) compared lexicographically) and in the head-first order (pairs compared by (j, i)). We prove this by induction on $|F|$, for every set F of pairs from a set of tails to a disjoint set of heads; for $F = \emptyset$ both matchings are empty. Otherwise, let t be the smallest tail of an edge of F, and let u be the smallest head of an edge of F at t. In the tail-first order, (t, u) comes first, so it is taken. In the head-first order, an edge processed before (t, u) has a head $u' < u$ (an edge with

head u and a tail smaller than t does not exist, by the choice of t), so it is not incident to u, and it is not incident to t either, because every edge at t has its head at least u. Hence t and u are still free when (t,u) is processed, and (t,u) is taken in both orders. Let F' be the set of edges of F not incident to t or u. In both orders, the edges of $F \setminus F'$ other than (t,u) are rejected, and the edges of F' are processed in the same relative order and with the same outcome as when the algorithm runs on F' alone, because their outcome does not depend on t and u. Both matchings are therefore $\{(t,u)\}$ together with the matching on F' in the respective order, and these agree by induction. Hence $M(-g) = \mathrm{rev}(M(g))$.

Since g and $-g$ have the same distribution, $\Pr[M(g) = \mu] = \Pr[M(-g) = \mu] = \Pr[M(g) = \mathrm{rev}(\mu)]$. A vertex i is a head of μ exactly when it is a tail of $\mathrm{rev}(\mu)$; summing over μ gives the last claim. □

The next lemma says that the algorithm either finds a good vertex separator in expectation or produces a large matching in expectation. It rests on the following estimate, which is similar to the analysis of hyperplane rounding in Lecture 4.

Homework 11.7. Show that there is an absolute constant $c > 0$ such that for all $i, j \in [n]$ with $d(i,j) \geq 1/20$,

$$\Pr[y_i \leq -1,\ y_j \geq 1] \geq c \cdot d(i,j).$$

Remark 11.8. Some lower bound on $d(i,j)$ is necessary in Homework 11.7. The event requires $y_j - y_i \geq 2$, and $y_j - y_i$ is Gaussian with mean 0 and variance $4d(i,j)$. So the probability is at most $\Pr[N(0,1) \geq 1/\sqrt{d(i,j)}] \leq e^{-1/(2d(i,j))}$, which is not $\Omega(d(i,j))$ as $d(i,j) \to 0$. The pair (y_i, y_j) is Gaussian with unit variances and correlation $1 - 2d(i,j)$, so the probability depends only on $d(i,j)$; numerically, at $d(i,j) = 1/20$ it is about $1.5 \cdot 10^{-7}$.

Lemma 11.9 (Vertex separator or large matching)**.** *There is an absolute constant $c' > 0$ with the following property. Suppose that $\sum_{i,j \in [n]} d(i,j) \geq n^2/10$, and let $\Delta > 0$. If $\mathbb{E}\,|A||B| < c'n^2$, then $\mathbb{E}\,|M| > c'n$.*

Proof. Let c be the constant of Homework 11.7; we show that $c' = c/40$ works. Let P be the set of pairs $(i,j) \in [n]^2$ with $d(i,j) \geq 1/20$. There are at most n^2 pairs outside P, each with $d(i,j) < 1/20$, so together they contribute at most $n^2/20$ to $\sum_{i,j} d(i,j)$, and therefore $\sum_{(i,j)\in P} d(i,j) \geq n^2/10 - n^2/20 = n^2/20$. Hence

$$\mathbb{E}\,|A^0||B^0| = \sum_{i,j\in[n]} \Pr[y_i \leq -1,\ y_j \geq 1] \geq \sum_{(i,j)\in P} c \cdot d(i,j) \geq \frac{c\,n^2}{20} = 2c'n^2.$$

Every edge of M has one endpoint in A^0 and one in B^0, so $|A| = |A^0| - |M|$ and $|B| = |B^0| - |M|$. Since $|A^0| + |B^0| \leq n$,

$$|A||B| = |A^0||B^0| - |M|(|A^0| + |B^0|) + |M|^2 \geq |A^0||B^0| - n|M|.$$

Taking expectations, $\mathbb{E}\,|A||B| \geq \mathbb{E}\,|A^0||B^0| - n\,\mathbb{E}\,|M| \geq 2c'n^2 - n\,\mathbb{E}\,|M|$. So if $\mathbb{E}\,|A||B| < c'n^2$, then $n\,\mathbb{E}\,|M| > c'n^2$, that is, $\mathbb{E}\,|M| > c'n$. □

We will analyze $\mathbb{E}\,|M|$ and prove that it is small when Δ is small.

11.3 The main technical lemma

Lemma 11.10 (Main technical lemma)**.** *There is an absolute constant $c_0 > 0$ such that, for every $\Delta > 0$,*

$$\frac{c_0}{\Delta}\Big(\frac{\mathbb{E}\,|M|}{n}\Big)^3 \le \mathbb{E}\max_{i,j\in[n]}(y_j - y_i) \le 4\sqrt{\ln(2n^2)}.$$

In words, we analyze the expectation of the maximum of $\{y_j - y_i : i, j \in [n]\}$. By a standard bound, stated next, this expectation is $O(\sqrt{\log n})$. On the other hand, we will show that if there is a large matching, then this expected maximum is large. The variables $y_j - y_i$ are jointly Gaussian, so the maximum is the maximum of a *Gaussian process*.

Fact 11.11. *Let $Z_1, \dots, Z_t$ be jointly Gaussian random variables with $\mathbb{E}Z_i = 0$ and $\mathbb{E}Z_i^2 \le 1$ for all i. Then*

$$\mathbb{E}\max_{i\le t} Z_i \le 2\sqrt{\ln(2t)}.$$

Proof. Let $\lambda > 0$. By Jensen's inequality and since $\mathbb{E}e^{\lambda Z_i} = e^{\lambda^2 \mathbb{E}Z_i^2/2} \le e^{\lambda^2/2}$,

$$\mathbb{E}\max_{i\le t} Z_i \le \frac{1}{\lambda}\ln \mathbb{E}\, e^{\lambda \max_i Z_i} \le \frac{1}{\lambda}\ln \sum_{i=1}^{t} \mathbb{E}\, e^{\lambda Z_i} \le \frac{\ln t}{\lambda} + \frac{\lambda}{2}.$$

With $\lambda = \sqrt{2\ln(2t)}$ and $\ln t \le \ln(2t)$, the right side is at most $\sqrt{2\ln(2t)} \le 2\sqrt{\ln(2t)}$. □

Proof of the upper bound in Lemma 11.10. We apply Fact 11.11 to the n^2 jointly Gaussian variables $y_j - y_i$, which have mean 0, after scaling them by $\frac{1}{2}$ so that their variances $d(i,j)$ are at most 1: with $Z_{ij} = (y_j - y_i)/2$ we get $\mathbb{E}\max_{i,j}(y_j - y_i) = 2\,\mathbb{E}\max_{i,j} Z_{ij} \le 4\sqrt{\ln(2n^2)}$. □

The lower bound is proved in the next subsection. We first derive the structure theorem.

Proof of Theorem 11.1. Let c' be the constant of Lemma 11.9 and c_0 that of Lemma 11.10. Rearranging Lemma 11.10 gives, for every $\Delta > 0$,

$$\mathbb{E}\,|M| \le n\Big(\frac{4\sqrt{\ln(2n^2)}\cdot\Delta}{c_0}\Big)^{1/3}.$$

Choose $\Delta = c_0 c'^3/(4\sqrt{\ln(2n^2)})$. Then the right side equals $c'n$, so $\mathbb{E}\,|M| \le c'n$, and Lemma 11.9 gives $\mathbb{E}\,|A||B| \ge c'n^2$. Hence some outcome of g gives sets with $|A||B| \ge c'n^2$; by Proposition 11.5, (A, B) is a c'-good vertex separator in H_Δ. So $|A|, |B| \ge c'n$, and $d(i,j) > \Delta$ for all $i \in A$ and $j \in B$. Since $\ln(2n^2) \le 3\ln n$ for $n \ge 2$, we have $\Delta \ge b/\sqrt{\ln n}$ with $b = c_0 c'^3/(4\sqrt{3})$.

For the algorithm, note that A and B are disjoint, so $|A||B| \le n^2/4$. Let q be the probability that $|A||B| \ge c'n^2/2$. Then $c'n^2 \le \mathbb{E}\,|A||B| \le qn^2/4 + c'n^2/2$, so $q \ge 2c'$. Repeating Algorithm 11.4 with independent Gaussian vectors until $|A||B| \ge c'n^2/2$ takes at most $1/(2c')$ runs in expectation, and each run takes polynomial time. The output satisfies $|A|, |B| \ge c'n/2$. The theorem therefore holds with $a = c'/2$ and $b = c_0 c'^3/(4\sqrt{3})$. □

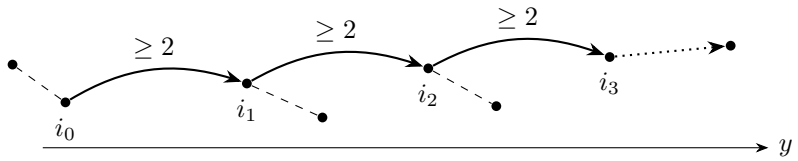


Figure 17: Chaining, as a heuristic picture. Each solid arrow is an edge of H along which y increases by at least 2, as it does along an edge of M, so the endpoints of a path of k such arrows differ by at least $2k$ in y. A single matching contains no two arrows with a common vertex; the argument chains matching edges on average over g. Dashed: other edges of H at the vertices of the path. Horizontal position indicates the value of y.

11.4 Chaining

It remains to prove the lower bound in Lemma 11.10. The idea is that a large matching forces some difference $y_j - y_i$ to be large. A matching edge $i \to j$ has $y_j - y_i \geq 2$ by (11.1). We chain such edges together through H to form long paths, as in Fig. 17; the endpoints of a long path are then very far apart in y. We run this argument on average over all starting points.

Definition 11.12. For $k \geq 0$ and $i \in [n]$, let $H^k(i)$ be the set of vertices reachable from i in at most k steps in H, and let

$$Z_i^{(k)} = \max_{j \in H^k(i)} (y_j - y_i), \qquad \Psi(k) = \sum_{i=1}^{n} \mathbb{E} Z_i^{(k)}.$$

Since $i \in H^k(i)$, we have $Z_i^{(k)} \geq 0$; since $H^0(i) = \{i\}$, we have $Z_i^{(0)} = 0$ and $\Psi(0) = 0$. Each $Z_i^{(k)}$ is at most $\max_{i,j \in [n]}(y_j - y_i)$, so

$$\frac{\Psi(k)}{n} \leq \mathbb{E} \max_{i,j \in [n]} (y_j - y_i) \qquad \text{for every } k \geq 0. \tag{11.2}$$

Our goal is to prove that $\Psi(k)$ is large if $\mathbb{E}\,|M|$ is large.

Lemma 11.13 (Chaining lemma)**.** *There is an absolute constant $C \geq 0$ such that, for every $\Delta > 0$ and every integer $k \geq 0$,*

$$\Psi(k+1) \geq \Psi(k) + 4\,\mathbb{E}\,|M| - Cn \max_{i \in [n],\, j \in H^{k+1}(i)} \big(\mathbb{E}(y_j - y_i)^2\big)^{1/2}.$$

The term $4\,\mathbb{E}\,|M|$ is large if M is large on average; the last term is an error term, controlled by the variances of the differences $y_j - y_i$ within distance $k+1$ in H.

Proof of the lower bound in Lemma 11.10, assuming Lemma 11.13. Let $j \in H^{k+1}(i)$, and let $i = i_0, i_1, \ldots, i_l = j$ with $l \leq k+1$ be a path in H. Each step has $d(i_{s-1}, i_s) \leq \Delta$, so the triangle inequality for d (the squared triangle inequality for the vectors) gives $d(i,j) \leq (k+1)\Delta$. Hence

$$\mathbb{E}(y_i - y_j)^2 = 4\,d(i,j) \leq 4(k+1)\Delta,$$

and Lemma 11.13 gives

$$\Psi(k+1) \geq \Psi(k) + 4\,\mathbb{E}\,|M| - 2Cn\sqrt{(k+1)\Delta}.$$

We may assume $C > 0$, by increasing C if necessary. Write $\mu = \mathbb{E}\,|M|/n$, and let

$$k_0 = \frac{1}{4C^2} \cdot \frac{\mu^2}{\Delta}.$$

For every integer $k \geq 0$ with $k+1 \leq k_0$ we have $2Cn\sqrt{(k+1)\Delta} \leq 2Cn\sqrt{k_0\Delta} = n\mu = \mathbb{E}\,|M|$, so $\Psi(k+1) \geq \Psi(k) + 3\,\mathbb{E}\,|M|$.

Suppose first that $k_0 \geq 1$, and let $K = \lfloor k_0 \rfloor$, so that $K \geq k_0/2$. Since $\Psi(0) = 0$, applying this for $k = 0, 1, \ldots, K-1$ gives $\Psi(K) \geq 3K\,\mathbb{E}\,|M|$, and so, by (11.2),

$$\mathbb{E} \max_{i,j\in[n]} (y_j - y_i) \geq \frac{\Psi(K)}{n} \geq \frac{3}{2} k_0 \mu = \frac{3}{8C^2} \cdot \frac{\mu^3}{\Delta} \geq \frac{1}{8C^2} \cdot \frac{\mu^3}{\Delta}.$$

Suppose next that $k_0 < 1$. Then $\frac{1}{8C^2}\mu^3/\Delta = k_0\mu/2 < \mu/2$. On the other hand, $|M| \leq n/2$, so $\Pr[M \neq \emptyset] \geq 2\mathbb{E}\,|M|/n = 2\mu$; when $M \neq \emptyset$ the maximum of $y_j - y_i$ is at least 2 by (11.1), and it is never negative. Hence $\mathbb{E} \max_{i,j}(y_j - y_i) \geq 4\mu \geq \frac{1}{8C^2}\mu^3/\Delta$. In both cases the lower bound in Lemma 11.10 holds with $c_0 = 1/(8C^2)$. □

The proof of the chaining lemma uses a concentration property of the maximum of Gaussian variables.

Fact 11.14 (Borell [Bor75]; Sudakov and Tsirelson [ST78])**.** *There is an absolute constant C_B such that, for all jointly Gaussian random variables $Z_1, \ldots, Z_t$ with mean 0,*

$$\mathrm{Var}\big[\max\{Z_1, \ldots, Z_t\}\big] \leq C_B \cdot \max\{\mathrm{Var}(Z_1), \ldots, \mathrm{Var}(Z_t)\}.$$

This is a consequence of Borell's inequality, which can be proved by concentration of Lipschitz functions of Gaussian vectors; we do not prove it here.

Proof of Lemma 11.13. Fix $k \geq 0$, write $Z_i = Z_i^{(k)}$ and $Z_i' = Z_i^{(k+1)}$, and let

$$\sigma = \max_{i\in[n],\, j\in H^{k+1}(i)} \big(\mathbb{E}(y_j - y_i)^2\big)^{1/2}.$$

Step 1: inequalities for a fixed outcome. For every edge $(i,j) \in E(H)$,

$$Z_i' \geq Z_j + y_j - y_i. \tag{11.3}$$

Indeed, for every $r \in H^k(j)$ there is a path of length at most $k+1$ from i to r, through j, so $r \in H^{k+1}(i)$ and $Z_i' \geq y_r - y_i = (y_r - y_j) + (y_j - y_i)$; maximizing over r gives (11.3). Applying (11.3) to the edges of M and using (11.1),

$$Z_i' \geq Z_j + 2 \qquad \text{for every } (i,j) \in M.$$

For the vertices not covered by M we use the *fluff inequalities*

$$\tfrac{1}{2}Z_i' \geq \tfrac{1}{2}Z_i \qquad \text{for every } i \notin V(M),$$

which hold because $H^k(i) \subseteq H^{k+1}(i)$. Equivalently, we may pair up the uncovered vertices by an arbitrary matching N of *fluff edges* and add the two inequalities of each pair, obtaining

$\frac{1}{2}Z'_i + \frac{1}{2}Z'_j \geq \frac{1}{2}Z_i + \frac{1}{2}Z_j$ for $(i,j) \in N$; when the number of uncovered vertices is odd, one of them keeps its own inequality. Define

$$L_i = \begin{cases} 1 & \text{if } i \text{ is a tail of } M, \\ \frac{1}{2} & \text{if } i \notin V(M), \\ 0 & \text{if } i \text{ is a head of } M, \end{cases} \qquad R_i = \begin{cases} 1 & \text{if } i \text{ is a head of } M, \\ \frac{1}{2} & \text{if } i \notin V(M), \\ 0 & \text{if } i \text{ is a tail of } M. \end{cases}$$

Adding all the inequalities above gives

$$\sum_{i=1}^{n} L_i Z'_i \geq \sum_{j=1}^{n} R_j Z_j + 2|M|,$$

and taking expectations,

$$\sum_{i=1}^{n} \mathbb{E}[L_i Z'_i] \geq \sum_{j=1}^{n} \mathbb{E}[R_j Z_j] + 2\,\mathbb{E}\,|M|. \tag{11.4}$$

Step 2: removing the weights. We would like to replace $\mathbb{E}[L_i Z'_i]$ by $\mathbb{E}L_i \cdot \mathbb{E}Z'_i$, and $\mathbb{E}[R_j Z_j]$ by $\mathbb{E}R_j \cdot \mathbb{E}Z_j$. By Lemma 11.6, $\Pr[i \text{ is a head}] = \Pr[i \text{ is a tail}]$, so

$$\mathbb{E}L_i = \Pr[i \text{ is a tail}] + \tfrac{1}{2}(1 - \Pr[i \text{ is a tail}] - \Pr[i \text{ is a head}]) = \tfrac{1}{2},$$

and likewise $\mathbb{E}R_i = \frac{1}{2}$. The error incurred by the replacement is a covariance, which we bound by the Cauchy–Schwarz inequality:

$$\left|\mathbb{E}[L_i Z'_i] - \mathbb{E}Z'_i \cdot \mathbb{E}L_i\right| = \left|\mathbb{E}\left[(Z'_i - \mathbb{E}Z'_i)(L_i - \mathbb{E}L_i)\right]\right| \leq \sqrt{\mathbb{E}(Z'_i - \mathbb{E}Z'_i)^2} \cdot \sqrt{\mathbb{E}(L_i - \mathbb{E}L_i)^2}.$$

Since $L_i \in [0,1]$, the second factor is at most 1. The variable Z'_i is the maximum of the jointly Gaussian variables $y_j - y_i$ with $j \in H^{k+1}(i)$, which have mean 0. By Fact 11.14,

$$\mathbb{E}(Z'_i - \mathbb{E}Z'_i)^2 \leq C_B \max_{j \in H^{k+1}(i)} \mathbb{E}(y_j - y_i)^2 \leq C_B \sigma^2.$$

Hence $|\mathbb{E}[L_i Z'_i] - \frac{1}{2}\mathbb{E}Z'_i| \leq \sqrt{C_B}\,\sigma$. Similarly, since $H^k(j) \subseteq H^{k+1}(j)$, $|\mathbb{E}[R_j Z_j] - \frac{1}{2}\mathbb{E}Z_j| \leq \sqrt{C_B}\,\sigma$.

Step 3: conclusion. Inserting these bounds into (11.4) gives

$$\frac{1}{2}\Psi(k+1) + n\sqrt{C_B}\,\sigma \geq \frac{1}{2}\Psi(k) - n\sqrt{C_B}\,\sigma + 2\,\mathbb{E}\,|M|,$$

that is, $\Psi(k+1) \geq \Psi(k) + 4\,\mathbb{E}\,|M| - 4\sqrt{C_B}\,n\sigma$. This is the claim with $C = 4\sqrt{C_B}$. □

With the chaining lemma, the proof of Lemma 11.10, and hence of the structure theorem Theorem 11.1, is complete.

12 Local Search Algorithms for Facility Location

Lecture of September 27, 2021, by Anupam Gupta. Based on the lecturer's handwritten notes.

The algorithms of the earlier lectures relaxed a problem to a linear or semidefinite program and rounded the solution (Lectures 2 and 4), ran a greedy algorithm (Lecture 1), grew regions and embedded metrics (Lectures 6 and 7), or used eigenvalues (Lectures 8 and 9). This lecture and Lectures 13 and 14 introduce three further techniques (local search, LP rounding beyond basic randomized rounding, and primal-dual algorithms) on two related clustering problems, facility location and k-median. Here we study local search for k-median, which starts from any solution and exchanges an open facility for a closed one as long as this lowers the cost. Following Gupta and Tangwongsan [GT08] (see also Section 9.2 of [WS11]), we prove the result of Arya et al. [AGK+04] that every solution which no single exchange improves costs at most five times the optimum. We also give an instance on which such a solution costs almost three times the optimum, and we show how to make the algorithm run in polynomial time.

12.1 Facility location and k-median

Both problems ask where to open facilities so that every client is close to an open facility, as in Fig. 18. They share the following data.

Definition 12.1. An *instance* consists of a finite metric space (V, d) with $n = |V|$ points, a nonempty set $C \subseteq V$ of *clients*, and a *facility cost* $f_i \geq 0$ for each $i \in V$. For $j \in V$ and a nonempty set $F \subseteq V$, we write $d(j, F) := \min_{i \in F} d(j, i)$.

Definition 12.2. In the *facility location problem*, we choose a nonempty set $F \subseteq V$ of facilities to open, and we minimize

$$\text{cost}(F) := \sum_{i \in F} f_i + \sum_{j \in C} d(j, F).$$

In the *k-median problem*, there are no facility costs. For an integer $1 \leq k \leq n$, we choose $F \subseteq V$ with $|F| = k$, and we minimize

$$\text{cost}(F) := \sum_{j \in C} d(j, F).$$

In both problems, OPT denotes the optimal value.

In both problems, each client is served by a nearest open facility, and $d(j, F)$ is the *connection cost* of client j. The two problems are closely related; Lecture 14 explains how. If $d(j, F)$ is replaced by $d(j, F)^2$ in the k-median objective, we obtain the commonly studied *k-means* problem.

12.2 Local search

Local search starts from a feasible solution and repeatedly makes a small change, called a *move*, as long as the change lowers the cost.

Algorithm 12.3 (Local search)**.** Start with any feasible solution, preferably a reasonable one (Section 12.5 makes this precise). While some move produces a solution of strictly smaller cost, perform such a move, or a move that lowers the cost the most. Output the final solution.

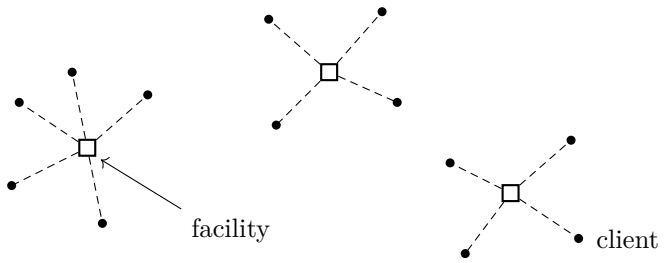


Figure 18: A solution to a facility location or k-median instance: three open facilities (squares), and each client (dot) connected to a nearest open facility.

For k-median, the natural moves exchange open facilities for closed ones.

Definition 12.4. Let $F \subseteq V$ with $|F| = k$. For $u \in F$ and $v \in V \setminus F$, the *swap* (u, v) moves from F to

$$F - u + v \coloneqq (F \setminus \{u\}) \cup \{v\}.$$

More generally, for an integer $p \geq 1$, a *p-swap* moves from F to $(F \setminus U) \cup W$, where $U \subseteq F$ and $W \subseteq V \setminus F$ satisfy $|U| = |W| \leq p$. The set F is a *local optimum* with respect to swaps (respectively, p-swaps) if no swap (respectively, p-swap) produces a set of strictly smaller cost.

Swaps are exactly the 1-swaps. We care about constant values of p: there are $n^{O(p)}$ p-swaps, so for $p = O(1)$ an improving p-swap, or the best one, can be found in polynomial time. The ratio between the cost of the worst local optimum and OPT, maximized over all instances, is the *locality gap* of the moves.

Theorem 12.5 (Arya et al. [AGK+04])**.** *Consider an instance of k-median.*

1) Every local optimum with respect to swaps has cost at most $5 \cdot \mathrm{OPT}$.

2) For every integer $p \geq 1$, every local optimum with respect to p-swaps has cost at most $(3 + 2/p) \cdot \mathrm{OPT}$.

We prove part 1 in Section 12.4. Part 2 is stated without proof; see [AGK+04]. Conversely, no fixed size of swaps brings the locality gap below 3; we prove this for swaps in Section 12.3.

Theorem 12.6. *For every $\varepsilon > 0$, there is an instance of k-median with a local optimum with respect to swaps whose cost is at least $(3 - \varepsilon) \cdot \mathrm{OPT}$.*

Remark 12.7. Theorem 12.5 says nothing about how many moves local search makes before it stops. Starting from a reasonable solution and always performing the best move (or at least a move that makes large progress) gives a $(3 + \varepsilon)$-approximation in time $n^{O(1/\varepsilon)}$ when the moves are the p-swaps with $p = O(1/\varepsilon)$ [AGK+04]; we state this without proof. Section 12.5 proves the analogous statement for swaps with the factor $5 + \varepsilon$.

12.3 A lower bound on the locality gap

We now prove Theorem 12.6. Distances in the instance are shortest-path distances in the graph of Fig. 19.

Definition 12.8. Let $k \geq 2$ be an integer and $x \geq 1$. Let $A = \{a_1, \ldots, a_k\}$, $O = \{o_1, \ldots, o_k\}$ and $C = \{c_{il} : i, l \in [k]\}$ be disjoint sets of points. Take the complete bipartite graph on $A \times O$ and subdivide each edge $a_i o_l$ by the client c_{il}, with an edge of length x between a_i and c_{il} and an edge of length 1 between c_{il} and o_l. Let $V = A \cup O \cup C$, let d the shortest-path metric of this graph, and let C be the set of clients.

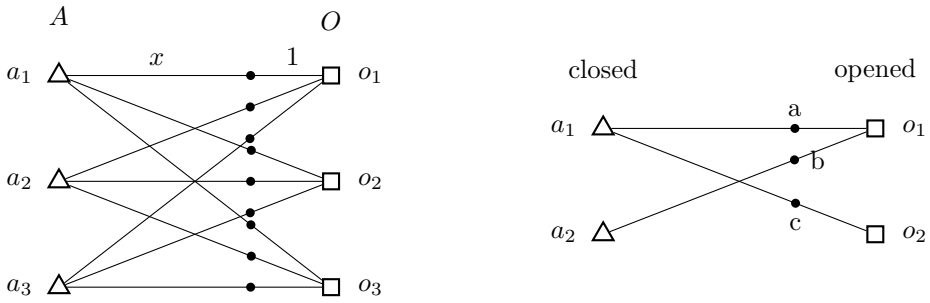


Figure 19: Left: the instance of Definition 12.8 for $k = 3$. The local optimum opens A (triangles), the optimal solution opens O (squares), and every edge carries a client (dot) at distance x from its endpoint in A and 1 from its endpoint in O. Right: the swap (a_1, o_1) and the three types of affected clients.

We need a few distances. Every path from a client c_{il} leaves through a_i or through o_l. Through o_l, it reaches any a_m with $m \neq i$ at length $1 + 1 + x$, via the client c_{ml}; through a_i, it reaches any o_m with $m \neq l$ at length $x + x + 1$, via c_{im}. Hence, for $m \neq i$ and $m' \neq l$,

$$d(c_{il}, a_i) = x, \quad d(c_{il}, o_l) = 1, \quad d(c_{il}, a_m) = x + 2, \quad d(c_{il}, o_{m'}) = 2x + 1, \tag{12.1}$$

where the last value uses $x + 2 + x + 1 \geq 2x + 1$ for the path through o_l and a_m. In particular, $\text{cost}(A) = k^2 x$ and $\text{cost}(O) = k^2$.

Lemma 12.9. *Let $k \geq 2$ and $x = 3 - 2/k$. In the instance of Definition 12.8, the set A is a local optimum with respect to swaps.*

Proof. Consider a swap that closes a_i. By the symmetry of the instance, we may assume $i = 1$. Note that $x \geq 2$ because $k \geq 2$.

Case 1: the swap opens an optimal facility, by symmetry o_1. Only three types of clients change their connection cost (see Fig. 19), by (12.1).

- Type a, the client c_{11}: its cost drops from x to 1.
- Type b, the $k - 1$ clients c_{m1} with $m \neq 1$: their cost drops from x to 1.
- Type c, the $k - 1$ clients c_{1l} with $l \neq 1$: they lose a_1, and their nearest open facility is now some a_m with $m \neq 1$ at distance $x + 2$, since o_1 is at distance $2x + 1 \geq x + 2$. Their cost rises from x to $x + 2$.

Every other client c_{ml} keeps a_m at distance x, and o_1 is farther. So k clients improve by $x - 1$ each and $k - 1$ clients lose 2 each. The total improvement is

$$k(x - 1) - 2(k - 1) = k(x - 3) + 2 = 0,$$

so the swap does not lower the cost.

Case 2: the swap opens a client location c_{ml}. Clients that share the endpoint o_l with c_{ml} are at distance 2 from it, clients that share the endpoint a_m are at distance $2x$, and all other clients are at distance $2x+2$. If $m=1$, the client c_{1l} saves x, the $k-1$ clients $c_{m'l}$ with $m' \neq 1$ save $x-2$ each (moving to c_{1l}), and the $k-1$ clients $c_{1l'}$ with $l' \neq l$ pay 2 more each (moving to some $a_{m'}$, because $2x \geq x+2$). If $m \neq 1$, the client c_{ml} saves x, the $k-1$ clients $c_{m'l}$ with $m' \neq m$ save $x-2$ each, and the $k-1$ clients $c_{1l'}$ with $l' \neq l$ pay 2 more each. In both cases, the cost changes by

$$-x-(k-1)(x-2)+2(k-1) = -x+(k-1)(4-x) = k-2 \geq 0.$$

Hence no swap lowers the cost, and A is a local optimum. □

Proof of Theorem 12.6. Let $k \geq \max\{2, 2/\varepsilon\}$ and $x = 3-2/k$. By Lemma 12.9, the set A is a local optimum, and $\mathrm{OPT} \leq \mathrm{cost}(O) = k^2$. Hence

$$\mathrm{cost}(A) = k^2 x \geq x \cdot \mathrm{OPT} = (3-2/k)\cdot \mathrm{OPT} \geq (3-\varepsilon)\cdot \mathrm{OPT}.$$ □

Remark 12.10. Similarly, for every fixed p, the same instance with $x = 3-2p/k$ and k large has a local optimum with respect to p-swaps of cost at least $(3-\varepsilon)\cdot\mathrm{OPT}$. For a p-swap that closes p facilities of A and opens p facilities of O, the kp clients next to an opened facility improve by $x-1$ each, and the $p(k-p)$ other clients of the closed facilities lose 2 each, so the total improvement is $p\,(k(x-3)+2p) = 0$. We omit the verification for the remaining p-swaps, including those that open client locations.

12.4 Local optima are 5-approximations

We now prove part 1 of Theorem 12.5. The generic idea, used in most analyses of local search, is this: at a local optimum all moves are non-improving, so we exhibit a set of moves whose non-improvement, added up, bounds the cost of the local optimum by the cost of the optimal solution.

Throughout this section, F is a local optimum with respect to swaps and F^* is an optimal solution, with $|F| = |F^*| = k$. For each client j, fix a nearest facility $\varphi(j) \in F$ and a nearest facility $\varphi^*(j) \in F^*$ (breaking ties arbitrarily), and write

$$A_j \coloneqq d(j,F) = d(j,\varphi(j)), \qquad O_j \coloneqq d(j,F^*) = d(j,\varphi^*(j)).$$

Thus $\mathrm{cost}(F) = \sum_{j\in C} A_j$ and $\mathrm{OPT} = \sum_{j\in C} O_j$. For $a \in F$ and $b \in F^*$, let

$$C(a) \coloneqq \{j \in C : \varphi(j) = a\}, \qquad C^*(b) \coloneqq \{j \in C : \varphi^*(j) = b\}$$

be the clients that a serves in F and that b serves in F^*. For $a \in F$ and any $b \in V$, we keep writing $F - a + b$ for $(F \setminus \{a\}) \cup \{b\}$.

Lemma 12.11. *For every $a \in F$ and every $b \in V$, $\mathrm{cost}(F-a+b) \geq \mathrm{cost}(F)$.*

Proof. If $b \notin F$, this is the local optimality of F. If $b \in F$, then $F - a + b \subseteq F$, and closing facilities never lowers the cost. □

For each $b \in F^*$, let $\pi(b) \in F$ be a facility of F nearest to b (breaking ties arbitrarily). The swaps we use open some $b \in F^*$ and close a facility $a \in F$ that is the image under π of no other optimal facility.

Lemma 12.12. *Let $a \in F$ and $b \in F^*$ be such that $\pi(b') \neq a$ for every $b' \in F^* \setminus \{b\}$. Then*

$$\operatorname{cost}(F - a + b) - \operatorname{cost}(F) \leq \sum_{j \in C^*(b)} (O_j - A_j) + \sum_{j \in C(a) \setminus C^*(b)} 2O_j. \tag{12.2}$$

Proof. The cost of $F - a + b$ is at most the cost of any assignment of the clients to facilities of $F - a + b$. We use the following assignment, illustrated in Fig. 20.

- Each client $j \in C^*(b)$ goes to b, so its cost changes from A_j to O_j.
- Let $j \in C(a) \setminus C^*(b)$, and let $b' = \varphi^*(j)$, so $b' \neq b$ and $j \in C^*(b')$. Then $a' = \pi(b')$ is not a, by the hypothesis, so a' is still open, and j goes to a'.
- Every other client j stays at $\varphi(j)$, which is not a.

For a client j of the second kind,

$$\begin{aligned} d(j, a') &\leq d(j, b') + d(b', a') && \text{by the triangle inequality} \\ &\leq d(j, b') + d(b', a) && \text{since } a' = \pi(b') \text{ is nearest to } b' \text{ in } F \\ &\leq d(j, b') + d(b', j) + d(j, a) && \text{by the triangle inequality} \\ &= 2O_j + A_j, \end{aligned}$$

so its cost rises by at most $2O_j$. Adding up the changes gives (12.2). □

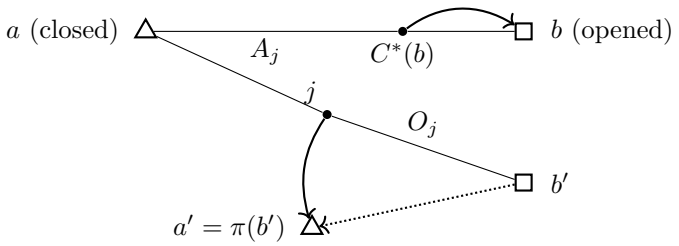


Figure 20: The reassignment in Lemma 12.12. Clients of $C^*(b)$ move to b. A client $j \in C(a) \cap C^*(b')$ moves to $a' = \pi(b')$ (dotted arrow: π), which stays open.

The bijective case. Suppose first that $\pi\colon F^* \to F$ is a bijection. Then each pair $(\pi(b), b)$ with $b \in F^*$ satisfies the hypothesis of Lemma 12.12, and the left side of (12.2) is nonnegative by Lemma 12.11. Since $O_j \geq 0$, we may enlarge the second sum to all of $C(\pi(b))$. Summing over $b \in F^*$, and using that each client lies in exactly one set $C^*(b)$ and, since π is onto F, in exactly one set $C(\pi(b))$, we get

$$0 \leq \sum_{j \in C} (O_j - A_j) + \sum_{j \in C} 2O_j = 3 \cdot \mathrm{OPT} - \operatorname{cost}(F).$$

So in this case $\operatorname{cost}(F) \leq 3 \cdot \mathrm{OPT}$.

The general case. In general, $\pi(b) = \pi(b')$ may hold for some $b \neq b'$. Consider the bipartite graph on $F^* \cup F$ with the k edges $\{b, \pi(b)\}$. Every $b \in F^*$ has degree 1. Call $a \in F$ *good* if it has degree 1, *empty* if it has degree 0, and *dangerous* if it has degree at least 2.

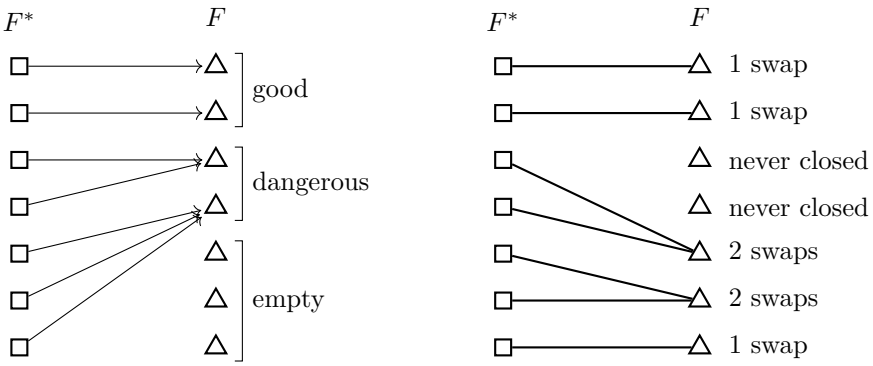
Lemma 12.13. *There is a set $P \subseteq F \times F^*$ of pairs such that every $b \in F^*$ lies in exactly one pair, every $a \in F$ lies in at most two pairs, and every pair $(a, b) \in P$ satisfies $\pi(b') \neq a$ for all $b' \in F^* \setminus \{b\}$.*

Proof. Let g, e and δ be the numbers of good, empty and dangerous facilities, so $g + e + \delta = k$. Pair each good a with the unique b such that $\pi(b) = a$. The other $k - g$ optimal facilities are mapped to dangerous facilities, each of which receives at least two of them, so $k - g \geq 2\delta$. Hence

$$2e = 2(k - g - \delta) \geq 2(k - g) - (k - g) = k - g.$$

So we can pair each of the $k - g$ remaining optimal facilities with an empty facility, using each empty facility at most twice. (In particular, $e \geq \delta$: there are at least as many empty facilities as dangerous ones.) A good facility a is paired only with its unique preimage, and an empty facility has no preimage, so the last property holds. Dangerous facilities are never paired. □

Figure 21 shows an example. The construction closes each good facility in one swap, each empty facility in at most two swaps, and never closes a dangerous facility.



(a) the map π (b) the pairs P

Figure 21: The pairing of Lemma 12.13 for $k = 7$: two good, two dangerous and three empty facilities. The five optimal facilities mapped to dangerous facilities are paired with empty facilities, at most two per empty facility.

Proof of part 1 of Theorem 12.5. Let P be as in Lemma 12.13. Each pair $(a, b) \in P$ satisfies the hypothesis of Lemma 12.12, and the left side of (12.2) is nonnegative by Lemma 12.11. Enlarging the second sum in (12.2) to all of $C(a)$, which is allowed since $O_j \geq 0$, and summing over the k pairs gives

$$0 \leq \sum_{(a,b) \in P} \sum_{j \in C^*(b)} (O_j - A_j) + \sum_{(a,b) \in P} \sum_{j \in C(a)} 2O_j.$$

Each $b \in F^*$ is opened in exactly one pair, so the first double sum equals $\sum_{j \in C}(O_j - A_j)$ exactly; this matters because its terms may be negative. Each $a \in F$ is closed in at most two pairs and

the terms of the second double sum are nonnegative, so it is at most $2\sum_{j\in C} 2O_j$. Hence

$$0 \le \sum_{j\in C}(O_j - A_j) + 4\sum_{j\in C} O_j = 5\cdot \mathrm{OPT} - \mathrm{cost}(F),$$

that is, $\mathrm{cost}(F) \le 5\cdot \mathrm{OPT}$. □

12.5 Polynomial running time

To turn Algorithm 12.3 into a polynomial-time algorithm we need two pieces, both generic in local search: make big steps when possible, and start at a reasonable solution.

Big steps. The argument above did not use local optimality until the very end, so it also measures how much progress the best swap makes from any set.

Lemma 12.14. *Let $F \subseteq V$ be any set with $|F| = k$ and $\mathrm{cost}(F) > 5\cdot\mathrm{OPT}$. Then some swap (a, b) satisfies*

$$\mathrm{cost}(F - a + b) \le \mathrm{cost}(F) - \frac{\mathrm{cost}(F) - 5\cdot\mathrm{OPT}}{k}.$$

Proof. Define φ, φ^*, π and the pairs P for F as above; Lemmas 12.12 and 12.13 do not use local optimality. Summing (12.2) over the k pairs as in the proof of Theorem 12.5 gives

$$\sum_{(a,b)\in P} (\mathrm{cost}(F - a + b) - \mathrm{cost}(F)) \le 5\cdot\mathrm{OPT} - \mathrm{cost}(F).$$

Hence some pair $(a, b) \in P$ has $\mathrm{cost}(F - a + b) - \mathrm{cost}(F) \le -(\mathrm{cost}(F) - 5\cdot\mathrm{OPT})/k < 0$. Then $b \notin F$, since otherwise $F - a + b \subseteq F$ and the cost could not drop, so (a, b) is a swap. □

Theorem 12.15. *Run Algorithm 12.3 on a k-median instance from a set F_0 with $|F_0| = k$, always performing a best swap, and let F_t be the solution after t steps (with $F_t = F_{t-1}$ once the algorithm has stopped). Then for every $t \ge 0$,*

$$\mathrm{cost}(F_t) \le 5\cdot\mathrm{OPT} + \left(1 - \frac{1}{k}\right)^t \cdot \mathrm{cost}(F_0).$$

Proof. Let $D_t = \max\{\mathrm{cost}(F_t) - 5\cdot\mathrm{OPT}, 0\}$, the distance of the current cost to $5\cdot\mathrm{OPT}$. We show that $D_{t+1} \le (1 - 1/k)D_t$. If $D_t = 0$, then $D_{t+1} = 0$, because the cost never increases. If $D_t > 0$, then by Lemma 12.14 an improving swap exists, so the algorithm has not stopped, and the best swap gives

$$\mathrm{cost}(F_{t+1}) - 5\cdot\mathrm{OPT} \le \left(1 - \frac{1}{k}\right)(\mathrm{cost}(F_t) - 5\cdot\mathrm{OPT}).$$

Thus the distance to $5\cdot\mathrm{OPT}$ shrinks by a constant factor in every k steps, and $D_t \le (1-1/k)^t D_0 \le (1 - 1/k)^t \,\mathrm{cost}(F_0)$. □

A reasonable starting solution. If $\text{cost}(F_0) \le n \cdot \text{OPT}$, few steps suffice.

Corollary 12.16. *Let* $0 < \varepsilon \le 1$*, and suppose that* $\text{cost}(F_0) \le n \cdot \text{OPT}$*. Then* $\text{cost}(F_t) \le (5+\varepsilon) \cdot \text{OPT}$ *for* $t = \lceil (k/\varepsilon) \log_2 n \rceil$*.*

Proof. If $n = 1$, then already $\text{cost}(F_0) \le \text{OPT}$, so let $n \ge 2$. By Theorem 12.15 and $1 - 1/k \le e^{-1/k}$,

$$\text{cost}(F_t) - 5 \cdot \text{OPT} \le e^{-t/k}\, n \cdot \text{OPT} \le n^{1-1/(\varepsilon \ln 2)} \cdot \text{OPT}.$$

Since $\varepsilon \le 1 < 1/\ln 2$, the exponent is negative, so the right side is largest at $n = 2$, where it equals $2e^{-1/\varepsilon} \cdot \text{OPT}$. Finally, $2e^{-1/\varepsilon} \le \varepsilon$, because $e^u \ge 2u$ for all real u, applied to $u = 1/\varepsilon$. □

Each step examines $k(n-k)$ swaps, and the cost of each can be computed in $O(nk)$ time, so the algorithm runs in polynomial time. A reasonable starting solution is easy to find.

Homework 12.17. Assume that $|C| \ge k$. Pick any client f_1, set $F_1 = \{f_1\}$, and for $i = 2, \ldots, k$ let f_i be a client at largest distance from F_{i-1}, and $F_i = F_{i-1} \cup \{f_i\}$. Show that $\text{cost}(F_k) \le n \cdot \text{OPT}$.

The choices must be clients: if f_1 may be any point of V, or if f_i is the farthest point of V, a facility may land on a point far from all clients. For instance, with clients c_1, c_2 at distance 1, a third point at distance 100 from both and $k = 2$, the farthest point of V from $\{c_1\}$ is the far point, which gives cost 1 while $\text{OPT} = 0$. Other, better starting solutions exist; finding one is another exercise.

12.6 Summary

Local search is a simple algorithmic idea. To use it, one defines a set of moves such that

(a) an improving move, or the best one, can be found in polynomial time, and

(b) local optima with respect to these moves are good.

Local search is widely used in practice, but there it often comes with no guarantee on the quality of local optima, and its convergence time may be unbounded or exponential, even to reach a near-optimal solution. In theoretical analyses, one exhibits a subset of moves that helps compare the local optimum to an optimal solution, as we did with the pairs of Lemma 12.13. This has several nice applications, sometimes with exponentially large sets of moves (capacitated facility location, labeling problems in vision), and gave the first algorithms for bounded-degree spanning trees and k-median with outliers. Clean algorithms often admit clean analyses.

13 LP Rounding Algorithms for Facility Location

Lecture of September 29, 2021, by Anupam Gupta. Based on the lecturer's handwritten notes.

Lecture 12 analyzed local search for k-median, which gives a constant factor approximation; local search also gives a constant factor for facility location. This lecture turns to linear programming for facility location. LP-based algorithms do better, are more versatile, and compare the solution with the value of the LP relaxation rather than with the optimal integer solution. We write the LP relaxation and its dual, recall the facts about LP duality that we need, and use an optimal pair of primal and dual solutions to round: the dual values tell each client how far it may travel. A deterministic clustering gives a 4-approximation, and a randomized version of the same clustering gives $2 + 2/e$; Sections 4.5 and 5.8 of [WS11] cover these algorithms.

13.1 The linear program

We focus on facility location (Definition 12.2): we are given a finite metric space (V, d), clients $C \subseteq V$ and facility costs $f_i \geq 0$, and we open a nonempty set $F \subseteq V$ minimizing $\text{cost}(F) = \sum_{i \in F} f_i + \sum_{j \in C} d(j, F)$. Throughout, i denotes a potential facility, j a client, and $d_{ij} := d(i, j)$.

One approach is to cast facility location as set cover, where a set consists of a center together with a subset of clients that it serves.

Homework 13.1. For each $i \in V$ and each nonempty $S \subseteq C$, let the set S with center i have cost $f_i + \sum_{j \in S} d_{ij}$. Show that the optimal value of this set cover instance (with elements C) equals OPT, and that the greedy algorithm for weighted set cover (Lecture 2) can be run on this exponentially large instance in polynomial time. Conclude that it gives an $O(\log n)$-approximation for facility location.

We can do better. As in Lecture 2, we write a linear program and round it. The variable y_i indicates that facility i is open, and x_{ij} indicates that client j is served by facility i:

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in V} f_i y_i + \sum_{i \in V} \sum_{j \in C} d_{ij} x_{ij} & & \\ \text{subject to} \quad & \sum_{i \in V} x_{ij} = 1 & & \text{for all } j \in C, \\ & x_{ij} \leq y_i & & \text{for all } i \in V,\ j \in C, \\ & x_{ij} \geq 0,\ y_i \geq 0 & & \text{for all } i \in V,\ j \in C. \end{aligned} \tag{13.1}$$

We write LP for its optimal value.

Lemma 13.2. *The integer program obtained from (13.1) by requiring $x_{ij}, y_i \in \{0, 1\}$ is an exact formulation of facility location: its optimal value is* OPT. *Consequently,* LP $\leq$ OPT.

Proof. Given F, set $y_i = 1$ exactly for $i \in F$ and $x_{ij} = 1$ exactly when i is a chosen nearest facility of F to j; this is feasible with objective value $\text{cost}(F)$. Conversely, given a feasible integral (x, y), let $F = \{i : y_i = 1\}$, which is nonempty because each client has some $x_{ij} = 1 \leq y_i$. Each client j has exactly one i with $x_{ij} = 1$, and this i lies in F, so the objective value is at least $\sum_{i \in F} f_i + \sum_j d(j, F) = \text{cost}(F)$. The LP is a relaxation of the integer program, so LP $\leq$ OPT. □

LP rounding gives constant factor approximations with respect to LP: one can round an optimal solution of (13.1) to a set $F \subseteq V$ with $\text{cost}(F) \le c \cdot \text{LP}$ for a constant c. The best constant known, as of 2021, is $c = 1.488$, due to Li [Li13]. In the other direction, the integrality gap of (13.1) is at least 1.463; we state this without proof. The remaining gap is small, but our focus is on the algorithmic ideas, not on the numerical values.

13.2 LP duality

The earlier applications of linear programming did not use the dual program, but it carries a lot of information. The dual of (13.1) has a variable α_j for each client, from the constraint $\sum_i x_{ij} = 1$, and a variable β_{ij} for each pair of a facility and a client, from the constraint $x_{ij} \le y_i$:

$$\begin{aligned} \text{maximize} \quad & \sum_{j \in C} \alpha_j \\ \text{subject to} \quad & \alpha_j - \beta_{ij} \le d_{ij} && \text{for all } i \in V,\ j \in C, \\ & \sum_{j \in C} \beta_{ij} \le f_i && \text{for all } i \in V, \\ & \beta_{ij} \ge 0 && \text{for all } i \in V,\ j \in C, \end{aligned} \tag{13.2}$$

with α_j unconstrained in sign. Since this is a maximization problem and α_j appears only in the constraints $\alpha_j \le d_{ij} + \beta_{ij}$, an optimal solution may take $\alpha_j = \min_{i \in V}(d_{ij} + \beta_{ij})$, which is nonnegative. Lecture 14 interprets this program in detail. Here we recall the basic facts about LP duality, for a matrix $A \in \mathbb{R}^{m \times n}$ and vectors $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$:

$$\text{(P)} \quad \min \{c^\top x : Ax \ge b,\ x \ge 0\}, \qquad \text{(D)} \quad \max \{b^\top u : A^\top u \le c,\ u \ge 0\}. \tag{13.3}$$

Theorem 13.3 (Weak duality)**.** *If x is feasible for (P) and u is feasible for (D), then $c^\top x \ge b^\top u$.*

Proof. We have

$$\begin{aligned} c^\top x &\ge (A^\top u)^\top x && \text{since } A^\top u \le c \text{ and } x \ge 0 \\ &= u^\top A x \\ &\ge u^\top b && \text{since } Ax \ge b \text{ and } u \ge 0. \end{aligned}$$

□

Theorem 13.4 (Strong duality)**.** *If (P) and (D) both have feasible solutions, then both have optimal solutions x^* and u^*, and $c^\top x^* = b^\top u^*$.*

We do not prove this here; see, for example, [Sch86, MG07].

Corollary 13.5 (Complementary slackness)**.** *If x^* and u^* are optimal solutions of (P) and (D), then*

$$(x^*)^\top (c - A^\top u^*) = 0 \qquad \text{and} \qquad (u^*)^\top (Ax^* - b) = 0.$$

Proof. By Theorem 13.4, $c^\top x^* = b^\top u^*$, so both inequalities in the proof of Theorem 13.3 are equalities. □

Both products are sums of nonnegative terms, so every term vanishes. In other words, if a dual variable is nonzero, then the corresponding primal constraint is tight; and if a primal variable is nonzero, then the corresponding dual constraint is tight. A nonzero dual variable thus marks an important constraint. For facility location, this gives the following.

Lemma 13.6. *Let (x, y) be an optimal solution of (13.1) and (α, β) an optimal solution of (13.2). Then $\sum_{j \in C} \alpha_j = \mathrm{LP}$, and for all $i \in V$ and $j \in C$,*

$$x_{ij} > 0 \quad \textit{implies} \quad \alpha_j = \beta_{ij} + d_{ij}.$$

Proof. For feasible solutions of the two programs, using $\sum_i x_{ij} = 1$ in the first step,

$$\begin{aligned}
\sum_j \alpha_j = \sum_{i,j} \alpha_j x_{ij} &\le \sum_{i,j} (d_{ij} + \beta_{ij}) x_{ij} && \text{since } \alpha_j \le d_{ij} + \beta_{ij},\ x_{ij} \ge 0 \\
&\le \sum_{i,j} d_{ij} x_{ij} + \sum_i y_i \sum_j \beta_{ij} && \text{since } x_{ij} \le y_i,\ \beta_{ij} \ge 0 \\
&\le \sum_{i,j} d_{ij} x_{ij} + \sum_i f_i y_i && \text{since } \textstyle\sum_j \beta_{ij} \le f_i,\ y_i \ge 0.
\end{aligned}$$

Both programs are feasible (open one facility and assign every client to it; take $\alpha = 0$ and $\beta = 0$), and Theorem 13.4 applies to them after the standard rewriting of each equality constraint as two inequalities. So for optimal solutions the two ends of the chain agree, which proves the first claim, and every inequality is an equality. The first inequality is a sum of the nonnegative terms $x_{ij}(d_{ij} + \beta_{ij} - \alpha_j)$, so each of them vanishes. □

Weak duality has a further consequence. For every feasible dual solution,

$$\sum_{j \in C} \alpha_j \le (\text{optimal dual value}) \le \mathrm{LP} \le \mathrm{OPT},$$

where the middle inequality is weak duality and the last one holds because the LP is a relaxation (Lemma 13.2). Hence if an algorithm finds a solution of cost at most $\rho \cdot \sum_j \alpha_j$ for some feasible dual solution, it is a ρ-approximation. The dual is an accounting device. It therefore suffices to construct a solution to facility location together with a dual solution such that the cost is at most a constant times the dual value.

13.3 Clustering and a deterministic 4-approximation

Fix an optimal solution (x, y) of (13.1) and an optimal solution (α, β) of (13.2), so that $\sum_j \alpha_j = \mathrm{LP}$ by Lemma 13.6.

Definition 13.7. The *cluster* of a client $j \in C$ is $N(j) := \{i \in V : x_{ij} > 0\}$, the set of facilities to which the LP sends j fractionally.

Each cluster is nonempty, since $\sum_i x_{ij} = 1$. By Lemma 13.6 and $\beta \ge 0$,

$$d_{ij} \le \alpha_j \qquad \text{for every } j \in C \text{ and } i \in N(j): \tag{13.4}$$

the connection cost of j to any facility of its cluster is at most the dual contribution of j. A natural idea is to open the cheapest facility of each cluster and send its client there. For disjoint clusters this works well.

Lemma 13.8. *Let $C' \subseteq C$ be a set of clients whose clusters are pairwise disjoint, and for $j \in C'$ let $i(j)$ be a facility of least cost in $N(j)$. Then*

$$\sum_{j \in C'} f_{i(j)} \le \sum_{i \in V} f_i y_i \qquad \textit{and} \qquad d_{i(j)j} \le \alpha_j \textit{ for every } j \in C'.$$

Proof. For $j \in C'$,

$$f_{i(j)} \le \sum_{i \in N(j)} f_i x_{ij} \le \sum_{i \in N(j)} f_i y_i,$$

where the first inequality holds because a minimum is at most a weighted average (the weights x_{ij}, $i \in N(j)$, sum to 1), and the second because $x_{ij} \le y_i$. The right side is the LP's facility cost inside the cluster $N(j)$. Since the clusters are disjoint, summing over $j \in C'$ gives at most $\sum_i f_i y_i$. The second claim is (13.4). □

When clusters overlap, we pick a family of pairwise disjoint clusters of small radius, open the cheapest facility in each selected cluster, and show that the clients of the unselected clusters can be rerouted to these open facilities at small cost. Small radius means small α_j, by (13.4).

Algorithm 13.9 (Deterministic clustering)**.**

1) Solve (13.1) and (13.2) optimally.
2) Order the clients so that $\alpha_{j_1} \le \alpha_{j_2} \le \cdots \le \alpha_{j_{|C|}}$.
3) Set $S = \emptyset$. For $t = 1, \ldots, |C|$: if $N(j_t)$ is disjoint from $N(j')$ for every $j' \in S$, add j_t to S.
4) Open $F = \{i(j) : j \in S\}$, where $i(j)$ is a facility of least cost in $N(j)$.

Theorem 13.10. *The set F of Algorithm 13.9 satisfies*

$$\text{cost}(F) \le \sum_{i \in V} f_i y_i + 3 \sum_{j \in C} \alpha_j \le 4 \cdot \text{LP} \le 4 \cdot \text{OPT}.$$

Proof. Facility cost. The clusters $N(j)$, $j \in S$, are pairwise disjoint by construction, so the facilities $i(j)$ are distinct, and

$$\begin{aligned}
\sum_{i \in F} f_i = \sum_{j \in S} \min_{i \in N(j)} f_i &\le \sum_{j \in S} \sum_{i \in N(j)} f_i x_{ij} && \text{since a minimum is at most an average} \\
&\le \sum_{j \in S} \sum_{i \in N(j)} f_i y_i && \text{since } x_{ij} \le y_i \\
&\le \sum_{i \in V} f_i y_i && \text{since the clusters are disjoint} \\
&\le \text{LP}.
\end{aligned}$$

Connection cost of the cluster centers. For $j \in S$, the facility $i(j) \in N(j)$ is open, so $d(j, F) \le d_{i(j)j} \le \alpha_j$ by (13.4).

Connection cost of the other clients. Let $j \notin S$. When j was considered, $N(j)$ met $N(j')$ for some j' already in S; let i lie in $N(j) \cap N(j')$, and recall that $i(j') \in N(j')$ is open (see Fig. 22). Since j' came earlier in the order, $\alpha_{j'} \le \alpha_j$. Then

$$\begin{aligned}
d(j, F) \le d(j, i(j')) &\le d(j, i) + d(i, j') + d(j', i(j')) && \text{by the triangle inequality} \\
&\le \alpha_j + \alpha_{j'} + \alpha_{j'} && \text{by (13.4)} \\
&\le 3\alpha_j && \text{by the sorted order.}
\end{aligned}$$

Adding up, the connection cost is at most $3 \sum_j \alpha_j = 3 \cdot \text{LP}$, and the facility cost is at most $\sum_i f_i y_i \le \text{LP}$. Finally, $\text{LP} \le \text{OPT}$ by Lemma 13.2. □

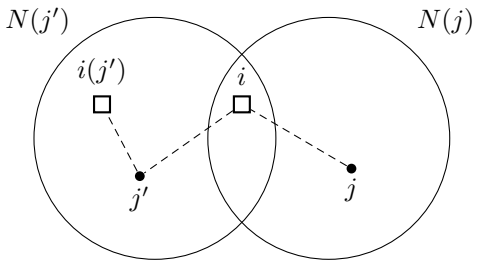


Figure 22: Rerouting a client $j \notin S$ in Theorem 13.10. Its cluster meets the cluster of an earlier center $j' \in S$ in a facility i, and j travels to the open facility $i(j')$ along the dashed path, of length at most $\alpha_j + 2\alpha_{j'} \le 3\alpha_j$.

Remark 13.11. The properties of the dual solution and its relation to the primal one gave a quick approximation algorithm for facility location. The rounding is deterministic. The clustering keeps the facility opening cost small, and the triangle inequality keeps the connection cost of the clients that are not cluster centers small.

13.4 Randomized rounding

Theorem 13.10 shows that the integrality gap of (13.1) is at most 4. Randomized rounding improves this. We may assume that $y_i \le 1$ for all i: replacing y_i by $\min\{y_i, 1\}$ keeps (13.1) feasible (since $x_{ij} \le 1$) and does not increase the cost, so the solution stays optimal.

Algorithm 13.12 (Randomized clustering)**.** Compute the set S of cluster centers as in Algorithm 13.9. Then open facilities as follows, with all random choices independent.

1) For each $j \in S$, open exactly one facility of $N(j)$, choosing $i \in N(j)$ with probability x_{ij}.

2) For each $i \in V$, open i with probability $y_i - \sum_{j \in S} x_{ij}$.

The probabilities in step 1 sum to 1. Those in step 2 lie in $[0, 1]$: since the clusters of S are disjoint, at most one $j \in S$ has $x_{ij} > 0$, and $x_{ij} \le y_i \le 1$.

Lemma 13.13. *The expected facility cost of Algorithm 13.12 is at most* $\sum_{i \in V} f_i y_i$*, which is at most* LP.

Proof. By the union bound over the two steps, facility i is open with probability at most $\sum_{j \in S} x_{ij} + (y_i - \sum_{j \in S} x_{ij}) = y_i$. By linearity of expectation, the expected facility cost is at most

$$\sum_{j \in S} \sum_{i \in N(j)} f_i x_{ij} + \sum_{i \in V} f_i \Big(y_i - \sum_{j \in S} x_{ij} \Big) = \sum_{i \in V} f_i y_i \le \mathrm{LP}. \qquad \square$$

As before, every cluster center $j \in S$ has an open facility in $N(j)$, at distance at most α_j, its dual contribution. A client $j \notin S$ has a cluster that meets the cluster of an earlier center $j' \in S$, with $\alpha_{j'} \le \alpha_j$. If some facility of $N(j)$ is open, j goes there, at distance at most α_j; otherwise it goes to the open facility of $N(j')$, at distance at most $3\alpha_j$ as in Theorem 13.10. The first case is likely.

Lemma 13.14. *For every client* j*, the probability that Algorithm 13.12 opens some facility of* $N(j)$ *is at least* $1 - 1/e$.

Proof. No facility of $N(j)$ is open exactly when, for each $j' \in S$, the facility chosen in $N(j')$ lies outside $N(j)$, and no $i \in N(j)$ is opened in step 2. These events, one for each $j' \in S$ and one for each $i \in N(j)$, are independent. The event for j' has probability $1 - \sum_{i \in N(j) \cap N(j')} x_{ij'}$, and the event for i has probability $1 - (y_i - \sum_{j' \in S} x_{ij'})$. Using $1 - z \le e^{-z}$ for each factor of the product, we get

$$\begin{aligned}\Pr[\text{no facility of } N(j) \text{ is open}] &\le \exp\Big(-\sum_{j' \in S} \sum_{i \in N(j) \cap N(j')} x_{ij'} - \sum_{i \in N(j)} \Big(y_i - \sum_{j' \in S} x_{ij'}\Big)\Big) \\ &= \exp\Big(-\sum_{i \in N(j)} y_i\Big) \le \exp\Big(-\sum_{i \in N(j)} x_{ij}\Big) = e^{-1},\end{aligned}$$

where the equality holds because $x_{ij'} = 0$ for $i \notin N(j')$. □

Lemma 13.15. *For every client j, the expected connection cost of j is at most $(1 + 2/e)\,\alpha_j$.*

Proof. If $j \in S$, its connection cost is always at most α_j. Let $j \notin S$, and let E be the event that some facility of $N(j)$ is open. On E, the connection cost is at most α_j by (13.4); otherwise it is at most $3\alpha_j$. Since $\alpha_j \ge 0$ and $\Pr[E] \ge 1 - 1/e$ by Lemma 13.14,

$$\mathbb{E}[d(j, F)] \le \alpha_j \Pr[E] + 3\alpha_j(1 - \Pr[E]) = 3\alpha_j - 2\alpha_j \Pr[E] \le \Big(1 + \frac{2}{e}\Big)\alpha_j. \qquad \square$$

Theorem 13.16. *The expected cost of the solution of Algorithm 13.12 is at most*

$$\sum_{i \in V} f_i y_i + \Big(1 + \frac{2}{e}\Big) \sum_{j \in C} \alpha_j \le \Big(2 + \frac{2}{e}\Big)\mathrm{LP}.$$

In particular, the integrality gap of (13.1) is at most $2 + 2/e < 2.74$.

Proof. The bound follows from Lemmas 13.13 and 13.15 and $\sum_j \alpha_j = \mathrm{LP}$: the facility cost contributes at most $1 \cdot \mathrm{LP}$ and the connection cost at most $(1 + 2/e) \cdot \mathrm{LP}$. Some outcome costs at most the expectation, which bounds the integrality gap. □

With more care, one can avoid losing the additive $1 \cdot \mathrm{LP}$ for the facility cost and obtain a $(1 + 2/e)$-approximation, by choosing the cluster centers differently; see Section 5.8 of [WS11] and Chudak and Shmoys [CS03]. There are more ideas of this kind. Lecture 14 uses duality differently, building primal and dual solutions together, without solving the LP.

14 Hardness for Facility Location; Primal-Dual Algorithms

Lecture of October 1, 2021, by Anupam Gupta. Based on the lecturer's handwritten notes.

The first of this lecture's two parts reduces max-coverage (Lecture 3) to facility location, following Guha and Khuller [GK99], and shows that facility location is hard to approximate within any factor smaller than about 1.463. This nearly matches the best known algorithms (Lecture 13). The second part introduces the primal-dual method. Where Lecture 13 solved the LP and its dual and then rounded, a primal-dual algorithm builds an integral primal solution and a feasible dual solution together, by hand, and pays for the first with the second. We warm up with weighted vertex cover and then give the 3-approximation of Jain and Vazirani [JV01] for facility location (Sections 7.6 and 7.7 of [WS11]), whose stronger guarantee also leads to algorithms for k-median.

Facility location is as in Definition 12.2: a finite metric space (V, d), clients $C \subseteq V$, facility costs $f_i \geq 0$, and $\text{cost}(F) = \sum_{i \in F} f_i + \sum_{j \in C} d(j, F)$ for a nonempty $F \subseteq V$. We write $d_{ij} = d(i, j)$.

14.1 Hardness of facility location

We start from the hardness of max-coverage. For $\varepsilon > 0$, the 1-*vs*-$(1 - 1/e + \varepsilon)$ *problem for max-coverage* is the following decision problem. We are given a universe U with $|U| = n$, sets $S_1, \ldots, S_m \subseteq U$, and an integer $k \geq 1$, and we must distinguish YES instances, in which some k of the sets cover U, from NO instances, in which every k of the sets cover at most $(1 - 1/e + \varepsilon)n$ elements.

Fact 14.1 (Feige [Fei98])**.** *For every $\varepsilon > 0$, the* 1-*vs*-$(1 - 1/e + \varepsilon)$ *problem for max-coverage is NP-hard.*

Lecture 3 proves the weaker statement with $3/4 + \varepsilon$ in place of $1 - 1/e + \varepsilon$. We use Fact 14.1 to show that a similar gap problem for facility location is hard. That is, we give a map from max-coverage instances to facility location instances, sketched in Fig. 23, which sends YES instances to instances of cost at most $n + fk$ and NO instances, in a strong sense made precise below, to instances of cost at least roughly $1.463\,(n + fk)$. This shows that approximating facility location within 1.463 is hard as well.

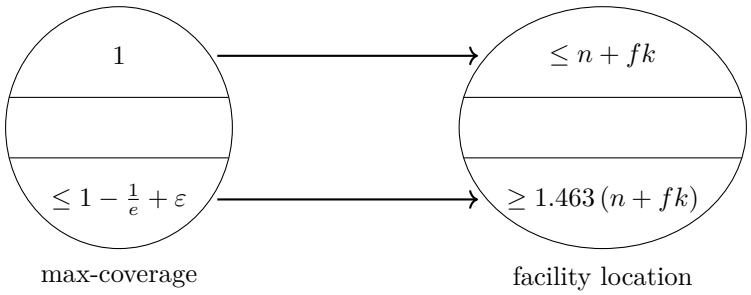


Figure 23: A gap-preserving map. Max-coverage instances in which k sets cover the whole universe go to facility location instances of small cost; instances in which every few sets cover little go to instances of large cost (the bound 1.463 holds up to an arbitrarily small loss).

Definition 14.2. Let $S_1, \dots, S_m$ be sets with $S_1 \cup \cdots \cup S_m = U$, $|U| = n$, and let $f > 0$. The facility location instance $I(U, S, f)$ has the point set $V = L \cup R$, where $L = U$ (the elements) and $R = \{s_1, \dots, s_m\}$ (one potential facility location per set). The clients are $C = L$. Opening a facility at $s_i \in R$ costs f, and facilities may not be opened in L ($f_j = \infty$ for $j \in L$). The distances are $d(j, s_i) = 1$ if $j \in S_i$ and $d(j, s_i) = 3$ if $j \notin S_i$, and $d(p, q) = 2$ for distinct p, q that both lie in L or both lie in R.

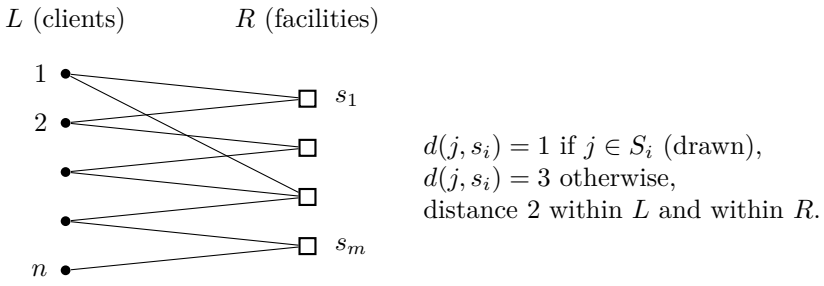


Figure 24: The instance $I(U, S, f)$ of Definition 14.2. An element is joined to the sets that contain it.

Lemma 14.3. *The distances of Definition 14.2 form a metric.*

Proof. The distances are symmetric and positive between distinct points. All of them lie in $\{1, 2, 3\}$, so a triangle inequality $d(p, q) \le d(p, r) + d(r, q)$ with distinct p, q, r can fail only if $d(p, q) = 3$, that is, if $p = j \in L$ and $q = s_i \in R$ with $j \notin S_i$ (or the reverse). If $r \in L$, then $d(j, r) = 2$ and $d(r, s_i) \ge 1$; if $r \in R$, then $d(j, r) \ge 1$ and $d(r, s_i) = 2$. In both cases the right side is at least 3. □

A solution opens a nonempty set $F \subseteq R$, that is, a family of sets. A client j is at distance 1 from F if some set of F contains it, and at distance 3 otherwise. Hence, if the sets of F cover c_F elements,

$$\operatorname{cost}(F) = f|F| + c_F + 3(n - c_F) = f|F| + 3n - 2c_F. \tag{14.1}$$

YES instances. If k sets cover U, opening them gives cost $kf + n \cdot 1$, so $\mathrm{OPT} \le kf + n$.

NO instances. Here we need more than Fact 14.1 provides: a bound on how much any number of sets can cover. We say that (U, S, k) is an *ε-strong NO instance* if, for every $\alpha \ge 0$ such that αk is an integer, every αk of the sets cover at most $(1 - e^{-\alpha} + \varepsilon)\, n$ elements. For $\alpha = 1$ this is the NO case of the max-coverage problem. Write $x = kf/n$ for the ratio of the YES cost of the facilities to that of the connections.

Lemma 14.4. *Let (U, S, k) be an ε-strong NO instance, let $f > 0$ with $x = kf/n \le 2$, and let $F \subseteq R$ be nonempty. Then*

$$\operatorname{cost}(F) \ge n\Big(1 + x + x \ln \frac{2}{x} - 2\varepsilon\Big).$$

Proof. Let $\alpha = |F|/k$. The sets of F cover $c_F \le (1 - e^{-\alpha} + \varepsilon)n$ elements, and (14.1) decreases in c_F, so

$$\begin{aligned}\operatorname{cost}(F) &\ge f\alpha k + (1 - e^{-\alpha} + \varepsilon)\, n \cdot 1 + (e^{-\alpha} - \varepsilon)\, n \cdot 3 \\ &= f\alpha k + \Big(1 + \frac{2}{e^{\alpha}}\Big)n - 2\varepsilon n = n(1 + x\alpha + 2e^{-\alpha} - 2\varepsilon).\end{aligned}$$

Regardless of how many facilities are chosen, this is at least the minimum over all real $\alpha \ge 0$. The function $\alpha \mapsto x\alpha + 2e^{-\alpha}$ is convex with derivative $x - 2e^{-\alpha}$, so it is minimized at $\alpha^* = \ln(2/x) = \ln(2n/(fk))$, which is nonnegative because $x \le 2$. Its minimum value is $x\ln(2/x) + x$. □

In the original parameters, the hardness becomes the problem of distinguishing cost at most $kf + n$ from cost at least $kf\ln(2n/(fk)) + n + kf$ (up to $2\varepsilon n$), that is, $n(1 + x)$ from $n(x\ln(2/x) + 1 + x)$. We set f to maximize the ratio

$$\frac{x\ln(2/x) + 1 + x}{1 + x} = 1 + \frac{x\ln(2/x)}{1 + x}.$$

Lemma 14.5. *On $0 < x \le 2$, the function $h(x) = x\ln(2/x)/(1 + x)$ attains its maximum, and $\max_x(1 + h(x)) = \gamma = 1.46305\ldots$. More precisely, the maximum is attained at the unique solution x^* of $\ln(2/x) = 1 + x$, and $h(x^*) = x^*$, so $\gamma = 1 + x^*$ is the solution of $\gamma = 1 + 2e^{-\gamma}$.*

Proof. A direct computation gives $h'(x) = (\ln(2/x) - 1 - x)/(1 + x)^2$. The numerator decreases in x, tends to $+\infty$ as $x \to 0$, and equals -3 at $x = 2$, so it has a unique zero x^*, where h attains its maximum. At x^*, we have $\ln(2/x^*) = 1 + x^*$, so $h(x^*) = x^*$. Writing $\gamma = 1 + x^*$, the equation $\ln(2/x^*) = \gamma$ says $x^* = 2e^{-\gamma}$, that is, $\gamma = 1 + 2e^{-\gamma}$. □

Theorem 14.6. *Let $\varepsilon > 0$ and $0 < x \le 2$, and map each max-coverage instance (U, S, k) to $I(U, S, f)$ with $f = xn/k$. YES instances go to facility location instances with $\mathrm{OPT} \le (1 + x)\, n$, and ε-strong NO instances go to instances with $\mathrm{OPT} \ge (1 + x + x\ln(2/x) - 2\varepsilon)\, n$. Consequently, for every $\rho < \gamma$, with γ as in Lemma 14.5, a polynomial-time ρ-approximation algorithm for facility location would distinguish, in polynomial time, YES instances from ε-strong NO instances, for some $\varepsilon > 0$.*

Proof. The two bounds are the YES case above and Lemma 14.4. Given $\rho < \gamma$, Lemma 14.5 lets us choose a rational x and an $\varepsilon > 0$ with $1 + x + x\ln(2/x) - 2\varepsilon > \rho(1 + x)$. On a YES instance, the algorithm returns a solution of cost at most $\rho(1 + x)n$; on an ε-strong NO instance, every solution costs more than that. Comparing the cost of the returned solution with $\rho(1 + x)n$ therefore decides which case holds. □

This is the best hardness result known for facility location (as of 2021). It is due to Guha and Khuller [GK99]. They applied this reduction to set cover instances from Feige's work [Fei98], and concluded that, for every $\rho < 1.463$, facility location has no polynomial-time ρ-approximation unless NP $\subseteq$ DTIME$(n^{O(\log\log n)})$.

14.2 The primal-dual method

In Lecture 13, we solved the primal and dual LPs and used both optimal solutions to round the primal one. A primal-dual algorithm instead constructs solutions to the primal-dual pair by hand, such that

1) the primal and dual solutions are feasible,

2) the primal solution is integral, and

3) the cost of the primal solution is at most ρ times the value of the dual solution (for a minimization problem as the primal).

Lemma 14.7. *If a primal-dual algorithm satisfies these three properties, then its primal solution is a ρ-approximation.*

Proof. The cost of the primal solution is at most ρ times the dual value by property 3, which is at most ρ times the optimal primal LP value by weak duality (Theorem 13.3), which is at most $\rho \cdot \mathrm{OPT}$ because the LP is a relaxation of the integer program. □

14.3 Warm-up: vertex cover

Let $G = (V, E)$ be a graph. A *vertex cover* is a set $S \subseteq V$ that contains at least one endpoint of every edge. For unit costs, a greedy algorithm (from the homework) repeatedly picks an edge not yet covered and adds both of its endpoints.

Proposition 14.8. *The greedy algorithm outputs a vertex cover with at most twice the minimum number of vertices.*

Proof. The picked edges form a matching M, since a picked edge has no endpoint in common with an earlier one, and the output has $2|M|$ vertices. Every vertex cover contains an endpoint of each edge of M, and these endpoints are distinct, so every vertex cover has at least $|M|$ vertices. □

Now let the vertices have costs $c_v \geq 0$; we want a vertex cover of least total cost. The LP relaxation and its dual are

$$\begin{array}{lll} \text{minimize} & \sum_{v \in V} c_v x_v & \\ \text{subject to} & x_u + x_v \geq 1 \quad \text{for all } uv \in E, & \\ & x_v \geq 0 \quad \text{for all } v \in V, & \end{array} \qquad \begin{array}{lll} \text{maximize} & \sum_{e \in E} y_e & \\ \text{subject to} & \sum_{e \ni v} y_e \leq c_v \quad \text{for all } v \in V, & (14.2) \\ & y_e \geq 0 \quad \text{for all } e \in E. & \end{array}$$

If $c_v = 1$ for all v, then every matching M is a feasible dual solution ($y_e = 1$ if $e \in M$ and $y_e = 0$ otherwise), and the greedy solution, both endpoints of a maximal matching M, has cost at most $2|M|$. So the greedy algorithm is already a primal-dual algorithm. For general costs, we raise dual variables until dual constraints become tight. Call a vertex v *tight* if $\sum_{e \ni v} y_e = c_v$.

Algorithm 14.9 (Primal-dual vertex cover)**.** Start with $x = 0$ and $y = 0$, and let S always be the set of tight vertices ($x_v = 1$ for $v \in S$). While some edge $e = uv$ has neither endpoint in S, raise y_e until u or v becomes tight. Output S.

If $c_v = 1$ for all v, each step sets $y_e = 1$ and makes both endpoints tight, so this is the greedy algorithm above.

Theorem 14.10 (Bar-Yehuda and Even [BYE81])**.** *Algorithm 14.9 stops after at most $|V|$ steps, and outputs a vertex cover S with*

$$\sum_{v \in S} c_v \leq 2 \sum_{e \in E} y_e \leq 2 \cdot \mathrm{OPT}.$$

Proof. In each step, neither endpoint of e is tight, so y_e rises by a positive amount, the dual stays feasible, and at least one more vertex becomes tight. A tight vertex stays tight, since its edges are covered and never raised again. So there are at most $|V|$ steps, and at the end every edge has an endpoint in S. Thus the primal and dual solutions are feasible, and the primal one is integral. Since every vertex of S is tight,

$$\sum_{v \in S} c_v = \sum_{v \in S} \sum_{e \ni v} y_e = \sum_{e \in E} y_e \cdot |e \cap S| \leq 2 \sum_{e \in E} y_e,$$

and Lemma 14.7 gives the second inequality. □

Remark 14.11. The algorithm handcrafts a dual solution by raising money, and it maintains one half of complementary slackness: if a primal variable is positive, then its dual constraint is tight. Each edge offers money to its endpoints, and a tight vertex can open. But each edge offers its money to both endpoints; this double dipping is where the factor 2 comes from. Primal-dual algorithms often have this double-dipping flavor, which is where the approximation factor arises. Here we raise a single dual variable at a time; for other problems, it helps to raise many variables simultaneously.

14.4 A primal-dual algorithm for facility location

We use the LP relaxation of Lecture 13 with the equality constraints relaxed to inequalities, which is still a relaxation of facility location, and its dual:

$$\begin{array}{lll} \min & \sum_i f_i y_i + \sum_{i,j} d_{ij} x_{ij} & \\ \text{s.t.} & x_{ij} \leq y_i \quad \text{for all } i \in V,\ j \in C, & \\ & \sum_i x_{ij} \geq 1 \quad \text{for all } j \in C, & \\ & x, y \geq 0, & \end{array} \qquad \begin{array}{lll} \max & \sum_{j \in C} \alpha_j & \\ \text{s.t.} & \alpha_j - \beta_{ij} \leq d_{ij} \quad \text{for all } i \in V,\ j \in C, & \\ & \sum_j \beta_{ij} \leq f_i \quad \text{for all } i \in V, & \\ & \alpha, \beta \geq 0. & \end{array} \tag{14.3}$$

By weak duality, $\sum_j \alpha_j \leq \mathrm{OPT}$ for every feasible dual solution.

The dual has a natural reading. The value α_j is the money that client j offers to be connected. If j connects to facility i, a part d_{ij} of it pays for the connection, and the part β_{ij} is a contribution toward opening i. The total contribution $\sum_j \beta_{ij}$ offered to a location i should never exceed its cost f_i; once it reaches f_i, we can open i. The most money we can raise this way is the optimal dual value. We now raise this money and build a solution simultaneously.

Algorithm 14.12 (Jain and Vazirani)**.**

1) *Dual growth.* A time τ grows continuously from 0. Every client is *active* at the start. Each active client j has $\alpha_j = \tau$; a frozen client keeps its value. At all times, $\beta_{ij} = (\alpha_j - d_{ij})^+ := \max\{\alpha_j - d_{ij}, 0\}$.

   - When $\sum_j \beta_{ij} = f_i$ for a facility i that is not yet tentatively open, i becomes *tentatively open*; let t_i be this time.
   - When an active client j touches a tentatively open facility i, that is, $\alpha_j \geq d_{ij}$, client j freezes.

   The phase ends when all clients are frozen. Let F' be the set of tentatively open facilities.

2) *Clean-up.* Let H be the graph on F' in which $i \neq i'$ are adjacent if some client j has $\beta_{ij} > 0$ and $\beta_{i'j} > 0$, that is, makes a nonzero contribution to both. Open a maximal independent set F of H.

The clean-up is needed because many facilities may be tentatively open. For example, with a single client whose own location has facility cost at least 2 and many facilities at distance 1 of cost 1, all of them become tentatively open at the same time $\tau = 2$: the double dipping becomes many-times dipping. The metric property lets us close most of them.

Lemma 14.13. *The dual growth phase ends, and the final (α, β) is a feasible solution of the dual in (14.3).*

Proof. The constraints $\alpha_j - \beta_{ij} \leq d_{ij}$ and $\alpha, \beta \geq 0$ hold by the definition of β_{ij}. For a facility i, the sum $\sum_j \beta_{ij}$ grows continuously, and i becomes tentatively open when it reaches f_i. From then on it stays f_i: at time t_i, every active client j with $\alpha_j \geq d_{ij}$ touches i and freezes, and every other active client freezes as soon as α_j reaches d_{ij}, with $\beta_{ij} = 0$. The phase ends, since a client j that is still active at time $\tau = d_{ij} + f_i$ (for any i) has $\beta_{ij} = f_i$ by then, so i is tentatively open and j touches it. □

Connecting the clients. Assign clients to the open facilities in F as follows.

(a) If $\beta_{ij} > 0$ for some $i \in F$, client j contributes to i, and we assign j to i. Since F is independent in H, each client contributes to at most one facility of F.

(b) Otherwise, if $\alpha_j = d_{ij}$ for some $i \in F$, we assign j to one such i.

Let $N(i)$ be the set of clients assigned to $i \in F$. In both cases,

$$\alpha_j = \beta_{ij} + d_{ij} \qquad \text{for } j \in N(i). \tag{14.4}$$

The remaining clients, for which neither a facility that they contributed to nor a facility with $\alpha_j = d_{ij}$ is open, are called *lonely*.

Lemma 14.14 (Three-hop lemma)**.** *For every lonely client j, there is an open facility at distance at most $3\alpha_j$ from j.*

Proof. Consider the dual growth phase. We stopped raising α_j because j touched some tentatively open facility $i \in F'$, so $t_i \leq \alpha_j$ and $d_{ij} \leq \alpha_j$. If i were in F, then j would be assigned by rule (a) if $\alpha_j > d_{ij}$, and by rule (b) if $\alpha_j = d_{ij}$. Since j is lonely, $i \notin F$. Because F is a maximal independent set, i has a neighbor $i' \in F$ in H: some client j' contributes to both i and i', that is, $\beta_{ij'} > 0$ and $\beta_{i'j'} > 0$. Then $d_{ij'} < \alpha_{j'}$ and $d_{i'j'} < \alpha_{j'}$, and (see Fig. 25)

$$d(j, F) \leq d(j, i) + d(i, j') + d(j', i') \leq \alpha_j + \alpha_{j'} + \alpha_{j'}.$$

We claim that $\alpha_{j'} \leq \alpha_j$. Since all active clients rise simultaneously, and facility i was tentatively open at time α_j (if not before), client j' stopped at time t_i because of i, if it had not stopped earlier. Indeed, j' touches i by time t_i: otherwise $\alpha_{j'}$ would reach $d_{ij'}$ only after t_i, and j' would freeze at that moment with $\beta_{ij'} = 0$. Hence $\alpha_{j'} \leq t_i \leq \alpha_j$, and $d(j, F) \leq 3\alpha_j$. □

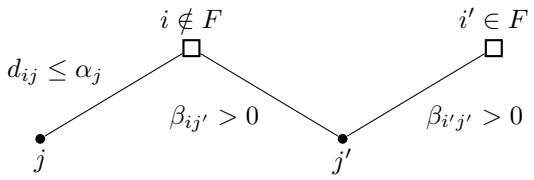


Figure 25: The three hops in Lemma 14.14: from the lonely client j to the closed facility i that froze it, to a client j' that contributed to i and to an open neighbor i' of i in H, and on to i'.

Theorem 14.15 (Jain and Vazirani [JV01])**.** *The set F of Algorithm 14.12 satisfies*

$$\operatorname{cost}(F) \le 3 \sum_{j \in C} \alpha_j \le 3 \cdot \mathrm{OPT}.$$

Moreover, $3\sum_{i\in F} f_i + \sum_{j\in C} d(j,F) \le 3\sum_{j\in C}\alpha_j$; in other words, $\sum_{j\in C} d(j,F) \le 3(\sum_{j\in C}\alpha_j - \sum_{i\in F} f_i)$.

Proof. Each $i \in F$ is tentatively open, so $f_i = \sum_j \beta_{ij}$, and by rule (a) every client with $\beta_{ij} > 0$ belongs to $N(i)$. Hence $f_i = \sum_{j\in N(i)} \beta_{ij}$. Connecting each client that is not lonely to its assigned facility, the cost of the open facilities and of these clients is at most

$$\sum_{i\in F}\Big(f_i + \sum_{j\in N(i)} d_{ij}\Big) = \sum_{i\in F}\Big(\sum_{j\in N(i)} \beta_{ij} + \sum_{j\in N(i)} d_{ij}\Big) = \sum_{i\in F}\sum_{j\in N(i)} \alpha_j = \sum_{j \text{ not lonely}} \alpha_j,$$

by (14.4). The lonely clients cost at most $3\sum_{j \text{ lonely}} \alpha_j$ by Lemma 14.14. So $\operatorname{cost}(F) \le 3\sum_j \alpha_j$, and $\sum_j \alpha_j \le \mathrm{OPT}$ by Lemma 14.13 and weak duality.

For the stronger inequality, let $i \in F$. For $j \in N(i)$ we have $d(j,F) \le d_{ij} = \alpha_j - \beta_{ij}$ by (14.4), and $\beta_{ij} \le \alpha_j$ because $d_{ij} \ge 0$. Hence

$$3f_i + \sum_{j\in N(i)} d(j,F) \le \sum_{j\in N(i)} (3\beta_{ij} + \alpha_j - \beta_{ij}) = \sum_{j\in N(i)} (\alpha_j + 2\beta_{ij}) \le 3 \sum_{j\in N(i)} \alpha_j.$$

Summing over $i \in F$ and adding $d(j,F) \le 3\alpha_j$ for the lonely clients gives the claim. □

Remark 14.16. The stronger guarantee, with the factor 3 on the facility cost, is what makes the algorithm useful beyond facility location: it gives an $O(1)$-approximation for k-median [JV01]. Such algorithms are called *Lagrangian multiplier preserving*. They relate a constrained problem to its Lagrangian version. The LP relaxation of k-median is

$$\min\Big\{\sum_{i,j} d_{ij}x_{ij} : \sum_i x_{ij} = 1,\ x_{ij} \le y_i,\ \sum_i y_i \le k,\ x,y \ge 0\Big\}.$$

Moving the constraint $\sum_i y_i \le k$ into the objective with a multiplier $f \ge 0$ gives

$$\min\Big\{\sum_{i,j} d_{ij}x_{ij} + f\Big(\sum_i y_i - k\Big) : \sum_i x_{ij} = 1,\ x_{ij} \le y_i,\ x,y \ge 0\Big\},$$

which, up to the constant $-fk$, is the facility location LP with every facility cost equal to f. See Section 7.7 of [WS11] for the details.

14.5 Summary

The primal-dual technique raises primal and dual solutions in coordinated ways. It constructs a primal solution, and at the same time it raises a dual solution that proves that every solution must cost at least that much; alternatively, the dual is money that pays for the primal. It can also be viewed as enforcing approximate complementary slackness. Several problems have primal-dual algorithms: Steiner trees and forests, facility location and k-median, and network design problems. There are also primal-dual algorithms for problems in P. Since they do not use an LP solver, primal-dual algorithms can be faster than LP rounding algorithms.

15 Introduction to Hardness of Approximation: Vertex Cover and Sparsest Cut

Lecture of October 4, 2021, by Pravesh K. Kothari. Based on the lecturer's handwritten notes.

Approximation algorithms show how well an NP-hard problem can be approximated in polynomial time; hardness of approximation shows, assuming P $\neq$ NP, which approximation ratios are out of reach. This lecture recalls mapping reductions and NP-completeness, introduces gap versions of 3SAT, and takes as starting points the PCP theorem and Håstad's optimal 7/8 hardness for 3SAT. One reduction, the conflict graph, then shows that approximating Maximum Independent Set within $7/8 + \varepsilon$ and Minimum Vertex Cover within $49/48 - \delta$ is NP-hard, and we survey the stronger results known for Vertex Cover and Max-Cut. Finally, a reduction from Max-Cut on graphs of maximum degree 3 shows that the non-uniform sparsest cut problem of Lecture 7 is NP-hard to approximate within some constant factor $c > 1$, and we compare the known algorithms and hardness results for the problems of the course so far.

15.1 Reductions and NP-completeness

We recall the basic notions of undergraduate algorithms and complexity: the class P of languages decidable in polynomial time, the class NP, and polynomial-time mapping reductions between languages, such as the classical reduction from 3SAT to Vertex Cover, written

$$\text{3SAT} \leq_m^p \text{Vertex Cover}.$$

Throughout this lecture, a *3SAT formula* ψ on the variables $x_1, \ldots, x_n$ is a conjunction of $m \geq 1$ clauses $C_1, \ldots, C_m$, each of which is a disjunction of exactly three literals on three distinct variables; a literal is a variable x_i or its negation $\bar{x}_i$. An assignment gives each variable a value in $\{0, 1\}$, and 3SAT is the language of satisfiable 3SAT formulas.

Definition 15.1 (Mapping reduction)**.** Let A and B be languages. We write $A \leq_m^p B$ if there is a polynomial-time algorithm R such that, for every input x,

$$x \in A \iff R(x) \in B;$$

that is, $R(x) \in B$ for every $x \in A$, and $R(x) \notin B$ for every $x \notin A$. The algorithm R is a *(polynomial-time mapping) reduction* from A to B.

Definition 15.2. Let $G = (V, E)$ be a graph. A set $S \subseteq V$ is *independent* if no edge of G has both endpoints in S. We write $\mathrm{mis}(G)$ for the largest size of an independent set of G. The *Maximum Independent Set* problem asks, given G, for an independent set of size $\mathrm{mis}(G)$, and the language k-Independent Set consists of the pairs (G, k) with $\mathrm{mis}(G) \geq k$.

For $A =$ 3SAT and $B = k$-Independent Set, a reduction maps each formula ψ to a pair $R(\psi) = (G, k)$ such that G has an independent set of size at least k if ψ is satisfiable, and $\mathrm{mis}(G) < k$ if ψ is unsatisfiable. Theorem 15.9 below gives such a reduction, with $k = m$: an unsatisfiable formula ψ has $\mathrm{val}(\psi) \leq 1 - 1/m$, so its graph has $\mathrm{mis}(G) \leq m - 1$.

Proposition 15.3. *Let $A \leq_m^p B$. If there is a polynomial-time algorithm for B, then there is one for A. Consequently, if* P $\neq$ NP*, then there is no polynomial-time algorithm for k-Independent Set.*

Proof. To decide whether $x \in A$, compute $R(x)$, whose length is polynomial in the length of x, and run the algorithm for B on it; by Definition 15.1 its answer is correct for x. For the second statement, if P $\neq$ NP then 3SAT has no polynomial-time algorithm, because it is NP-complete (Fact 15.4), and the first statement applied to the reduction from 3SAT to k-Independent Set shows that k-Independent Set has none either. □

Fact 15.4 (Cook–Levin [Coo71])**.** *SAT is NP-complete: it lies in* NP*, and $A \leq_m^p$ SAT for every language $A \in$ NP. The same holds for 3SAT.*

The proof works at the level of machines: it encodes the computation of a nondeterministic polynomial-time Turing machine on a given input as a formula. Two lessons came out of this theorem. First, SAT served as an excellent starting point for reductions: to show that a problem is NP-hard, it suffices to reduce SAT to it. Second, most natural problems turned out either to be NP-complete or to have polynomial-time algorithms, so NP-completeness sorts them into essentially two types.

15.2 Hardness of approximation and the PCP theorem

Hardness of approximation gives a more granular picture, with many classes of problems. Some NP-hard problems have constant-factor approximation algorithms, some have $O(\log n)$-approximations, and some only n^{ε}-approximations; for others one studies the whole curve of pairs (c, s) for which instances of value at least c can be told apart from instances of value at most s, or bicriteria approximations. To prove such results we need two things: a good starting point, and reductions that preserve approximation.

Definition 15.5. For a 3SAT formula ψ with m clauses, $\mathrm{val}(\psi)$ is the largest fraction of the clauses of ψ that an assignment satisfies. For $0 \leq s < c \leq 1$, the (c, s)*-3SAT* problem is: given a formula ψ with $\mathrm{val}(\psi) \geq c$ or $\mathrm{val}(\psi) \leq s$, decide which of the two holds. We say that (c, s)-3SAT is *NP-hard* if there is a polynomial-time algorithm R, mapping 3SAT formulas to 3SAT formulas, such that $\mathrm{val}(R(\psi)) \geq c$ for every satisfiable ψ and $\mathrm{val}(R(\psi)) \leq s$ for every unsatisfiable ψ.

This is the c-vs-s decision problem of Lecture 1 for Max 3SAT, except that the second case allows $\mathrm{val}(\psi) = s$. If (c, s)-3SAT is NP-hard, then a polynomial-time algorithm for it would decide 3SAT, through R, and hence P = NP by Fact 15.4. For a maximization problem with optimum OPT, a *ρ-approximation* ($\rho \leq 1$) outputs a solution of value at least $\rho \cdot \mathrm{OPT}$, and for a minimization problem ($\rho \geq 1$) a solution of value at most $\rho \cdot \mathrm{OPT}$, as in Lecture 1. We say that *approximating a problem within ρ is NP-hard* if a polynomial-time ρ-approximation algorithm for it would imply P = NP.

Theorem 15.6 (PCP theorem; Arora, Lund, Motwani, Sudan and Szegedy, and Arora and Safra [ALM+98, AS98])**.** *There is a constant $s < 1$ such that $(1, s)$-3SAT is NP-hard. Equivalently, there is a polynomial-time algorithm R such that $R(\psi)$ is satisfiable for every satisfiable 3SAT formula ψ, and no assignment satisfies more than an s fraction of the clauses of $R(\psi)$ for every unsatisfiable ψ.*

The constant s that the original proofs give is very close to 1. It cannot be smaller than 7/8: by Proposition 15.8 below, every 3SAT formula has value at least 7/8, while R maps each unsatisfiable formula to a formula of value at most s, and unsatisfiable formulas exist, such as the conjunction of the eight clauses on x_1, x_2, x_3. The PCP theorem led to a huge number of new

hardness of approximation results; see the lecture notes of Guruswami and O'Donnell [GO05] for its proof and Moshkovitz's survey [Mos12] for its story. It raises two questions: whether one can prove a better hardness result, that is, NP-hardness of $(1, s)$-3SAT for a smaller s, and whether some polynomial-time algorithm matches it. Raz's parallel repetition theorem [Raz98] yields strong hardness of approximation for the label cover problem (Lecture 3), and from it Håstad [Hås01] proved optimal inapproximability results for 3SAT, 3XOR, k-SAT and other problems.

Theorem 15.7 (Håstad [Hås01])**.** *For every $\varepsilon \in (0, 1/8)$, $(1, 7/8 + \varepsilon)$-3SAT is NP-hard.*

We state this theorem without proof. The threshold $7/8$ is optimal because of the random assignment.

Proposition 15.8. *Let ψ be a 3SAT formula with m clauses. A uniformly random assignment satisfies each clause of ψ with probability exactly $7/8$, and hence satisfies $7m/8$ clauses in expectation. In particular, $\mathrm{val}(\psi) \geq 7/8$. Moreover, an assignment satisfying at least $7m/8$ clauses can be found in deterministic polynomial time.*

Proof. A clause is unsatisfied exactly when its three literals are all false. Its three variables are distinct, so they receive independent uniform values, and this happens with probability $(1/2)^3 = 1/8$. Linearity of expectation gives $7m/8$ satisfied clauses in expectation, so some assignment satisfies at least $7m/8$ clauses. To find one, we use the method of conditional expectations. For a partial assignment a to $x_1, \ldots, x_i$, let $f(a)$ be the expected number of satisfied clauses when the remaining variables are set uniformly at random. A clause contributes 1 to $f(a)$ if a makes one of its literals true, 0 if a makes all three literals false, and $1 - 2^{-k}$ otherwise, where k is the number of its literals on unset variables; hence $f(a)$ is computable in polynomial time. Since $f(a)$ is the average of $f(a, x_{i+1} = 0)$ and $f(a, x_{i+1} = 1)$, one of these is at least $f(a)$. Setting $x_1, \ldots, x_n$ one at a time to the better value never decreases f, which starts at $7m/8$, so the final assignment satisfies at least $7m/8$ clauses. □

Thus the threshold $7/8$ in Theorem 15.7 cannot be lowered, and the algorithm of Proposition 15.8 is a $7/8$-approximation for Max 3SAT, while by Theorem 15.7 approximating Max 3SAT within $7/8 + \varepsilon$ is NP-hard for every $\varepsilon \in (0, 1/8)$, by the argument of Corollary 15.13 below. The 3XOR problem is similar: its instances are systems of linear equations $x_i + x_j + x_k \equiv b \pmod 2$ on three distinct variables each, a random assignment satisfies each equation with probability $1/2$, and Håstad [Hås01] proved that for every $\varepsilon \in (0, 1/4)$ it is NP-hard to tell instances in which a $1 - \varepsilon$ fraction of the equations can be satisfied from instances in which at most a $1/2 + \varepsilon$ fraction can.

15.3 Independent set and vertex cover

We now start from Theorem 15.7 and prove hardness of approximation, first for Maximum Independent Set and then, using it, for Minimum Vertex Cover.

Theorem 15.9. *There is a polynomial-time algorithm that takes as input a 3SAT formula ψ on n variables with m clauses and outputs a graph G on $7m$ vertices such that the following hold.*

1) If ψ is satisfiable, then G has an independent set of size m.

2) For every $s \in [0,1]$, if $\mathrm{val}(\psi) \le s$, then every independent set of G has size at most sm. In particular, for every $\varepsilon \in (0, 1/8)$, if $\mathrm{val}(\psi) \le 7/8 + \varepsilon$, then every independent set of G has size at most $(7/8 + \varepsilon)m$.

The graph is the conflict graph of ψ. For a clause C, let $\mathrm{var}(C)$ be its set of three variables. A *local assignment* to C is a map $\alpha\colon \mathrm{var}(C) \to \{0,1\}$; it *satisfies* C if it makes at least one literal of C true. Every clause, for instance $x_i \vee \bar{x}_j \vee x_k$, has exactly 7 satisfying local assignments: of the 8 local assignments, only the one making all three literals false fails. Two local assignments α to C and α' to C' *conflict* if $\alpha(x) \neq \alpha'(x)$ for some variable $x \in \mathrm{var}(C) \cap \mathrm{var}(C')$.

Definition 15.10 (Conflict graph)**.** Let ψ be a 3SAT formula with clauses $C_1, \dots, C_m$. Its *conflict graph* G_ψ has a vertex (j, α) for every $j \in [m]$ and every local assignment α to C_j that satisfies C_j. Two distinct vertices (j, α) and (j', α') are adjacent if $j = j'$, or if $j \neq j'$ and α and α' conflict.

Thus the 7 vertices of each clause form a clique, and vertices of different clauses are adjacent when their local assignments disagree on a shared variable. The graph has $7m$ vertices, and it can be built in time polynomial in m by checking each of the fewer than $(7m)^2$ pairs of vertices.

Claim 15.11 (Completeness)**.** *If ψ has a satisfying assignment x, then G_ψ has an independent set of size m.*

Proof. For each $j \in [m]$, let x_{C_j} be the local assignment to C_j induced by x, that is, the restriction of x to $\mathrm{var}(C_j)$. It satisfies C_j because x does, so (j, x_{C_j}) is a vertex. Let S consist of these m vertices. They belong to distinct clauses, and any two of them are restrictions of the same assignment x, so x_{C_j} does not conflict with $x_{C_{j'}}$ for any j'. Hence S is an independent set of size m. □

Claim 15.12 (Soundness)**.** *Let $s \in [0,1]$. If G_ψ has an independent set S of size at least sm, then some assignment satisfies at least an s fraction of the clauses of ψ.*

Proof. For each j, the vertices $(j, \cdot)$ form a clique, so S contains at most one of them. Let J be the set of indices j for which S contains a vertex (j, α_j); then $|J| = |S|$. Since S is independent, the local assignments α_j with $j \in J$ pairwise do not conflict: every variable that lies in $\mathrm{var}(C_j)$ for some $j \in J$ receives the same value from all of them. Let x give each such variable this common value and every other variable the value 0. Then x agrees with α_j on $\mathrm{var}(C_j)$ for every $j \in J$, so x satisfies every clause C_j with $j \in J$. These are $|S| \ge sm$ clauses. □

Proof of Theorem 15.9. The algorithm outputs $G = G_\psi$, which has $7m$ vertices and is computed in polynomial time. Part 1 is Claim 15.11. For part 2, suppose that $\mathrm{val}(\psi) \le s$ and that S is an independent set of G. By Claim 15.12 applied with $|S|/m$ in place of s, some assignment satisfies at least $|S|$ clauses, so $|S| \le \mathrm{val}(\psi) \cdot m \le sm$. □

Corollary 15.13. *For every $\varepsilon \in (0, 1/8)$, approximating Maximum Independent Set within $7/8 + \varepsilon$ is NP-hard.*

Proof. Suppose that a polynomial-time algorithm A outputs, on every graph G, an independent set of size at least $(7/8 + \varepsilon)\,\mathrm{mis}(G)$, and let $\varepsilon' = \varepsilon/2$. Given an instance ψ of $(1, 7/8 + \varepsilon')$-3SAT with m clauses, we build $G = G_\psi$, run A on it, and answer that $\mathrm{val}(\psi) = 1$ if and only if the output has more than $(7/8 + \varepsilon')m$ vertices. If ψ is satisfiable, then $\mathrm{mis}(G) \ge m$ by Theorem 15.9,

so the output has at least $(7/8+\varepsilon)m > (7/8+\varepsilon')m$ vertices. If $\mathrm{val}(\psi) \le 7/8+\varepsilon'$, then every independent set of G has at most $(7/8+\varepsilon')m$ vertices. So $(1, 7/8+\varepsilon')$-3SAT is solved in polynomial time, and P = NP by Theorem 15.7. □

Definition 15.14. Let $G = (V, E)$ be a graph with n vertices and m edges. A set $S \subseteq V$ is a *vertex cover* if it covers every edge, that is, $S \cap \{i, j\} \neq \emptyset$ for every $\{i, j\} \in E$. We write $\mathrm{vc}(G)$ for the smallest size of a vertex cover, and the *Minimum Vertex Cover* problem asks for a vertex cover of size $\mathrm{vc}(G)$.

Lemma 15.15. *A set $S \subseteq V$ is a vertex cover of G if and only if $V \setminus S$ is an independent set of G. Thus $\mathrm{vc}(G) = n - \mathrm{mis}(G)$.*

Proof. Both conditions say that no edge has both endpoints in $V \setminus S$. Taking complements therefore matches vertex covers of size k with independent sets of size $n-k$, and the smallest cover corresponds to the largest independent set. □

Corollary 15.16. *For every $\delta \in (0, 1/48)$, approximating Minimum Vertex Cover within $49/48-\delta$ is NP-hard. In particular, approximating Minimum Vertex Cover within 1.02 is NP-hard.*

Proof. We use the same reduction $\psi \mapsto G = G_\psi$ as for independent set; G has $7m$ vertices. Let $\varepsilon \in (0, 1/8)$. If ψ has a satisfying assignment, then G has an independent set of size at least m, so $\mathrm{vc}(G) \le 7m - m = 6m$ by Lemma 15.15. If no assignment satisfies more than a $7/8+\varepsilon$ fraction of the clauses of ψ, then every independent set of G has size at most $(7/8+\varepsilon)m$, so $\mathrm{vc}(G) \ge 7m - (7/8+\varepsilon)m = 6m + \frac{1}{8}m - \varepsilon m$. The ratio of the two thresholds is

$$\frac{6m + \frac{1}{8}m - \varepsilon m}{6m} = \frac{49}{48} - \frac{\varepsilon}{6}.$$

Now let $\delta \in (0, 1/48)$, apply the above with $\varepsilon = 3\delta \in (0, 1/16)$, and suppose that a polynomial-time algorithm A outputs, on every graph G, a vertex cover of size at most $(49/48 - \delta)\,\mathrm{vc}(G)$. Given an instance ψ of $(1, 7/8+\varepsilon)$-3SAT, we run A on G_ψ and answer that ψ is satisfiable if and only if the output has fewer than $(6 + \frac{1}{8} - \varepsilon)m$ vertices. If ψ is satisfiable, the output has at most

$$\Big(\frac{49}{48} - \delta\Big)6m < \Big(\frac{49}{48} - \frac{\delta}{2}\Big)6m = \Big(6 + \frac{1}{8} - \varepsilon\Big)m$$

vertices. If $\mathrm{val}(\psi) \le 7/8+\varepsilon$, every vertex cover has at least $(6 + \frac{1}{8} - \varepsilon)m$ vertices. So A solves $(1, 7/8+\varepsilon)$-3SAT, and P = NP by Theorem 15.7. Finally, $49/48 = 1.0208\ldots > 1.02$, so a polynomial-time 1.02-approximation would be a $(49/48-\delta)$-approximation for $\delta = 49/48 - 1.02$, and the second statement follows. □

Using other constraint satisfaction problems in place of 3SAT as the starting point improves the factor 1.02 to 1.16 [Hås01]. Stronger results are known.

Fact 15.17. *The following hold.*

1) *(Dinur and Safra [DS05].) Approximating Minimum Vertex Cover within 1.34 is NP-hard. The proof requires some heavy lifting.*

2) *(Khot, Minzer and Safra, and Dinur, Khot, Kindler, Minzer and Safra [KMS17, DKK+18a, KMS18].) For every $\varepsilon > 0$, approximating Minimum Vertex Cover within $\sqrt{2}-\varepsilon$ is NP-hard. This follows from the proof of the 2-to-2 Games Conjecture.*

15.4 Max-Cut

Recall from Lecture 4 that $\mathrm{maxcut}(G)$ is the largest fraction of the edges of a graph G that a cut $(S, \bar{S})$ separates, and that the Goemans–Williamson algorithm [GW95] is an α_{GW}-approximation with $\alpha_{\mathrm{GW}} \approx 0.878$.

Theorem 15.18 (Håstad [Hås01]; Trevisan, Sorkin, Sudan and Williamson [TSSW00])**.** *For every $\varepsilon > 0$, approximating Max-Cut within $16/17 + \varepsilon$ is NP-hard. Here $16/17 = 0.9411\ldots$.*

The proof, which we omit, is a reduction from 3XOR: each equation is replaced by a small graph, a gadget whose vertices can be labeled by the points of a hypercube, found by Trevisan, Sorkin, Sudan and Williamson [TSSW00]. Max-Cut remains hard even on graphs of bounded degree.

Fact 15.19 (Alimonti and Kann [AK00])**.** *There is a constant $\varepsilon_0 \in (0, 1)$ such that approximating Max-Cut within $1 - \varepsilon_0$ on graphs whose vertices all have degree at most 3 is NP-hard.*

15.5 Non-uniform sparsest cut

We use the sparsest cut problem with general demands of Lecture 7. An instance is a graph $H = (V_H, E_H)$ with capacities $c_e \geq 0$ on its edges and demands $D_{ij} = D_{ji} \geq 0$ on the pairs of distinct vertices, not all zero. For $S \subseteq V_H$, $\mathrm{cap}(S, \bar{S})$ is the total capacity of the edges with exactly one endpoint in S, and $\mathrm{dem}(S, \bar{S}) = \sum_{i \in S,\, j \in \bar{S}} D_{ij}$. A cut $(S, \bar{S})$ with $\mathrm{dem}(S, \bar{S}) > 0$ has *sparsity*

$$\Phi(S) = \frac{\mathrm{cap}(S, \bar{S})}{\mathrm{dem}(S, \bar{S})},$$

and $\Phi^*(H)$ is the minimum of $\Phi(S)$ over all such cuts. Since cap and dem are symmetric, $\Phi(S) = \Phi(V_H \setminus S)$. A ρ-approximation algorithm, with $\rho \geq 1$, outputs a cut S with $\mathrm{dem}(S, \bar{S}) > 0$ and $\Phi(S) \leq \rho\, \Phi^*(H)$.

Theorem 15.20. *There is a constant $c > 1$ such that approximating the non-uniform sparsest cut problem within c is NP-hard. More precisely, with ε_0 as in Fact 15.19, this holds for every $c \in [1, 1 + \varepsilon_0/18)$.*

The proof is a reduction from Max-Cut. Let $G = (V_G, E_G)$ be an instance of Max-Cut with $n = |V_G|$ vertices and $m = |E_G| \geq 1$ edges.

Definition 15.21. The *sparsest cut instance* H of G has vertex set $V_H = V_G \cup \{s, t\}$, where s and t are two new vertices, a special source and sink. Its edges are $\{s, v\}$ and $\{t, v\}$ for every $v \in V_G$, each of capacity 1. The demands are $D_{uv} = 1$ for every edge $\{u, v\} \in E_G$ and $D_{st} = 6m$, and all other demands are 0 (Fig. 26).

The instance H has $n + 2$ vertices, $2n$ edges and $m + 1$ pairs with nonzero demand, so it is computed from G in polynomial time. For $\gamma \in [1/2, 1]$ we set

$$\tau(\gamma) = \frac{n}{6m + \gamma m}.$$

Lemma 15.22 (Completeness)**.** *If $\mathrm{maxcut}(G) \geq \gamma$, then H has a cut of sparsity at most $\tau(\gamma)$.*

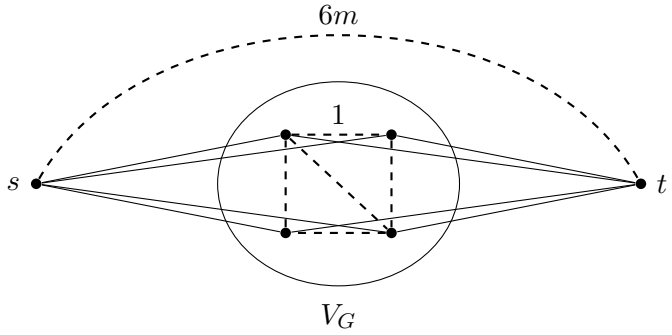


Figure 26: The sparsest cut instance H of Definition 15.21, for a graph G on four vertices. Solid lines are the edges of H, each of capacity 1. Dashed lines are the pairs with nonzero demand: the edges of G, with demand 1 each, and the pair $\{s,t\}$, with demand $6m$.

Proof. Let $S \subseteq V_G$ be an optimal max-cut of G, so that $|E_G(S, V_G \setminus S)| \geq \gamma m$, and consider the cut defined by $S' = \{s\} \cup S$ in H. For every $v \in V_G$, exactly one of the edges $\{s,v\}$ and $\{t,v\}$ crosses the cut: the first if $v \notin S$, and the second if $v \in S$. Hence $\mathrm{cap}(S', \bar{S}') = n$. The demand pairs separated by the cut are $\{s,t\}$ and the edges of $E_G(S, V_G \setminus S)$, so the separated demand is $6m + |E_G(S, V_G \setminus S)| \geq 6m + \gamma m$. Therefore

$$\Phi(S') = \frac{n}{6m + |E_G(S, V_G \setminus S)|} \leq \frac{n}{6m + \gamma m} = \tau(\gamma). \qquad \square$$

Lemma 15.23 (Soundness)**.** *Suppose that every vertex of G has degree at most 3 and that G has no isolated vertices. Let $\gamma \in [1/2, 1]$ and $\varepsilon \in (0,1)$. If $\mathrm{maxcut}(G) \leq (1-\varepsilon)\gamma$, then every cut of H has sparsity at least $(1 + \varepsilon/18)\,\tau(\gamma)$. More precisely, if a cut S' of H with $\mathrm{dem}(S', \bar{S}') > 0$ has $\Phi(S') < (1 + \varepsilon/18)\,\tau(\gamma)$, then S' separates s from t, and the vertices of G on the side of s form a set S with $|E_G(S, V_G \setminus S)| > (1-\varepsilon)\gamma m$.*

The first statement follows from the second, since under its hypothesis no cut of G separates more than $(1-\varepsilon)\gamma m$ edges.

Proof. Let S' be a cut of H with $\mathrm{dem}(S', \bar{S}') > 0$. We distinguish two cases.

Case 1: S' separates s from t. Since $\Phi(S') = \Phi(V_H \setminus S')$, we may assume that $s \in S'$. As in the proof of Lemma 15.22, $\mathrm{cap}(S', \bar{S}') = n$, and with $S = S' \cap V_G$ the separated demand is $6m + |E_G(S, V_G \setminus S)|$. Suppose that $|E_G(S, V_G \setminus S)| \leq (1-\varepsilon)\gamma m$. Then

$$\begin{aligned} \Phi(S') &\geq \frac{n}{6m + (1-\varepsilon)\gamma m} = \frac{n}{(1-\varepsilon)(6m + \gamma m) + 6\varepsilon m} \\ &= \frac{\tau(\gamma)}{1 - \varepsilon\gamma/(6+\gamma)} \geq \Big(1 + \frac{\varepsilon\gamma}{6+\gamma}\Big)\tau(\gamma) \geq \Big(1 + \frac{\varepsilon}{13}\Big)\tau(\gamma) \geq \Big(1 + \frac{\varepsilon}{18}\Big)\tau(\gamma). \end{aligned}$$

The second inequality is $1/(1-y) \geq 1 + y$ for $y \in [0,1)$, and the third holds because $\gamma/(6+\gamma)$ is increasing in γ and equals $1/13$ at $\gamma = 1/2$. Hence, if $\Phi(S') < (1 + \varepsilon/18)\tau(\gamma)$, then $|E_G(S, V_G \setminus S)| > (1-\varepsilon)\gamma m$.

Case 2: s and t lie on the same side. Replacing S' by its complement if necessary, we may assume that $s, t \notin S'$, so that $\emptyset \neq S' \subseteq V_G$. The edges of H that cross the cut are $\{s,v\}$ and $\{t,v\}$ for $v \in S'$, so $\mathrm{cap}(S', \bar{S}') = 2|S'|$. The separated demand is the number of edges of G with

exactly one endpoint in S', which is at most the sum of the degrees of the vertices of S'; since all degrees are at most 3, it is at most $3|S'|$. Hence $\Phi(S') \geq 2/3$. On the other hand, G has no isolated vertices, so $2m = \sum_{v \in V_G} \deg(v) \geq n$, that is, $m \geq n/2$; together with $\gamma \geq 1/2$ this gives

$$\tau(\gamma) = \frac{n}{6m + \gamma m} \leq \frac{2}{6 + \gamma} \leq \frac{4}{13}.$$

Therefore

$$\Phi(S') \geq \frac{2}{3} = \frac{13}{6} \cdot \frac{4}{13} \geq \frac{13}{6}\,\tau(\gamma) \geq \Big(1 + \frac{\varepsilon}{18}\Big)\tau(\gamma),$$

because $1 + \varepsilon/18 < 19/18 < 13/6$. So a cut with $\Phi(S') < (1 + \varepsilon/18)\tau(\gamma)$ falls under Case 1, which proves both statements. □

Proof of Theorem 15.20. Let $c \in [1, 1 + \varepsilon_0/18)$, and suppose that a polynomial-time algorithm A outputs, on every instance of the non-uniform sparsest cut problem, a cut of sparsity at most c times the optimum. We build a polynomial-time $(1 - \varepsilon_0)$-approximation for Max-Cut on graphs of maximum degree 3, which implies P = NP by Fact 15.19.

Let G be such a graph. Deleting its isolated vertices changes neither m nor the number of edges that any cut separates, and a cut of the smaller graph extends to G by placing the deleted vertices on either side; if $m = 0$, every cut is optimal. So we may assume that $m \geq 1$ and that G has no isolated vertices. We build H as in Definition 15.21, run A on it to get a cut S', and output the set S of vertices of G on the side of s. Let $\gamma = \operatorname{maxcut}(G)$. A uniformly random set separates each edge with probability $1/2$, so $\gamma \geq 1/2$ (Lecture 4). By Lemma 15.22, $\Phi^*(H) \leq \tau(\gamma)$, so

$$\Phi(S') \leq c\,\tau(\gamma) < \Big(1 + \frac{\varepsilon_0}{18}\Big)\tau(\gamma).$$

By Lemma 15.23 with $\varepsilon = \varepsilon_0$, the cut S' separates s from t, and $|E_G(S, V_G \setminus S)| > (1 - \varepsilon_0)\gamma m = (1 - \varepsilon_0)\operatorname{maxcut}(G) \cdot m$, as required. □

15.6 What is known

The following table compares the best known approximation ratio with a ratio within which approximation is NP-hard, for every $\varepsilon > 0$ where ε appears (Theorems 15.18 and 15.20 and Fact 15.17). For Max-Cut the ratios are at most 1, and for the other problems at least 1.

Problem	Algorithm	NP-hardness
Max-Cut	$\alpha_{\mathrm{GW}} = 0.878\ldots$ [GW95]	$16/17 + \varepsilon$, with $16/17 = 0.941\ldots$
Vertex Cover	2	$\sqrt{2} - \varepsilon$, with $\sqrt{2} = 1.414\ldots$
Sparsest Cut (non-uniform)	$O(\sqrt{\log n}\,\log\log n)$ [ALN08]	some $c > 1$ (Theorem 15.20)
Sparsest Cut (uniform)	$O(\sqrt{\log n})$ [ARV09]	none known

The uniform sparsest cut algorithm is the Arora–Rao–Vazirani algorithm of Lectures 10 and 11, and the non-uniform bound is due to Arora, Lee and Naor, who build on the same semidefinite program and structure theorem. For some problems, such as 3SAT and 3XOR, the hardness of approximation is optimal: it matches the guarantee of the random assignment (Theorem 15.7 and Proposition 15.8). For the problems in the table, the algorithms and the hardness results are still far apart.

16 The Unique Games Conjecture and Boolean Fourier Analysis

Lecture of October 6, 2021, by Pravesh K. Kothari. Based on the lecturer's handwritten notes.

This lecture introduces the Unique Games Conjecture of Khot, which asserts that it is NP-hard to tell nearly satisfiable systems of linear equations modulo a large prime, with two variables per equation, from systems in which only a small fraction of the equations can be satisfied. Assuming it, Khot, Kindler, Mossel and O'Donnell proved that it is NP-hard to approximate Max-Cut better than the Goemans–Williamson algorithm of Lecture 4 does; this lecture and Lectures 17 and 18 develop the proof. The reduction is built from gadgets, as in Lecture 15, and the gadget for Max-Cut is the hypercube graph of Lecture 5, whose cuts are functions on the Boolean cube. To analyze such functions we develop the basics of their Fourier analysis: the expansion in the basis of parity functions, Parseval's identity, the influence of a variable, and low-degree influences, with the lemma that a bounded function has only a few variables of large low-degree influence.

16.1 The problem and the conjecture

The Unique Games Conjecture posits that a natural approximation problem is NP-hard. Together with P $\neq$ NP, it implies that this problem has no polynomial-time algorithm, a hypothesis that in turn implies P $\neq$ NP; in this sense it strengthens the hypothesis P $\neq$ NP. Its truth is not known. If it holds, it yields optimal hardness of approximation results for many problems, among them Max-Cut [KKMO07] and Vertex Cover [KR08]. Around it has grown a rich theory that connects hardness of approximation, the design of algorithms based on semidefinite programs, and methods from the analysis of Boolean functions and from geometry.

Definition 16.1. Let p be a prime, and let $\mathbb{F}_p = \{0, 1, \ldots, p-1\}$ be the field of integers modulo p. An instance I of *Max-2-Lin*(p) consists of variables $x_1, \ldots, x_n$ with values in $\mathbb{F}_p$ and a list of $m \geq 1$ equations, each of the form

$$x_i + x_j = b \pmod{p}, \qquad \text{with } i \neq j \text{ and } b \in \mathbb{F}_p.$$

An assignment $x \in \mathbb{F}_p^n$ *satisfies* an equation if the equation holds at x. The *value* $\mathrm{val}(I)$ is the largest fraction of the equations of I that an assignment satisfies.

In the name, p is the size of the field and 2 is the number of variables in each equation. The goal is to find an assignment that satisfies as large a fraction of the equations as possible. A uniformly random assignment satisfies a $1/p$ fraction of the equations in expectation: whatever the value of x_i, exactly one of the p values of x_j satisfies $x_i + x_j = b$, so each equation holds with probability exactly $1/p$. In particular, $\mathrm{val}(I) \geq 1/p$ for every instance I. Satisfiable instances, on the other hand, are easy.

Theorem 16.2. *There is a polynomial-time algorithm that, given an instance of Max-2-Lin*(p) *of value* 1*, finds an assignment that satisfies all of its equations.*

Proof. An instance is a system of m linear equations in n unknowns over the field $\mathbb{F}_p$, and value 1 means that the system has a solution. Gaussian elimination over $\mathbb{F}_p$ finds a solution of a solvable linear system with $O(mn \min(m, n))$ arithmetic operations in $\mathbb{F}_p$, each of which takes time polynomial in $\log p$. □

The conjecture asserts that the problem becomes hard as soon as a small fraction of the equations cannot be satisfied. We state it with the gap problems of Lecture 15.

Definition 16.3. Let p be a prime and $0 \le s < c \le 1$. The (c, s)-*Max-2-Lin*(p) problem is: given an instance I of Max-2-Lin(p) with $\mathrm{val}(I) \ge c$ or $\mathrm{val}(I) \le s$, decide which of the two holds. As in Lecture 15, we say that it is *NP-hard* if there is a polynomial-time algorithm R, mapping 3SAT formulas to instances of Max-2-Lin(p), such that $\mathrm{val}(R(\psi)) \ge c$ for every satisfiable ψ and $\mathrm{val}(R(\psi)) \le s$ for every unsatisfiable ψ.

Conjecture 16.4 (Unique Games Conjecture [Kho02])**.** *For every ε with $0 < \varepsilon < 1/2$ there is p_0 such that for every prime $p \ge p_0$, the $(1-\varepsilon, \varepsilon)$-Max-2-Lin($p$) problem is NP-hard.*

In words, assuming the conjecture, for every small $\varepsilon > 0$ and every large prime p, no polynomial-time algorithm that is given an instance of Max-2-Lin(p) of value at least $1-\varepsilon$ finds an assignment satisfying more than an ε fraction of its equations, unless P = NP. Indeed, such an algorithm would decide the $(1-\varepsilon, \varepsilon)$ problem, because on an instance of value at most ε no assignment satisfies more than an ε fraction of the equations.

Remark 16.5. Khot [Kho02] formulated the conjecture for *unique games*. A unique game with *alphabet size* k has variables with values in a set Σ of size k and a list of constraints on pairs of variables, each of the form $x_j = \pi(x_i)$ for a bijection $\pi\colon \Sigma \to \Sigma$; its value is the largest fraction of the constraints that an assignment satisfies. An equation $x_i + x_j = b$ is such a constraint, with $\pi(a) = b - a$, so an instance of Max-2-Lin(p) is a unique game with alphabet size p. Khot, Kindler, Mossel and O'Donnell [KKMO07] showed that Khot's conjecture is equivalent to the following statement: for every $\varepsilon > 0$ and all large q, it is NP-hard to distinguish systems of equations of the form $x_i - x_j = b \pmod q$ of value at least $1-\varepsilon$ from those of value at most ε. Such a system I reduces to a system I' of equations of the form used here, with $\mathrm{val}(I) \le \mathrm{val}(I') \le 2\,\mathrm{val}(I)$: give every variable x_i two copies x_i^+ and x_i^-, standing for x_i and $-x_i$, and replace each equation $x_i - x_j = b$ by the two equations $x_i^+ + x_j^- = b$ and $x_j^+ + x_i^- = -b$. An assignment x of I gives the assignment $x_i^+ = x_i$, $x_i^- = -x_i$ of I', which satisfies both copies of every equation that x satisfies. Conversely, given an assignment of I', set each x_i to the value of x_i^+ or to minus the value of x_i^-, with probability $1/2$ each, independently. Each satisfied equation of I' then yields a satisfied equation of I with probability at least $1/4$, through disjoint events for the two copies, so the expected number of satisfied equations of I is at least a quarter of the number of satisfied equations of I', which has twice as many equations. Hence the formulation above is equivalent to Khot's.

The best polynomial-time algorithm known is due to Charikar, Makarychev and Makarychev [CMM06]. Given a unique game with alphabet size k and value at least $1-\varepsilon$, it finds an assignment of value at least $1 - O(\sqrt{\varepsilon \log k})$. The loss depends on the alphabet size k. In particular, for fixed ε the guarantee exceeds ε only when $\log k = O(1/\varepsilon)$, so the algorithm does not contradict the conjecture, which concerns large p.

Unique games can also be solved faster than by brute-force search over all k^n assignments. Arora, Barak and Steurer [ABS15] gave, for an absolute constant c and every $0 < \varepsilon < 1$, an algorithm that runs in time $\exp(kn^{\varepsilon})$ and, given a unique game with alphabet size k on n variables of value at least $1-\varepsilon^c$, outputs an assignment of value at least $1-\varepsilon$. For fixed ε the running time is subexponential in n, and, unlike the previous guarantee, the value of the output does not depend on the alphabet size.

There is much further work, on both algorithms and lower bounds. One of the most important lower bounds, proved in 2018, is the following.

Theorem 16.6 (2-to-2 theorem [KMS17, DKK+18b, DKK+18a, KMS18])**.** *For every $\varepsilon > 0$ there are k and a polynomial-time algorithm R, mapping 3SAT formulas to unique games with alphabet size k, such that $\mathrm{val}(R(\psi)) \geq \frac{1}{2} - \varepsilon$ for every satisfiable ψ and $\mathrm{val}(R(\psi)) \leq \varepsilon$ for every unsatisfiable ψ.*

We state this theorem without proof. Raising the completeness $\frac{1}{2} - \varepsilon$ in it to $1 - \varepsilon$ would prove the Unique Games Conjecture, in Khot's form (Remark 16.5).

16.2 Optimal hardness for Max-Cut and gadget reductions

This lecture and Lectures 17 and 18 give a glimpse of this theory. We prove optimal NP-hardness of approximating Max-Cut, assuming the Unique Games Conjecture, in a way that shows a principled theory of such optimal hardness reductions. This theory builds hardness reductions directly from integrality gaps of semidefinite programs, for a large class of problems [Rag08]: a failure of the semidefinite program yields hardness assuming the Unique Games Conjecture.

Theorem 16.7 (Khot, Kindler, Mossel and O'Donnell [KKMO07])**.** *Assume the Unique Games Conjecture. Then for every $\varepsilon > 0$, approximating Max-Cut within $\alpha_{\mathrm{GW}} + \varepsilon$ is NP-hard, where*

$$\alpha_{\mathrm{GW}} = \min_{-1 \leq \rho < 0} \frac{\arccos(\rho)/\pi}{\frac{1}{2} - \frac{1}{2}\rho}$$

is the constant of the Goemans–Williamson algorithm [GW95].

In Lecture 5 the minimum defining α_{GW} is taken over $-1 \leq \rho < 1$; it is attained at $\rho_* \approx -0.689 < 0$, so the two definitions agree, and $\alpha_{\mathrm{GW}} \approx 0.878$. Throughout, ρ_* denotes this minimizer.

Like the examples of Lecture 15, we prove this theorem by gadget reductions; the proof is completed in Lecture 18. A *gadget reduction* R from a problem P_1 to a problem P_2 produces the instance of P_2 as a local transformation of the instance of P_1: it modifies constant-size portions of the input. For example, the reduction in Lecture 15 from 3SAT to Independent Set, which yields the hardness of Vertex Cover, replaces each clause by 7 vertices, one for each local assignment that satisfies the clause, and each edge of the resulting graph depends on at most two clauses of the 3SAT formula.

For Max-Cut, our gadgets must produce exactly the gap between $\frac{1}{2} - \frac{1}{2}\rho$ and $\arccos(\rho)/\pi$: the reduction must map instances of value close to 1 to graphs with max-cut about $\frac{1}{2} - \frac{1}{2}\rho_*$, and instances of small value to graphs with max-cut about $\arccos(\rho_*)/\pi$, whose ratio is α_{GW}. So some geometry has to show up in them. In fact, in a precise sense our gadgets come from the integrality gap and rounding gap instances for Max-Cut of Lecture 5, which we now recall.

16.3 Integrality gap and rounding gap instances

Both instances are embedded graphs (Lecture 5): weighted graphs, given by a probability distribution on pairs of vertices (the edge distribution), in which each vertex u comes with a unit vector z_u labeling it. The *objective value* of the embedding is $\mathbb{E}_{\{u,v\}}[\frac{1}{2} - \frac{1}{2}\langle z_u, z_v\rangle]$, the objective of the semidefinite relaxation of Lecture 4 at the solution $X_{uv} = \langle z_u, z_v\rangle$; so it is at most the optimum $\mathrm{SDP}(G)$ of the relaxation. The value of a cut is the probability that an edge drawn from the edge distribution crosses it.

The integrality gap instance. This is the sphere graph of Feige and Schechtman [FS02]. Its vertices are the points of the unit sphere $\mathbb{S}^{d-1} \subseteq \mathbb{R}^d$, or of a fine discretization of it, each labeled by itself. An edge is drawn by choosing u and v independently and uniformly from $\mathbb{S}^{d-1}$, conditioned on $\langle u, v\rangle \le \rho_*$. Every edge has $\langle u, v\rangle \le \rho_*$, so the objective value of the embedding is

$$\mathbb{E}_{\{u,v\}}\Big[\frac{1}{2} - \frac{1}{2}\langle u, v\rangle\Big] \ge \frac{1}{2} - \frac{1}{2}\rho_*,$$

and by Lecture 5 it exceeds $\frac{1}{2} - \frac{1}{2}\rho_*$ by only $O(1/d)$. The analysis of the max-cut of this graph has two steps, both carried out in Lecture 5.

1) The hemisphere cuts $\{u : \langle u, h\rangle \ge 0\}$, for unit vectors $h \in \mathbb{R}^d$, are maximum cuts of the graph.

2) The edge distribution is invariant under rotations, so every hemisphere cuts the same fraction of the edges as a uniformly random hemisphere, which is the random hyperplane rounding of Lecture 4. By the analysis of that rounding, a hemisphere cut has value $\mathbb{E}_{\{u,v\}}[\arccos\langle u, v\rangle]/\pi$, which tends to $\arccos(\rho_*)/\pi$ as $d \to \infty$.

Hence the ratio of the max-cut to $\mathrm{SDP}(G)$ is at most $\alpha_{\mathrm{GW}} + o(1)$ as $d \to \infty$.

The rounding gap instance. This is the hypercube graph of Lecture 5. Its vertices are the corners $\{\pm 1/\sqrt{d}\}^d$ of the hypercube, that is, the points of $\{\pm 1\}^d$ scaled down to unit vectors, each labeled by itself. An edge is drawn in three steps.

1) Choose $u \in \{\pm 1/\sqrt{d}\}^d$ uniformly at random.

2) Obtain v by flipping (negating) each coordinate of u independently with probability $\frac{1}{2} - \frac{1}{2}\rho_*$.

3) Output $\{u, v\}$.

The product $u_i v_i$ is $1/d$ with probability $\frac{1}{2} + \frac{1}{2}\rho_*$ and $-1/d$ otherwise, so $\mathbb{E}[u_i v_i] = \rho_*/d$ and $\mathbb{E}\langle u, v\rangle = \rho_*$. Hence the objective value of the embedding is $\frac{1}{2} - \frac{1}{2}\rho_*$. The true max-cut is also $\frac{1}{2} - \frac{1}{2}\rho_*$. The *dictator cuts*

$$D_i = \Big\{u : u_i = +\frac{1}{\sqrt{d}}\Big\}, \qquad i \in [d],$$

attain this value: an edge $\{u, v\}$ crosses D_i exactly when the i-th coordinate was flipped, so it crosses D_i with probability $\frac{1}{2} - \frac{1}{2}\rho_*$; and by Lecture 5 no cut does better, since $\rho_* < 0$. The random hyperplane rounding of this embedding, on the other hand, cuts in expectation only an $\arccos(\rho_*)/\pi + o(1)$ fraction of the edges as $d \to \infty$ (Lecture 5), because almost every edge has $\langle u, v\rangle$ close to ρ_*. So on this graph the relaxation is exact, and the loss comes from the rounding.

The gadget. Our gadget for Max-Cut is the hypercube graph. To analyze cuts in such graphs, it is useful to view them as functions. We identify the vertex set with $\{\pm 1\}^d$, dropping the scaling by $1/\sqrt{d}$. A set S of vertices, that is, a cut $(S, \bar{S})$, corresponds to the function $f\colon \{\pm 1\}^d \to \{\pm 1\}$ with

$$f(x) = 1 \text{ if } x \in S, \qquad f(x) = -1 \text{ if } x \notin S,$$

and every such function arises from exactly one set. The value of the cut is

$$\Pr_{\{x,y\}}[f(x) \neq f(y)],$$

where $\{x, y\}$ is drawn from the edge distribution. For example, the dictator cut D_i corresponds to the function $f(x) = x_i$. To reason about such quantities we need some machinery, the Fourier analysis of Boolean functions, which occupies the rest of this lecture.

16.4 Fourier analysis of Boolean functions

Throughout the rest of this lecture, $n \geq 1$ is the number of coordinates, and x is uniformly distributed on $\{-1,1\}^n$. A *Boolean function* is a function $f\colon \{-1,1\}^n \to \mathbb{R}$; it is *Boolean-valued* if $f(x) \in \{-1,1\}$ for every x. Fourier analysis writes f as a linear combination of some simple functions and uses this expansion to prove properties of f; see O'Donnell's book [O'D14] for much more.

Definition 16.8. For $f, g\colon \{-1,1\}^n \to \mathbb{R}$, the *inner product* and the *norm* are

$$\langle f, g\rangle \coloneqq \mathbb{E}_x[f(x)g(x)] = 2^{-n} \sum_{x\in\{-1,1\}^n} f(x)g(x), \qquad \|f\|_2 \coloneqq \sqrt{\langle f, f\rangle},$$

so that $\|f\|_2^2 = \mathbb{E}_x[f(x)^2]$.

This treats f and g as vectors of length 2^n, indexed by the points of $\{-1,1\}^n$, takes their standard inner product, and rescales it by 2^{-n}. It is an inner product on the real vector space of functions $\{-1,1\}^n \to \mathbb{R}$, which has dimension 2^n: the indicator functions of the 2^n points form a basis. Note also that $\mathbb{E}_x[x_i] = 0$ for every $i \in [n]$, and that the coordinates $x_1, \ldots, x_n$ are independent.

Definition 16.9. For $S \subseteq [n]$, the *parity function*, or *monomial*, on S is

$$\chi_S(x) \coloneqq \prod_{i\in S} x_i, \qquad \text{with } \chi_\emptyset = 1.$$

Thus $\chi_S(x) = -1$ if an odd number of the coordinates x_i with $i \in S$ equal -1, and $\chi_S(x) = 1$ if an even number do.

Lemma 16.10 (Orthonormality)**.** *Let $S, T \subseteq [n]$.*

1) If $S \neq \emptyset$, then $\mathbb{E}_x[\chi_S(x)] = 0$.

2) If $S \neq T$, then $\langle \chi_S, \chi_T\rangle = 0$; and $\langle \chi_S, \chi_S\rangle = \|\chi_S\|_2^2 = 1$.

Proof. For the first part, the coordinates are independent with mean 0, so $\mathbb{E}_x[\chi_S(x)] = \prod_{i\in S} \mathbb{E}_x[x_i] = 0$ when $S \neq \emptyset$. For the second part, since $x_i^2 = 1$,

$$\chi_S(x)\chi_T(x) = \prod_{i\in S} x_i \prod_{i\in T} x_i = \prod_{i\in S\cap T} x_i^2 \prod_{i\in S\triangle T} x_i = \prod_{i\in S\triangle T} x_i = \chi_{S\triangle T}(x),$$

where $S\triangle T$ is the symmetric difference. If $S \neq T$, then $S\triangle T \neq \emptyset$, and the first part gives $\langle \chi_S, \chi_T\rangle = \mathbb{E}_x[\chi_{S\triangle T}(x)] = 0$. If $S = T$, then $\chi_S\chi_S = \chi_\emptyset = 1$, so $\langle \chi_S, \chi_S\rangle = 1$. □

Proposition 16.11. *The 2^n functions χ_S, $S \subseteq [n]$, form an orthonormal basis of the space of functions $\{-1,1\}^n \to \mathbb{R}$, with the inner product of Definition 16.8.*

Proof. By Lemma 16.10, each χ_S has length $\|\chi_S\|_2 = 1$ and $\langle \chi_S, \chi_T \rangle = 0$ for all $S \neq T$. Orthonormal vectors are linearly independent: if $\sum_S a_S \chi_S = 0$, then taking the inner product with χ_T gives $a_T = 0$. So these 2^n functions are linearly independent in a space of dimension 2^n, and hence form a basis. □

Thus every function on $\{-1,1\}^n$ can be expanded as a linear combination of the parity functions.

Definition 16.12 (Fourier transform)**.** Let $f\colon \{-1,1\}^n \to \mathbb{R}$. By Proposition 16.11, there are unique real numbers $\hat{f}(S)$, for $S \subseteq [n]$, such that

$$f(x) = \sum_{S \subseteq [n]} \hat{f}(S)\, \chi_S(x) \qquad \text{for every } x \in \{-1,1\}^n.$$

The number $\hat{f}(S)$ is the *Fourier coefficient* of f on S, and the right-hand side is the *Fourier expansion* of f: it writes f in the basis of the parity functions.

Lemma 16.13. *For every $f\colon \{-1,1\}^n \to \mathbb{R}$ and every $T \subseteq [n]$,*

$$\hat{f}(T) = \mathbb{E}_x\big[f(x)\chi_T(x)\big] = \langle f, \chi_T \rangle.$$

So $\hat{f}(T)$ is the correlation of f with the parity function χ_T.

Proof. By the Fourier expansion of f and linearity of expectation,

$$\mathbb{E}_x\big[f(x)\chi_T(x)\big] = \mathbb{E}_x\Big[\sum_S \hat{f}(S)\chi_S(x)\chi_T(x)\Big] = \sum_S \hat{f}(S)\mathbb{E}_x\big[\chi_S(x)\chi_T(x)\big] = \hat{f}(T),$$

because, by Lemma 16.10, $\mathbb{E}_x[\chi_S(x)\chi_T(x)]$ is 0 for $S \neq T$ and 1 for $S = T$. □

Theorem 16.14 (Parseval's identity)**.** *For every $f\colon \{-1,1\}^n \to \mathbb{R}$,*

$$\mathbb{E}_x\big[f(x)^2\big] = \sum_{S \subseteq [n]} \hat{f}(S)^2.$$

Proof. Expanding both factors of $f(x)^2$,

$$\mathbb{E}_x\big[f(x)^2\big] = \frac{1}{2^n}\sum_x \sum_{S,T} \hat{f}(S)\hat{f}(T)\chi_S(x)\chi_T(x) = \sum_{S,T} \hat{f}(S)\hat{f}(T)\mathbb{E}_x\big[\chi_S(x)\chi_T(x)\big] = \sum_S \hat{f}(S)^2,$$

by Lemma 16.10. □

16.5 Influence of variables

For $x \in \{-1,1\}^n$ and $i \in [n]$, let $x^{(i)}$ be the point obtained from x by flipping its i-th coordinate. The *influence* $\mathrm{Inf}_i(f)$ of the i-th variable on f measures how much f depends on that variable.

Definition 16.15. Let $f\colon \{-1,1\}^n \to \{-1,1\}$ be Boolean-valued and $i \in [n]$. The *influence* of the i-th variable on f is

$$\mathrm{Inf}_i(f) \coloneqq \Pr_x\big[f(x) \neq f(x^{(i)})\big],$$

the probability that, at a uniformly random x, flipping the i-th bit changes the value of f.

Since f takes the values ± 1, the number $\frac{1}{4}(f(x) - f(x^{(i)}))^2$ is 1 if $f(x) \neq f(x^{(i)})$ and 0 otherwise, so

$$\mathrm{Inf}_i(f) = \mathbb{E}_x\Big[\frac{1}{4}\big(f(x) - f(x^{(i)})\big)^2\Big]. \tag{16.1}$$

For an arbitrary $f\colon \{-1,1\}^n \to \mathbb{R}$ we take the right-hand side of (16.1) as the definition of $\mathrm{Inf}_i(f)$.

Lemma 16.16. *For every $f\colon \{-1,1\}^n \to \mathbb{R}$ and every $i \in [n]$,*

$$\mathrm{Inf}_i(f) = \frac{1}{4}\mathbb{E}_x\big[\big(f(x) - f(x^{(i)})\big)^2\big] = \sum_{S \ni i} \hat{f}(S)^2.$$

Proof. Flipping x_i negates $\chi_S(x)$ if $i \in S$ and leaves it unchanged otherwise. Hence, by the Fourier expansion of f,

$$f(x^{(i)}) = \sum_S \hat{f}(S)\chi_S(x^{(i)}) = \sum_{S \not\ni i} \hat{f}(S)\chi_S(x) - \sum_{S \ni i} \hat{f}(S)\chi_S(x),$$

and therefore $f(x) - f(x^{(i)}) = 2\sum_{S \ni i} \hat{f}(S)\chi_S(x)$. Let

$$g(x) = \frac{1}{2}\big(f(x) - f(x^{(i)})\big) = \sum_{S \ni i} \hat{f}(S)\chi_S(x).$$

By the uniqueness of the Fourier expansion, $\hat{g}(S) = \hat{f}(S)$ for $S \ni i$ and $\hat{g}(S) = 0$ otherwise. By Parseval's identity (Theorem 16.14),

$$\frac{1}{4}\mathbb{E}_x\big[\big(f(x) - f(x^{(i)})\big)^2\big] = \mathbb{E}_x\big[g(x)^2\big] = \sum_{S \ni i} \hat{f}(S)^2. \qquad \square$$

Example 16.17 (Dictator function)**.** For $j \in [n]$, the *dictator function* $f(x) = x_j$ depends only on the j-th bit. Flipping x_j always changes its value and flipping any other bit never does, so $\mathrm{Inf}_j(f) = 1$ and $\mathrm{Inf}_i(f) = 0$ for $i \neq j$. The j-th variable is as influential as possible.

Example 16.18 (Mean function)**.** The *mean function* $f(x) = \frac{1}{n}\sum_{i=1}^n x_i$ is real-valued. For every i and x we have $f(x) - f(x^{(i)}) = 2x_i/n$, so

$$\mathrm{Inf}_i(f) = \frac{1}{4}\mathbb{E}_x\Big[\Big(\frac{2x_i}{n}\Big)^2\Big] = \frac{1}{n^2}.$$

Every variable has low influence: the mean function is a *low-influence function*.

Example 16.19 (Majority)**.** Let n be odd, and let $\mathrm{Maj}_n(x) = \mathrm{sign}(\sum_{i=1}^n x_i)$; the sum is odd, and hence nonzero. Flipping x_i changes the sum by ± 2. Flipping a bit can therefore change the value only when $\sum_j x_j = \pm 1$, and at such an x it changes the value exactly for the coordinates

x_i with the same sign as the sum. Equivalently, flipping x_i changes $\mathrm{Maj}_n(x)$ if and only if $\sum_{j\neq i} x_j = 0$. Since the $n-1$ coordinates other than x_i are uniform and independent,

$$\mathrm{Inf}_i(\mathrm{Maj}_n) = \Pr_x\Big[\sum_{j\neq i} x_j = 0\Big] = 2^{-(n-1)}\binom{n-1}{(n-1)/2} \qquad \text{for every } i \in [n].$$

For $n \geq 3$ this value lies between $1/\sqrt{2(n-1)}$ and $1/\sqrt{n}$. To see this, write $n-1 = 2k$ with $k \geq 1$. Since $(2k)! = \prod_{j=1}^{k}(2j-1)(2j)$ and $2^k k! = \prod_{j=1}^{k} 2j$, the value is

$$a_k = \frac{1}{4^k}\binom{2k}{k} = \frac{(2k)!}{(2^k k!)^2} = \prod_{j=1}^{k}\frac{2j-1}{2j}.$$

Since $(2j-1)(2j+1) \leq (2j)^2$ and $(2j-1)^2 \geq 2j(2j-2)$,

$$\Big(\frac{2j-1}{2j}\Big)^2 \leq \frac{2j-1}{2j}\cdot\frac{2j}{2j+1} = \frac{2j-1}{2j+1} \qquad \text{for } j \geq 1,$$
$$\Big(\frac{2j-1}{2j}\Big)^2 \geq \frac{2j-2}{2j-1}\cdot\frac{2j-1}{2j} = \frac{j-1}{j} \qquad \text{for } j \geq 2.$$

Multiplying the first bounds over $j = 1, \ldots, k$, the product telescopes to $a_k^2 \leq 1/(2k+1) = 1/n$. Multiplying the second bounds over $j = 2, \ldots, k$ and the factor $1/4$ for $j = 1$ gives $a_k^2 \geq 1/(4k) = 1/(2(n-1))$. So all n variables have influence $\Theta(1/\sqrt{n})$, and majority is also a low-influence function.

Example 16.20 (Parity function)**.** For the parity function $f(x) = \prod_{i=1}^{n} x_i = \chi_{[n]}(x)$, flipping any bit at any x changes the value. So every variable has influence 1: $\mathrm{Inf}_i(f) = 1$ for all $i \in [n]$.

16.6 Low-degree influence

By Lemma 16.16, $\mathrm{Inf}_i(f) = \sum_{S\ni i}\hat{f}(S)^2$. The low-degree influence keeps only the terms with $|S|$ small, discounting the effect of the parity functions of high degree.

Definition 16.21. Let $f\colon \{-1,1\}^n \to \mathbb{R}$, $i \in [n]$ and $C > 0$. The *low-degree influence* of the i-th variable on f is

$$\mathrm{Inf}_i^{C}(f) \coloneqq \sum_{S\ni i,\ |S|\leq C}\hat{f}(S)^2.$$

Example 16.22. Let $S \subseteq [n]$. By the uniqueness of the Fourier expansion, the only nonzero Fourier coefficient of χ_S is $\hat{\chi}_S(S) = 1$. Hence

$$\mathrm{Inf}_i^{C}(\chi_S) = \begin{cases} 1 & \text{if } i \in S \text{ and } |S| \leq C, \\ 0 & \text{otherwise.} \end{cases}$$

So at most C variables of a parity function have positive low-degree influence, and if $|S| > C$, all its low-degree influences vanish, although each of its $|S|$ variables has influence 1.

Lemma 16.23 (Only a few variables can be influential)**.** *Let $f\colon \{-1,1\}^n \to [-1,1]$, $C > 0$ and $\varepsilon > 0$. Then*

$$\big|\{i \in [n] : \mathrm{Inf}_i^{C}(f) \geq \varepsilon\}\big| \leq \frac{C}{\varepsilon}.$$

Proof. Each set S with $|S| \le C$ appears in the definitions of exactly $|S|$ of the low-degree influences, so

$$\sum_{i=1}^{n} \mathrm{Inf}_i^{C}(f) = \sum_{i=1}^{n} \sum_{S \ni i,\ |S| \le C} \hat{f}(S)^2 = \sum_{|S| \le C} |S|\, \hat{f}(S)^2 \le C \sum_{S} \hat{f}(S)^2 \le C,$$

since $\sum_S \hat{f}(S)^2 = \mathbb{E}_x[f(x)^2] \le 1$ by Parseval's identity (Theorem 16.14) and $|f(x)| \le 1$. Let $I = \{i : \mathrm{Inf}_i^{C}(f) \ge \varepsilon\}$. All low-degree influences are nonnegative, so by Markov's inequality, in its counting form,

$$\varepsilon |I| \le \sum_{i \in I} \mathrm{Inf}_i^{C}(f) \le \sum_{i=1}^{n} \mathrm{Inf}_i^{C}(f) \le C,$$

that is, $|I| \le C/\varepsilon$. □

Remark 16.24. The restriction to low degrees is needed: for $0 < \varepsilon \le 1$, the number of variables with $\mathrm{Inf}_i(f) \ge \varepsilon$ has no bound independent of n. Indeed, the parity function $\chi_{[n]}$ takes values in $\{-1, 1\}$, and all n of its variables have influence 1 (Example 16.20), while its low-degree influences are all 0 once $n > C$ (Example 16.22).

17 UGC-Based Hardness from Dictator vs. Spread-Out Function Tests I

Lecture of October 8, 2021, by Pravesh K. Kothari. Based on the lecturer's handwritten notes.

In Lecture 16 the hypercube graph appeared as the gadget for Max-Cut: its dictator cuts are its best cuts. This lecture turns that view around. A dictator test reads a few values of a Boolean function and should accept every dictator with high probability, and every function that is far from a dictator, in the sense that all its coordinates have small low-degree influence, with smaller probability. We study the two-query test behind Max-Cut, which accepts dictators with probability $\frac{1}{2} - \frac{1}{2}\rho$. By the central limit theorem the majority function passes it with probability tending to $\arccos(\rho)/\pi$, and the invariance principle extends this bound to every function with small low-degree influences when $\rho \leq 0$. We then state the theorem that turns a dictator test into hardness of approximation, assuming the Unique Games Conjecture, and build the Max-Cut instance used in its proof by composing the test with a unique game. The analysis of this reduction is the subject of Lecture 18.

17.1 Dictator tests

Throughout this lecture, $n \geq 1$, the point x is uniformly distributed on $\{-1,1\}^n$ unless stated otherwise, and we use the Fourier expansion $f = \sum_{S \subseteq [n]} \hat{f}(S)\chi_S$ of a function $f\colon \{-1,1\}^n \to \mathbb{R}$ from Lecture 16. For $i \in [n]$, the *dictator* $D_i\colon \{-1,1\}^n \to \{-1,1\}$ is the function $D_i(x) = x_i$. In Lecture 16 we asked that a large cut of the hypercube graph be close to a dictator cut. A dictator test is the same idea seen from the other side: a procedure that inspects a function at a few points and accepts dictators more often than functions that are far from dictators.

Definition 17.1. Let $q \geq 1$. A *q-query dictator test* on $\{-1,1\}^n$ is a finite probability distribution T, with rational probabilities, on pairs $(P, (x^1, \ldots, x^q))$, where $P\colon \{-1,1\}^q \to \{0,1\}$ is a *predicate* and $x^1, \ldots, x^q \in \{-1,1\}^n$. The test is given access to the truth table of a function $f\colon \{-1,1\}^n \to \{-1,1\}$. It draws $(P, (x^1, \ldots, x^q))$ from T, queries f at the q points, and *accepts*, or f *passes*, if $P(f(x^1), \ldots, f(x^q)) = 1$. We write

$$\mathrm{acc}_T(f) \coloneqq \Pr_{(P,(x^1,\ldots,x^q))\sim T}\left[P(f(x^1), \ldots, f(x^q)) = 1\right]$$

for the probability that f passes.

We hope for a test with two properties, for some numbers $s < c$.

1) *Completeness*: every dictator passes with probability at least c.

2) *Soundness*: every function that is far from a dictator passes with probability at most about s.

We call a function *far from a dictator* if all its coordinates have small influence. It is convenient to measure influence by its low-degree part, recalled from Lecture 16.

Definition 17.2. Let $f\colon \{-1,1\}^n \to \mathbb{R}$, $i \in [n]$ and $C > 0$. The *C-degree influence* of the i-th coordinate on f is

$$\mathrm{Inf}_i^{C}(f) \coloneqq \sum_{S \ni i,\ |S| \leq C} \hat{f}(S)^2.$$

The plain influence $\mathrm{Inf}_i(f) = \sum_{S \ni i} \hat{f}(S)^2$ allows the somewhat unintuitive phenomenon that many coordinates have large influence: every coordinate of the parity function $\chi_{[n]}$ has influence 1. Its C-degree influences, by contrast, all vanish once $n > C$. Low-degree influences rule this phenomenon out.

Lemma 17.3. *Let $f\colon \{-1,1\}^n \to [-1,1]$, $C > 0$ and $\varepsilon > 0$. Then*

$$\bigl|\{i \in [n] : \mathrm{Inf}_i^{C}(f) \geq \varepsilon\}\bigr| \leq \frac{C}{\varepsilon}.$$

Proof. Each set S with $|S| \leq C$ occurs in exactly $|S|$ of the sums $\mathrm{Inf}_1^{C}(f), \dots, \mathrm{Inf}_n^{C}(f)$. Hence

$$\sum_{i=1}^{n} \mathrm{Inf}_i^{C}(f) = \sum_{|S| \leq C} |S|\, \hat{f}(S)^2 \leq C \sum_{S} \hat{f}(S)^2 = C\, \mathbb{E}_x\bigl[f(x)^2\bigr] \leq C,$$

by Parseval's identity and $|f(x)| \leq 1$. All the terms on the left are nonnegative, so by Markov's inequality (in its counting form) at most C/ε of them are at least ε. □

We allow values in $[-1,1]$ rather than $\{-1,1\}$ in Lemma 17.3 on purpose: in Lecture 18 the lemma is applied to averages of Boolean-valued functions.

17.2 A two-query test

To see how dictator tests relate to hardness reductions, consider the following two-query test. For $x, \mu \in \{-1,1\}^n$ we write $x \cdot \mu := (x_1\mu_1, \dots, x_n\mu_n)$ for the coordinatewise product.

Algorithm 17.4 (Two-query test with parameter $\rho \in [-1,1]$)**.** Given a function $f\colon \{-1,1\}^n \to \{-1,1\}$:

1) choose $x \in \{-1,1\}^n$ uniformly at random;
2) choose $\mu \in \{-1,1\}^n$ with independent coordinates, each equal to -1 with probability $\frac{1}{2} - \frac{1}{2}\rho$ and to $+1$ with probability $\frac{1}{2} + \frac{1}{2}\rho$;
3) accept if $f(x) \neq f(x \cdot \mu)$.

This is a 2-query dictator test in the sense of Definition 17.1, with the single predicate $P_{\neq}(a,b) = 1$ if $a \neq b$ and 0 otherwise (for rational ρ its probabilities are rational). Write $y = x \cdot \mu$. Then y is uniformly distributed, and the pairs (x_i, y_i) are independent with $\mathbb{E}[x_i y_i] = \mathbb{E}[\mu_i] = \rho$; we call (x,y) a *ρ-correlated pair*. This is exactly the edge distribution of the hypercube graph, the rounding gap instance for Max-Cut of Lectures 5 and 16, with ρ in place of ρ_*, and the check asks whether the edge $\{x,y\}$ is cut by the cut that f describes. So f passes with probability equal to the value of its cut in the hypercube graph.

Lemma 17.5 (Dictators on the two-query test)**.** *Let $\rho \in [-1,1]$ and $i \in [n]$. The dictator D_i passes the test of Algorithm 17.4 with probability exactly $\frac{1}{2} - \frac{1}{2}\rho$.*

Proof. With $y = x \cdot \mu$,

$$\Pr_{(x,y)}\bigl[D_i(x) \neq D_i(y)\bigr] = \Pr_{(x,y)}\bigl[x_i \neq y_i\bigr] = \Pr[\mu_i = -1] = \frac{1}{2} - \frac{1}{2}\rho.$$ □

Analyzing the test therefore amounts to proving that a function that is far from a dictator passes with a much smaller probability. We look at some examples.

Example 17.6. The majority of three bits, $\mathrm{Maj}_3(x) = \mathrm{sign}(x_1 + x_2 + x_3)$, passes the test with probability

$$\frac{1}{2} - \frac{3}{8}\rho - \frac{1}{8}\rho^3.$$

To see this, note that for $a, b \in \{-1, 1\}$ we have $\mathbf{1}[a \neq b] = \frac{1}{2} - \frac{1}{2}ab$, so every $f\colon \{-1,1\}^n \to \{-1,1\}$ passes with probability $\frac{1}{2} - \frac{1}{2}\mathbb{E}[f(x)f(y)]$. For $S, S' \subseteq [n]$,

$$\mathbb{E}\big[\chi_S(x)\chi_{S'}(y)\big] = \mathbb{E}\Big[\prod_{i \in S \triangle S'} x_i\Big]\mathbb{E}\Big[\prod_{i \in S'} \mu_i\Big] = \begin{cases} \rho^{|S|} & \text{if } S = S', \\ 0 & \text{otherwise,} \end{cases}$$

since $y_i = x_i\mu_i$ and $x_i^2 = 1$. Expanding f in the Fourier basis therefore gives

$$\Pr[f(x) \neq f(y)] = \frac{1}{2} - \frac{1}{2}\sum_{S \subseteq [n]} \rho^{|S|}\hat{f}(S)^2. \tag{17.1}$$

The Fourier expansion of the majority of three bits is $\mathrm{Maj}_3(x) = \frac{1}{2}(x_1 + x_2 + x_3) - \frac{1}{2}x_1x_2x_3$, as one verifies on the four values of $x_1 + x_2 + x_3$. By (17.1) it passes with probability $\frac{1}{2} - \frac{1}{2}\big(3 \cdot \frac{1}{4}\rho + \frac{1}{4}\rho^3\big)$, the value stated.

For $\rho < 0$ this is smaller than the probability $\frac{1}{2} - \frac{1}{2}\rho$ with which dictators pass, but only by $\frac{1}{8}\rho(\rho^2 - 1)$, a margin that vanishes as $\rho \to -1$ or $\rho \to 0$. The variables of Maj_3 still have large influence. The basic example of a function in which every variable has low influence is the majority function on many bits.

Lemma 17.7. *Let $\rho \in [-1, 1]$ and $\varepsilon > 0$. There is n_0 such that for every odd $n \geq n_0$, the majority function $\mathrm{Maj}_n(x) = \mathrm{sign}(\sum_{i=1}^n x_i)$ passes the test of Algorithm 17.4 with probability at most $\arccos(\rho)/\pi + \varepsilon$.*

For odd n the sum $\sum_i x_i$ is odd, hence nonzero, so Maj_n takes values in $\{-1, 1\}$. The idea of the proof is the following. The test picks x uniformly and μ with the biased distribution above, and checks whether $\langle x, \mathbf{1}\rangle = \sum_i x_i$ and $\langle x \cdot \mu, \mathbf{1}\rangle = \sum_i x_i\mu_i$ have different signs. Let

$$a = \frac{1}{\sqrt{n}}\mathbf{1}, \qquad b = \frac{1}{\sqrt{n}}\mu,$$

which are unit vectors, so that $\sum_i x_i = \sqrt{n}\,\langle a, x\rangle$ and $\sum_i x_i\mu_i = \sqrt{n}\,\langle b, x\rangle$. Suppose that we pick μ first and fix it at a typical value. In particular this fixes

$$\langle a, b\rangle = \frac{1}{n}\sum_{i=1}^n \mu_i,$$

which is close to ρ by the law of large numbers. Conditioned on μ, the test chooses x uniformly and checks whether $\langle a, x\rangle$ and $\langle b, x\rangle$ have different signs; now a and b are fixed and only x is random. The key idea is the central limit theorem: each of $\langle a, x\rangle = \frac{1}{\sqrt{n}}\sum_i x_i$ and $\langle b, x\rangle = \frac{1}{\sqrt{n}}\sum_i \mu_i x_i$ is approximately a Gaussian with the same mean 0 and variance 1. A better way to state this is that $\langle a, x\rangle$ is approximately distributed as $\langle a, g\rangle$, where g is a standard Gaussian vector, and

likewise for b. The two-dimensional central limit theorem, an instance of an *invariance principle*, says more:

$$(\langle a, x\rangle, \langle b, x\rangle) \text{ is approximately distributed as } (\langle a, g\rangle, \langle b, g\rangle) \tag{17.2}$$

as a pair. For the Gaussian pair the probability of different signs is exactly $\arccos(\langle a, b\rangle)/\pi$, by the following two facts.

Fact 17.8. *If g is a standard Gaussian vector in $\mathbb{R}^d$, then $g/\|g\|_2$ is uniformly distributed on the unit sphere of $\mathbb{R}^d$.*

Lemma 17.9. *Let $a, b \in \mathbb{R}^d$ be unit vectors and g a standard Gaussian vector in $\mathbb{R}^d$. Then*

$$\Pr[\operatorname{sign}\langle a, g\rangle \neq \operatorname{sign}\langle b, g\rangle] = \frac{\arccos\langle a, b\rangle}{\pi}.$$

Proof. The events $\langle a, g\rangle = 0$ and $\langle b, g\rangle = 0$ have probability 0. By Fact 17.8, the hyperplane orthogonal to g is the random hyperplane of the Goemans–Williamson rounding (Lecture 4), and the signs differ exactly when this hyperplane separates a from b. By the analysis in Lecture 4, this happens with probability θ/π, where $\theta = \arccos\langle a, b\rangle$ is the angle between a and b. □

Proof of Lemma 17.7. We make the argument precise without conditioning on μ. The vectors $Z_i = (x_i, x_i\mu_i) \in \mathbb{R}^2$, $i = 1, \dots, n$, are independent and identically distributed, with mean 0, $\mathbb{E}[x_i^2] = \mathbb{E}[(x_i\mu_i)^2] = 1$ and $\mathbb{E}[x_i \cdot x_i\mu_i] = \mathbb{E}[\mu_i] = \rho$. By the multidimensional central limit theorem, $n^{-1/2}\sum_{i=1}^n Z_i$ converges in distribution to a centered Gaussian vector (G_1, G_2) with $\mathbb{E}[G_1^2] = \mathbb{E}[G_2^2] = 1$ and $\mathbb{E}[G_1G_2] = \rho$. Let $A = \{(u, w) \in \mathbb{R}^2 : uw < 0\}$. For odd n, Maj_n passes exactly when $n^{-1/2}\sum_i Z_i \in A$. The boundary of A lies in the union of the two coordinate axes, which has probability 0 under the law of (G_1, G_2) because $\Pr[G_1 = 0] = \Pr[G_2 = 0] = 0$. By the portmanteau theorem,

$$\Pr[\mathrm{Maj}_n \text{ passes}] \longrightarrow \Pr[(G_1, G_2) \in A] \qquad \text{as } n \to \infty \text{ through odd values.}$$

Finally, choose unit vectors $a', b' \in \mathbb{R}^2$ with $\langle a', b'\rangle = \rho$ and let g be a standard Gaussian vector in $\mathbb{R}^2$. The pair $(\langle a', g\rangle, \langle b', g\rangle)$ is a centered Gaussian vector with the same covariances as (G_1, G_2), so it has the same law, and by Lemma 17.9 $\Pr[(G_1, G_2) \in A] = \arccos(\rho)/\pi$. The convergence gives n_0 for every $\varepsilon > 0$. □

Remark 17.10. For a fixed unit vector a, the approximation of $\langle a, x\rangle$ by $\langle a, g\rangle$ is made quantitative by the Berry–Esseen theorem. O'Donnell's notes [O'D07] prove a version of it by the Lindeberg replacement method, which replaces the bits $x_1, \dots, x_n$ by Gaussians one at a time and bounds the change caused by each replacement. The same method applies to pairs such as (17.2) and to low-degree polynomials, and this is how the invariance principle below is proved.

17.3 Low-influence functions and the invariance principle

Such invariance principles generalize greatly, from the linear functions $\langle a, x\rangle$ to low-degree polynomials with small influences. The resulting theorem says that, when $\rho \leq 0$, among functions with small low-degree influences, none passes the two-query test much more often than majority does.

Theorem 17.11 (Majority is stablest [MOO10, KKMO07])**.** *Let $\rho \in (-1,0]$ and $\varepsilon > 0$. There exist $\tau > 0$ and $\delta > 0$ such that for every $n \geq 1$ and every $f\colon \{-1,1\}^n \to \{-1,1\}$ with $\mathrm{Inf}_i^{1/\delta}(f) \leq \tau$ for all $i \in [n]$,*

$$\Pr[f \text{ passes the test of Algorithm 17.4}] \leq \frac{\arccos(\rho)}{\pi} + \varepsilon.$$

We state this theorem without proof. Its proof has two steps. An invariance principle, which extends (17.2) to low-degree polynomials whose variables all have small influence, shows that the pair $(f(x), f(y))$ behaves like $(F(g), F(g'))$, where F is the polynomial $\sum_S \hat{f}(S) \prod_{i \in S} z_i$ in real variables z and (g, g') is a ρ-correlated pair of standard Gaussian vectors; in particular, $F(g)$ lies in $[-1,1]$ up to a small error. So the passing probability $\mathbb{E}[\frac{1}{2} - \frac{1}{2}f(x)f(y)]$ is close to $\mathbb{E}[\frac{1}{2} - \frac{1}{2}F(g)F(g')]$. The largest possible value of this expectation, over functions F with values in $[-1,1]$, is the max-cut of the ρ-correlated Gaussian analogue of the sphere graph, and its analysis is the analysis of the Feige–Schechtman integrality gap instance [FS02] of Lecture 5: halfspaces through the origin are optimal [Bor85], and they cut an $\arccos(\rho)/\pi$ fraction.

Remark 17.12. The hypothesis $\rho \leq 0$ is needed. For $\rho \in (0,1)$, the parity function $\chi_{[n]}$ with $n > 1/\delta$ has $\mathrm{Inf}_i^{1/\delta}(\chi_{[n]}) = 0$ for every i, and by (17.1) it passes with probability $\frac{1}{2} - \frac{1}{2}\rho^n$, which tends to $\frac{1}{2} > \arccos(\rho)/\pi$. For $\rho \leq 0$ there is no counterexample of this kind: the parity function passes with probability at most $\frac{1}{2} + \frac{1}{2}|\rho|^n$, while $\arccos(\rho)/\pi \geq \frac{1}{2}$.

To summarize, for $\rho \in (-1,0)$ the test of Algorithm 17.4 has

1) completeness $c = \frac{1}{2} - \frac{1}{2}\rho$: every dictator passes with probability c (Lemma 17.5);
2) soundness $s = \arccos(\rho)/\pi$, the soundness of the Feige–Schechtman integrality gap: by Theorem 17.11, for every $\varepsilon > 0$ there are $\tau, \delta > 0$ such that every function whose coordinates all have $(1/\delta)$-degree influence at most τ passes with probability at most $\arccos(\rho)/\pi + \varepsilon$.

Here $s < c$, because $\rho \mapsto \arccos(\rho)/\pi$ is strictly convex on $[-1,0]$ and agrees with $\frac{1}{2} - \frac{1}{2}\rho$ at $\rho = -1$ and $\rho = 0$. At $\rho = \rho_*$ the ratio s/c is the Goemans–Williamson constant α_{GW} of Lecture 16.

17.4 From dictator tests to hardness

A dictator test becomes a hardness result once we compose it with a unique game. We first fix the kind of problem the test produces.

Definition 17.13. Let $q \geq 1$ and let $\mathcal{F}$ be a finite family of predicates $P\colon \{-1,1\}^q \to \{0,1\}$. An instance Φ of *Max-$\mathcal{F}$* consists of a finite set U of variables and a finite probability distribution, with rational probabilities, on *constraints*; a constraint is a pair $(P, (u_1, \ldots, u_q))$ with $P \in \mathcal{F}$ and $u_1, \ldots, u_q \in U$. An assignment $a\colon U \to \{-1,1\}$ *satisfies* the constraint if $P(a(u_1), \ldots, a(u_q)) = 1$. The *value* $\mathrm{val}(\Phi)$ is the largest probability, over assignments a, that a random constraint is satisfied by a. For $0 \leq s < c \leq 1$, the (c,s)-*Max-$\mathcal{F}$* problem and its NP-hardness are defined as for Max-2-Lin(p) in Lecture 16: given Φ with $\mathrm{val}(\Phi) \geq c$ or $\mathrm{val}(\Phi) \leq s$, decide which holds.

Max-Cut is Max-$\{P_{\neq}\}$: the variables are the vertices of a graph, a constraint $(P_{\neq}, (u,w))$ is an edge with weight equal to its probability, and an assignment is a cut. (A constraint $(P_{\neq}, (u,u))$, a loop, is never satisfied.) The checks of a dictator test with predicates in $\mathcal{F}$ look exactly like the constraints of Max-$\mathcal{F}$, and the reduction below replaces each query point by a variable.

The reduction needs the soundness of the test for functions with values in $[-1,1]$ as well. Such a function $h\colon \{-1,1\}^n \to [-1,1]$ is run on a test as a randomized function: each query x^j is answered by an independent random sign Z_j with $\mathbb{E}[Z_j] = h(x^j)$, and $\mathrm{acc}_T(h)$ is the probability that $P(Z_1,\dots,Z_q) = 1$. For a Boolean-valued h this is the acceptance probability of Definition 17.1. For the test of Algorithm 17.4, $\Pr[Z_1 \neq Z_2] = \frac{1}{2} - \frac{1}{2}h(x)h(y)$, so $\mathrm{acc}(h) = \mathbb{E}\big[\frac{1}{2} - \frac{1}{2}h(x)h(x\cdot\mu)\big]$.

Theorem 17.14 (Reduction)**.** *Let $q \geq 1$, let $\mathcal{F}$ be a finite family of predicates on $\{-1,1\}^q$, and let $0 \leq s < c \leq 1$. Suppose that for every $\eta > 0$ there exist $\tau > 0$ and $\delta > 0$ such that for every $n \geq 1$ there is a q-query dictator test T on $\{-1,1\}^n$, all of whose predicates lie in $\mathcal{F}$, with the following properties.*

1) All n dictators pass with probability at least c.

2) Every $h\colon \{-1,1\}^n \to [-1,1]$ with $\mathrm{Inf}_i^{1/\delta}(h) \leq \tau$ for all $i \in [n]$ satisfies $\mathrm{acc}_T(h) \leq s + \eta$.

Assume the Unique Games Conjecture. Then for every η with $0 < \eta < (c-s)/2$, the $(c-\eta, s+\eta)$-Max-$\mathcal{F}$ problem is NP-hard.

We have discussed such a test for the predicate $P_{\neq}$, the Max-Cut type predicate. In Lecture 18 we prove the theorem for Max-Cut, with the test of Algorithm 17.4; the construction and the analysis are general and apply to every family $\mathcal{F}$. Lecture 18 also explains why property 2 is needed for $[-1,1]$-valued functions and not only for Boolean-valued ones: the proof applies it to averages of Boolean-valued functions.

17.5 The reduction for Max-Cut

We now construct the Max-Cut instance. The input is an instance I of Max-2-Lin(p), as in Lecture 16: variables x_v indexed by a finite set V, taking values in $\mathbb{F}_p = \{0,1,\dots,p-1\}$, and a list of equations $x_v + x_w = b_{vw} \pmod p$ with $v \neq w$. We view I as a graph $G = (V,E)$ with an edge $\{v,w\}$ for each equation, labeled by $b_{vw} = b_{wv} \in \mathbb{F}_p$. To keep the notation light we assume that no two equations involve the same pair of variables; the general case is handled by treating such equations as parallel edges. Let $\deg(v)$ be the number of edges at v. Regularity and bipartiteness of G are not needed below.

We call the values of the variables *labels*. Labels i for v and j for w *satisfy* the edge $\{v,w\}$ if $i + j \equiv b_{vw} \pmod p$.

The vertices of H. The graph H has $|V| \cdot 2^p$ vertices: for each $v \in V$, a copy $\{v\} \times \{-1,1\}^p$ of the p-dimensional hypercube, whose coordinates we index by the labels $0,1,\dots,p-1$. Before choosing the edges, note that a cut of H is described by a family of functions

$$\{f_v\colon \{-1,1\}^p \to \{-1,1\}\}_{v\in V},$$

where $f_v(x)$ records the side of the vertex (v,x) of the v-th hypercube; every such family describes exactly one cut.

Encoding labels by dictators. We use the tester to check that the equations of I are satisfied, so that the probability of passing is related to the value of I. In fact, we would like to be able to learn a good assignment for I by looking at a large cut $\{f_v\}_{v\in V}$. The *intended* cuts are those in which every f_v is a dictator, $f_v = D_i$ for some $i \in \mathbb{F}_p$; we read this as assigning the label i to v.

Suppose that all the f_v are dictators. We still have to check that they are the correct dictators, that is, that the labels they encode satisfy the equations. For this we compare the function at v with the functions at the neighbors of v.

Definition 17.15 (Opinion of a neighbor)**.** For an edge $\{v, w\}$ of G, let $\sigma_{v\to w}\colon \mathbb{F}_p \to \mathbb{F}_p$ be the permutation $\sigma_{v\to w}(i) = b_{vw} - i \pmod p$, and for $x \in \{-1,1\}^p$ let $\sigma_{v\to w}(x) \in \{-1,1\}^p$ be the point with coordinates $\sigma_{v\to w}(x)_i = x_{\sigma_{v\to w}(i)}$. Given a cut $\{f_u\}_{u\in V}$, the *opinion of w about v* is the function

$$g_v^w := f_w \circ \sigma_{v\to w}\colon \{-1,1\}^p \to \{-1,1\}.$$

Since $b_{vw} = b_{wv}$, we have $\sigma_{v\to w} = \sigma_{w\to v}$, and $\sigma_{v\to w}$ is an involution: $b_{vw} - (b_{vw} - i) = i$. For each label i of v, $\sigma_{v\to w}(i)$ is the unique label of w that satisfies the edge together with i.

Lemma 17.16. *Let $\{v, w\}$ be an edge of G, and suppose $f_v = D_i$ and $f_w = D_j$, where the labels i and j satisfy $\{v, w\}$. Then $g_v^w = f_v$.*

Proof. For every $x \in \{-1,1\}^p$, $g_v^w(x) = D_j(\sigma_{v\to w}(x)) = x_{\sigma_{v\to w}(j)} = x_{b_{vw}-j} = x_i = f_v(x)$, because $i + j \equiv b_{vw}$. □

So if the f_u are dictators that encode labels satisfying the equations on the edges at v, then every neighbor's opinion about v agrees with f_v. Otherwise the opinions g_v^w need not be dictators, nor agree with each other. The test of Algorithm 17.4, applied to two opinions about the same vertex, checks for such agreement.

Algorithm 17.17 (The test, that is, the edge distribution of H)**.** Let $\rho \in (-1, 0)$ be rational.

1) Pick $v \in V$ with probability $\deg(v)/(2|E|)$.

2) Pick two random edges $\{v, w_1\}$ and $\{v, w_2\}$ at v, independently and uniformly among the $\deg(v)$ edges at v.

3) Apply the tester of Algorithm 17.4 to $g_v^{w_1}$ and $g_v^{w_2}$: choose $x \in \{-1,1\}^p$ uniformly and μ as there, and check whether $g_v^{w_1}(x) \neq g_v^{w_2}(x \cdot \mu)$.

When G is d-regular, the vertex v in step 1 is uniformly distributed. With the degree-proportional choice, each of $\{v, w_1\}$ and $\{v, w_2\}$ is a uniformly random edge of G for every graph G: the ordered pair (v, w_1) equals a given orientation (v, w) of a given edge with probability $\deg(v)/(2|E|) \cdot 1/\deg(v) = 1/(2|E|)$.

By Definition 17.15, the check in step 3 compares the side of the vertex $(w_1, \sigma_{v\to w_1}(x))$ with the side of the vertex $(w_2, \sigma_{v\to w_2}(x \cdot \mu))$. We let H be the weighted graph on the vertex set above in which each such pair is an edge whose weight is the probability that the test produces it (Fig. 27). Then, for every cut $\{f_u\}_{u\in V}$ of H, the probability that the test passes equals the weight of the edges that the cut cuts, the value of the cut; the max-cut of H is the largest passing probability over all cuts. For fixed p, the graph H has $|V| \cdot 2^p$ vertices and at most $2|E| \cdot (\max_v \deg(v)) \cdot 4^p$ edges with rational weights, so it is computed from I in polynomial time.

We stress that the edge distribution typically produces edges between vertices of two *different* hypercubes, those of two different neighbors w_1 and w_2 of v, and not edges inside one hypercube. The hypercube graph of Lecture 16 enters only through the test, which is applied to the opinions $g_v^{w_1}$ and $g_v^{w_2}$ of two neighbors about the same vertex. Lecture 18 proves the completeness and the soundness of this reduction and discusses the generalization to other predicates.

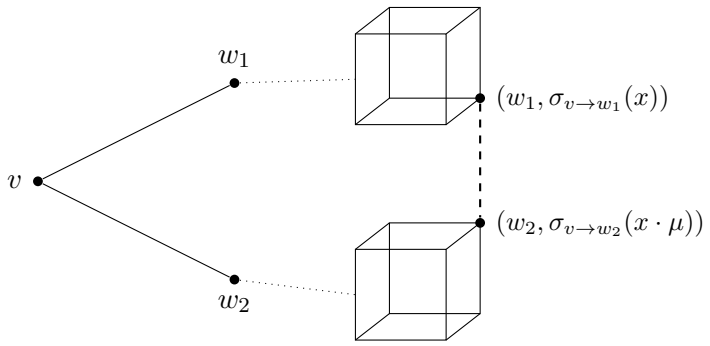


Figure 27: An edge of H (dashed). The test picks a vertex v of G and two neighbors w_1, w_2, and joins a vertex of the hypercube of w_1 to a vertex of the hypercube of w_2.

18 UGC-Based Hardness from Dictator vs. Spread-Out Function Tests II

Lecture of October 11, 2021, by Pravesh K. Kothari. Based on the lecturer's handwritten notes.

This lecture completes the proof, begun in Lecture 17, that the Unique Games Conjecture implies optimal hardness of approximation for Max-Cut. We analyze the reduction of Lecture 17, which composes the two-query dictator test with a unique game. A nearly satisfiable unique game yields a graph with a large cut (completeness). Conversely, from any cut that passes the test noticeably more often than low-influence functions can, we decode an assignment that satisfies a fraction of the equations bounded below independently of the number of labels (soundness). The decoding averages the opinions of the neighbors of a vertex and uses low-degree influences. Together with the Majority Is Stablest theorem this gives the theorem of Khot, Kindler, Mossel and O'Donnell stated in Lecture 16. We end with Raghavendra's theorem, which builds such a dictator test from any integrality gap of a natural semidefinite program.

18.1 The reduction

We recall the setting of Lecture 17. For $f\colon \{-1,1\}^n \to \mathbb{R}$ with Fourier expansion $f = \sum_S \hat{f}(S)\chi_S$, $i \in [n]$ and $C > 0$, the C-degree influence of the i-th coordinate is $\mathrm{Inf}_i^{C}(f) = \sum_{S \ni i,\, |S| \le C} \hat{f}(S)^2$. We use the following lemma, proved in Lectures 16 and 17.

Lemma 18.1. *Let $f\colon \{-1,1\}^n \to [-1,1]$, $C > 0$ and $\varepsilon > 0$. Then $|\{i \in [n] : \mathrm{Inf}_i^{C}(f) \ge \varepsilon\}| \le C/\varepsilon$.*

Throughout, $\rho \in (-1,0)$ is rational and $c = \frac{1}{2} - \frac{1}{2}\rho$. For $x, \mu \in \{-1,1\}^p$, $x \cdot \mu$ is the coordinatewise product; in expectations, x is uniform on $\{-1,1\}^p$ and μ has independent coordinates, each equal to -1 with probability $\frac{1}{2} - \frac{1}{2}\rho$. For $h\colon \{-1,1\}^p \to [-1,1]$ let

$$\mathrm{acc}(h) \coloneqq \mathbb{E}_{x,\mu}\Big[\frac{1}{2} - \frac{1}{2}h(x)h(x \cdot \mu)\Big].$$

For Boolean-valued h this is the probability that h passes the two-query test, $\Pr[h(x) \neq h(x \cdot \mu)]$; for general h it is the passing probability when h is run as a randomized function (Lecture 17). Dictators pass with probability exactly c.

The input is an instance I of Max-2-Lin(p) with variables indexed by V, viewed as a graph $G = (V, E)$ whose edge $\{v, w\}$ carries the equation $x_v + x_w = b_{vw} \pmod p$, with $b_{vw} = b_{wv}$. Labels i for v and j for w satisfy the edge if $i + j \equiv b_{vw}$. For an edge $\{v, w\}$, $\sigma_{v\to w}(i) = b_{vw} - i \pmod p$, and $\sigma_{v\to w}(x)_i = x_{\sigma_{v\to w}(i)}$ for $x \in \{-1,1\}^p$; the label $\sigma_{v\to w}(i)$ of w is the one that satisfies the edge together with the label i of v, and $\sigma_{v\to w}$ is an involution. The graph H has a copy $\{u\} \times \{-1,1\}^p$ of the hypercube for each $u \in V$, and a cut of H is a family $\{f_u\colon \{-1,1\}^p \to \{-1,1\}\}_{u \in V}$. Given a cut, the opinion of w about v is $g_v^w = f_w \circ \sigma_{v\to w}$. A *random neighbor* w of v is the other endpoint of an edge chosen uniformly among the $\deg(v)$ edges at v. The edges of H and their weights are given by the test:

1) pick $v \in V$ with probability $\deg(v)/(2|E|)$;

2) pick independent random neighbors w_1 and w_2 of v;

3) pick x and μ, and check whether $g_v^{w_1}(x) \neq g_v^{w_2}(x \cdot \mu)$.

Each of $\{v, w_1\}$ and $\{v, w_2\}$ is then a uniformly random edge of G. The *value* of a cut is the probability that the test passes, and $\mathrm{val}(H)$, the max-cut of H, is the largest value of a cut.

18.2 Completeness

Lemma 18.2 (Completeness)**.** *Let $0 \le \lambda \le 1$, and suppose that $\mathrm{val}(I) \ge 1 - \lambda$. Then H has a cut of value at least $c(1 - 2\lambda) \ge c - 2\lambda$.*

Here c is the completeness of the dictator test.

Proof. Let $A\colon V \to \mathbb{F}_p$ be an assignment that satisfies at least a $(1-\lambda)$ fraction of the equations. We give a cut $\{f_u\}$ of H and analyze its performance on the edge distribution of H: let $f_u = D_{A(u)}$, the dictator on the $A(u)$-th coordinate, for every $u \in V$.

Each of $\{v, w_1\}$ and $\{v, w_2\}$ is a uniformly random edge of G, so each is violated by A with probability at most λ. By the union bound, with probability at least $1 - 2\lambda$ over the choice of v and of two random neighbors w_1, w_2, the assignment A satisfies both $\{v, w_1\}$ and $\{v, w_2\}$. Condition on this event. If A satisfies $\{v, w\}$, then $\sigma_{v\to w}(A(v)) = A(w)$, so for every x,

$$g_v^w(x) = D_{A(w)}(\sigma_{v\to w}(x)) = x_{\sigma_{v\to w}(A(w))} = x_{A(v)} = f_v(x).$$

Hence $g_v^{w_1} = g_v^{w_2} = f_v$, and f_v is a dictator. The pair (x, μ) is independent of (v, w_1, w_2), so the dictator test on v, w_1, w_2 passes with probability c. In total the test passes with probability at least $c(1 - 2\lambda)$, which is the fraction of the edge weight of H that the cut cuts. Finally $c(1 - 2\lambda) \ge c - 2\lambda$ because $c \le 1$. □

18.3 Soundness

We now show that a cut of H of large value yields a good assignment for I. The key object is the average of the opinions about a vertex.

Definition 18.3 (Average opinion)**.** Given a cut $\{f_u\}_{u\in V}$ of H and $v \in V$ with $\deg(v) \ge 1$, the *average opinion about v* is

$$h_v \coloneqq \mathbb{E}_w\big[g_v^w\big]\colon \{-1,1\}^p \to [-1,1],$$

the expectation over a random neighbor w of v.

Note that h_v takes values in the interval $[-1, 1]$ instead of $\{-1, 1\}$; this is why the soundness of the test is needed for $[-1, 1]$-valued functions.

Fact 18.4 (Averaging)**.** *Let $s \ge 0$ and $\eta > 0$, and let Z be a random variable with $|Z| \le 1$ and $\mathbb{E}Z \ge s + \eta$. Then $\Pr[Z \ge s + \eta/2] \ge \eta/2$.*

Proof. Let $t = s + \eta/2$. On the event $Z < t$ we have $Z \le t$, and $Z \le 1$ always, so

$$s + \eta \le \mathbb{E}Z \le \Pr[Z < t] \cdot t + \Pr[Z \ge t] \cdot 1 \le t + \Pr[Z \ge t],$$

where the last step uses $t \ge 0$. Hence $\Pr[Z \ge t] \ge s + \eta - t = \eta/2$. □

Lemma 18.5. *Let $\{v, w\}$ be an edge of G, $f\colon \{-1,1\}^p \to \mathbb{R}$, and $g = f \circ \sigma_{v\to w}$. Then $\hat{g}(S) = \hat{f}(\sigma_{v\to w}^{-1}(S))$ for every $S \subseteq \mathbb{F}_p$, where $\sigma_{v\to w}^{-1}(S) = \{\sigma_{v\to w}^{-1}(i) : i \in S\}$. Consequently, for every $i \in \mathbb{F}_p$ and $C > 0$, $\mathrm{Inf}_i^C(g) = \mathrm{Inf}_{\sigma_{v\to w}(i)}^C(f)$.*

Proof. Write $\sigma = \sigma_{v\to w}$. For every T and x, $\chi_T(\sigma(x)) = \prod_{k\in T} x_{\sigma(k)} = \chi_{\sigma(T)}(x)$. Hence $g = \sum_T \hat{f}(T)\chi_{\sigma(T)}$, and by the uniqueness of the Fourier expansion, $\hat{g}(\sigma(T)) = \hat{f}(T)$ for every T, that is, $\hat{g}(S) = \hat{f}(\sigma^{-1}(S))$. Since σ is an involution, $\sigma^{-1} = \sigma$, and $S \mapsto \sigma(S)$ is a bijection that preserves sizes and maps the sets containing i onto the sets containing $\sigma(i)$. Therefore

$$\mathrm{Inf}_i^C(g) = \sum_{S\ni i,\,|S|\le C} \hat{f}(\sigma(S))^2 = \sum_{T\ni\sigma(i),\,|T|\le C} \hat{f}(T)^2 = \mathrm{Inf}_{\sigma(i)}^C(f). \qquad \square$$

Lemma 18.6 (Soundness)**.** *Let $s \ge 0$, $\eta > 0$, $\tau > 0$ and $\delta > 0$, and suppose that every $h\colon \{-1,1\}^p \to [-1,1]$ with $\mathrm{Inf}_i^{1/\delta}(h) \le \tau$ for all $i \in \mathbb{F}_p$ satisfies $\mathrm{acc}(h) \le s + \eta/4$. If some cut $\{f_u\}_{u\in V}$ of H passes the test with probability at least $s+\eta$, then there is an assignment that satisfies at least a*

$$\frac{\eta\tau^3\delta^2}{36}$$

fraction of the equations of I. This bound depends on η, τ and δ, but not on p.

Proof. Write $C = 1/\delta$. We first express the passing probability through the average opinions, then find many vertices whose average opinion has an influential coordinate, and finally decode an assignment from the influential coordinates.

Step 1: average opinions. For fixed v, w_1, w_2, x, μ, the check passes exactly when $g_v^{w_1}(x) \ne g_v^{w_2}(x\cdot\mu)$, and the indicator of this event is $\frac{1}{2} - \frac{1}{2}g_v^{w_1}(x)g_v^{w_2}(x\cdot\mu)$. Since the cut passes with probability at least $s+\eta$,

$$s + \eta \le \mathbb{E}_{v,w_1,w_2}\mathbb{E}_{x,\mu}\Big[\frac{1}{2} - \frac{1}{2}g_v^{w_1}(x)\,g_v^{w_2}(x\cdot\mu)\Big] = \mathbb{E}_{v,x,\mu}\Big[\frac{1}{2} - \frac{1}{2}\mathbb{E}_{w_1,w_2}\big[g_v^{w_1}(x)\,g_v^{w_2}(x\cdot\mu)\big]\Big]. \quad (18.1)$$

Given v, the neighbors w_1 and w_2 are independent and each is a random neighbor of v, so $\mathbb{E}_{w_1,w_2}[g_v^{w_1}(x)g_v^{w_2}(y)] = h_v(x)h_v(y)$ for all x, y. Therefore

$$s + \eta \le \mathbb{E}_v[Z_v], \qquad Z_v \coloneqq \mathbb{E}_{x,\mu}\Big[\frac{1}{2} - \frac{1}{2}h_v(x)h_v(x\cdot\mu)\Big] = \mathrm{acc}(h_v), \quad (18.2)$$

where v is distributed as in step 1 of the test. The test thus only sees the average opinions.

Step 2: good vertices. Since $|h_v| \le 1$, we have $0 \le Z_v \le 1$. Apply Fact 18.4 to the random variable Z_v, where the randomness is over v: by (18.2), a random v satisfies

$$s + \frac{\eta}{2} \le \mathrm{acc}(h_v)$$

with probability at least $\eta/2$. Call such vertices v *good*.

Step 3: an influential coordinate. Let v be good. Then $\mathrm{acc}(h_v) \ge s + \eta/2 > s + \eta/4$, so by the hypothesis, the soundness of the test for $[-1,1]$-valued functions, h_v has an influential coordinate: there is $i = i_v \in \mathbb{F}_p$ with $\mathrm{Inf}_i^C(h_v) > \tau$. The Fourier coefficients are linear in the function, so $\hat{h}_v(S) = \mathbb{E}_w[\hat{g}_v^w(S)]$, and

$$\begin{aligned}
\tau < \mathrm{Inf}_i^C(h_v) &= \sum_{S\ni i,\,|S|\le C} \big(\mathbb{E}_w[\hat{g}_v^w(S)]\big)^2 \\
&\le \sum_{S\ni i,\,|S|\le C} \mathbb{E}_w[\hat{g}_v^w(S)^2] && \text{by the Cauchy–Schwarz inequality} \\
&= \mathbb{E}_w\big[\mathrm{Inf}_i^C(g_v^w)\big] = \mathbb{E}_w\big[\mathrm{Inf}_{\sigma_{v\to w}(i)}^C(f_w)\big] && \text{by Lemma 18.5.}
\end{aligned}$$

Step 4: good neighbors. The random variable $Y_w = \mathrm{Inf}^C_{\sigma_{v\to w}(i)}(f_w)$, over a random neighbor w of v, satisfies $0 \le Y_w \le \sum_T \hat{f}_w(T)^2 = 1$ and $\mathbb{E}_w Y_w > \tau$. By Fact 18.4 (with $s = 0$ and $\eta = \tau$), for at least a $\tau/2$ fraction of the edges $\{v, w\}$ at v,

$$\mathrm{Inf}^C_{\sigma_{v\to w}(i)}(f_w) \ge \frac{\tau}{2}.$$

Call such w *good neighbors* of v. So a vertex v drawn as in step 1 of the test is good with probability at least $\eta/2$, and then h_v has an influential coordinate i_v; and for an influential coordinate i_v of h_v, at least a $\tau/2$ fraction of the neighbors w of v have the influential coordinate $\sigma_{v\to w}(i_v)$ in f_w. The pair of labels $(i_v, \sigma_{v\to w}(i_v))$ satisfies the edge $\{v, w\}$ of G: it is a good assignment for this edge.

Step 5: decoding. For every $u \in V$ let

$$\mathrm{Cand}(u) = \{i : \mathrm{Inf}^C_i(f_u) \ge \tau/2\} \cup \{i : \mathrm{Inf}^C_i(h_u) > \tau\},$$

where the second set is empty if $\deg(u) = 0$, since then h_u is not defined. By Lemma 18.1, applied to f_u with $\varepsilon = \tau/2$ and to h_u with $\varepsilon = \tau$, both of which take values in $[-1, 1]$,

$$|\mathrm{Cand}(u)| \le \frac{C}{\tau/2} + \frac{C}{\tau} = \frac{3}{\tau\delta}.$$

Assign to every vertex u an element $A(u)$ of $\mathrm{Cand}(u)$ chosen uniformly at random, independently for different vertices (and the label 0 if $\mathrm{Cand}(u)$ is empty). Let v be drawn as in step 1 of the test and let w be a random neighbor of v; then $\{v, w\}$ is a uniformly random edge of G. With probability at least $\frac{\eta}{2} \cdot \frac{\tau}{2}$, the vertex v is good and w is a good neighbor of v. In that case $i_v \in \mathrm{Cand}(v)$ by Step 3, and $\sigma_{v\to w}(i_v) \in \mathrm{Cand}(w)$ by Step 4. Since $v \ne w$ and the choices are independent, the random choice agrees with this pair with probability

$$\Pr[A(v) = i_v \text{ and } A(w) = \sigma_{v\to w}(i_v)] \ge \left(\frac{\tau\delta}{3}\right)^2,$$

and then A satisfies $\{v, w\}$. Hence the expected fraction of the equations of I that A satisfies is at least

$$\frac{\eta}{2} \cdot \frac{\tau}{2} \cdot \frac{\tau^2\delta^2}{9} = \frac{\eta\tau^3\delta^2}{36},$$

and some assignment satisfies at least this fraction. □

Remark 18.7. The decoding returns a random influential coordinate, and this is a good assignment only if the number of influential coordinates is small. This is why the soundness of the test is stated for low-degree influences: by Lemma 18.1, a function with values in $[-1, 1]$ has at most C/ε coordinates of C-degree influence at least ε, independently of p. With the plain influences, all p coordinates of a parity function have influence 1.

18.4 Hardness of Max-Cut

The soundness lemma needs the soundness of the two-query test for functions with values in $[-1, 1]$. It holds, by the Majority Is Stablest theorem of Lecture 17 in its form for bounded functions.

Theorem 18.8 (Majority is stablest [MOO10, KKMO07])**.** *Let $\rho \in (-1,0]$ and $\varepsilon > 0$. There exist $\tau > 0$ and $\delta > 0$ such that for every $n \geq 1$ and every $h\colon \{-1,1\}^n \to [-1,1]$ with $\mathrm{Inf}_i^{1/\delta}(h) \leq \tau$ for all $i \in [n]$,*

$$\mathbb{E}_{x,\mu}\Big[\frac{1}{2} - \frac{1}{2}h(x)h(x\cdot\mu)\Big] \leq \frac{\arccos(\rho)}{\pi} + \varepsilon,$$

where x is uniform on $\{-1,1\}^n$ and μ is as in the two-query test.

We state this theorem without proof. Combining it with the completeness and soundness lemmas gives the hardness of Max-Cut. In words: choose the soundness λ of the unique game below the bound of Lemma 18.6, and then the number p of labels large enough for the Unique Games Conjecture to apply.

Theorem 18.9. *Assume the Unique Games Conjecture. Let $\rho \in (-1,0)$ be rational, and let $c = \frac{1}{2} - \frac{1}{2}\rho$ and $s = \arccos(\rho)/\pi$. Then for every η with $0 < \eta < (c-s)/2$, the $(c-\eta, s+\eta)$-Max-Cut problem is NP-hard.*

Proof. Here $s < c$ by Lecture 17. Let $\tau, \delta > 0$ be given by Theorem 18.8 with $\varepsilon = \eta/4$. Let $\lambda = \min\{\eta/2, \eta\tau^3\delta^2/72\}$; then $0 < \lambda < 1/2$, because $c - s \leq \frac{1}{2}$ (as $c \leq 1$ and $s \geq \frac{1}{2}$) gives $\eta < \frac{1}{4}$. By the Unique Games Conjecture (Lecture 16) there is p_0 such that the $(1-\lambda, \lambda)$-Max-2-Lin(p) problem is NP-hard for every prime $p \geq p_0$; fix such a prime p. Given an instance I of Max-2-Lin(p), construct the graph H of Lecture 17; since p is fixed and ρ is rational, this takes polynomial time.

If $\mathrm{val}(I) \geq 1 - \lambda$, then by Lemma 18.2, $\mathrm{val}(H) \geq c - 2\lambda \geq c - \eta$.

If $\mathrm{val}(I) \leq \lambda$, then $\mathrm{val}(I) < \eta\tau^3\delta^2/36$. By Theorem 18.8, the hypothesis of Lemma 18.6 holds with $s = \arccos(\rho)/\pi \geq 0$, so no cut of H passes the test with probability at least $s + \eta$; otherwise the lemma would give an assignment satisfying more than a λ fraction of the equations. Hence $\mathrm{val}(H) < s + \eta$.

Composing a polynomial-time reduction from 3SAT to $(1-\lambda, \lambda)$-Max-2-Lin(p) with the map $I \mapsto H$ therefore shows that $(c - \eta, s + \eta)$-Max-Cut is NP-hard. □

Corollary 18.10 (Khot, Kindler, Mossel and O'Donnell [KKMO07])**.** *Assume the Unique Games Conjecture, and let $\varepsilon > 0$. There are numbers $0 \leq s' < c' \leq 1$ with $s' < (\alpha_{\mathrm{GW}} + \varepsilon)c'$ such that (c', s')-Max-Cut is NP-hard. Consequently, unless* P = NP*, no polynomial-time algorithm finds, in every weighted graph, a cut of value at least $\alpha_{\mathrm{GW}} + \varepsilon$ times the maximum.*

Proof. The function $r(\rho) = (\arccos(\rho)/\pi)/(\frac{1}{2} - \frac{1}{2}\rho)$ is continuous on $(-1,0)$ and $r(\rho_*) = \alpha_{\mathrm{GW}}$ (Lecture 16), so there is a rational $\rho \in (-1,0)$ with $r(\rho) < \alpha_{\mathrm{GW}} + \varepsilon/2$. With c and s as in Theorem 18.9, $(s+\eta)/(c-\eta) \to s/c = r(\rho)$ as $\eta \to 0$, so some $\eta \in (0, (c-s)/2)$ gives $s + \eta < (\alpha_{\mathrm{GW}} + \varepsilon)(c - \eta)$. With $c' = c - \eta$ and $s' = s + \eta$ we have $0 \leq s' < c' \leq 1$, and (c', s')-Max-Cut is NP-hard by Theorem 18.9. An algorithm as in the second statement would decide the (c', s') problem: on an instance of value at least c' it finds a cut of value at least $(\alpha_{\mathrm{GW}} + \varepsilon)c' > s'$, while on an instance of value at most s' every cut has value at most s'. □

18.5 Beyond Max-Cut

The proof used only two properties of the Max-Cut problem:

1) a dictator test whose checks are Max-Cut constraints, with completeness c;

2) the soundness of this test for functions with small low-degree influences and with values in $[-1, 1]$.

For a family $\mathcal{F}$ of predicates of arity q and a q-query test, the same construction picks q independent random neighbors $w_1, \dots, w_q$ of v and queries the opinion $g_v^{w_j}$ at the j-th query point. Given v and the query points, the answers are independent random signs with means $h_v(x^1), \dots, h_v(x^q)$, so the test sees exactly the average opinion h_v run as a randomized function, as in (18.2). The completeness and soundness proofs then go through with 2λ replaced by $q\lambda$, and they prove the reduction theorem of Lecture 17 in general. The remaining question is where good dictator tests come from.

Recall that a *predicate* is a function $P\colon \{-1,1\}^q \to \{0,1\}$. For example, the 3SAT predicate is $P(z_1, z_2, z_3) = z_1 \vee z_2 \vee z_3$, where -1 stands for true, so that $P(z) = 0$ exactly when $z = (1, 1, 1)$. Max-3SAT is Max-$\mathcal{F}$ for the family $\mathcal{F}$ of the eight predicates obtained from P by negating some of its inputs. There is a natural generalization of the Goemans–Williamson semidefinite program (Lecture 4) to Max-$\mathcal{F}$ problems.

Definition 18.11. Let Φ be an instance of Max-$\mathcal{F}$ with variables U. The *basic semidefinite program* of Φ asks for vectors $v_\emptyset$ and v_u, $u \in U$, in some $\mathbb{R}^N$, with $\|v_\emptyset\|_2 = 1$, and for every constraint $\gamma = (P, (u_1, \dots, u_q))$ a probability distribution λ_γ on $\{-1,1\}^q$, such that for all γ and all $j, k \in [q]$,

$$\mathbb{E}_{z\sim\lambda_\gamma}[z_j] = \langle v_\emptyset, v_{u_j}\rangle, \qquad \mathbb{E}_{z\sim\lambda_\gamma}[z_j z_k] = \langle v_{u_j}, v_{u_k}\rangle.$$

Its value $\mathrm{SDP}(\Phi)$ is the supremum, over such solutions, of $\mathbb{E}_\gamma \Pr_{z\sim\lambda_\gamma}[P_\gamma(z) = 1]$, where γ is a random constraint of Φ and P_γ its predicate.

Taking $j = k$ shows that v_u is a unit vector for every variable u that occurs in a constraint. Every assignment a gives a solution, with $v_u = a(u)v_\emptyset$ and λ_γ the point mass at $(a(u_1), \dots, a(u_q))$, so $\mathrm{SDP}(\Phi) \geq \mathrm{val}(\Phi)$. For Max-Cut, $\Pr_{z\sim\lambda_\gamma}[z_1 \neq z_2] = \frac{1}{2} - \frac{1}{2}\langle v_{u_1}, v_{u_2}\rangle$, and conversely, unit vectors v_u together with a unit vector $v_\emptyset$ orthogonal to all of them admit distributions λ_γ (uniform z_1, and $z_2 = z_1$ with probability $\frac{1}{2} + \frac{1}{2}\langle v_{u_1}, v_{u_2}\rangle$); so the program has the value of the Goemans–Williamson relaxation. An instance Φ with $\mathrm{SDP}(\Phi) \geq c$ and $\mathrm{val}(\Phi) \leq s$ is a (c, s)-*integrality gap*.

Theorem 18.12 (Raghavendra [Rag08])**.** *Let $\mathcal{F}$ be a finite family of predicates of arity q, and suppose that there is a (c, s)-integrality gap for the basic semidefinite program of Max-$\mathcal{F}$. Then for every $\eta > 0$ there exist $\tau > 0$ and $\delta > 0$ such that for every $n \geq 1$ there is a q-query dictator test on $\{-1,1\}^n$, all of whose predicates lie in $\mathcal{F}$, with completeness $c - \eta$ and soundness $s + \eta$: every dictator passes with probability at least $c - \eta$, and every $h\colon \{-1,1\}^n \to [-1,1]$ with $\mathrm{Inf}_i^{1/\delta}(h) \leq \tau$ for all $i \in [n]$ passes with probability at most $s + \eta$.*

We state this theorem without proof. Combined with the reduction theorem of Lecture 17, applied with completeness $c - \eta/2$ and soundness $s + \eta/2$ and with the tests given for $\eta/2$, it shows that, assuming the Unique Games Conjecture, the $(c - \eta, s + \eta)$-Max-$\mathcal{F}$ problem is NP-hard for every η with $0 < \eta < (c - s)/2$. Thus every integrality gap of this semidefinite program becomes a hardness result, and for Max-Cut the Feige–Schechtman gap gives back Corollary 18.10.

19 Scheduling Problems: LP Structure and Strengthening LPs

Lecture of October 13, 2021, by Anupam Gupta. Based on the lecturer's handwritten notes.

This lecture treats three scheduling problems, both to survey the kind of problems that arise in scheduling and to show several techniques at work. For a single machine with release dates, we solve the preemptive problem exactly and turn its schedule into a non-preemptive one at a loss of a factor of 2. For a single machine with precedence constraints, we strengthen a linear program by exponentially many valid inequalities and obtain a 2-approximation. For the generalized assignment problem, where the natural LP has an unbounded integrality gap, we guess the optimal makespan and exploit the structure of the LP solutions, in two ways, to obtain a 2-approximation. Sections 4.1 and 4.2 of [WS11] treat single-machine problems of this kind, and the survey [KSW99] covers scheduling algorithms more broadly.

19.1 A single machine with release dates

Scheduling problems are often named in the three-field notation $\alpha \,|\, \beta \,|\, \gamma$ of Graham, Lawler, Lenstra and Rinnooy Kan [GLLRK79]. The first field describes the machines (1 for a single machine), the second lists the features of the jobs (r_j for release dates, prec for precedence constraints, pmtn for preemption), and the third gives the objective. Our first problem is $1 \,|\, r_j \,|\, \sum_j C_j$: a single machine, release dates, and the sum of completion times as the objective.

Definition 19.1. An instance of $1 \,|\, r_j \,|\, \sum_j C_j$ consists of n jobs; job $j \in [n]$ has a *processing time* $p_j > 0$ and a *release date* $r_j \geq 0$. A *non-preemptive schedule* assigns to each job a start time $S_j \geq r_j$ so that the intervals $[S_j, S_j + p_j)$ are pairwise disjoint; the *completion time* of job j is $C_j = S_j + p_j$. In a *preemptive schedule*, a job may be interrupted and resumed later: each job j is processed during a set $T_j \subseteq [r_j, \infty)$, a finite union of intervals of total length p_j, the sets T_j are pairwise disjoint, and $C_j = \sup T_j$. The goal is to minimize $\sum_j C_j$. We write OPT and $\mathrm{OPT}_{\mathrm{pmtn}}$ for the optimal values over non-preemptive and over preemptive schedules; the preemptive problem is $1 \,|\, r_j, \mathrm{pmtn} \,|\, \sum_j C_j$.

Every non-preemptive schedule is also a preemptive one, so $\mathrm{OPT}_{\mathrm{pmtn}} \leq \mathrm{OPT}$. The plan is to solve the preemptive problem first and then use its solution to solve the general non-preemptive problem approximately.

Definition 19.2. The *shortest remaining processing time* (SRPT) rule processes, at each time, a released and unfinished job whose remaining processing time is smallest, breaking ties by index; the machine idles only when no released job is unfinished.

The SRPT schedule changes the job being processed only at release dates and completion times, so it has at most $2n$ events and can be computed in polynomial time.

Fact 19.3 (Schrage [Sch68])**.** *For the preemptive problem $1 \,|\, r_j, \mathrm{pmtn} \,|\, \sum_j C_j$, the SRPT rule gives an optimal schedule.*

Proof sketch. The proof is an exchange argument. Let $\rho_j(t)$ denote the remaining processing time of job j at time t in a schedule σ. Suppose that at time t the schedule σ processes a job k although a released, unfinished job j has $\rho_j(t) < \rho_k(t)$. Let U be the set of times after t at which σ processes j or k; it has measure $\rho_j(t) + \rho_k(t)$. Change σ on U only: process j during the first $\rho_j(t)$ units of U and k during the rest. The job among j, k that finishes first in σ, at

time $\min(C_j, C_k)$, had at least $\rho_j(t)$ units of work left at time t, all processed in U before that time; so j now finishes by $\min(C_j, C_k)$, and k finishes at $\sup U = \max(C_j, C_k)$. No other job moves, so the sum of completion times does not increase. Repeating such exchanges turns an optimal schedule into the SRPT schedule. This sketch leaves out the argument that finitely many exchanges suffice, which can be made precise, for example, by discretizing time when the data are integers. □

Now take an optimal preemptive schedule, with completion times $\bar{C}_1, \dots, \bar{C}_n$, and number the jobs so that $\bar{C}_1 \le \bar{C}_2 \le \cdots \le \bar{C}_n$. This order is the input to the following algorithm.

Algorithm 19.4. Schedule the jobs non-preemptively in the order $1, 2, \dots, n$, each as early as possible: if job j is not released yet when job $j-1$ finishes, the machine waits. In formulas, with $C_0 := 0$,

$$C_j = \max(C_{j-1}, r_j) + p_j \qquad (j = 1, \dots, n).$$

Lemma 19.5. *For every job j, the schedule of Algorithm 19.4 satisfies $C_j \le \bar{C}_j + \sum_{i \le j} p_i \le 2\bar{C}_j$.*

Proof. Take the preemptive schedule and, for each job j, insert at time $\bar{C}_j$ a block of length p_j in which a second copy of job j runs without interruption; the insertion delays everything after time $\bar{C}_j$ by p_j. (Blocks inserted at the same time are placed in index order.) Before the second copy of job j ends, exactly the blocks of jobs $1, \dots, j$ have been inserted, so it completes at $C'_j := \bar{C}_j + \sum_{i \le j} p_i$. Deleting the original pieces leaves the second copies, which form a non-preemptive schedule in the order $1, \dots, n$; the copy of j starts after time $\bar{C}_j \ge r_j$, so release dates are respected. Algorithm 19.4 processes the jobs in the same order, each as early as possible, so $C_j \le C'_j$ by induction on j: if $C_{j-1} \le C'_{j-1}$, then $C_j = \max(C_{j-1}, r_j) + p_j \le \max(C'_{j-1}, r_j) + p_j \le C'_j$, because the copy of j starts no earlier than C'_{j-1} and no earlier than r_j. Finally, jobs $1, \dots, j$ are all processed completely within $[0, \bar{C}_j]$ in the preemptive schedule, so $\sum_{i \le j} p_i \le \bar{C}_j$, and $C_j \le C'_j \le 2\bar{C}_j$. □

Theorem 19.6 (Phillips, Stein and Wein [PSW98])**.** *Computing the SRPT schedule and applying Algorithm 19.4 to its order is a polynomial-time 2-approximation algorithm for $1 \,|\, r_j \,|\, \sum_j C_j$.*

Proof. By Lemma 19.5 and Fact 19.3, $\sum_j C_j \le 2 \sum_j \bar{C}_j = 2\,\mathrm{OPT}_{\mathrm{pmtn}} \le 2\,\mathrm{OPT}$. □

19.2 A single machine with precedence constraints

We now consider a more difficult problem, $1 \,|\, \mathrm{prec} \,|\, \sum_j w_j C_j$. There are no release dates (the approach extends to them with a little more effort), but the jobs carry weights and precedence constraints.

Definition 19.7. An instance of $1 \,|\, \mathrm{prec} \,|\, \sum_j w_j C_j$ consists of n jobs, where job j has a processing time $p_j > 0$ and a weight $w_j \ge 0$ (its importance), together with a partial order $\prec$ on $[n]$; $i \prec j$ means that job i must finish before job j starts. All jobs are available at time 0. A feasible schedule is a non-preemptive schedule on one machine (Definition 19.1, with all $r_j = 0$) that respects $\prec$, and the goal is to minimize $\sum_j w_j C_j$. We write OPT for the optimal value.

For a relaxation, we take the completion times C_j as variables and minimize $\sum_j w_j C_j$. For $i \prec j$, job j starts at $C_j - p_j \ge C_i$, so we want the constraint

$$C_j \ge C_i + p_j \qquad \text{for all jobs } i \prec j.$$

For two jobs j, k that are not comparable, one of them runs first, so we would like

$$C_j \geq C_k + p_j \quad \text{or} \quad C_k \geq C_j + p_k.$$

This disjunction is not a convex constraint, let alone a linear one: the points $(C_j, C_k) = (p_j, 0)$ and $(0, p_k)$ satisfy it, but their midpoint does not. So we use a different family of constraints.

For a set $S \subseteq [n]$ of jobs, let

$$p(S) \coloneqq \sum_{j \in S} p_j \qquad \text{and} \qquad p^2(S) \coloneqq \sum_{j \in S} p_j^2.$$

Lemma 19.8. *For every non-preemptive schedule on one machine (with or without idle time) and every $S \subseteq [n]$, the completion times satisfy*

$$\sum_{j \in S} p_j C_j \geq \frac{1}{2}\left[p(S)^2 + p^2(S)\right]. \tag{19.1}$$

In other words, (19.1) is a *valid* constraint: it holds for the completion times of every schedule, in particular of the optimal one.

Proof. For $S = \emptyset$ both sides are 0. Otherwise, relabel so that $S = \{1, \dots, m\}$ and $C_1 \leq C_2 \leq \cdots \leq C_m$. The jobs $1, \dots, j$ are processed in disjoint intervals inside $[0, C_j]$, so $C_j \geq \sum_{k \leq j} p_k$. Hence

$$\sum_{j \in S} p_j C_j \geq \sum_{j=1}^{m} \sum_{k=1}^{j} p_j p_k = \frac{1}{2}\left[p(S)^2 + p^2(S)\right],$$

where the equality holds because $p(S)^2 = \sum_{j,k \in S} p_j p_k$ counts each pair $k < j$ twice and each pair $k = j$ once. □

So here is a linear program:

$$\begin{aligned} \text{minimize} \quad & \sum_j w_j C_j \\ \text{subject to} \quad & C_j \geq C_i + p_j && \text{for all } i \prec j, \\ & \sum_{j \in S} p_j C_j \geq \frac{1}{2}\left[p^2(S) + p(S)^2\right] && \text{for all } S \subseteq [n]. \end{aligned} \tag{19.2}$$

The completion times of every feasible schedule satisfy both families of constraints, the second by Lemma 19.8; hence the optimal value LP of (19.2) satisfies LP $\leq$ OPT. The LP has exponentially many constraints, but that is not an obstacle: Lecture 25 shows how to solve such LPs given a separation oracle, and the next remark provides one.

Remark 19.9 (Queyranne [Que93]). For a given vector C, the most violated constraint of the second family is attained by a set of the form $\{j : C_j \leq t\}$, so it suffices to check the at most n sets $\{j : C_j \leq C_k\}$. Indeed, let $f(S) \coloneqq \sum_{j \in S} p_j C_j - \frac{1}{2}[p(S)^2 + p^2(S)]$ and let S minimize f, with $P = p(S)$. A short computation gives $f(S \setminus \{j\}) - f(S) = p_j(P - C_j)$ for $j \in S$ and $f(S \cup \{k\}) - f(S) = p_k(C_k - P - p_k)$ for $k \notin S$. Both are nonnegative by minimality, so $C_j \leq P$ for $j \in S$ and $C_k \geq P + p_k > P$ for $k \notin S$; that is, $S = \{j : C_j \leq P\}$.

Suppose that we solve (19.2) and get an optimal solution $\bar{C}$, and renumber the jobs so that $\bar{C}_1 \leq \bar{C}_2 \leq \cdots \leq \bar{C}_n$. If $i \prec j$, then $\bar{C}_j \geq \bar{C}_i + p_j > \bar{C}_i$, so this order respects the precedence constraints. Now schedule the jobs in this order without idle time, so that $C_j = p_1 + p_2 + \cdots + p_j$.

Lemma 19.10. *For each job j, $C_j \le 2\bar{C}_j$.*

Proof. Let $S = \{1, 2, \dots, j\}$. Then $C_j = p_1 + \cdots + p_j = p(S)$. The constraint of (19.2) for S gives $\sum_{i \in S} p_i \bar{C}_i \ge \frac{1}{2} p(S)^2$. On the other hand, the jobs are sorted in the order of increasing $\bar{C}$, so $\sum_{i \in S} p_i \bar{C}_i \le \sum_{i \in S} p_i \bar{C}_j = p(S)\, \bar{C}_j$. Together, $p(S)\, \bar{C}_j \ge \frac{1}{2} p(S)^2$, and since $p(S) > 0$, $\bar{C}_j \ge \frac{1}{2} p(S) = \frac{1}{2} C_j$. □

Hence each job finishes at most twice as late as in the fractional schedule.

Theorem 19.11 (Hall, Schulz, Shmoys and Wein [HSSW97])**.** *Scheduling the jobs in the order of an optimal solution of (19.2) is a 2-approximation algorithm for $1 \,|\, \text{prec} \,|\, \sum_j w_j C_j$.*

Proof. Since the weights are nonnegative, Lemma 19.10 gives $\sum_j w_j C_j \le 2 \sum_j w_j \bar{C}_j = 2\,\text{LP} \le 2\,\text{OPT}$. □

Remark 19.12. The lesson of this section is to find valid inequalities, satisfied by the schedules we are looking for, and to add them to the relaxation: we *strengthen* the LP. This idea returns in Lecture 21.

19.3 The generalized assignment problem

Definition 19.13. An instance of the *generalized assignment problem* has n jobs and m machines; job j has size $p_{ij} \ge 0$ when it is scheduled on machine i. An *assignment* is a map $\sigma \colon [n] \to [m]$. The *load* of machine i is $\text{load}(i) \coloneqq \sum_{j : \sigma(j) = i} p_{ij}$, and the *makespan* of σ is $\max_{i \in [m]} \text{load}(i)$. Moreover, we are given costs c_{ij}, and the *assignment cost* of σ is $\sum_{i,j} c_{ij}\, \mathbf{1}(\text{job } j \text{ is assigned to machine } i)$.

We first try to minimize the makespan alone, ignoring the assignment costs; Lecture 20 returns to them. Let OPT denote the minimum makespan.

A first attempt. The natural LP relaxation is

$$\begin{aligned} \text{minimize} \quad & L \\ \text{subject to} \quad & L \ge \sum_j p_{ij} x_{ij} && \text{for all } i \in [m], \\ & \sum_i x_{ij} = 1 && \text{for all } j \in [n], \\ & x_{ij} \ge 0 && \text{for all } i, j. \end{aligned} \tag{19.3}$$

It is not a good relaxation.

Example 19.14. Let there be a single job, of size $p_{i1} = 1$ on every machine i. Setting $x_{i1} = 1/m$ for all i and $L = 1/m$ is feasible for (19.3), but $\text{OPT} = 1$. So the integrality gap of (19.3) is at least m.

Guessing the optimal makespan. The fix is to guess the optimal load L^* and to forbid every pair with $p_{ij} > L^*$, which no assignment of makespan L^* can use. For $L^* \ge 0$, let $\text{LP}(L^*)$

denote the feasibility problem

$$\begin{aligned} &\sum_j p_{ij} x_{ij} \le L^* && \text{for all } i \in [m], \\ &\sum_i x_{ij} \ge 1 && \text{for all } j \in [n], \\ &x_{ij} \ge 0 \ \text{ for all } i,j, \qquad x_{ij} = 0 \ \text{ whenever } p_{ij} > L^*. \end{aligned} \tag{19.4}$$

If $L^* \ge \mathrm{OPT}$, then $\mathrm{LP}(L^*)$ is feasible: the indicator vector of an optimal assignment is a solution, since every job of size p_{ij} on machine i satisfies $p_{ij} \le \mathrm{load}(i) \le \mathrm{OPT}$. We aim to turn a feasible solution of $\mathrm{LP}(L^*)$ into an assignment of makespan at most $2L^*$. If the sizes are integers, then OPT is an integer, and since feasibility of $\mathrm{LP}(L^*)$ is monotone in L^*, a binary search over the integers finds the smallest integer L^* for which $\mathrm{LP}(L^*)$ is feasible; this L^* is at most OPT, so an assignment of makespan at most $2L^*$ is a 2-approximation.

For a solution x of (19.4), the *support graph* of x is the bipartite graph on the machines and the jobs with an edge between machine i and job j whenever $x_{ij} > 0$. Every edge ij of it has $p_{ij} \le L^*$.

Approach 1: an acyclic support. The first approach, due to Lenstra, Shmoys and Tardos [LST90], uses a solution whose support graph is a forest. If $p_{ij} = 0$ for some pair, we assign job j to machine i at the start, which changes no load and keeps the rest of x feasible; so we may assume that $p_{ij} > 0$ for all i, j.

Homework 19.15. Assume $p_{ij} > 0$ for all i, j. Show that for every feasible solution x of (19.4) there is a feasible solution x' whose support graph is a forest and is contained in the support graph of x.

Remark 19.16. An extreme point of the feasible region of (19.4) need not have an acyclic support. With two machines and three jobs, sizes $(p_{11}, p_{12}, p_{13}) = (1, 1, 2)$ and $(p_{21}, p_{22}, p_{23}) = (1, 2, 1)$, and $L^* = 2$, the solution with $x_{12} = x_{13} = 2/3$, $x_{21} = 1$, $x_{22} = x_{23} = 1/3$ and $x_{11} = 0$ is a vertex of (19.4) (the six tight constraints are linearly independent), and its support contains the cycle through machine 1, job 2, machine 2 and job 3.

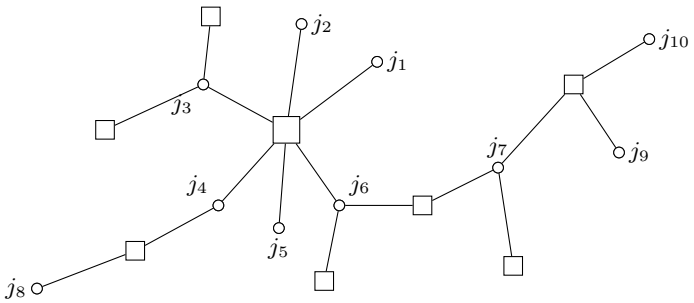


Figure 28: A support graph that is a tree: squares are machines (the large one is adjacent to $j_1, \ldots, j_6$) and circles are jobs. The jobs $j_1, j_2, j_5, j_8, j_9, j_{10}$ are leaves and go to their unique machine.

Lemma 19.17. *Let x be a feasible solution of (19.4) whose support graph is a forest. Then there is an assignment, computable in polynomial time, in which every machine i has load at most $\sum_j p_{ij}x_{ij} + L^* \le 2L^*$.*

Proof. Every job has degree at least 1 in the support graph, since $\sum_i x_{ij} \ge 1$. Some jobs go only to a single machine, that is, they are leaves of the forest (in Fig. 28: j_1, j_2, j_5, j_8, j_9 and j_{10}), and they are easy: a leaf job j with unique neighbor i has $x_{ij} = \sum_{i'} x_{i'j} \ge 1$, and we assign it to i, which adds $p_{ij} \le p_{ij}x_{ij}$ to the load of i.

Delete the leaf jobs. In the remaining forest every job has degree at least 2, so all jobs are internal nodes and the leaves are machines. Root each component at a machine, and assign each job to one of its children. A job has exactly one parent and at least two neighbors, so it has a child, which is a machine. Every machine has at most one parent, so it receives at most one job in this step. That job j has $x_{ij} > 0$, hence $p_{ij} \le L^*$. So each machine gets at most one extra job, of size at most $\max_{j:x_{ij}>0} p_{ij} \le L^*$, and the load of machine i is at most $\sum_j p_{ij}x_{ij} + L^* \le 2L^*$. □

Approach 2: the bipartite matching polytope. Another approach, due to Shmoys and Tardos [ST93], uses the integrality of the bipartite matching polytope.

Fact 19.18. *Let H be a bipartite graph with sides U and J and edge set $E(H)$, and let*

$$K(H) := \Big\{ y \in \mathbb{R}_{\ge 0}^{E(H)} : \sum_{u:uj\in E(H)} y_{uj} \ge 1 \text{ for all } j \in J, \quad \sum_{j:uj\in E(H)} y_{uj} \le 1 \text{ for all } u \in U \Big\}.$$

Every vertex (extreme point) of $K(H)$ is integral, that is, lies in $\{0,1\}^{E(H)}$. Since $K(H)$ is a bounded polytope, every $y \in K(H)$ is a convex combination of integral points of $K(H)$.

An integral point of $K(H)$ is a set of edges that covers every vertex of J at least once and every vertex of U at most once. When H is the support graph of a given $y \in K(H)$, which amounts to imposing $y_{uj} = 0$ on all other pairs, these integral points use only pairs in the support of y.

To use this, we must deal with the constraints $\sum_j p_{ij}x_{ij} \le L^*$ of (19.4), whose coefficients are not all 1. So we do the following.

Definition 19.19. Let x be a feasible solution of (19.4). Each machine i gets some *mass* $M_i := \sum_j x_{ij}$. Split machine i into $\lceil M_i \rceil$ *mini-machines* $i_1, \dots, i_{\lceil M_i \rceil}$; for example, $M_i = 3.5$ gives four mini-machines. Sort the jobs j with $x_{ij} > 0$ in order of decreasing p_{ij} (so the order differs from machine to machine), and pour their amounts x_{ij}, in this order, into the mini-machines: one unit into i_1, one unit into i_2, and so on, and the remaining $M_i - (\lceil M_i \rceil - 1)$ into the last one; a job whose amount straddles two mini-machines is split between them. Let $y_{i_s j} \ge 0$ be the amount of job j poured into mini-machine i_s.

Example 19.20. Let machine i carry four jobs with $p_{i1} \ge p_{i2} \ge p_{i3} \ge p_{i4}$ and $x_{i1}, x_{i2}, x_{i3}, x_{i4} = 0.6, 0.5, 0.4, 0.9$, so $M_i = 2.4$ and there are three mini-machines (Fig. 29). Mini-machine i_1 receives 0.6 of job 1 and 0.4 of job 2; mini-machine i_2 receives 0.1 of job 2, 0.4 of job 3 and 0.5 of job 4; and i_3 receives the remaining 0.4 of job 4.

Lemma 19.21. *The amounts y of Definition 19.19 satisfy the following.*

1) For every job j, the total amount of j over all mini-machines is $\sum_i x_{ij} \ge 1$.

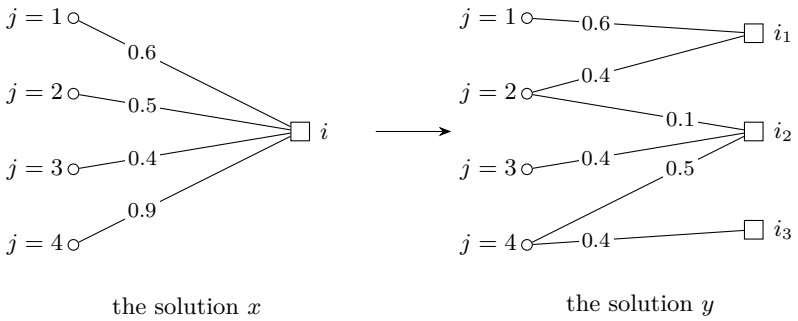


Figure 29: The splitting of Example 19.20: machine i with mass $M_i = 2.4$ becomes three mini-machines; the jobs are sorted by decreasing size $p_{i1} \geq p_{i2} \geq p_{i3} \geq p_{i4}$.

2) Every mini-machine receives total amount at most 1*, and every mini-machine other than the last one of its machine receives exactly* 1*.*

3) For $s \geq 2$, every job with $y_{i_s j} > 0$ has size p_{ij} at most the size of every job with $y_{i_{s-1} j'} > 0$.

Proof. Parts 1 and 2 hold because the pouring preserves each amount x_{ij} and fills the mini-machines one unit at a time. Part 3 holds because the jobs are poured in order of decreasing size, so the jobs that reach i_s come no earlier in this order than those that reach i_{s-1}. □

So the new instance has each job going fractionally to at least one unit of mini-machines in total, and each mini-machine getting at most one unit of mass. Let H be the bipartite graph between the mini-machines and the jobs with an edge $i_s j$ whenever $y_{i_s j} > 0$; then $y \in K(H)$, and $y_{i_s j} = 0$ for many of the pairs.

Theorem 19.22 (Shmoys and Tardos [ST93])**.** *Given a feasible solution x of (19.4), one can find in polynomial time an assignment in which every machine i has load at most $\sum_j p_{ij} x_{ij} + L^* \leq 2L^*$.*

Proof. By Fact 19.18, y is a convex combination of integral points of $K(H)$, and such a point z can be found in polynomial time (as a vertex of $K(H)$). In z, each mini-machine gets at most one job, and each job goes to at least one mini-machine; assign each job to one of them, and to the machine i that this mini-machine belongs to. Every edge used has $y_{i_s j} > 0$, so $x_{ij} > 0$ and $p_{ij} \leq L^*$.

It remains to bound the load of machine i. It receives at most one job from each of its mini-machines. The job on i_1 has size at most $\max_{j : x_{ij} > 0} p_{ij} \leq L^*$. For $s \geq 2$, by part 3 of Lemma 19.21, the job on i_s is no larger than any job that reaches i_{s-1}, and by part 2 the mini-machine i_{s-1} is full, so the size of the job on i_s is at most the weighted average $\sum_j p_{ij} y_{i_{s-1} j}$ of the sizes on i_{s-1}. Summing over $s \geq 2$, the jobs on all mini-machines of i except the first have total size at most

$$\sum_{s=1}^{\lceil M_i \rceil - 1} \sum_j p_{ij} y_{i_s j} \leq \sum_{s=1}^{\lceil M_i \rceil} \sum_j p_{ij} y_{i_s j} = \sum_j p_{ij} x_{ij} \leq L^*,$$

the fractional load of i. Hence the load of i is at most $\sum_j p_{ij} x_{ij} + L^* \leq 2L^*$. □

19.4 Lessons

Three ideas carried this lecture. The structure of LP solutions helped: an acyclic support, or the integrality of the bipartite matching polytope. It is useful to add inequalities that strengthen the LP, such as the exponentially many constraints (19.1), or even the simple constraints $x_{ij} = 0$ whenever $p_{ij} > L^*$. And guessing a parameter of the optimal solution, such as its makespan L^*, can be useful.

Lecture 20 uses the structure of optimal LP solutions once again, in a different way. Systematic ways to add valid constraints and strengthen an LP automatically include hierarchies, lift-and-project methods and knapsack cover inequalities; Lecture 21 begins the study of hierarchies.

20 Iterated Rounding

Lecture of October 15, 2021, by Anupam Gupta. Based on the lecturer's handwritten notes.

A vertex of an n-dimensional linear program is determined by n linearly independent constraints that are tight at it. Counting these constraints often shows that a vertex has an integral coordinate, or that some constraint can be dropped at little cost; fixing that coordinate or dropping that constraint, solving again and repeating is called *iterated rounding*. This lecture applies such rank-based arguments three times: to prove the Beck–Fiala theorem on the discrepancy of set systems, to pack bins with k types of items using at most $k-1$ bins more than the optimum, and to approximate the generalized assignment problem of Lecture 19, now with assignment costs, within a factor of 3 in the makespan. Chapter 11 of [WS11] and the monograph of Lau, Ravi and Singh [LRS11] treat iterated rounding in depth.

20.1 Vertices and tight constraints

Let $A \in \mathbb{R}^{m\times n}$ have rows $a_1^\top, \dots, a_m^\top$, let $b \in \mathbb{R}^m$, and let $P = \{x \in \mathbb{R}^n : Ax \geq b\}$. A point of P is a *vertex* (or *extreme point*, or *basic solution*) if it is not a convex combination of two other points of P. Equality constraints are allowed; each counts as two inequalities.

Fact 20.1. *A point $x^* \in P$ is a vertex of P if and only if there are n rows $a_{i_1}^\top, \dots, a_{i_n}^\top$ of A that are linearly independent and tight at x^*, that is, $a_{i_k}^\top x^* = b_{i_k}$ for $k = 1, \dots, n$. If P is nonempty and contains no line (for example, if P is bounded or $P \subseteq \mathbb{R}^n_{\geq 0}$), then P has a vertex, and every linear function that is bounded below on P attains its minimum over P at a vertex; such a vertex can be computed in polynomial time.*

So if x^* is a vertex solution of $\min\{c^\top x : Ax \geq b\}$, we can use the n tight, linearly independent constraints to argue what x^* looks like.

20.2 The Beck–Fiala theorem

Suppose that we have a universe $U = [n]$ of n elements and m sets $S_1, \dots, S_m \subseteq U$ over these elements, and that each element belongs to at most k of the sets. We want to color the elements red and blue so that every set is nearly balanced. Formally, a *coloring* is a map $\chi\colon U \to \{-1, +1\}$, with $\chi(S) := \sum_{e\in S} \chi(e)$ (the number of red elements of S minus the number of blue ones), and its *discrepancy* is

$$\operatorname{disc}(\chi) := \max_{i\in[m]} |\chi(S_i)|.$$

Theorem 20.2 (Beck and Fiala [BF81])**.** *Let $S_1, \dots, S_m \subseteq [n]$ be sets such that every element of $[n]$ belongs to at most k of them, where $k \geq 1$. Then there is a coloring $\chi\colon [n] \to \{-1, +1\}$ with $\operatorname{disc}(\chi) \leq 2k-1$, and it can be found in polynomial time.*

We write an LP relaxation. Let $a_i \in \{0,1\}^n$ be the incidence vector of S_i, and let $A \in \{0,1\}^{m\times n}$ be the matrix with rows $a_1^\top, \dots, a_m^\top$; every column of A has at most k ones. We would love to have $Ax = 0$ with $x \in \{-1, 1\}^n$, which is a coloring of discrepancy zero. The LP relaxation is $Ax = 0$, $-1 \leq x_j \leq 1$ for all j: it has n variables and $m+n$ constraints. Consider a basic solution x; among its n tight, linearly independent constraints, if any is of the form $x_j = -1$ or $x_j = 1$, we fix that coordinate. To keep track, let $F \subseteq [n]$ be the set of fixed variables, with

fixed values $\hat{x}_j \in \{-1, 1\}$ for $j \in F$, and let $M \subseteq [m]$ be the set of constraints still in play. They define the system

$$\sum_{j \notin F} a_{ij} x_j = -\sum_{j \in F} a_{ij} \hat{x}_j \quad \text{for all } i \in M, \qquad -1 \le x_j \le 1 \quad \text{for all } j \in [n] \setminus F, \tag{20.1}$$

in the variables x_j, $j \in [n] \setminus F$. Its solution set is a bounded polytope.

Algorithm 20.3. Start with $F \leftarrow \emptyset$ and $M \leftarrow [m]$. While $F \neq [n]$, repeat the following.

1) Find a vertex x of the polytope (20.1).

2) If $x_j \in \{-1, 1\}$ for some $j \notin F$, then for every such j set $\hat{x}_j \leftarrow x_j$ and add j to F. This gives a new system with fewer variables.

3) Otherwise, choose $i \in M$ such that S_i has at most k elements outside F, drop its constraint, and remove i from M. This gives a new system with fewer constraints.

Output the coloring $\chi(j) \coloneqq \hat{x}_j$.

Lemma 20.4. *At every iteration of Algorithm 20.3, the polytope (20.1) is nonempty, and if step 2 does not apply, then a set i as in step 3 exists.*

Proof. Initially $x = 0$ is a solution. In step 2 the vertex x, restricted to the variables that remain free, solves the new system, because the fixed values move to the right-hand side unchanged; in step 3, dropping a constraint keeps x feasible. So the system stays feasible, and being bounded, it has a vertex.

Suppose that step 2 does not apply, and let $n' = n - |F| \ge 1$ be the number of free variables. By Fact 20.1, the vertex x has n' linearly independent tight constraints. None of them is of the form $x_j = \pm 1$, so they are n' of the equations of (20.1), say $a_{i_1}, \dots, a_{i_{n'}}$ restricted to the free columns: $A_{1:n'} x = b$ for the square submatrix $A_{1:n'}$ formed by these rows and the free columns, with b the right-hand side, whatever it is. The matrix $A_{1:n'}$ has at most k ones per column, by the property of the set system, and it is square, so it has at most kn' ones in its n' rows. Hence some row has at most k ones, which means that the corresponding set S_i has at most k elements outside F. □

Proof of Theorem 20.2. Each iteration of Algorithm 20.3 adds an element to F or removes a set from M, so the algorithm stops after at most $n + m$ iterations, by Lemma 20.4, and each iteration solves one LP.

Consider a set S_i. If i is still in M at the end, then $\chi(S_i) = 0$ by its constraint. Otherwise, i was removed from M in step 3 of some iteration, with vertex x and fixed set F. At that moment $\sum_{j \in S_i \cap F} \hat{x}_j + \sum_{j \in S_i \setminus F} x_j = 0$, so the discrepancy of S_i is 0 if the values x_j do not change later. But they may change: each of the at most k elements $j \in S_i \setminus F$ has $x_j \in (-1, 1)$, since step 2 did not apply, and its final color $\chi(j) \in \{-1, 1\}$ differs from x_j by less than 2 (from, say, $-1 + \varepsilon$ to $+1$), while the elements of $S_i \cap F$ keep their values. Hence

$$|\chi(S_i)| = \Big| \sum_{j \in S_i \setminus F} (\chi(j) - x_j) \Big| < 2k.$$

Since $\chi(S_i)$ is an integer, $|\chi(S_i)| \le 2k - 1$. □

Remark 20.5. Improving the bound to $(2-\varepsilon)k$ for a constant $\varepsilon > 0$ was an open problem as of 2021. The best bound depending on k alone was $2k - \log^* k$ for large k, due to Bukh [Buk16], and Beck and Fiala conjectured that the discrepancy is $O(\sqrt{k})$.

This idea, using the fact that a vertex of an n-dimensional LP has n tight, linearly independent constraints, so that the tight submatrix A_I with $A_I x = b_I$ and $|I| = n$ has rank n, is called a *rank argument*. Rank arguments often give algorithms of the same type as Algorithm 20.3, which are called *iterative* or *iterated rounding*.

20.3 Bin packing with few item types

Definition 20.6. In *bin packing with k item types*, where $k \geq 1$, we are given sizes $s_1, \dots, s_k \in (0, 1]$ and multiplicities $n_1, \dots, n_k \in \mathbb{Z}_{\geq 0}$: there are n_i copies of an item of size s_i. The goal is to find the smallest number of bins of size 1 such that all items can be packed into them; we write OPT for this number. A *pattern* is a configuration of items that fits into a bin, written as a vector $X_p = (a_1, \dots, a_k)$ of nonnegative integers, where a_i is the number of copies of item i in the pattern, such that $\sum_i a_i s_i \leq 1$. Let $\mathcal{P}$ be the set of patterns and $N = |\mathcal{P}|$.

Since every $s_i > 0$, each pattern has $a_i \leq 1/s_i$, so N is finite. We want to choose, for each pattern p, the number x_p of bins packed according to p. This gives the LP

$$\begin{aligned} \text{minimize} \quad & \sum_{p \in \mathcal{P}} x_p \\ \text{subject to} \quad & \sum_{p \in \mathcal{P}} x_p X_p \geq \begin{pmatrix} n_1 \\ \vdots \\ n_k \end{pmatrix}, \\ & x_p \geq 0 \quad \text{for all } p \in \mathcal{P}. \end{aligned} \tag{20.2}$$

A packing into b bins gives a solution of value b (each bin follows some pattern), so its optimal value LP satisfies LP $\leq$ OPT. Conversely, an integral solution x packs all items into $\sum_p x_p$ bins: the bins provide at least n_i slots for items of type i, and unused slots stay empty.

Lemma 20.7. *Every basic solution of (20.2) has at most k nonzero coordinates.*

Proof. The LP has N variables and $k + N$ constraints. By Fact 20.1, a basic solution has N linearly independent tight constraints. At most k of them are covering constraints, so at least $N - k$ of them are nonnegativity constraints $x_p \geq 0$. Hence at least $N - k$ of the variables are 0. □

Theorem 20.8. *Given a basic optimal solution x of (20.2), rounding each x_p up to $\lceil x_p \rceil$ packs all items into at most $\lceil \text{LP} + (k-1) \rceil \leq \text{OPT} + k - 1$ bins.*

Proof. The rounded vector is integral and still satisfies the covering constraints, so it is a packing. By Lemma 20.7, at most k coordinates are fractional, and rounding each of them up adds less than 1, so the packing opens at most LP $+ k$ bins, and in fact fewer if some coordinate is fractional. In that case the number of bins is an integer smaller than LP $+ k$, hence at most $\lceil \text{LP} + k \rceil - 1 = \lceil \text{LP} + (k-1) \rceil$. If no coordinate is fractional, the number of bins is LP $\leq \lceil \text{LP} + (k-1) \rceil$, since $k \geq 1$. Finally, LP $\leq$ OPT and OPT is an integer, so $\lceil \text{LP} + (k-1) \rceil \leq \text{OPT} + k - 1$. □

How to solve (20.2), which has one variable per pattern, is a separate matter that we do not discuss; the standard method is column generation [GG61]. It can indeed be solved, and a basic optimal solution found, which by Theorem 20.8 gives a solution with at most $\mathrm{OPT} + (k-1)$ bins.

20.4 The generalized assignment problem with costs

Our third example is scheduling on unrelated machines again (Lecture 19), but now we add assignment costs; this is the generalized assignment problem. We are given jobs J, machines $[m]$, sizes $p_{ij} \geq 0$ and costs c_{ij} for job j on machine i, and a target T for the load. As in Lecture 19, an assignment of makespan at most T never puts job j on machine i if $p_{ij} > T$, so we work with the set of *permissible edges* $E \coloneqq \{ij : p_{ij} \leq T\}$ of the bipartite graph between machines and jobs. For a job j, let $\delta(j) \subseteq E$ be the set of edges at j, and write $x(\delta(j)) = \sum_{e \in \delta(j)} x_e$. With budgets T_i (initially $T_i = T$ for every i) and a set $M \subseteq [m]$ of machines whose load constraints are in play (initially $M = [m]$), consider the LP

$$\begin{aligned} \text{minimize} \quad & \sum_{ij \in E} c_{ij} x_{ij} & & \\ \text{subject to} \quad & x(\delta(j)) = 1 & & \text{for all } j \in J, \\ & \sum_{j : ij \in E} p_{ij} x_{ij} \leq T_i & & \text{for all } i \in M, \\ & x \in [0,1]^E. & & \end{aligned} \tag{20.3}$$

The indicator vector of any assignment of makespan at most T is a solution of the initial LP, so its optimal value $\mathrm{LP}(T)$ is at most the cost of every such assignment.

Algorithm 20.9. Start with $F \leftarrow \emptyset$ (the fixed assignments), J the set of all jobs, $M \leftarrow [m]$, $T_i \leftarrow T$ for all i, and E the set of permissible edges. While $J \neq \emptyset$, compute a basic optimal solution x of (20.3) and do the first of the following that applies.

1) If some edge $e \in E$ has $x_e = 0$, drop it: $E \leftarrow E \setminus \{e\}$.
2) Else, if some edge $ij \in E$ has $x_{ij} = 1$, assign j to machine i and fix this value: $F \leftarrow F \cup \{j \to i\}$, $T_i \leftarrow T_i - p_{ij}$, $J \leftarrow J \setminus \{j\}$ and $E \leftarrow E \setminus \{ij\}$.
3) Else, if some machine $i \in M$ has at most 2 edges of E incident to it, remove its load constraint: $M \leftarrow M \setminus \{i\}$.

Then resolve the LP and repeat.

In step 2 no edge has value 0, and $x(\delta(j)) = 1$ with $x_{ij} = 1$, so ij is the only edge at j; hence every edge of E always joins a job of J to a machine. Each round that finds an $x_e \in \{0,1\}$ or a machine of degree at most 2 in M makes progress: it removes an edge, a job or a load constraint.

Lemma 20.10. *If the initial LP is feasible, then the LP (20.3) stays feasible throughout Algorithm 20.9, and the cost of the fixed assignments plus the optimal value of the current LP never increases. In particular, if the algorithm assigns all jobs, the total cost of the assignment is at most* $\mathrm{LP}(T)$.

Proof. Let x be the current basic optimal solution. After step 1 or step 3, x (restricted to the remaining edges) is still feasible, with the same cost. After step 2, the restriction of x to $E \setminus \{ij\}$ is feasible for the new LP: the jobs other than j are unaffected, and if $i \in M$, its load constraint holds because $\sum_{j' \neq j} p_{ij'} x_{ij'} \leq T_i - p_{ij}$. Its cost is the old optimal value minus c_{ij}, while the fixed cost grows by c_{ij}. When $J = \emptyset$, the LP has value 0, so the fixed cost is at most the initial optimal value $\mathrm{LP}(T)$. □

Lemma 20.11. *If $J \neq \emptyset$, the LP is feasible and x is a basic solution of (20.3), then $x_e \in \{0, 1\}$ for some $e \in E$, or some machine of M has at most 2 edges of E incident to it. Hence Algorithm 20.9 always makes progress.*

Proof. Suppose not. For a vertex v, let $d_E(v)$ be its degree, the number of edges of E incident to it. None of the constraints $0 \leq x_e \leq 1$ is tight, so by Fact 20.1 there are $|E|$ tight, linearly independent constraints given by the jobs of some $J' \subseteq J$ and the machines of some $M' \subseteq M$, with $|J'| + |M'| = |E|$. Every job $j \in J$ has $d_E(j) \geq 2$, because $x(\delta(j)) = 1$ and every $x_e < 1$; every machine $i \in M$ has $d_E(i) \geq 3 \geq 2$. Every edge has exactly one endpoint in J and at most one in M. Now

$$|J'| + |M'| = |E| \geq \frac{\sum_{j \in J} d_E(j) + \sum_{i \in M} d_E(i)}{2} \geq |J| + |M| \geq |J'| + |M'|,$$

where the middle inequality holds because all degrees are at least 2. So all these inequalities are tight, which means:

(a) all edges are incident to machines in M, and none to machines whose constraints were dropped;

(b) all machines in M and all jobs in J have degree exactly 2;

(c) $J' = J$ and $M' = M$.

So the assignment graph has degree 2 on both sides: $d_E(j) = d_E(i) = 2$ for all $i \in M$ and $j \in J$. But $d_E(i) \geq 3$ for every $i \in M$, so $M = \emptyset$; and by (a) the jobs of $J \neq \emptyset$, which have degree 2, cannot go to machines outside M. This is a contradiction. □

Theorem 20.12. *If the initial LP (20.3) is feasible, then Algorithm 20.9 terminates after at most $|E| + |J| + m$ iterations and assigns all jobs, so that every machine has load at most $3T$ and the total cost is at most $\mathrm{LP}(T)$, which is at most the cost of every assignment of makespan at most T.*

Proof. By Lemmas 20.10 and 20.11, every iteration removes an edge, a job or a load constraint, so the algorithm terminates, with $J = \emptyset$. Fix a machine i. Until its constraint is dropped, the jobs fixed to i have total size $T - T_i \leq T$, because the load constraint, with $x \geq 0$, keeps $T_i \geq 0$. If the constraint is never dropped, the load of i is at most T. Otherwise, at the moment it is dropped, at most 2 edges of E are incident to i, and E only shrinks afterward, so at most 2 extra jobs are assigned to i later; each has $p_{ij} \leq T$ because ij is permissible. Hence the load of i is at most $T + 2T = 3T$. The cost bound is Lemma 20.10. □

This 3-approximation of the makespan is slightly worse than the factor 2 of Lecture 19. One can do better by slightly more careful arguments; we leave the details to the book of Lau, Ravi and Singh [LRS11].

Iterated rounding is a very powerful idea. For example, for survivable network design, Jain [Jai01] gives an extremely elegant 2-approximation: he shows that every basic solution of the LP has an edge of value at least 1/2, takes that edge and recurses. The book of Lau, Ravi and Singh [LRS11] contains many more elegant ideas.

21 Strengthening Convex Relaxations: SDP Hierarchies I

Lecture of October 18, 2021, by Pravesh K. Kothari. Based on the lecturer's handwritten notes.

When the approximation ratio of a rounding algorithm matches the integrality gap of its relaxation, no algorithm analyzed against that relaxation can do better, and improving the ratio needs a stronger relaxation. This lecture and Lectures 22 and 23 develop a systematic way to strengthen relaxations, the sum-of-squares hierarchy, and apply it to the Knapsack problem. We start with Knapsack itself: a greedy algorithm, analyzed through the natural linear programming relaxation, is a 1/2-approximation, and the integrality gap of that relaxation is exactly 2. We then view solutions of semidefinite programs as relaxations of probability distributions. We define pseudo-distributions on the hypercube and their degree, prove a Cauchy–Schwarz inequality for them, and show that optimizing over pseudo-distributions of degree $2t$ is a semidefinite program of size $n^{O(t)}$. For Max-Cut, degree 2 gives the Goemans–Williamson relaxation of Lecture 4, and degree $2n$ gives the exact problem.

21.1 The knapsack problem

Definition 21.1. An instance of *Knapsack* consists of n items, where item $i \in [n]$ has a *size* $c_i > 0$ and a *value* $v_i \geq 0$, and a *capacity* C with $c_i \leq C$ for every i. The goal is to choose a set of items of total size at most C and maximum total value:

$$\mathrm{OPT} := \max\Big\{\sum_{i=1}^{n} v_i x_i \;\Big|\; x \in \{0,1\}^n,\ \sum_{i=1}^{n} c_i x_i \leq C\Big\}.$$

A vector $x \in \{0,1\}^n$, or the set $\{i : x_i = 1\}$, is *feasible* if $\sum_i c_i x_i \leq C$.

In words, we collect as much value as possible while fitting the chosen items into the knapsack. The assumption $c_i \leq C$ only removes items that fit in no feasible solution. Knapsack is NP-hard to solve exactly [Kar72].

21.2 A greedy algorithm and the linear programming relaxation

After renumbering the items we may assume that

$$\frac{v_1}{c_1} \geq \frac{v_2}{c_2} \geq \cdots \geq \frac{v_n}{c_n}, \tag{21.1}$$

that is, the items are sorted by their *density of value*.

Algorithm 21.2 (Greedy)**.** With the items sorted as in (21.1), let $j \geq 0$ be the largest index with $\sum_{i=1}^{j} c_i \leq C$, and output the set $\{1, \ldots, j\}$.

We write GREEDY for the value $\sum_{i \leq j} v_i$ of this set.

Theorem 21.3. *The better of the greedy set and a single item of maximum value is a* 1/2*-approximation for Knapsack:*

$$\max\Big\{\mathrm{GREEDY},\ \max_{i \in [n]} v_i\Big\} \geq \frac{1}{2}\,\mathrm{OPT}.$$

We prove the theorem by first analyzing the natural linear programming relaxation of Knapsack,

$$\mathrm{LP} := \max\Big\{\sum_{i=1}^{n} v_i y_i \;\Big|\; \sum_{i=1}^{n} c_i y_i \le C,\ 0 \le y_i \le 1 \text{ for all } i\Big\}. \tag{21.2}$$

Every feasible $x \in \{0,1\}^n$ is a feasible solution of (21.2), so $\mathrm{OPT} \le \mathrm{LP}$.

Lemma 21.4. *Let $c_1, \dots, c_n > 0$, $v_1, \dots, v_n \ge 0$ and $C \ge 0$. The lemma does not assume $c_i \le C$; this matters when it is applied to the smaller instances in Lecture 23. Then*

$$\mathrm{LP} \le \mathrm{OPT} + \max_{i \in [n]} v_i.$$

More precisely, with the items sorted as in (21.1): if $\sum_{i=1}^{n} c_i \le C$, then $\mathrm{LP} = \mathrm{OPT} = \sum_i v_i$. Otherwise let k be the smallest index with $\sum_{i \le k} c_i > C$. Then the vector y^* with

$$y_i^* = \begin{cases} 1 & \text{for } i \le k-1, \\ \dfrac{C - \sum_{j=1}^{k-1} c_j}{c_k} & \text{for } i = k, \\ 0 & \text{for } i > k \end{cases}$$

is an optimal solution of (21.2), $0 \le y_k^* < 1$, and

$$\mathrm{LP} \le \sum_{i=1}^{k-1} v_i + v_k \le \mathrm{OPT} + \max_{i \ge k} v_i.$$

Proof. If $\sum_i c_i \le C$, then $y = \mathbf{1}$ is feasible and gives every item its largest possible value, and so does the feasible set $[n]$; hence $\mathrm{LP} = \mathrm{OPT} = \sum_i v_i$. Assume from now on that $\sum_i c_i > C$, so that k is defined. By the choice of k we have $\sum_{i<k} c_i \le C < \sum_{i \le k} c_i$, which gives $0 \le y_k^* < 1$, and y^* is feasible.

Claim 21.5. *Let y be feasible for (21.2), and let $i < j$ with $y_i < 1$ and $y_j > 0$. Set $\delta := \min\{y_j c_j / c_i,\ 1 - y_i\}$, and let y' agree with y except that $y'_i = y_i + \delta$ and $y'_j = y_j - \delta c_i / c_j$. Then y' is feasible, its value is at least that of y, and $y'_i = 1$ or $y'_j = 0$.*

Proof. By the choice of δ we have $0 \le y'_i \le 1$ and $0 \le y'_j \le 1$, and $c_i y'_i + c_j y'_j = c_i y_i + c_j y_j$, so y' is feasible. The value changes by $\delta v_i - \delta c_i v_j / c_j = \delta c_i (v_i/c_i - v_j/c_j) \ge 0$, by (21.1). If $\delta = 1 - y_i$, then $y'_i = 1$; otherwise $\delta = y_j c_j / c_i$ and $y'_j = 0$. □

In words, moving size from a less dense item to a denser one never decreases the value. Iterating this exchange shows that an optimal solution has the form of y^*. To make this precise, note that the optimal solutions of (21.2) form a nonempty compact set, and let y be the lexicographically largest of them. If some $i < j$ had $y_i < 1$ and $y_j > 0$, then the vector y' of Claim 21.5 would be optimal and lexicographically larger than y. Hence there is an index m with $y_i = 1$ for $i < m$ and $y_i = 0$ for $i > m$. Since $\sum_{i<m} c_i \le \sum_i c_i y_i \le C$, the definition of k gives $m \le k$. If $m < k$, then $y_m \le 1 = y_m^*$; if $m = k$, then $c_k y_k \le C - \sum_{i<k} c_i$, that is, $y_k \le y_k^*$. In both cases $y \le y^*$ coordinatewise, and since $v \ge 0$, the value of y^* is at least LP. Hence y^* is optimal.

Consequently,

$$\mathrm{LP} = \sum_{i=1}^{k-1} v_i + y_k^* v_k \le \sum_{i=1}^{k-1} v_i + v_k \le \mathrm{OPT} + \max_{i \ge k} v_i,$$

where the last step holds because the set $\{1, \dots, k-1\}$ is feasible, so its value is at most OPT. □

Proof of Theorem 21.3. If $\sum_i c_i \le C$, the greedy set is $[n]$, which is optimal. Otherwise the greedy set is the largest prefix that fits, $\{1, \dots, k-1\}$ with k as in Lemma 21.4, so $\mathrm{GREEDY} = \sum_{i<k} v_i$. By Lemma 21.4,

$$\mathrm{OPT} \le \mathrm{LP} \le \mathrm{GREEDY} + \max_i v_i \le 2 \max\Big\{\mathrm{GREEDY},\ \max_i v_i\Big\}.$$

Every single item is feasible because $c_i \le C$, so both candidates are feasible solutions. □

The same chain bounds the linear program. Both GREEDY and $\max_i v_i$ are values of feasible solutions, so $\mathrm{LP} \le \mathrm{GREEDY} + \max_i v_i \le 2\,\mathrm{OPT}$. For a maximization problem, the *integrality gap* of a relaxation is the supremum of LP/OPT over all instances with $\mathrm{OPT} > 0$ (compare Lecture 4), so the integrality gap of (21.2) is at most 2. The next example shows that it is exactly 2.

Example 21.6. Let $n \ge 2$ and $0 < \varepsilon < 1$, and take $c_i = v_i = 1$ for every i and $C = 2 - \varepsilon$. Any two items have total size $2 > C$, so a feasible set has at most one item and $\mathrm{OPT} = 1$. The vector $y = (1, 1-\varepsilon, 0, \dots, 0)$ is feasible for (21.2) with value $2 - \varepsilon$, and since $v_i = c_i$, every feasible y has value $\sum_i c_i y_i \le C = 2 - \varepsilon$. Hence $\mathrm{LP} = 2 - \varepsilon$, and the integrality gap of (21.2) is at least $2 - \varepsilon$ for every $\varepsilon > 0$.

So the linear program cannot certify a ratio better than $1/2$. Knapsack nevertheless admits arbitrarily good approximations. Since Knapsack is a maximization problem, a $(1+\varepsilon)$-approximation here is a feasible solution of value at least $\mathrm{OPT}/(1+\varepsilon)$.

Theorem 21.7. *For every $\varepsilon > 0$ there is an algorithm that runs in time $n^{O(1/\varepsilon)}$ and outputs a $(1+\varepsilon)$-approximation for Knapsack.*

This theorem has several proofs. The standard one rounds the values and uses dynamic programming, and even runs in time polynomial in n and $1/\varepsilon$; see [WS11, Section 3.1]. We do not repeat it. Instead, Lectures 22 and 23 give a different proof that the value OPT can be approximated within a factor $1 + \varepsilon$ in time $n^{O(1/\varepsilon)}$, through a stronger relaxation. This is our introduction to hierarchies of convex relaxations.

21.3 Strengthening relaxations

The approach of this course to an optimization problem has been to write a basic relaxation, round its solutions, and compare the result with the integrality gap. Suppose that the approximation ratio of the rounding is close to the integrality gap. One option is to prove that no efficient algorithm does better, as for Max-Cut (Lectures 16 to 18). The other is to look for a better algorithm through a better relaxation, as the triangle inequalities did for Sparsest Cut in the algorithm of Arora, Rao and Vazirani [ARV09] (Lectures 10 and 11).

We now study a mechanical method to construct stronger relaxations. It adds extra variables and additional constraints, and the original variables are recovered as a projection of the new ones; such methods are often called *lift-and-project* methods. Specifically, we study the *sum-of-squares*

method [Par03, Las01]; see also the lecture notes [BS16]. Its starting point is a different way to think about semidefinite programs, one that suggests how to strengthen them: we regard a solution of a semidefinite program as a relaxation of a probability distribution over solutions.

21.4 Pseudo-distributions

Throughout this section, functions are defined on the hypercube $\{-1,1\}^n$. For $S \subseteq [n]$ we write $x_S \coloneqq \prod_{i \in S} x_i$, with $x_\emptyset = 1$. Since $x_i^2 = 1$ on the hypercube, $x_S x_T = x_{S \triangle T}$. By the Fourier expansion (Lecture 16), every function $f\colon \{-1,1\}^n \to \mathbb{R}$ can be written in a unique way as a multilinear polynomial,

$$f(x) = \sum_{S \subseteq [n]} \hat{f}(S)\, x_S.$$

In particular, every function on the hypercube is a polynomial of degree at most n. The *degree* of f is the largest $|S|$ with $\hat{f}(S) \neq 0$. A polynomial of degree d in $x_1, \dots, x_n$, restricted to the hypercube, is a function of degree at most d.

Definition 21.8. A *pseudo-distribution* on $\{-1,1\}^n$ is a function $D\colon \{-1,1\}^n \to \mathbb{R}$ with $\sum_x D(x) = 1$. For a function $f\colon \{-1,1\}^n \to \mathbb{R}$, the *pseudo-expectation* of f with respect to D is

$$\widetilde{\mathbb{E}}_D f \coloneqq \sum_{x \in \{-1,1\}^n} f(x)\, D(x).$$

A pseudo-distribution is the analog of the mass function of a discrete probability distribution, except that it may take negative values. We write $\widetilde{\mathbb{E}}_D f(x)$ or $\widetilde{\mathbb{E}} f$ when convenient.

Example 21.9. Let $p\colon \{-1,1\}^n \to \mathbb{R}_{\geq 0}$ with $\sum_x p(x) = 1$, the mass function of a probability distribution. Then p is a pseudo-distribution, and its pseudo-expectation is the expectation with respect to p: $\widetilde{\mathbb{E}}_p f = \mathbb{E}_{x \sim p} f(x)$.

As defined, a pseudo-distribution has none of the properties of a probability distribution except total mass 1. We now force some nontrivial constraints of being a probability distribution on it.

Definition 21.10. Let $d \geq 0$ be an integer. A pseudo-distribution D on $\{-1,1\}^n$ is *of degree d* if

$$\widetilde{\mathbb{E}}_D f^2 \geq 0 \quad \text{for every polynomial } f \text{ of degree at most } d/2.$$

A pseudo-distribution of degree d is also of every degree $d' \leq d$. Pseudo-expectation is *linear*: for all functions f, g and reals a, b, $\widetilde{\mathbb{E}}(af + bg) = a\,\widetilde{\mathbb{E}} f + b\,\widetilde{\mathbb{E}} g$, directly from the definition.

The mass function of a probability distribution is a pseudo-distribution of every degree, in particular of degree $2n$, since $\mathbb{E} f^2 \geq 0$ for every f. Degree $2n$ is a characterization of probability distributions.

Lemma 21.11. *Every pseudo-distribution D of degree $2n$ on $\{-1,1\}^n$ is the mass function of a probability distribution: $D(z) \geq 0$ for every $z \in \{-1,1\}^n$, and $\widetilde{\mathbb{E}}_D f = \mathbb{E}_{x \sim D} f(x)$ for every f.*

Proof. For $z \in \{-1,1\}^n$, let $\mathbf{1}_z\colon \{-1,1\}^n \to \{0,1\}$ be the indicator of z, that is, $\mathbf{1}_z(x) = 1$ if and only if $x = z$. It is the polynomial

$$\mathbf{1}_z(x) = \prod_{i=1}^{n} \frac{1 + z_i x_i}{2}$$

of degree n. Since D has degree $2n$, the number $p_z := \widetilde{\mathbb{E}}_D \mathbf{1}_z^2$ is nonnegative. As functions, $\sum_z \mathbf{1}_z^2 = \sum_z \mathbf{1}_z = 1$, so by linearity

$$\sum_z p_z = \widetilde{\mathbb{E}}_D \sum_z \mathbf{1}_z^2 = \widetilde{\mathbb{E}}_D 1 = \sum_x D(x) = 1.$$

So the numbers p_z form a probability distribution on $\{-1,1\}^n$. Moreover $\mathbf{1}_z^2 = \mathbf{1}_z$, so $p_z = \widetilde{\mathbb{E}}_D \mathbf{1}_z = D(z)$. Finally, every f satisfies $f(x) = \sum_z f(z)\mathbf{1}_z(x)$, so

$$\widetilde{\mathbb{E}}_D f = \sum_z f(z)\, \widetilde{\mathbb{E}}_D \mathbf{1}_z = \sum_z f(z)\, p_z = \mathbb{E}_{x\sim p} f(x).$$

Thus $\widetilde{\mathbb{E}}_D$ is the expectation with respect to the distribution $p = D$. □

Remark 21.12. Lemma 21.11 is an instance of a duality between probability distributions and nonnegative polynomials. On the hypercube every nonnegative function f is a square, $f = g^2$ with $g = \sqrt{f}$, and g is a polynomial of degree at most n. Hence a function D with total mass 1 has $\widetilde{\mathbb{E}}_D f \geq 0$ for every nonnegative f, which is what being a distribution means, exactly when it has degree $2n$. Results that relate nonnegative polynomials to sums of squares are studied in real algebraic geometry under the name *Positivstellensatz* [Ste74].

21.5 The Cauchy–Schwarz inequality for pseudo-distributions

Many inequalities about expectations remain true for pseudo-expectations of low-degree polynomials, with the same proofs, as long as the proofs only use that squares have nonnegative expectation. The first example is the Cauchy–Schwarz inequality.

Lemma 21.13 (Cauchy–Schwarz inequality)**.** *Let D be a pseudo-distribution on $\{-1,1\}^n$ of degree at least $2t$. Then for all polynomials f and g of degree at most t,*

$$\widetilde{\mathbb{E}}_D fg \leq \sqrt{\widetilde{\mathbb{E}}_D f^2}\, \sqrt{\widetilde{\mathbb{E}}_D g^2}.$$

The square roots are defined because D has degree at least $2t$.

Proof. For every real c, the polynomials $cf + g$ and $cf - g$ have degree at most t, so their squares have nonnegative pseudo-expectation. We use linearity throughout.

Case 1: $\widetilde{\mathbb{E}} g^2 = 0$. Let $c > 0$. Since $(cf+g)^2 - (cf-g)^2 = 4cfg$,

$$\widetilde{\mathbb{E}} fg = \frac{\widetilde{\mathbb{E}}(cf+g)^2 - \widetilde{\mathbb{E}}(cf-g)^2}{4c} \leq \frac{\widetilde{\mathbb{E}}(cf+g)^2}{4c} = \frac{c^2\,\widetilde{\mathbb{E}} f^2 + 2c\,\widetilde{\mathbb{E}} fg}{4c},$$

where the last step uses $\widetilde{\mathbb{E}} g^2 = 0$. Rearranging gives $\widetilde{\mathbb{E}} fg \leq \frac{c}{2}\widetilde{\mathbb{E}} f^2$ for every $c > 0$. Letting $c \to 0$ gives $\widetilde{\mathbb{E}} fg \leq 0$, which is the claim. The case $\widetilde{\mathbb{E}} f^2 = 0$ is the same with f and g exchanged.

Case 2: $\widetilde{\mathbb{E}} f^2 > 0$ *and* $\widetilde{\mathbb{E}} g^2 > 0$. Write $\|f\| := (\widetilde{\mathbb{E}} f^2)^{1/2}$ and $\|g\| := (\widetilde{\mathbb{E}} g^2)^{1/2}$, and let $\bar{f} = f/\|f\|$ and $\bar{g} = g/\|g\|$. Then $\widetilde{\mathbb{E}}\bar{f}^2 = \widetilde{\mathbb{E}}\bar{g}^2 = 1$ and

$$0 \leq \widetilde{\mathbb{E}}(\bar{f} - \bar{g})^2 = \widetilde{\mathbb{E}}\bar{f}^2 + \widetilde{\mathbb{E}}\bar{g}^2 - 2\,\widetilde{\mathbb{E}}\bar{f}\bar{g} = 2 - 2\,\widetilde{\mathbb{E}}\bar{f}\bar{g}.$$

Hence $\widetilde{\mathbb{E}}\bar{f}\bar{g} \leq 1$, that is, $\widetilde{\mathbb{E}} fg \leq \|f\|\,\|g\|$. □

Remark 21.14. The proof suggests a general strategy. Commit to the language of pseudo-distributions: to guess a true statement about them, and often to prove it, first ask what is true of actual distributions, and then check that the proof for distributions only uses the nonnegativity of expectations of low-degree squares.

21.6 Pseudo-moments and the moment matrix

Definition 21.15. The *pseudo-moments of degree t* of a pseudo-distribution D on $\{-1,1\}^n$ are the numbers $\{\widetilde{\mathbb{E}}_D x_S : S \subseteq [n],\ |S| \le t\}$, the pseudo-expectations of the monomials of degree at most t.

We write $\binom{n}{\le t} \coloneqq \sum_{i=0}^{t} \binom{n}{i}$ for the number of sets $S \subseteq [n]$ with $|S| \le t$. Since every function is a linear combination of monomials, the pseudo-expectation of a polynomial of degree at most t is determined by the pseudo-moments of degree t.

Lemma 21.16. *Let $\{\gamma_S : S \subseteq [n],\ |S| \le 2t\}$ be real numbers with $\gamma_\emptyset = 1$. Let M_t be the $\binom{n}{\le t} \times \binom{n}{\le t}$ matrix, with rows and columns indexed by the sets of size at most t, defined by*

$$M_t(S,T) \coloneqq \gamma_{S \triangle T}.$$

Then the γ_S are the pseudo-moments of degree $2t$ of a pseudo-distribution of degree $2t$ on $\{-1,1\}^n$, that is, $\gamma_S = \widetilde{\mathbb{E}}_D x_S$ for $|S| \le 2t$, if and only if $M_t \succeq 0$.

We call M_t the *moment matrix*. When $\gamma_S = \widetilde{\mathbb{E}}_D x_S$, its entries are $M_t(S,T) = \widetilde{\mathbb{E}}_D x_{S\triangle T} = \widetilde{\mathbb{E}}_D x_S x_T$.

Proof. Let D be a pseudo-distribution with $\widetilde{\mathbb{E}}_D x_S = \gamma_S$ for every $|S| \le 2t$. For a vector $v \in \mathbb{R}^{\binom{n}{\le t}}$, indexed like the rows of M_t, we have

$$v^\top M_t v = \sum_{S,T} v_S v_T \, \widetilde{\mathbb{E}}_D x_{S\triangle T} = \sum_{S,T} v_S v_T \, \widetilde{\mathbb{E}}_D x_S x_T = \widetilde{\mathbb{E}}_D \sum_{S,T} v_S v_T \, x_S x_T = \widetilde{\mathbb{E}}_D \Big(\sum_S v_S x_S\Big)^2.$$

Every polynomial f of degree at most t has the form $f = \sum_{|S|\le t} v_S x_S$. Hence $v^\top M_t v \ge 0$ for every v if and only if $\widetilde{\mathbb{E}}_D f^2 \ge 0$ for every f of degree at most t, that is, $M_t \succeq 0$ if and only if D has degree $2t$.

It remains to see that some pseudo-distribution has the prescribed pseudo-moments. Let $D(x) \coloneqq 2^{-n} \sum_{|U|\le 2t} \gamma_U \, x_U$. Since $\sum_x x_U x_S$ equals 2^n if $U = S$ and 0 otherwise, we get $\widetilde{\mathbb{E}}_D x_S = \gamma_S$ for every $|S| \le 2t$, and in particular $\sum_x D(x) = \gamma_\emptyset = 1$. So the γ_S are the pseudo-moments of a pseudo-distribution of degree $2t$ if and only if $M_t \succeq 0$. □

Thus, to check whether $\binom{n}{\le 2t}$ given numbers are the pseudo-moments of a pseudo-distribution of degree $2t$, it suffices to check whether the moment matrix M_t is positive semidefinite.

Corollary 21.17 (Degree-$2t$ sum-of-squares)**.** *Let p be a polynomial of degree at most $2t$ on $\{-1,1\}^n$. Then the problem*

$$\mathrm{SOS}_{2t}(p) \coloneqq \max\{\widetilde{\mathbb{E}}_D\, p \mid D \text{ a pseudo-distribution on } \{-1,1\}^n \text{ of degree } 2t\} \tag{21.3}$$

is a semidefinite program with $\binom{n}{\le 2t}$ variables and one semidefinite constraint of size $\binom{n}{\le t}$. It can be written down in time $n^{O(t)}$.

Proof. Write $p = \sum_{|S|\le 2t} \hat{p}(S)\, x_S$. Then $\widetilde{\mathbb{E}}_D\, p = \sum_S \hat{p}(S)\, \widetilde{\mathbb{E}}_D x_S$ depends only on the pseudo-moments of degree $2t$, linearly. By Lemma 21.16, (21.3) equals the semidefinite program

$$\max\Big\{ \sum_{|S|\le 2t} \hat{p}(S)\, \gamma_S \;\Big|\; \gamma_\emptyset = 1,\ M_t \succeq 0 \Big\}$$

in the variables γ_S, $|S| \le 2t$. The maximum is attained: the diagonal entries of M_t are $\gamma_\emptyset = 1$, so every entry of a feasible M_t lies in $[-1, 1]$, and every γ_S with $|S| \le 2t$ is an entry of M_t. Hence the feasible set is compact. □

As usual, such a semidefinite program is solved up to an arbitrarily small additive error (Lectures 4 and 25).

Since every probability distribution is a pseudo-distribution of every degree (Example 21.9),

$$\mathrm{SOS}_{2t}(p) \ge \max_{D \text{ a distribution on } \{-1,1\}^n} \mathbb{E}_D\, p = \max_{x \in \{-1,1\}^n} p(x).$$

So the degree-$2t$ sum-of-squares program for maximizing $\widetilde{\mathbb{E}}\, p$ is a relaxation of the problem of maximizing p over $\{-1, 1\}^n$.

21.7 Example: Max-Cut

Example 21.18. Let $G = ([n], E)$ be a graph with $m = |E| \ge 1$ edges, and let

$$p_G(x) := \frac{1}{2} - \frac{1}{2m} \sum_{\{i,j\} \in E} x_i x_j = \frac{1}{m} \sum_{\{i,j\} \in E} \frac{1 - x_i x_j}{2}.$$

For $x \in \{-1, 1\}^n$, $p_G(x)$ is the fraction of edges cut by the cut $\{i : x_i = 1\}$, so $\max_x p_G(x) = \mathrm{maxcut}(G)$ in the notation of Lecture 4. The polynomial p_G has degree 2, and its degree-2 sum-of-squares relaxation is

$$\mathrm{SOS}_2(p_G) = \max\Big\{ \frac{1}{2} - \frac{1}{2m} \sum_{\{i,j\} \in E} M_1(\{i\}, \{j\}) \;\Big|\; M_1 \succeq 0 \Big\},$$

where M_1 is the moment matrix of degree 2, indexed by $\emptyset$ and the singletons, with $M_1(\{i\}, \{j\}) = \widetilde{\mathbb{E}}\, x_i x_j$ and $M_1(\{i\}, \{i\}) = \widetilde{\mathbb{E}}\, x_\emptyset = 1$. This is the Goemans–Williamson semidefinite program for Max-Cut (Lecture 4, [GW95]),

$$\max\Big\{ \frac{1}{m} \sum_{\{i,j\} \in E} \frac{1 - X_{ij}}{2} \;\Big|\; X \in \mathbb{R}^{n \times n},\ X \succeq 0,\ X_{ii} = 1 \text{ for all } i \Big\}.$$

Indeed, write $M_1 = \begin{pmatrix} 1 & \mu^\top \\ \mu & X \end{pmatrix}$ with $\mu_i = \widetilde{\mathbb{E}}\, x_i$ and $X_{ij} = \widetilde{\mathbb{E}}\, x_i x_j$. If $M_1 \succeq 0$, then its principal submatrix X is positive semidefinite with unit diagonal, and the two objectives agree. Conversely, for X feasible for the Goemans–Williamson program, the choice $\mu = 0$ gives $M_1 = \begin{pmatrix} 1 & 0 \\ 0 & X \end{pmatrix} \succeq 0$ with the same objective value. Hence the two programs have the same optimal value. Thus the Goemans–Williamson relaxation is simply the degree-2 sum-of-squares relaxation of Max-Cut.

21.8 The sum-of-squares hierarchy

Let p be a polynomial of degree at most $2t_0$. At the other extreme from degree 2, by Lemma 21.11 the pseudo-distributions of degree $2n$ are exactly the probability distributions, so $\mathrm{SOS}_{2n}(p)$ is the exact, integral problem $\max_x p(x)$. A pseudo-distribution of degree $2t + 2$ is also one of degree $2t$,

so raising the degree can only shrink the feasible set. We get a hierarchy of relaxations, shown in Fig. 30:

$$\max_{x\in\{-1,1\}^n} p(x) = \mathrm{SOS}_{2n}(p) \le \cdots \le \mathrm{SOS}_{2t_0+2}(p) \le \mathrm{SOS}_{2t_0}(p). \tag{21.4}$$

Increasing the degree gives tighter relaxations at the cost of larger running times: by Corollary 21.17, degree $2t$ costs time $n^{O(t)}$. In particular, every constant degree gives a polynomial-time relaxation. This is the mechanical method for constructing stronger relaxations that we set out to find.

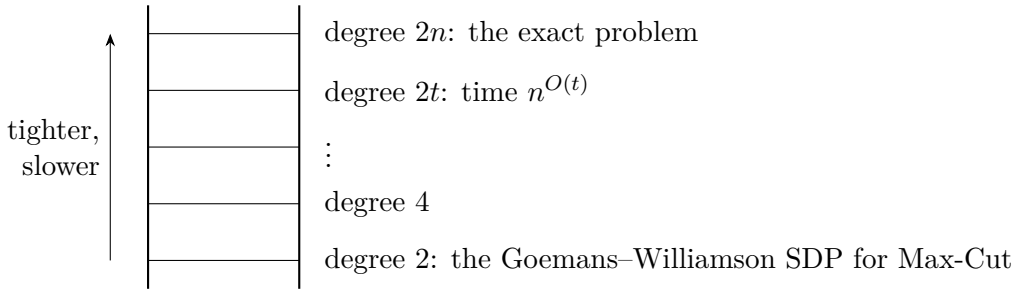


Figure 30: The sum-of-squares hierarchy: each rung is a relaxation at least as tight as the rungs below it.

22 Strengthening Convex Relaxations: SDP Hierarchies II

Lecture of October 20, 2021, by Pravesh K. Kothari. Based on the lecturer's handwritten notes.

This lecture extends the pseudo-distributions of Lecture 21 in three ways and then applies them to Knapsack. We impose polynomial constraints on pseudo-distributions, which keeps optimization over them a semidefinite program; we move from the cube $\{-1,1\}^n$ to $\{0,1\}^n$ by a change of variables; and we show that a pseudo-distribution agrees with an actual distribution on every small set of variables. We then start the proof that the degree-$(2t+2)$ sum-of-squares relaxation of Knapsack has integrality gap at most $1+1/(t-1)$, which approximates OPT within a factor $1+\varepsilon$ in time $n^{O(1/\varepsilon)}$. The idea is that a feasible solution contains fewer than t items of large value, so the pseudo-distribution behaves like an actual distribution on the large items. Conditioning on which large items are chosen then leaves feasible solutions of the linear relaxation of the remaining instance, which Lecture 23 compares with integral solutions through the bound of Lecture 21.

We use the notation of Lecture 21: $x_S = \prod_{i\in S} x_i$, the degree of a function on the cube, pseudo-distributions D, their pseudo-expectations $\widetilde{\mathbb{E}}_D$ and their degree, and the Cauchy–Schwarz inequality for pseudo-distributions.

22.1 Constrained pseudo-distributions

Definition 22.1. Let D be a pseudo-distribution of degree $2t$ on $\{-1,1\}^n$, and let q be a polynomial. We say that D *satisfies the constraint* $q(x)\geq 0$ if

$$\widetilde{\mathbb{E}}_D\, p^2 q \geq 0 \quad \text{for every polynomial } p \text{ with } 2\deg(p)+\deg(q)\leq 2t.$$

If D is a probability distribution supported on $\{x : q(x)\geq 0\}$, then $p^2 q\geq 0$ on its support, so D satisfies the constraint $q\geq 0$. Optimizing over constrained pseudo-distributions is still a semidefinite program.

Corollary 22.2. *Let $p, q_1,\dots,q_m$ be polynomials of degree at most $2t$ on $\{-1,1\}^n$. The problem of finding a pseudo-distribution D of degree $2t$ on $\{-1,1\}^n$ that maximizes $\widetilde{\mathbb{E}}_D\, p$ subject to satisfying the constraints $q_1\geq 0,\dots,q_m\geq 0$ is a semidefinite program of size $\mathrm{poly}(m, n^{O(t)})$, which can be written down in time $\mathrm{poly}(m, n^{O(t)})$.*

Proof. As in Lecture 21, the variables are the pseudo-moments $\gamma_S = \widetilde{\mathbb{E}}_D x_S$, $|S|\leq 2t$, with $\gamma_\emptyset = 1$, the objective is linear in them, and D has degree $2t$ if and only if the moment matrix M_t is positive semidefinite. Fix a constraint $q = q_i = \sum_U \hat{q}(U)\, x_U$ and let $\ell = \lfloor (2t-\deg q)/2\rfloor$. Define the *localizing matrix* M^q, indexed by the sets of size at most ℓ, by

$$M^q(S,T) \coloneqq \widetilde{\mathbb{E}}_D\, x_S x_T q = \sum_U \hat{q}(U)\, \gamma_{S\triangle T\triangle U},$$

which only involves pseudo-moments of degree at most $2\ell+\deg q\leq 2t$. For $p=\sum_{|S|\leq\ell} w_S x_S$ the same computation as for the moment matrix gives $\widetilde{\mathbb{E}}_D\, p^2 q = w^\top M^q w$. Hence D satisfies $q\geq 0$ if and only if $M^q\succeq 0$, a semidefinite constraint that is linear in the variables. The program has $\binom{n}{\leq 2t}$ variables and $m+1$ semidefinite constraints of size at most $\binom{n}{\leq t}$. □

22.2 Pseudo-distributions on $\{0,1\}^n$ and low-degree transformations

Knapsack is naturally stated over $\{0,1\}^n$. On $\{0,1\}^n$ we have $x_i^2 = x_i$, so $x_S x_T = x_{S\cup T}$. Every function $f\colon \{0,1\}^n \to \mathbb{R}$ is a multilinear polynomial, $f(x) = \sum_z f(z) \prod_{i:z_i=1} x_i \prod_{i:z_i=0}(1-x_i)$, and this representation is unique, because the 2^n monomials x_S span the 2^n-dimensional space of functions on $\{0,1\}^n$. The degree of a function, pseudo-distributions, pseudo-expectations, their degree, and constrained pseudo-distributions are defined exactly as on $\{-1,1\}^n$. The proof of the Cauchy–Schwarz inequality in Lecture 21 uses only linearity and the nonnegativity of $\widetilde{\mathbb{E}} h^2$ for $\deg h \le t$, so it holds on $\{0,1\}^n$ as well. Applied to f and to $-f$, it gives $|\widetilde{\mathbb{E}}\, fg| \le (\widetilde{\mathbb{E}} f^2)^{1/2} (\widetilde{\mathbb{E}} g^2)^{1/2}$ for $\deg f, \deg g \le t$ and D of degree at least $2t$.

Proposition 22.3. *Let D be a pseudo-distribution of degree $2t$ on $\{-1,1\}^n$. Define a pseudo-distribution D' on $\{0,1\}^n$ by*

$$\widetilde{\mathbb{E}}_{D'} X_S \coloneqq \widetilde{\mathbb{E}}_D \prod_{i\in S} \frac{1+x_i}{2} \qquad \text{for every } S \subseteq [n],$$

extended linearly to all functions of $X \in \{0,1\}^n$. Then D' is a pseudo-distribution of degree $2t$ on $\{0,1\}^n$.

Proof. The map $\varphi(x) = (1+x)/2$, applied coordinatewise, is a bijection from $\{-1,1\}^n$ to $\{0,1\}^n$. Let $D'(X) \coloneqq D(2X-1)$. Then $\widetilde{\mathbb{E}}_{D'} f = \sum_x f(\varphi(x))\, D(x) = \widetilde{\mathbb{E}}_D (f \circ \varphi)$ for every f; for $f = X_S$ this is the formula above, and since the monomials form a basis, D' is the unique pseudo-distribution with these pseudo-expectations. Its total mass is $\widetilde{\mathbb{E}}_D 1 = 1$. If f has degree at most t in X, then substituting $X_i = (1+x_i)/2$ gives a polynomial $f \circ \varphi$ of degree at most t in x, so $\widetilde{\mathbb{E}}_{D'} f^2 = \widetilde{\mathbb{E}}_D (f\circ\varphi)^2 \ge 0$. □

Remark 22.4. The same holds more generally for low-degree polynomial transformations. If $Y_j = \varphi_j(x)$ with each φ_j a polynomial of degree at most k, then $f \mapsto \widetilde{\mathbb{E}}_D f(\varphi(x))$ is nonnegative on squares of polynomials of degree at most t in Y whenever D has degree at least $2kt$, by the same argument. This property is often called the *basis invariance* of sum-of-squares relaxations. In the setting of Proposition 22.3, D' satisfies a constraint $q \ge 0$ if and only if D satisfies $q \circ \varphi \ge 0$, with the same degrees, so Corollary 22.2 holds on $\{0,1\}^n$ as well.

22.3 Local distributions

Lemma 22.5. *Let D be a pseudo-distribution on $\{-1,1\}^n$ of degree at least $2t$, and let $S \subseteq [n]$ with $|S| \le t$. Then there is a probability distribution μ on $\{-1,1\}^S$ such that $\widetilde{\mathbb{E}}_D x_T = \mathbb{E}_\mu x_T$ for every $T \subseteq S$.*

In words, D locally agrees with actual probability distributions.

Proof. For $z \in \{-1,1\}^S$, let $\mathbf{1}_z(x) = \prod_{i\in S}(1+z_i x_i)/2$, the indicator that $x_i = z_i$ for every $i \in S$. It has degree $|S| \le t$ and $\mathbf{1}_z^2 = \mathbf{1}_z$, so $\mu(z) \coloneqq \widetilde{\mathbb{E}}_D \mathbf{1}_z = \widetilde{\mathbb{E}}_D \mathbf{1}_z^2 \ge 0$. As functions, $\sum_z \mathbf{1}_z = 1$ and $\sum_z z_T \mathbf{1}_z(x) = x_T$ for $T \subseteq S$. By linearity, $\sum_z \mu(z) = 1$ and $\mathbb{E}_\mu x_T = \sum_z \mu(z) z_T = \widetilde{\mathbb{E}}_D x_T$. □

Lemma 22.6. *Let D be a pseudo-distribution on $\{0,1\}^n$ of degree at least 2. Then $\widetilde{\mathbb{E}}_D x_i \ge 0$ for every $i \in [n]$. More generally, if D has degree at least $2t$, then $\widetilde{\mathbb{E}}_D x_T \ge 0$ for every T with $|T| \le t$.*

Proof. On $\{0,1\}^n$ we have $x_i = x_i^2$, so $\widetilde{\mathbb{E}}\, x_i = \widetilde{\mathbb{E}}\, x_i^2 \ge 0$. Likewise $x_T = x_T^2$, and x_T has degree $|T| \le t$. □

22.4 Two facts about pseudo-distributions for Knapsack

From now on, we fix an instance of Knapsack as in Lecture 21: sizes $c_i > 0$, values $v_i \geq 0$ and a capacity C with $c_i \leq C$. A choice of items is a vector $x \in \{0,1\}^n$, and we write

$$q(x) \coloneqq C - \sum_{i=1}^{n} c_i x_i,$$

so that x is feasible if and only if $q(x) \geq 0$. A pseudo-distribution on $\{0,1\}^n$ *satisfies the knapsack constraint* if it satisfies $q \geq 0$. For $T \subseteq [n]$ we write $c(T) \coloneqq \sum_{i \in T} c_i$ and $v(T) \coloneqq \sum_{i \in T} v_i$.

Our goal is to show that pseudo-distributions of degree $O(t)$ satisfying the knapsack constraint approximate Knapsack within a factor $1+1/(t-1)$. In particular, for every $\varepsilon > 0$ this approximates OPT within $1+\varepsilon$ in time $n^{O(1/\varepsilon)}$. The proof needs two basic facts about pseudo-distributions.

Lemma 22.7. *Let $S \subseteq [n]$ and let $k \geq 0$ be an integer such that every feasible $x \in \{0,1\}^n$ satisfies $\sum_{i \in S} x_i \leq k$. Let D be a pseudo-distribution on $\{0,1\}^n$ of degree $2s$ with $s \geq k+2$ that satisfies the knapsack constraint. Then*

$$\widetilde{\mathbb{E}}_D x_T = 0 \quad \text{for every } T \subseteq S \text{ with } k+1 \leq |T| \leq 2s.$$

For example, if $c_1 = c_2 = 1$ and $C = 2.9$, every feasible x has $x_1 + x_2 \leq 2$: since the x_i are integers, the capacity 2.9 can be rounded down. The lemma is clear for an actual distribution supported on feasible points. Restricted to S, such a distribution is supported on vectors with at most k ones, so the event $x_i = 1$ for all $i \in T$ has probability 0 when $|T| \geq k+1$, and $\mathbb{E} x_T$ is exactly the probability of that event. The proof for pseudo-distributions conditions on this event.

Proof. Step 1: $|T| = k+1$. The set T is infeasible, since its indicator vector has $k+1$ ones in S; hence $c(T) > C$. On $\{0,1\}^n$, $x_T = x_T^2$, and $2|T| + \deg q = 2k+3 \leq 2s$, so the knapsack constraint gives $\widetilde{\mathbb{E}}\, x_T q = \widetilde{\mathbb{E}}\, x_T^2 q \geq 0$. Since $x_T x_i = x_T$ for $i \in T$,

$$0 \leq \widetilde{\mathbb{E}}\, x_T q = \big(C - c(T)\big)\, \widetilde{\mathbb{E}}\, x_T - \sum_{i \notin T} c_i\, \widetilde{\mathbb{E}}\, x_{T \cup \{i\}}.$$

By Lemma 22.6, $\widetilde{\mathbb{E}}\, x_{T\cup\{i\}} \geq 0$ because $|T \cup \{i\}| = k+2 \leq s$, and the c_i are positive. Hence $(C - c(T))\, \widetilde{\mathbb{E}}\, x_T \geq 0$. As $C - c(T) < 0$ and $\widetilde{\mathbb{E}}\, x_T \geq 0$ (again by Lemma 22.6), we get $\widetilde{\mathbb{E}}\, x_T = 0$. For an actual distribution, the inequality above says that the expectation of q conditioned on $x_T = 1$ is at most $C - c(T) < 0$, which is impossible for a distribution on feasible points.

Step 2: $k+1 \leq |T| \leq 2s$. Choose $A \subseteq T$ with $|A| = k+1$, so $\widetilde{\mathbb{E}}\, x_A = 0$ by Step 1. If $|T| \leq s$, then $x_T = x_A\, x_{T \setminus A}$ with both factors of degree at most s, and the Cauchy–Schwarz inequality gives

$$|\widetilde{\mathbb{E}}\, x_T| \leq \big(\widetilde{\mathbb{E}}\, x_A^2\big)^{1/2} \big(\widetilde{\mathbb{E}}\, x_{T\setminus A}^2\big)^{1/2} = \big(\widetilde{\mathbb{E}}\, x_A\big)^{1/2} \big(\widetilde{\mathbb{E}}\, x_{T\setminus A}\big)^{1/2} = 0.$$

If $s < |T| \leq 2s$, choose $A' \subseteq T$ with $A \subseteq A'$ and $|A'| = s$. By the previous case $\widetilde{\mathbb{E}}\, x_{A'} = 0$, and the same inequality applied to $x_T = x_{A'}\, x_{T \setminus A'}$, with $|T \setminus A'| \leq s$, gives $\widetilde{\mathbb{E}}\, x_T = 0$. □

Lemma 22.8. *Let $k \geq 0$ and $s \geq k+1$ be integers, and let D be a pseudo-distribution on $\{0,1\}^n$ of degree $2s$ such that $\widetilde{\mathbb{E}}_D x_T = 0$ for every T with $k+1 \leq |T| \leq 2s$. Then there is a probability distribution μ on $\{0,1\}^n$ such that*

$$\mathbb{E}_\mu x_T = \widetilde{\mathbb{E}}_D x_T \quad \text{for every } T \subseteq [n] \text{ with } |T| \leq 2s,$$

and $\mathbb{E}_\mu x_T = 0$ for every T with $|T| \geq k+1$.

That is, up to degree $2s$, D is an actual probability distribution. The restriction to degree $2s$ is necessary, because nothing constrains the pseudo-expectations of monomials of degree larger than $2s$. For example, on $\{0,1\}^3$ the function $D(x) = \mathbf{1}[x=0] - \frac{1}{20}(-1)^{x_1+x_2+x_3}$ has the same pseudo-moments of degree at most 2 as the point mass at 0, so it is a pseudo-distribution of degree 2 satisfying the hypothesis with $k=0$ and $s=1$, but $D(1,1,0) < 0$.

Proof. Since the monomials form a basis of the functions on $\{0,1\}^n$, there is a unique function μ on $\{0,1\}^n$ with $\mathbb{E}_\mu x_T = \widetilde{\mathbb{E}}_D x_T$ for $|T| \le 2s$ and $\mathbb{E}_\mu x_T = 0$ for $|T| > 2s$, where $\mathbb{E}_\mu f \coloneqq \sum_x f(x)\mu(x)$. By hypothesis, $\mathbb{E}_\mu x_T = 0$ for every $|T| \ge k+1$, and $\sum_x \mu(x) = \mathbb{E}_\mu x_\emptyset = 1$. It is enough to prove that $\mathbb{E}_\mu f^2 \ge 0$ for every $f\colon \{0,1\}^n \to \mathbb{R}$: applied to the indicator $f = \mathbf{1}_z$ of a point z, this gives $\mu(z) = \mathbb{E}_\mu \mathbf{1}_z^2 \ge 0$, so μ is a probability distribution.

Write $f = f_{\text{small}} + f_{\text{large}}$, where $f_{\text{small}} = \sum_{|T|\le k} f_T x_T$ and $f_{\text{large}} = \sum_{|T|>k} f_T x_T$. Squaring and taking $\mathbb{E}_\mu$,

$$\mathbb{E}_\mu f^2 = \mathbb{E}_\mu f_{\text{small}}^2 + 2\,\mathbb{E}_\mu f_{\text{small}} f_{\text{large}} + \mathbb{E}_\mu f_{\text{large}}^2.$$

Every monomial of $f_{\text{small}} f_{\text{large}}$ and of f_{large}^2 is $x_S x_T = x_{S\cup T}$ with $|T| \ge k+1$, so the last two terms vanish. The monomials of f_{small}^2 are $x_{S\cup T}$ with $|S\cup T| \le 2k \le 2s$, on which $\mathbb{E}_\mu$ and $\widetilde{\mathbb{E}}_D$ agree. Hence $\mathbb{E}_\mu f^2 = \widetilde{\mathbb{E}}_D f_{\text{small}}^2 \ge 0$, because f_{small} has degree at most $k \le s$. □

22.5 The sum-of-squares relaxation of Knapsack

For $s \ge 1$, the *degree-$2s$ sum-of-squares relaxation* of Knapsack is

$$\mathrm{SOS}_{2s} \coloneqq \max_D \widetilde{\mathbb{E}}_D \sum_{i=1}^n v_i x_i,$$

where the maximum is over the pseudo-distributions D on $\{0,1\}^n$ of degree $2s$ that satisfy the knapsack constraint. By Corollary 22.2 and Remark 22.4, it is a semidefinite program of size $n^{O(s)}$. The point mass at an optimal solution satisfies the knapsack constraint, so $\mathrm{SOS}_{2s} \ge \mathrm{OPT}$.

Theorem 22.9. *Let $t \ge 2$. For every pseudo-distribution D on $\{0,1\}^n$ of degree at least $2t+2$ that satisfies the knapsack constraint,*

$$\widetilde{\mathbb{E}}_D \sum_{i=1}^n v_i x_i \le \Big(1 + \frac{1}{t-1}\Big)\mathrm{OPT}.$$

Thus the integrality gap of the degree-$(2t+2)$ sum-of-squares relaxation, a semidefinite program of size $n^{O(t)}$, is at most $1 + 1/(t-1)$.

This bound, and the conditioning argument that proves it, are due to Karlin, Mathieu and Nguyen [KMN11].

Remark 22.10. The bound of Theorem 22.9 already holds for pseudo-distributions of degree $2t$. To see this, define the large items in the proof below by the strict inequality $v_i > \mathrm{OPT}/(t-1)$. A feasible solution then contains at most $t-2$ large items, since $t-1$ of them have total value larger than OPT, and the proof goes through with every threshold lowered by one: Lemma 22.7 is applied with $k = t-2$ and $s = t$, and Red deletes the monomials with at least $t-1$ large items. Every item outside S still has value at most $\mathrm{OPT}/(t-1)$, which is all that the last step (Lecture 23) needs. Karlin, Mathieu and Nguyen state the result in this form, for the t-th level of the Lasserre hierarchy, whose moment and localizing matrices are those of degree $2t$ here.

Corollary 22.11. *For every $\varepsilon \in (0,1]$, let $t = 1 + \lceil 1/\varepsilon \rceil$. Then SOS_{2t+2} is a semidefinite program of size $n^{O(1/\varepsilon)}$, and $\mathrm{OPT} \le \mathrm{SOS}_{2t+2} \le (1+\varepsilon)\,\mathrm{OPT}$.*

Proof. The size is $n^{O(t)} = n^{O(1/\varepsilon)}$, and $1/(t-1) = 1/\lceil 1/\varepsilon \rceil \le \varepsilon$, so the bounds follow from Theorem 22.9. Solving the program up to an additive error $\delta > 0$ gives a number in $[\mathrm{OPT} - \delta, (1+\varepsilon)\mathrm{OPT} + \delta]$. □

The proof of Theorem 22.9 takes the rest of this lecture and is completed in Lecture 23. A pseudo-distribution of degree at least $2t+2$ that satisfies the knapsack constraint is in particular a pseudo-distribution of degree $2t+2$ that satisfies it, so we assume that D has degree exactly $2t+2$. If $\mathrm{OPT} = 0$, then every $v_i = 0$, because each single item is feasible, and there is nothing to prove; so we assume $\mathrm{OPT} > 0$. Let

$$S \coloneqq \left\{ i \in [n] \;\middle|\; v_i \ge \frac{\mathrm{OPT}}{t-1} \right\}$$

be the set of *large* items. Every feasible x satisfies $\sum_{i \in S} x_i \le t-1$: a feasible set with at least t large items would collect value at least $t\,\mathrm{OPT}/(t-1) > \mathrm{OPT}$. By Lemma 22.7 with $k = t-1$ and $s = t+1 = k+2$,

$$\widetilde{\mathbb{E}}_D x_T = 0 \quad \text{for every } T \subseteq S \text{ with } t \le |T| \le 2t+2. \tag{22.1}$$

Let D_S be the restriction of D to the variables in S, the function on $\{0,1\}^S$ given by $D_S(z) = \sum_{x : x|_S = z} D(x)$, where $x|_S$ is the restriction of x to the coordinates in S. For every f that depends only on the coordinates in S we have $\widetilde{\mathbb{E}}_{D_S} f = \widetilde{\mathbb{E}}_D f$, so D_S is a pseudo-distribution of degree $2t+2$ on $\{0,1\}^S$. By (22.1) and Lemma 22.8, applied to D_S with $k = t-1$ and $s = t+1$, there is a probability distribution μ on $\{0,1\}^S$ with

$$\mathbb{E}_\mu x_T = \widetilde{\mathbb{E}}_D x_T \text{ for } T \subseteq S,\ |T| \le 2t+2, \quad \text{and} \quad \mathbb{E}_\mu x_T = 0 \text{ for } T \subseteq S,\ |T| \ge t. \tag{22.2}$$

Thus D, restricted to S, is a global distribution up to degree $2t+2$. Note that $|S|$ can be much larger than $2t+2$, and (22.2) still holds; so this is not the plain local property of Lemma 22.5.

For $U \subseteq S$, let $p_U \coloneqq \Pr_{z \sim \mu}[\{i \in S : z_i = 1\} = U]$ be the probability under μ that the large items chosen are exactly those in U, and let

$$f_{S,U}(x) \coloneqq \prod_{i \in U} x_i \prod_{i \in S \setminus U} (1 - x_i),$$

which is 1 if $x_i = 1$ for every $i \in U$ and $x_i = 0$ for every $i \in S \setminus U$, and 0 otherwise. Expanding $\prod_{i \in S}(x_i + (1 - x_i)) = 1$ gives

$$\sum_{U \subseteq S} f_{S,U} = 1. \tag{22.3}$$

22.6 Reduced polynomials and conditioning on the large items

The polynomial $f_{S,U}$ has degree $|S|$, which can exceed the degree of D. We replace it by a low-degree polynomial with the same behavior under $\widetilde{\mathbb{E}}_D$. First we extend (22.1) to monomials with one more variable outside S.

Lemma 22.12. *For every $T \subseteq S$ with $t \le |T| \le 2t-1$ and every $i \notin S$, we have $\widetilde{\mathbb{E}}_D x_T x_i = 0$.*

Proof. Let $T_1 \subseteq T$ with $|T_1| = t$. Then $x_T x_i = x_{T_1} \cdot x_{T \setminus T_1} x_i$, where the two factors have degrees t and $|T| - t + 1 \le t$, both at most $t+1$. By the Cauchy–Schwarz inequality and (22.1),

$$|\widetilde{\mathbb{E}}\, x_T x_i| \le (\widetilde{\mathbb{E}}\, x_{T_1}^2)^{1/2} (\widetilde{\mathbb{E}}\, (x_{T \setminus T_1} x_i)^2)^{1/2} = (\widetilde{\mathbb{E}}\, x_{T_1})^{1/2} (\widetilde{\mathbb{E}}\, (x_{T \setminus T_1} x_i)^2)^{1/2} = 0. \qquad \square$$

Definition 22.13. For a polynomial $f = \sum_{T \subseteq [n]} f_T\, x_T$ on $\{0,1\}^n$, written in its multilinear representation, the *reduction* of f is

$$\mathrm{Red}(f) \coloneqq \sum_{T : |T \cap S| \le t-1} f_T\, x_T.$$

That is, Red deletes every monomial that contains at least t large items.

The map Red is linear. The monomials of $f_{S,U}$ are the x_T with $U \subseteq T \subseteq S$, so $\mathrm{Red}(f_{S,U})$ has degree at most $t-1$, and $\mathrm{Red}(f_{S,U}) = 0$ when $|U| \ge t$. The monomials that Red deletes from $f_{S,U}$ have pseudo-expectation 0 by (22.1), even after multiplication by one more variable (Lemma 22.12). This suggests that $\mathrm{Red}(f_{S,U})$ can replace $f_{S,U}$, and the following lemmas make this precise.

Lemma 22.14. *$\sum_{U \subseteq S} \mathrm{Red}(f_{S,U}) = 1$, and $\sum_{U \subseteq S} \mathrm{Red}(f_{S,U})\, x_i = x_i$ for every $i \in [n]$.*

Proof. By linearity and (22.3), $\sum_U \mathrm{Red}(f_{S,U}) = \mathrm{Red}(\sum_U f_{S,U}) = \mathrm{Red}(1) = 1$. Multiplying this identity by x_i gives the second one. $\square$

For $U \subseteq S$ with $\widetilde{\mathbb{E}}_D \mathrm{Red}(f_{S,U}) > 0$ and $i \in [n]$, define

$$y_i^U \coloneqq \frac{\widetilde{\mathbb{E}}_D\, x_i\, \mathrm{Red}(f_{S,U})}{\widetilde{\mathbb{E}}_D\, \mathrm{Red}(f_{S,U})}.$$

Intuitively, y_i^U is the conditional expectation of x_i given $f_{S,U}(x) = 1$, that is, given that the large items chosen are exactly those in U.

Lemma 22.15. *For every $U \subseteq S$ and every $i \in [n]$,*

$$\widetilde{\mathbb{E}}_D\, \mathrm{Red}(f_{S,U})^2\, x_i = \widetilde{\mathbb{E}}_D\, \mathrm{Red}(f_{S,U})\, x_i \quad \text{and} \quad \widetilde{\mathbb{E}}_D\, \mathrm{Red}(f_{S,U})^2 = \widetilde{\mathbb{E}}_D\, \mathrm{Red}(f_{S,U}).$$

Proof. Write $f = f_{S,U}$, $g = \mathrm{Red}(f)$ and $h = f - g$. Since f takes only the values 0 and 1, $f^2 = f$. The idea is that $\mathrm{Red}(f)^2$ equals $\mathrm{Red}(f^2) = \mathrm{Red}(f)$ up to terms of high degree in the large items, whose pseudo-expectations vanish. Precisely, $f = f^2 = g^2 + 2gh + h^2$. Every monomial of gh and of h^2 contains a monomial of h, hence at least t large items, so $g = \mathrm{Red}(f) = \mathrm{Red}(g^2)$. The monomials of g^2 are $x_{T \cup T'}$ with $T, T' \subseteq S$ of size at most $t-1$. Hence $H \coloneqq g^2 - g = g^2 - \mathrm{Red}(g^2)$ is a linear combination of monomials x_V with $V \subseteq S$ and $t \le |V| \le 2t-2$. For each such V: $\widetilde{\mathbb{E}}\, x_V = 0$ by (22.1); if $i \notin S$, then $\widetilde{\mathbb{E}}\, x_V x_i = 0$ by Lemma 22.12; and if $i \in S$, then $x_V x_i = x_{V \cup \{i\}}$ with $V \cup \{i\} \subseteq S$ and $t \le |V \cup \{i\}| \le 2t-1$, so $\widetilde{\mathbb{E}}\, x_V x_i = 0$ by (22.1). Therefore $\widetilde{\mathbb{E}}\, H = \widetilde{\mathbb{E}}\, H x_i = 0$, which gives both identities. $\square$

Lemma 22.16. *For every $U \subseteq S$, $\widetilde{\mathbb{E}}_D\, \mathrm{Red}(f_{S,U}) = p_U$, and for every $i \in S$,*

$$\widetilde{\mathbb{E}}_D\, x_i\, \mathrm{Red}(f_{S,U}) = \begin{cases} p_U & \text{if } i \in U, \\ 0 & \text{if } i \in S \setminus U. \end{cases}$$

In particular, if $\widetilde{\mathbb{E}}_D\, \mathrm{Red}(f_{S,U}) > 0$, then $y_i^U = 1$ for $i \in U$ and $y_i^U = 0$ for $i \in S \setminus U$; and

$$\widetilde{\mathbb{E}}_D \sum_{i \in S} c_i x_i\, \mathrm{Red}(f_{S,U}) = \mathbb{E}_\mu \sum_{i \in S} c_i x_i f_{S,U} = c(U)\, \widetilde{\mathbb{E}}_D\, \mathrm{Red}(f_{S,U}). \tag{22.4}$$

Proof. Let $g = \mathrm{Red}(f_{S,U})$. It depends only on the coordinates in S, and its monomials x_T have $|T| \le t-1$, so by (22.2), $\widetilde{\mathbb{E}}_D\, g = \mathbb{E}_\mu\, g$. The difference $f_{S,U} - g$ consists of monomials x_T with $T \subseteq S$ and $|T| \ge t$, which have $\mathbb{E}_\mu x_T = 0$ by (22.2). Hence $\widetilde{\mathbb{E}}_D\, g = \mathbb{E}_\mu f_{S,U} = p_U$. Similarly, for $i \in S$ the monomials of $x_i g$ have degree at most t, so $\widetilde{\mathbb{E}}_D\, x_i g = \mathbb{E}_\mu x_i g = \mathbb{E}_\mu x_i f_{S,U}$. As functions on $\{0,1\}^S$, $x_i f_{S,U} = f_{S,U}$ if $i \in U$ and $x_i f_{S,U} = 0$ if $i \in S \setminus U$, which gives the formula and (22.4). □

In particular, the numbers $\widetilde{\mathbb{E}}_D\, \mathrm{Red}(f_{S,U}) = p_U$ are nonnegative and sum to 1, as Lemma 22.14 also shows.

Lemma 22.17. *Let $U \subseteq S$.*

1) If $\widetilde{\mathbb{E}}_D\, \mathrm{Red}(f_{S,U}) = 0$, then $\widetilde{\mathbb{E}}_D\, x_i\, \mathrm{Red}(f_{S,U}) = 0$ for every $i \in [n]$.

2) If $\widetilde{\mathbb{E}}_D\, \mathrm{Red}(f_{S,U}) > 0$, then $0 \le y_i^U \le 1$ for every $i \in [n]$, and

$$\sum_{i \notin S} c_i\, y_i^U \le C - c(U).$$

Proof. Write $g = \mathrm{Red}(f_{S,U})$, a polynomial of degree at most $t-1$.

(1) By the Cauchy–Schwarz inequality and Lemma 22.15,

$$|\widetilde{\mathbb{E}}\, g\, x_i| \le (\widetilde{\mathbb{E}}\, g^2)^{1/2} (\widetilde{\mathbb{E}}\, x_i^2)^{1/2} = (\widetilde{\mathbb{E}}\, g)^{1/2} (\widetilde{\mathbb{E}}\, x_i)^{1/2} = 0.$$

(2) By Lemma 22.15, $y_i^U = \widetilde{\mathbb{E}}\, g\, x_i / \widetilde{\mathbb{E}}\, g = \widetilde{\mathbb{E}}\, g^2 x_i / \widetilde{\mathbb{E}}\, g^2$. Since $x_i = x_i^2$ on $\{0,1\}^n$,

$$\widetilde{\mathbb{E}}\, g^2(1 - x_i) = \widetilde{\mathbb{E}}\, g^2(1 - 2x_i + x_i^2) = \widetilde{\mathbb{E}}\, \big(g(1-x_i)\big)^2 \ge 0,$$

because $g(1-x_i)$ has degree at most t. Thus $\widetilde{\mathbb{E}}\, g^2 x_i \le \widetilde{\mathbb{E}}\, g^2$, that is, $y_i^U \le 1$. Next, $\widetilde{\mathbb{E}}\, g = \widetilde{\mathbb{E}}\, g^2 > 0$ and $\widetilde{\mathbb{E}}\, g\, x_i = \widetilde{\mathbb{E}}\, g^2 x_i^2 = \widetilde{\mathbb{E}}\, (g x_i)^2 \ge 0$, so $y_i^U \ge 0$.

For the capacity bound, g^2 is the square of a polynomial of degree at most $t-1$, and $2(t-1)+1 \le 2t+2$. Since D satisfies the knapsack constraint, $\widetilde{\mathbb{E}}\, g^2 (C - \sum_{i \in [n]} c_i x_i) \ge 0$, which by Lemma 22.15 reads

$$C\, \widetilde{\mathbb{E}}\, g \ge \widetilde{\mathbb{E}} \sum_{i \in [n]} c_i x_i\, g. \tag{22.5}$$

Subtracting (22.4) from (22.5) gives

$$\widetilde{\mathbb{E}} \sum_{i \notin S} c_i x_i\, g \le C\, \widetilde{\mathbb{E}}\, g - c(U)\, \widetilde{\mathbb{E}}\, g.$$

Dividing by $\widetilde{\mathbb{E}}\, g > 0$ gives $\sum_{i \notin S} c_i\, y_i^U \le C - c(U)$. □

So for each U of positive weight, the vector $(y_i^U)_{i \notin S}$ is a feasible solution of the linear relaxation of Knapsack restricted to the items outside S, with capacity $C - c(U)$. Lecture 23 uses this to complete the proof of Theorem 22.9.

23 Strengthening Convex Relaxations: SDP Hierarchies III

Lecture of October 22, 2021, by Pravesh K. Kothari. Based on the lecturer's handwritten notes.

This lecture completes the proof, begun in Lecture 22, that the degree-$(2t+2)$ sum-of-squares relaxation of Knapsack has integrality gap at most $1+1/(t-1)$. We split the pseudo-expectation of the objective according to which large items are chosen. In each part, the large items contribute exactly their value, and the conditioned values of the remaining items form a feasible solution of the linear relaxation of a smaller Knapsack instance. The bound $\mathrm{LP} \le \mathrm{OPT} + \max_i v_i$ of Lecture 21 compares that solution with an integral one, and averaging over the parts gives the theorem. With it, the optimal value of Knapsack can be approximated within a factor $1+\varepsilon$ in time $n^{O(1/\varepsilon)}$.

23.1 The setting

We complete the proof of the following theorem of Lecture 22.

Theorem (Lecture 22)**.** *Let $t \ge 2$, and fix a Knapsack instance with sizes $c_i > 0$, values $v_i \ge 0$ and capacity C, where $c_i \le C$ for every $i \in [n]$. For every pseudo-distribution D on $\{0,1\}^n$ of degree at least $2t+2$ that satisfies the knapsack constraint $C - \sum_i c_i x_i \ge 0$,*

$$\widetilde{\mathbb{E}}_D \sum_{i=1}^{n} v_i x_i \le \Big(1 + \frac{1}{t-1}\Big)\mathrm{OPT}.$$

For $I \subseteq [n]$ and $C' \ge 0$, let $\mathrm{OPT}(C', I)$ be the largest value $v(J) = \sum_{i \in J} v_i$ of a set $J \subseteq I$ with $c(J) = \sum_{i \in J} c_i \le C'$, and let $\mathrm{LP}(C', I)$ be the optimal value of the linear relaxation of this instance, as in Lecture 21. Then $\mathrm{OPT} = \mathrm{OPT}(C, [n])$.

As in Lecture 22, we may assume that D has degree exactly $2t+2$ and that $\mathrm{OPT} > 0$. The set of large items is $S = \{i : v_i \ge \mathrm{OPT}/(t-1)\}$, so every item $i \notin S$ has $v_i < \mathrm{OPT}/(t-1)$. For $U \subseteq S$, $f_{S,U}$ is the indicator that the large items chosen are exactly those in U, and $\mathrm{Red}(f_{S,U})$ is obtained from it by deleting the monomials with at least t large items. We write

$$w_U := \widetilde{\mathbb{E}}_D \,\mathrm{Red}(f_{S,U}), \qquad y_i^U := \frac{\widetilde{\mathbb{E}}_D \, x_i \,\mathrm{Red}(f_{S,U})}{w_U} \quad \text{when } w_U > 0.$$

We use the following facts, proved in Lecture 22, and the last one, proved in Lecture 21.

1) For every $i \in [n]$, $\sum_{U \subseteq S} \mathrm{Red}(f_{S,U})\, x_i = x_i$.
2) $w_U \ge 0$ for every $U \subseteq S$, and $\sum_{U \subseteq S} w_U = 1$.
3) For every $U \subseteq S$ and $i \in S$, $\widetilde{\mathbb{E}}_D \, x_i \,\mathrm{Red}(f_{S,U})$ equals w_U if $i \in U$ and 0 otherwise.
4) If $w_U = 0$, then $\widetilde{\mathbb{E}}_D \, x_i \,\mathrm{Red}(f_{S,U}) = 0$ for every $i \in [n]$.
5) If $w_U > 0$, then $0 \le y_i^U \le 1$ for every i, and $\sum_{i \notin S} c_i \, y_i^U \le C - c(U)$.
6) For all sizes $c_i > 0$, values $v_i \ge 0$, every $C' \ge 0$ and every $I \subseteq [n]$, $\mathrm{LP}(C', I) \le \mathrm{OPT}(C', I) + \max_{i \in I} v_i$; this does not require $c_i \le C'$.

23.2 Completing the knapsack analysis

Proof of the theorem. By fact (1) and linearity,

$$\widetilde{\mathbb{E}} \sum_i v_i x_i = \widetilde{\mathbb{E}} \sum_{U \subseteq S} \operatorname{Red}(f_{S,U}) \sum_i v_i x_i = \sum_{U \subseteq S} \Big[\sum_{i \in S} v_i \, \widetilde{\mathbb{E}} \operatorname{Red}(f_{S,U}) \, x_i + \sum_{i \notin S} v_i \, \widetilde{\mathbb{E}} \operatorname{Red}(f_{S,U}) \, x_i \Big].$$

By fact (4), the terms with $w_U = 0$ vanish. For $w_U > 0$, fact (3) gives $\sum_{i \in S} v_i \, \widetilde{\mathbb{E}} \operatorname{Red}(f_{S,U}) \, x_i = v(U) \, w_U$, and the definition of y^U gives $\sum_{i \notin S} v_i \, \widetilde{\mathbb{E}} \operatorname{Red}(f_{S,U}) \, x_i = w_U \sum_{i \notin S} v_i \, y_i^U$. Hence

$$\widetilde{\mathbb{E}} \sum_i v_i x_i = \sum_{U \subseteq S \colon w_U > 0} w_U \Big(v(U) + \sum_{i \notin S} v_i \, y_i^U \Big). \tag{23.1}$$

Fix $U \subseteq S$ with $w_U > 0$. By fact (5), $0 \le \sum_{i \notin S} c_i \, y_i^U \le C - c(U)$, so $c(U) \le C$. Also by fact (5), the vector $(y_i^U)_{i \notin S}$ is a feasible solution of the linear relaxation for the items outside S with capacity $C - c(U)$. By fact (6), and since every item outside S has value less than $\mathrm{OPT}/(t-1)$,

$$\sum_{i \notin S} v_i \, y_i^U \le \mathrm{LP}(C - c(U), [n] \setminus S) \le \mathrm{OPT}(C - c(U), [n] \setminus S) + \frac{\mathrm{OPT}}{t-1}$$

(if $S = [n]$, the left-hand side is 0). The set U together with an optimal solution of the instance on $[n] \setminus S$ with capacity $C - c(U)$ has total size at most C, so it is feasible, and

$$v(U) + \mathrm{OPT}(C - c(U), [n] \setminus S) \le \mathrm{OPT}.$$

Hence every term of (23.1) with $w_U > 0$ satisfies

$$v(U) + \sum_{i \notin S} v_i \, y_i^U \le \max_{U' \subseteq S \colon w_{U'} > 0} \Big(v(U') + \mathrm{OPT}(C - c(U'), [n] \setminus S) \Big) + \frac{\mathrm{OPT}}{t-1} \le \Big(1 + \frac{1}{t-1} \Big) \mathrm{OPT}.$$

Since the weights w_U are nonnegative and sum to 1 by fact (2), (23.1) gives

$$\widetilde{\mathbb{E}} \sum_i v_i x_i \le \Big(1 + \frac{1}{t-1} \Big) \mathrm{OPT} \sum_{U \subseteq S} w_U = \Big(1 + \frac{1}{t-1} \Big) \mathrm{OPT}. \qquad \square$$

As shown in Lecture 22, choosing $t = 1 + \lceil 1/\varepsilon \rceil$ turns the theorem into a semidefinite program of size $n^{O(1/\varepsilon)}$ whose value lies between OPT and $(1+\varepsilon)$OPT. This is the second proof, announced in Lecture 21, that the value of Knapsack can be approximated within any factor $1 + \varepsilon$ in time $n^{O(1/\varepsilon)}$.

24 Gaussian Elimination, Linear Systems, and Related Topics

Lecture of October 25, 2021, by Anupam Gupta. Based on the handwritten notes of Pravesh K. Kothari and Anupam Gupta.

The algorithms of this course use a few numerical primitives as black boxes: eigenvalue computations, linear and semidefinite programming, and the solution of linear systems. Their inputs or outputs are often real numbers, while a computer works with finitely many bits. This lecture asks what it takes to implement these primitives on a digital computer. After a warning example (the sum of square roots problem), we show that linear systems and linear programs with rational data have solutions of polynomial bit size, by Cramer's rule, whereas eigenvalues can be irrational and optimal solutions of semidefinite programs can need exponentially many bits. We then analyze Gaussian elimination: it uses $O(n^3)$ arithmetic operations and, by a lemma of Edmonds [Edm67], every nonzero entry of the matrices it produces is a ratio of two determinants of submatrices of the input, so its running time is polynomial in the bit size of the input.

24.1 Computing with real numbers

The algorithms of the earlier lectures rest on a few basic techniques: eigenvalues of matrices (Cheeger's inequality, Lecture 8), semidefinite programs (max-cut, Lecture 4), linear programs (set cover, Lecture 2), the Cholesky decomposition of a positive semidefinite matrix, and the solution of systems of linear equations (satisfiable instances of unique games, Lecture 16). In several of these applications the inputs, the outputs, or both were real numbers. Our computers, and our models of them, are digital. Every algorithm should eventually run on a word RAM machine, which has

1) a random access memory,

2) a word size $w = \Theta(\log n)$ bits, where n is the input size, and

3) arithmetic operations on w-bit integers in $O(1)$ time.

Such a machine cannot store a real number. The best we can do is to approximate real numbers by rationals and control the error. Making this work can be easy, nontrivial or impossible, depending on the problem.

It is often assumed instead that basic operations on real numbers take $O(1)$ time, a model called the *real RAM*. One might hope that such operations can be implemented on Turing machines or digital computers with a polynomial slowdown, which would make the model a harmless convenience. The question is which operations count as basic. Addition and multiplication certainly do. Division, square roots, rounding to integers and the modulus operation are less clear: some combinations of these operations appear to make the model unreasonably powerful, able to solve PSPACE-complete problems in polynomial time. Even if we keep only the arithmetic operations, the model has problems, as the following example shows.

Definition 24.1. In the *sum of square roots problem*, we are given nonnegative integers $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$, and we must decide whether $\sum_{i=1}^n \sqrt{a_i} < \sum_{i=1}^n \sqrt{b_i}$.

Example 24.2. Given two polygons in the plane whose vertices have integer coordinates, we want to decide which one has the larger perimeter. Let the first polygon have vertices

$(x_1, y_1), \ldots, (x_n, y_n)$ and the second have vertices $(u_1, v_1), \ldots, (u_n, v_n)$, in order, and set $(x_0, y_0) \coloneqq (x_n, y_n)$ and $(u_0, v_0) \coloneqq (u_n, v_n)$. The first polygon has the smaller perimeter if and only if

$$\sum_{i=1}^{n} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} < \sum_{i=1}^{n} \sqrt{(u_i - u_{i-1})^2 + (v_i - v_{i-1})^2},$$

which is an instance of the sum of square roots problem. The same comparisons arise in the Euclidean traveling salesman problem and in Euclidean shortest path problems.

On a real RAM, the sum of square roots problem takes $O(n)$ time. Its complexity in the bit model is open. There is a polynomial-space algorithm, and the problem is known to lie in the fourth level of the counting hierarchy, $\mathrm{P}^{\mathrm{P}^{\mathrm{P}^{\mathrm{P}}}}$ (Allender, Bürgisser, Kjeldgaard-Pedersen and Miltersen [ABKPM09]), but it is not even known to lie in NP. Blömer [Blö91] gave a polynomial-time randomized algorithm that decides whether two such sums are equal, but it cannot decide which sum is larger. The moral is that we must be careful about numerical issues; we cannot expect them to work themselves out.

24.2 Rational inputs and the size of solutions

From now on, all input numbers are rationals, and a rational is described by a pair of integers. Throughout this lecture, log denotes the logarithm to base 2. For a real number $r \geq 0$, we say that an integer p has *at most r bits* if $|p| < 2^r$ (the sign is stored separately). A rational number has at most r bits if it can be written as p/q with integers p and $q \geq 1$ that both have at most r bits.

Lemma 24.3. *Let p_1/q_1 and p_2/q_2 be rationals whose numerators and denominators are integers with at most β bits. Then their sum, their product and, if $p_2 \neq 0$, their quotient are rationals with at most $2\beta + 2$ bits, and they can be computed in time polynomial in β.*

Proof. We have

$$\frac{p_1}{q_1} + \frac{p_2}{q_2} = \frac{p_1 q_2 + p_2 q_1}{q_1 q_2}, \qquad \frac{p_1}{q_1} \cdot \frac{p_2}{q_2} = \frac{p_1 p_2}{q_1 q_2}, \qquad \frac{p_1/q_1}{p_2/q_2} = \frac{p_1 q_2}{q_1 p_2},$$

where in the quotient we move the sign of p_2 to the numerator. Every product of two integers with at most β bits has absolute value less than $2^{2\beta}$, and $|p_1 q_2 + p_2 q_1| < 2^{2\beta+1}$. The integer multiplications and additions take time polynomial in β. □

We now look at the primitives one by one.

Linear systems. Given $A \in \mathbb{Q}^{n \times n}$ and $b \in \mathbb{Q}^n$, we want to find $x \in \mathbb{Q}^n$ with $Ax = b$, if one exists. The first question is whether there is always a solution that can be written down with polynomially many bits.

Fact 24.4 (Cramer's rule)**.** *Let $A \in \mathbb{Q}^{n \times n}$ be nonsingular and $b \in \mathbb{Q}^n$. For $i \in [n]$, let A_i be the matrix obtained from A by replacing its i-th column with b. Then the unique solution of $Ax = b$ is given by*

$$x_i = \frac{\det(A_i)}{\det(A)} \qquad (i \in [n]).$$

Proposition 24.5. *Let $A \in \mathbb{Z}^{n\times n}$ have entries that are integers with at most β bits. Then $\det(A)$ is an integer with at most $n\log n + n\beta$ bits.*

Proof. By the definition of the determinant as a sum over the permutations σ of $[n]$,

$$\det(A) = \sum_{\sigma} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} A(i,\sigma(i)).$$

Each of the $n!$ products has absolute value less than $2^{\beta n}$, so

$$|\det(A)| < n!\, 2^{\beta n} \le 2^{n\log n + n\beta},$$

since $n! \le n^n$. It is an integer because the entries of A are. □

Corollary 24.6. *Let $A \in \mathbb{Z}^{n\times n}$ be nonsingular and $b \in \mathbb{Z}^n$, with all entries integers of at most β bits. Then each coordinate of the solution of $Ax = b$ is a ratio of two integers with at most $n\log n + n\beta$ bits. In particular the solution can be written with $O(n^2\log n + n^2\beta)$ bits, which is polynomial in the input size.*

Proof. Apply Proposition 24.5 to A and to each A_i in Fact 24.4; the entries of A_i are entries of A or of b. □

Remark 24.7. If A is singular and $Ax = b$ has a solution, there is still a solution of the same size. For $I, J \subseteq [n]$, write $A[I,J]$ for the submatrix of A with rows in I and columns in J. Let $r = \operatorname{rank} A$, and choose sets $I, J \subseteq [n]$ of size r such that $A[I,J]$ is nonsingular. Then the rows of A indexed by I form a basis of the row space of A, and since the system is consistent, every equation of $Ax = b$ is a linear combination of the equations indexed by I. Hence setting $x_j = 0$ for $j \notin J$ and solving $A[I,J]\,x_J = b_I$ by Cramer's rule gives a solution of $Ax = b$, and Corollary 24.6 applies to $A[I,J]$.

If the entries of A are rationals rather than integers, we clear denominators first.

Lemma 24.8. *Let $r_1, \dots, r_N$ be rationals with at most β bits, and let D be the product of their denominators. Then D is a positive integer with at most $N\beta$ bits, and each Dr_k is an integer with at most $N\beta$ bits.*

Proof. Write $r_k = p_k/q_k$ with $|p_k|, q_k < 2^{\beta}$. Then $D = \prod_k q_k < 2^{N\beta}$, and $Dr_k = p_k \prod_{l\neq k} q_l$ is an integer of absolute value less than $2^{\beta}\cdot 2^{(N-1)\beta}$. □

For the n^2 entries of A, multiplying all entries by D turns rationals with at most β bits into integers with at most $n^2\beta$ bits, a polynomial blowup of the bit complexity. For the system $Ax = b$, we multiply both sides by the product of the denominators of all $n^2 + n$ entries of A and b; this does not change the solutions, and gives an integer system with entries of at most $(n^2+n)\beta$ bits.

Eigenvalues. Even integer matrices can have irrational eigenvalues.

Example 24.9. The matrix $A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ has characteristic polynomial $\lambda^2 - 2$, so its eigenvalues are $\pm\sqrt{2}$. They cannot be written down exactly as rationals.

So for eigenvalues we must resort to approximation. An approximation within ε is short: let $t = \lceil \log(1/\varepsilon) \rceil$; a real number λ is within $2^{-t} \le \varepsilon$ of the rational $p/2^t$, where p is an integer nearest to $2^t\lambda$. Its numerator has absolute value at most $2^t|\lambda|+1$. If λ is an eigenvalue of a matrix in $\mathbb{Z}^{n\times n}$ whose entries have at most β bits, then $|\lambda| < n2^\beta$ (the absolute value of an eigenvalue is at most the largest absolute row sum), so the approximation has $O(\log(1/\varepsilon) + \beta + \log n)$ bits.

Linear programs. Consider a linear program of the form

$$\max\ c^\top x \quad \text{subject to} \quad Ax \le b, \quad 0 \le x_i \le R \text{ for all } i \in [n], \tag{24.1}$$

where $A \in \mathbb{Q}^{m\times n}$, $b \in \mathbb{Q}^m$, $c \in \mathbb{Q}^n$, and R is a positive integer with $\mathrm{poly}(n)$ bits.

Fact 24.10. *If the linear program (24.1) is feasible, then it has an optimal solution whose coordinates are rationals with* $\mathrm{poly}(n, m, \beta)$ *bits, where β bounds the number of bits of the entries of A and b.*

The constraints $0 \le x_i \le R$ form a *bounding box*. Not every linear program comes with one, but in our applications $R = 1$ often works.

A linear program can also be written in *equational form*

$$\max\{c^\top x : x \in K\}, \qquad K \coloneqq \{x \in \mathbb{R}^n : Ax = b,\ x \ge 0\}, \tag{24.2}$$

where now $A \in \mathbb{Q}^{m\times n}$, $b \in \mathbb{Q}^m$ and $c \in \mathbb{Q}^n$. Let $M = \begin{pmatrix} A \\ I \end{pmatrix} \in \mathbb{Q}^{(m+n)\times n}$ and $d = \begin{pmatrix} b \\ 0 \end{pmatrix} \in \mathbb{Q}^{m+n}$, so that the constraints of K say that Mx agrees with d in the first m coordinates and $x \ge 0$. For $B \subseteq [m+n]$, let M_B be the submatrix of M with the rows indexed by B, and d_B the subvector of d indexed by B.

Definition 24.11. A point $x^* \in K$ is a *basic feasible solution* of (24.2) if there is a set $B \subseteq [m+n]$ with $|B| = n$, called a *basis*, such that M_B is nonsingular and $M_B x^* = d_B$. In words, n linearly independent constraints are tight at x^* and define it.

Theorem 24.12 (Fundamental theorem of linear programming [MG07])**.** *If K is nonempty and the optimal value of (24.2) is bounded, then there is a basic feasible solution x^* that achieves the optimum.*

By Cramer's rule, $x^* = M_B^{-1} d_B$ has polynomially bounded bit complexity: after clearing denominators (Lemma 24.8), each coordinate is a ratio of two determinants of matrices built from the input, and Proposition 24.5 bounds them. This shows that linear programming is in NP: if the optimal value is bounded and at least a given rational t, the optimal basic feasible solution x^* certifies this, and it can be checked exactly in polynomial time. If the optimal value is unbounded, a certificate is a basic feasible solution of the program with variables $(x, s) \ge 0$, constraints $Ax = b$ and $c^\top x - s = t$, and objective 0; this program is feasible, so such a solution exists by Theorem 24.12.

The same argument proves Fact 24.10. The feasible region of (24.1) is nonempty and bounded, so the maximum is attained at a vertex, and a vertex is the unique solution of n linearly independent constraints, taken from the rows of $Ax \le b$, $x \ge 0$ and $x \le R\mathbf{1}$, that are tight at it. Cramer's rule applied to these n equations, after clearing denominators, gives coordinates with $\mathrm{poly}(n, \beta)$ bits.

Semidefinite programs. Semidefinite programs have two kinds of problems. First, they capture eigenvalue problems. For a symmetric matrix A, the program

$$\min\ t \quad \text{subject to} \quad tI - A \succeq 0$$

has optimal value $\lambda_{\max}(A)$, since $tI - A \succeq 0$ holds exactly when $t \geq \lambda_{\max}(A)$. For the matrix of Example 24.9 the optimum is $\sqrt{2}$, so optimal solutions may be irrational. Second, even when the optimal solution is integral, its bit complexity may be exponential in the input size.

Example 24.13. Let $n \geq 1$, and consider the semidefinite program in the variables $x_1, \ldots, x_n$ that minimizes x_n subject to the block diagonal constraint

$$\begin{pmatrix} 1 & 2 \\ 2 & x_1 \end{pmatrix} \oplus \begin{pmatrix} 1 & x_1 \\ x_1 & x_2 \end{pmatrix} \oplus \begin{pmatrix} 1 & x_2 \\ x_2 & x_3 \end{pmatrix} \oplus \cdots \oplus \begin{pmatrix} 1 & x_{n-1} \\ x_{n-1} & x_n \end{pmatrix} \succeq 0.$$

A block diagonal matrix is positive semidefinite if and only if each block is, and a matrix $\left(\begin{smallmatrix} 1 & y \\ y & z \end{smallmatrix}\right)$ is positive semidefinite if and only if $z \geq y^2$. Hence the constraint says $x_1 \geq 4$ and $x_{i+1} \geq x_i^2$ for $1 \leq i \leq n-1$, so every feasible point satisfies $x_t \geq 2^{2^t}$ for all t, by induction on t. The point $x_t = 2^{2^t}$ is feasible, so the optimal value is 2^{2^n}. Writing it down takes $2^n + 1$ bits, while the constraint matrix has only $O(n)$ nonzero entries, each a small constant or a variable.

In general this is bad news for semidefinite programs: we must also argue that the solutions we want have polynomially bounded bit complexity, and that we can afford to work with approximations to them.

24.3 Gaussian elimination

We now give an algorithm that solves linear systems over $\mathbb{Q}$ in time polynomial in the size of the input. By Lemma 24.8, we may assume integer data: the input is a matrix $A = (a_{ij}) \in \mathbb{Z}^{n \times n}$ and a vector $b \in \mathbb{Z}^n$ whose entries are integers with at most β bits, and the goal is to find x with $Ax = b$, if one exists.

Lemma 24.14. *The following operations do not change the set of solutions of $Ax = b$:*

1) *multiplying an equation by a nonzero scalar s, that is, replacing $\sum_j a_{ij}x_j = b_i$ by $\sum_j sa_{ij}x_j = sb_i$;*

2) *subtracting a multiple of one equation from another equation;*

3) *reordering the equations.*

Proof. Each operation is undone by an operation of the same kind (multiplying by $1/s$, adding the multiple back, reordering back), and each operation maps solutions to solutions. □

The idea of Gaussian elimination is to subtract suitable multiples of the first equation from all the others, which makes the first column zero in rows $2, \ldots, n$. Then we subtract suitable multiples of the second equation to make the second column zero in rows $3, \ldots, n$, and so on.

Algorithm 24.15 (Gaussian elimination without pivoting)**.** Set $A^{(0)} := A$ and $b^{(0)} := b$. For $k = 1, 2, \ldots, n$, if $a_{kk}^{(k-1)} \neq 0$, obtain $A^{(k)}$ and $b^{(k)}$ from $A^{(k-1)}$ and $b^{(k-1)}$ by subtracting $a_{ik}^{(k-1)}/a_{kk}^{(k-1)}$ times equation k from equation i, for each $i = k+1, \ldots, n$. Here $a_{ij}^{(k)}$ denotes the (i, j) entry of $A^{(k)}$.

Step k makes column k zero below the diagonal. It changes only rows $k+1,\dots,n$, by subtracting multiples of row k, which is zero in columns $1,\dots,k-1$, so the zeros created in earlier steps stay zero. Step n changes nothing. At the end, $A^{(n)}$ is upper triangular:

$$A^{(n)}x = b^{(n)}, \qquad A^{(n)} = \begin{pmatrix} a_{11}^{(n)} & a_{12}^{(n)} & \cdots & a_{1n}^{(n)} \\ 0 & a_{22}^{(n)} & \cdots & a_{2n}^{(n)} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{nn}^{(n)} \end{pmatrix}.$$

By Lemma 24.14, this system has the same solutions as $Ax = b$, and it is easy to solve by *back substitution*: the last equation gives $x_n = b_n^{(n)}/a_{nn}^{(n)}$; substituting x_n into the second to last equation gives x_{n-1}; and so on, with

$$x_i = \frac{1}{a_{ii}^{(n)}}\Big(b_i^{(n)} - \sum_{j=i+1}^{n} a_{ij}^{(n)} x_j\Big) \qquad (i = n, n-1, \dots, 1).$$

Pivots. So far we have assumed that $a_{11} \neq 0$, then that $a_{22}^{(1)} \neq 0$ for the matrix $A^{(1)}$ obtained after zeroing the first column, and so on. The entry $a_{kk}^{(k-1)}$ by which step k divides is the *pivot*. Suppose that $a_{kk}^{(k-1)} = 0$.

1) If some equation i with $k \le i \le n$, that is, row i of $A^{(k-1)}$, has a nonzero entry in column k, we swap rows i and k, so that this row becomes row k of $A^{(k-1)}$.

2) If all rows i with $k \le i \le n$ have a zero in column k, but some column j with $k < j \le n$ has a nonzero entry in such a row, we swap columns j and k, and then rows as in the first case.

3) If the whole block of $A^{(k-1)}$ with rows $k \le i \le n$ and columns $k \le j \le n$ is zero, we stop. Equations $k,\dots,n$ now read $0 = b_i^{(k-1)}$. If one of these right-hand sides is nonzero, the system has no solution. Otherwise $x_k,\dots,x_n$ can be chosen freely, and back substitution in the first $k-1$ equations, whose coefficient matrix is upper triangular with nonzero diagonal, determines $x_1,\dots,x_{k-1}$. We have identified the affine subspace of solutions.

A row swap reorders the equations, and a column swap renames the variables, so neither affects the solutions beyond renaming.

Proposition 24.16. *Gaussian elimination, followed by back substitution, uses $O(n^3)$ arithmetic operations.*

Proof. Step k involves at most n divisions (one multiplier $a_{ik}^{(k-1)}/a_{kk}^{(k-1)}$ per row $i > k$) and, for each row $i > k$ and each of the at most n columns $j > k$ of A and the right-hand side, one multiplication and one subtraction: at most n^2 multiplications and n^2 subtractions in all. Hence each step uses $O(n^2)$ arithmetic operations, and the n steps use $O(n^3)$. Finding a nonzero pivot takes $O(n^2)$ comparisons per step. Back substitution computes x_i with $O(n-i+1)$ operations, so it takes $O(1) + O(2) + \cdots + O(n) = O(n^2)$ operations in all. □

24.4 Bit complexity of Gaussian elimination

Proposition 24.16 counts arithmetic operations, but this is not yet the running time. In the first step we do arithmetic on numbers with β bits, which takes $\mathrm{poly}(\beta)$ time per operation. The later steps work with the entries of $A^{(k)}$, and we must bound their size.

Throughout this subsection, we consider elimination without swaps, and whenever we speak of $A^{(k)}$ we assume that $a_{tt}^{(t-1)} \neq 0$ for $t = 1, \dots, k$; we return to swaps in Remark 24.21. Step k changes only rows $k+1, \dots, n$, and the row that it subtracts (row k) is zero in columns $1, \dots, k-1$. Hence, by induction on k,

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} \quad \text{for } i \le k, \qquad \text{and} \qquad a_{ij}^{(k)} = 0 \quad \text{for } i > k \text{ and } j \le k. \tag{24.3}$$

So row i does not change after step $i-1$, and column j is zero below the diagonal from step j on. For $i \ge k+1$ and $j \ge k$, step k sets

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{kj}^{(k-1)} \cdot \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}},$$

which ensures $a_{ik}^{(k)} = 0$ for all $i \ge k+1$. Equivalently,

$$a_{ij}^{(k)} = \frac{a_{ij}^{(k-1)} a_{kk}^{(k-1)} - a_{kj}^{(k-1)} a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} = \frac{1}{a_{kk}^{(k-1)}} \det \begin{pmatrix} a_{kk}^{(k-1)} & a_{kj}^{(k-1)} \\ a_{ik}^{(k-1)} & a_{ij}^{(k-1)} \end{pmatrix} \qquad (i \ge k+1,\ j \ge k). \tag{24.4}$$

The product $a_{kj}^{(k-1)} a_{ik}^{(k-1)}$ in (24.4) multiplies two entries of $A^{(k-1)}$, so the bit complexity can double in each step. The best bound that this recursion gives is that the entries of $A^{(n)}$ have about $2^n \beta$ bits, which would mean exponentially large numbers. This naive bound is pessimistic. The key observation is the following lemma of Edmonds [Edm67]. As in Remark 24.7, $A[S, C]$ is the submatrix of A with rows in S and columns in C; let $[0] = \emptyset$ and $\det(A[\emptyset, \emptyset]) = 1$.

Theorem 24.17 (Edmonds [Edm67]). *Let $A \in \mathbb{Q}^{n \times n}$, let $0 \le k \le n-1$, and suppose that $a_{tt}^{(t-1)} \neq 0$ for $t = 1, \dots, k$. For $i, j \ge k+1$, let*

$$D^{(k)} := A^{(k)}[[k], [k]], \qquad D_{ij}^{(k)} := A^{(k)}[[k] \cup \{i\}, [k] \cup \{j\}].$$

Then

1) $a_{ij}^{(k)} = \det(D_{ij}^{(k)}) / \det(D^{(k)})$, *and* $\det(D^{(k)}) \neq 0$;

2) $\det(D^{(k)}) = \det(A[[k], [k]])$ *and* $\det(D_{ij}^{(k)}) = \det(A[[k] \cup \{i\}, [k] \cup \{j\}])$.

Hence, for all $i, j \ge k+1$,

$$a_{ij}^{(k)} = \frac{\det(A[[k] \cup \{i\}, [k] \cup \{j\}])}{\det(A[[k], [k]])}. \tag{24.5}$$

By (24.3), every other entry of $A^{(k)}$ is either 0 or equal to $a_{ij}^{(i-1)}$ with $j \ge i$, to which (24.5) applies with k replaced by $i-1$. So every entry of $A^{(k)}$ is 0 or a ratio of the determinants of two square submatrices of A.

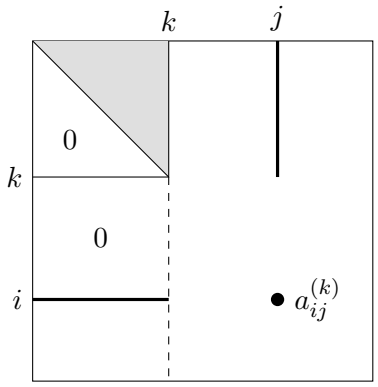


Figure 31: The matrix $A^{(k)}$. The matrix $D_{ij}^{(k)}$ consists of its first k rows and row i, and its first k columns and column j (thick lines and the shaded block). Its first k rows and columns form the upper triangular matrix $D^{(k)}$, and row i is zero in the first k columns, so $D_{ij}^{(k)}$ is upper triangular with last diagonal entry $a_{ij}^{(k)}$.

The proof uses the following key fact about row operations.

Fact 24.18. *Let M be a square matrix, and let M' be obtained from M by replacing its i-th row m_i with $m_i - s\,m_l$ for some row $l \neq i$ and some scalar s. Then $\det(M') = \det(M)$.*

Proof of Theorem 24.17. For the first part, see Fig. 31. By (24.3), $D^{(k)}$ is upper triangular with diagonal entries $a_{tt}^{(k)} = a_{tt}^{(t-1)}$ for $t = 1, \dots, k$, and row i of $A^{(k)}$ is zero in the first k columns because $i \geq k+1$. So $D_{ij}^{(k)}$ is upper triangular with diagonal entries $a_{11}^{(k)}, \dots, a_{kk}^{(k)}, a_{ij}^{(k)}$. The determinant of an upper (or lower) triangular matrix is the product of its diagonal entries, hence

$$\det(D^{(k)}) = a_{11}^{(k)} a_{22}^{(k)} \cdots a_{kk}^{(k)} \neq 0 \qquad \text{and} \qquad \det(D_{ij}^{(k)}) = a_{11}^{(k)} a_{22}^{(k)} \cdots a_{kk}^{(k)}\, a_{ij}^{(k)},$$

and the ratio of the two determinants is exactly $a_{ij}^{(k)}$.

For the second part, note that $A^{(k)}$ is obtained from A by row operations, that is, by subtracting scaled copies of rows. Let $S = [k] \cup \{i\}$ and $C = [k] \cup \{j\}$. In step $t \leq k$, a row r changes only if $r > t$, and then it changes by subtracting a multiple of the current row t; since $t \leq k$, the row t lies in S. Operations on rows outside S do not affect the rows in S. Restricting every operation to the columns in C, we see that $A^{(k)}[S, C] = D_{ij}^{(k)}$ is obtained from $A[S, C] = D_{ij}^{(0)}$ by a sequence of operations as in Fact 24.18. Hence $\det(D_{ij}^{(k)}) = \det(D_{ij}^{(0)})$, and in the same way (with $S = C = [k]$) $\det(D^{(k)}) = \det(D^{(0)})$. Both $D_{ij}^{(0)}$ and $D^{(0)}$ are submatrices of A. □

Corollary 24.19. *Let $A \in \mathbb{Z}^{n \times n}$ have entries with at most β bits, and suppose that the first k steps of Gaussian elimination need no swaps. Then every nonzero entry of $A^{(k)}$ is a ratio of two integers with at most $\log(n!\, 2^{\beta n}) \leq n \log n + n\beta$ bits.*

Proof. By Theorem 24.17, the numerator and the denominator are determinants of square submatrices of A of size at most n; apply Proposition 24.5. □

If each rational is stored in lowest terms, its stored numerator and denominator have at most the absolute values of these determinants, so they obey the same bound. A sharper bound on the determinants comes from Hadamard's inequality.

Fact 24.20 (Hadamard's inequality)**.** *For every matrix $M \in \mathbb{R}^{n\times n}$ with rows $m_1, \dots, m_n$,*

$$|\det(M)| \le \|m_1\|_2 \|m_2\|_2 \cdots \|m_n\|_2.$$

Geometrically, the volume of the parallelepiped spanned by the rows is at most the product of the lengths of its sides.

For an integer matrix with entries of at most β bits, each row has ℓ_2 norm less than $\sqrt{n}\,2^\beta$, so every determinant in Theorem 24.17 has at most $\frac{n}{2}\log n + n\beta$ bits: the numbers grow only to a limited extent, to roughly n times the bit complexity of the original entries.

Remark 24.21. The assumption that no swaps are needed is without loss of generality. Swapping rows $r, r' \ge k$, or columns $j, j' \ge k$, of $A^{(k-1)}$ commutes with the first $k-1$ steps: those steps treat the rows $k, \dots, n$ alike (each has multiples of the rows $1, \dots, k-1$ subtracted, with multipliers read from its own entries in columns $1, \dots, k-1$), and they act on all columns alike. Hence elimination with swaps on A produces the same matrices as elimination without swaps on PAQ for suitable permutation matrices P and Q, and the square submatrices of PAQ are square submatrices of A with permuted rows and columns, whose determinants agree up to sign. The right-hand side is updated by the same row operations, so Theorem 24.17 and Corollary 24.19 also hold for the augmented matrix $(A \mid b)$, with $j = n+1$ for the column b; the proof is the same.

Theorem 24.22. *Let $A \in \mathbb{Z}^{n\times n}$ and $b \in \mathbb{Z}^n$ have entries with at most β bits. Gaussian elimination with exact rational arithmetic decides whether $Ax = b$ has a solution, and finds one if so, in time polynomial in n and β.*

Proof. By Proposition 24.16, the algorithm uses $O(n^3)$ arithmetic operations, so it suffices to show that every number it handles has polynomially many bits. By Corollary 24.19 and Remark 24.21, the entries of every matrix $A^{(k)}$ and of the right-hand side are ratios of integers with at most $s := n\log n + n\beta$ bits. The multipliers $a_{ik}^{(k-1)}/a_{kk}^{(k-1)}$ and the intermediate products and differences in (24.4) are obtained from such ratios by at most three operations, so by Lemma 24.3 they have $O(s)$ bits. Let k be the step at which the elimination stops in the third case of the pivoting rules, or $k = n+1$ if it never does. In back substitution, the values $x_1, \dots, x_{k-1}$ (with the free variables set to 0) form the solution of the first $k-1$ equations of the eliminated system, which have the same solutions as a nonsingular square subsystem of the permuted system $PAQ\,y = Pb$; so each of them is a ratio of integers with at most s bits, by Corollary 24.6. Each partial sum in the formula for x_i is a sum of at most n products of two such ratios, hence a ratio of integers with $O(ns)$ bits. Keeping all fractions in lowest terms takes a polynomial number of gcd computations. So every operation acts on numbers with $O(ns)$ bits and takes polynomial time. □

There is a further way to keep all the numbers integral. For $0 \le k \le n-1$ and $i, j \ge k+1$, define

$$\Delta_{ij}^k := \det(D_{ij}^{(k)}) = \det(A[[k] \cup \{i\}, [k] \cup \{j\}]),$$

so that $\Delta_{ij}^0 = a_{ij}$, and set $\Delta_{00}^{-1} := 1$. Rows $1, \dots, k$ of $A^{(k)}$ and $A^{(k-1)}$ agree by (24.3), so $D^{(k)} = D_{kk}^{(k-1)}$ and $\det(D^{(k)}) = \Delta_{kk}^{k-1}$ for $k \ge 1$; this also holds for $k = 0$ by our conventions. Thus Theorem 24.17 reads $a_{ij}^{(k)} = \Delta_{ij}^k / \Delta_{kk}^{k-1}$.

Proposition 24.23. *Under the hypotheses of Theorem 24.17 with $k+1$ steps in place of k, for $0 \le k \le n-2$ and all $i, j \ge k+2$,*

$$\Delta_{ij}^{k+1} = \frac{\Delta_{ij}^{k}\Delta_{k+1,k+1}^{k} - \Delta_{i,k+1}^{k}\Delta_{k+1,j}^{k}}{\Delta_{kk}^{k-1}}.$$

Proof. Apply (24.4) to step $k+1$, and substitute $a_{pq}^{(k)} = \Delta_{pq}^{k}/\Delta_{kk}^{k-1}$ for the four entries of $A^{(k)}$ involved and $a_{ij}^{(k+1)} = \Delta_{ij}^{k+1}/\Delta_{k+1,k+1}^{k}$:

$$\frac{\Delta_{ij}^{k+1}}{\Delta_{k+1,k+1}^{k}} = \frac{\Delta_{ij}^{k}\Delta_{k+1,k+1}^{k} - \Delta_{k+1,j}^{k}\Delta_{i,k+1}^{k}}{(\Delta_{kk}^{k-1})^2} \cdot \frac{\Delta_{kk}^{k-1}}{\Delta_{k+1,k+1}^{k}}.$$

Multiplying by $\Delta_{k+1,k+1}^{k}$ gives the claim. □

Every Δ_{ij}^{k} is the determinant of a submatrix of A, so it is an integer when A is, and the division in Proposition 24.23 is exact. We can therefore use the proposition as a recurrence that works entirely within the integers, and recover the entries of the final matrix at the end: by (24.3) and Theorem 24.17, $a_{ij}^{(n)} = a_{ij}^{(i-1)} = \Delta_{ij}^{i-1}/\Delta_{i-1,i-1}^{i-2}$ for $j \ge i$. This integer-preserving form of elimination is also known from the work of Bareiss [Bar68].

The numbers do get larger, though only to the extent that Corollary 24.19 allows.

Example 24.24. For $n \ge 2$, let $A \in \mathbb{Z}^{n\times n}$ have 1 on the diagonal, -1 below the diagonal, 1 in the last column and 0 elsewhere; for $n = 5$,

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{pmatrix} \quad \text{gives} \quad A^{(n)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 & 8 \\ 0 & 0 & 0 & 0 & 16 \end{pmatrix}.$$

In general the last column of $A^{(n)}$ is $(1, 2, 4, \dots, 2^{n-1})$. Indeed, by induction on k, after step k every row $i > k$ has zeros in columns $1, \dots, k$, entries -1 in columns $k+1, \dots, i-1$, the entry 1 in column i if $i < n$, and 2^k in the last column. So for $k+1 < n$ the pivot row $k+1$ is the $(k+1)$-th unit vector plus 2^k in the last column, and step $k+1$ adds it to every row $i > k+1$, which clears column $k+1$ and doubles the last entry. Row i does not change after step $i-1$, so it ends with 2^{i-1} in the last column. The pivots $a_{kk}^{(k-1)}$ of steps $k = 1, \dots, n-1$ are all 1. The largest absolute value of an entry grows from 1 to 2^{n-1}.

Remark 24.25. The basics of Gaussian elimination are in Trefethen and Bau [TB97], and the notes of Lovász and Gács [LG18] give Edmonds' proof. Another approach is to compute modulo several primes and combine the results by the Chinese remainder theorem, which has its own issues.

25 The Ellipsoid Algorithm, This Time Without Ellipsoids

Lecture of October 27, 2021, by Anupam Gupta. Based on the lecturer's handwritten notes.

Linear programs can be solved in polynomial time by two kinds of algorithms, ellipsoid-type algorithms and interior point algorithms, and this lecture develops the ellipsoid-type approach. Its basic subroutine finds a point in a convex set K that is accessible only through a separation oracle, or certifies that K has tiny volume. The general scheme keeps a region that contains K, queries its center, and uses the answer to replace the region by a smaller one; if the volume shrinks by a factor $1-\delta$ in each step, then $O(\delta^{-1}\log(\text{volume ratio}))$ steps suffice. We compare three choices of the region (centroids of polytopes, ellipsoids and simplices) and analyze in full the simplices method of Yamnitsky and Levin [YL82], which keeps a simplex, needs no square roots, and shrinks the volume by a factor $1-\Theta(1/n^2)$ per step. For the feasibility of linear programs, two facts that we state without proof, one of them a bit-size bound as in Lecture 24, make the number of iterations polynomial in the input size.

25.1 Feasibility and separation oracles

The basic subroutine is the following feasibility problem.

Definition 25.1. In the *feasibility problem*, we are given a convex compact set $K \subseteq \mathbb{R}^n$ and a number $\varepsilon > 0$, and we must either find a point $x \in K$ or show that $\text{vol}(K) \le \varepsilon$.

The set K is not given explicitly; we can only ask questions about it.

Definition 25.2. A *(strong) separation oracle* for a convex compact set $K \subseteq \mathbb{R}^n$ takes as input a point $c \in \mathbb{R}^n$ (the letter stands for check, or center) and outputs

- YES if $c \in K$;
- NO if $c \notin K$, together with a hyperplane separating c from K, that is, a vector $a \in \mathbb{R}^n$ with $a \neq 0$ such that
$$K \subseteq \{x \in \mathbb{R}^n : a^\top x < a^\top c\}.$$

Such a vector a exists for every $c \notin K$. If $K = \emptyset$, every $a \neq 0$ works. Otherwise, if p is the point of K nearest to c, then $a = c - p$ satisfies $a^\top x \le a^\top p = a^\top c - \|a\|_2^2 < a^\top c$ for all $x \in K$. Figure 32 shows the three cases. In general one can even handle *weak* separation oracles, which tolerate errors of some $\varepsilon > 0$, but we do not pursue this here; the book of Grötschel, Lovász and Schrijver [GLS88] has many details.

25.2 The general approach

The general approach is a form of binary search. At each time t, we maintain a region $\mathcal{E}^{(t)} \subseteq \mathbb{R}^n$ such that the following hold.

(a) $\mathcal{E}^{(0)} \supseteq K$.

(b) At each time we test a point $c^{(t)}$, a center of $\mathcal{E}^{(t)}$ (in a sense that depends on the shape), with the separation oracle. If the answer is YES, we are done. If it is NO, with vector $a^{(t)}$, we find a region $\mathcal{E}^{(t+1)}$ that contains
$$\mathcal{E}^{(t)} \cap \{x \in \mathbb{R}^n : (a^{(t)})^\top x < (a^{(t)})^\top c^{(t)}\}.$$

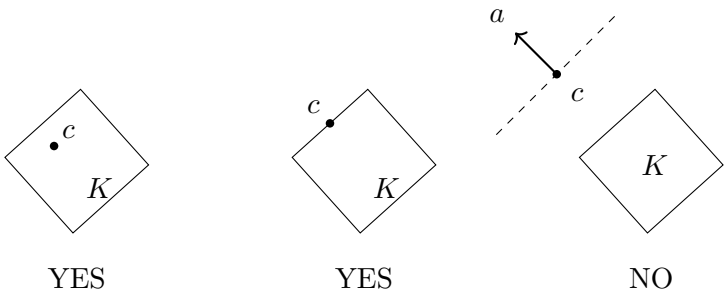


Figure 32: A separation oracle. It answers YES when c lies in K, including its boundary. When $c \notin K$, it answers NO and returns a vector $a \neq 0$; the hyperplane $\{x : a^\top x = a^\top c\}$ (dashed) passes through c, and K lies strictly on the side opposite to a.

(c) In the NO case, $\mathrm{vol}(\mathcal{E}^{(t+1)}) \leq (1-\delta)\,\mathrm{vol}(\mathcal{E}^{(t)})$, for a parameter $\delta \in (0, 1]$.

Lemma 25.3. *Suppose that the regions satisfy (a), (b) and (c), let $\varepsilon > 0$, and let T be an integer with*

$$T \geq \frac{1}{\delta} \ln\Big(\frac{\mathrm{vol}(\mathcal{E}^{(0)})}{\varepsilon}\Big).$$

If the oracle answers NO to the first T queries, then $\mathrm{vol}(K) \leq \varepsilon$. Hence the scheme solves the feasibility problem with at most T oracle calls.

Proof. We show by induction on t that $K \subseteq \mathcal{E}^{(t)}$ as long as all answers are NO. This holds for $t = 0$ by (a). If $K \subseteq \mathcal{E}^{(t)}$ and the answer at $c^{(t)}$ is NO, then K lies in the open halfspace $\{x : (a^{(t)})^\top x < (a^{(t)})^\top c^{(t)}\}$ by the definition of the oracle, so K lies in the intersection in (b), and hence in $\mathcal{E}^{(t+1)}$. If the first T answers are NO, then by (c),

$$\mathrm{vol}(\mathcal{E}^{(T)}) \leq (1-\delta)^T\,\mathrm{vol}(\mathcal{E}^{(0)}) \leq e^{-\delta T}\,\mathrm{vol}(\mathcal{E}^{(0)}) \leq \varepsilon,$$

and therefore $\mathrm{vol}(K) \leq \varepsilon$, since $K \subseteq \mathcal{E}^{(T)}$. If some call answers YES, we have already found a point of K. □

To make this scheme a polynomial-time algorithm, there are three things to do. Let L denote the bit complexity of the input.

1) Show that we can choose $\mathcal{E}^{(0)}$ with $\mathrm{vol}(\mathcal{E}^{(0)}) \leq 2^{\mathrm{poly}(L,n)}$. This is true for linear programs.

2) Ensure that $\delta \geq 1/\mathrm{poly}(L, n)$, for whatever choice of shape of the regions $\mathcal{E}^{(t)}$ and notion of center $c^{(t)}$ we make.

3) For the feasibility problem to be useful, show that small volume means something useful. Looking ahead to linear programs: suppose that K is nonempty and full-dimensional, so that $\mathrm{vol}(K) > 0$. Then one can show that $\mathrm{vol}(K) > 2^{-\mathrm{poly}(nL)}$. So if we set ε to this bound, the outcome $\mathrm{vol}(K) \leq \varepsilon$ is impossible, and hence some call to the oracle must return YES.

Items 1 and 3 are taken up again at the end of this lecture. For now we focus on solving the feasibility problem with a good δ.

25.3 Three choices of the region

Centroids. A. Y. Levin and D. J. Newman [New65] (in the 1960s) define $\mathcal{E}^{(t)} = \mathcal{E}^{(t-1)} \cap \{x : (a^{(t-1)})^\top x < (a^{(t-1)})^\top c^{(t-1)}\}$, starting with a ball $\mathcal{E}^{(0)}$ that contains K, and let $c^{(t)}$ be the centroid of $\mathcal{E}^{(t)}$. This gives $\delta = \Theta(1)$ (by a theorem of Grünbaum [Grü60], each side of a hyperplane through the centroid of a convex body contains at least a $(n/(n+1))^n \geq 1/e$ fraction of its volume, so $\delta \geq 1/e$), but centroids are difficult to compute.

Ellipsoids. Khachiyan [Kha80], in 1979, used an idea of Shor [Sho70] from 1970: let $\mathcal{E}^{(t)}$ be an ellipsoid that contains $\mathcal{E}^{(t-1)} \cap \{x : (a^{(t-1)})^\top x < (a^{(t-1)})^\top c^{(t-1)}\}$, and let $c^{(t)}$ be the center of the ellipsoid. This gives $\delta = \Theta(1/n)$, and it was the first polynomial-time algorithm for linear programs. It needs control over the bit precision of the numbers involved, since it computes square roots, among other things.

Simplices. The rest of this lecture uses the simplices idea of B. Yamnitsky and L. Levin [YL82] from 1982: let $\mathcal{E}^{(t)}$ be a simplex that contains $\mathcal{E}^{(t-1)} \cap \{x : (a^{(t-1)})^\top x < (a^{(t-1)})^\top c^{(t-1)}\}$, with $\mathcal{E}^{(0)}$ a simplex as well. Now $\delta = \Theta(1/n^2)$, which is worse, but square roots are no longer needed. Naively the bit precision of the vertices may still increase, so some truncation or rounding is needed; see Bartels [Bar00].

25.4 The simplices method

The name distinguishes this method from the simplex method of George Dantzig. Throughout this section, $n \geq 2$, and $K \subseteq \mathbb{R}^n$ is a convex compact set given by a separation oracle. The method starts with a simplex $S^0 = \operatorname{conv}(v_0^0, \ldots, v_n^0)$ of positive volume that contains K, and sets c^t to be the centroid of S^t. Each time the oracle answers NO, it computes a new simplex S^{t+1} by shifting some of the vertices of S^t, such that $\operatorname{vol}(S^{t+1}) \leq \operatorname{vol}(S^t)(1 - \Theta(1/n^2))$.

Algorithm 25.4 (Simplices method of Yamnitsky and Levin)**.** For $t = 0, 1, 2, \ldots$:

1) Let $S^t = \operatorname{conv}(v_0^t, \ldots, v_n^t)$.

2) Let $c^t = \frac{1}{n+1} \sum_{i=0}^n v_i^t$ be its centroid, and query the oracle at c^t. If the answer is YES, output c^t and stop.

3) If the answer is NO, we get back $a \in \mathbb{R}^n$ with $K \subseteq \{x : a^\top x < a^\top c^t\}$. (Naturally a depends on t; we do not write the superscript, to avoid clutter.)

   a) Define $e(x) \coloneqq \langle a, c^t - x \rangle$ for $x \in \mathbb{R}^n$, so that $K \subseteq \{x : e(x) > 0\}$.

   b) Choose $\ell \in \arg\max_i e(v_i^t)$.

   c) For every i, define

$$d_i \coloneqq 1 - \frac{e(v_i^t)}{n^2\, e(v_\ell^t)} \qquad \text{and} \qquad v_i^{t+1} \coloneqq \Big(1 - \frac{1}{d_i}\Big) v_\ell^t + \frac{1}{d_i} v_i^t = v_\ell^t + \frac{v_i^t - v_\ell^t}{d_i}.$$

In the analysis we write $v_0, \ldots, v_n$ for the vertices of S^t and $v_0', \ldots, v_n'$ for those of S^{t+1}.

Lemma 25.5. *In every step with answer NO, the following hold.*

1) $e(v_\ell) > 0$.

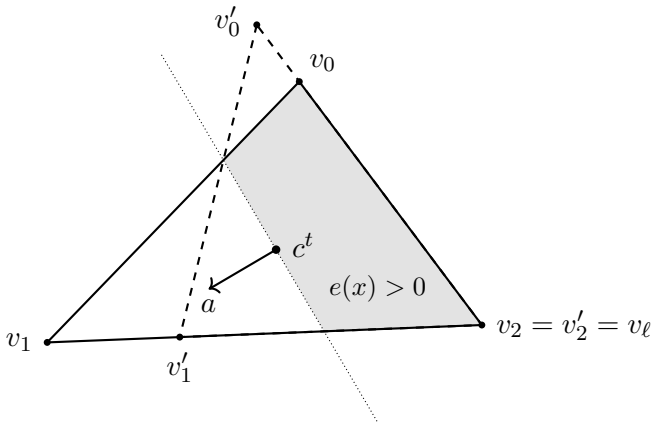


Figure 33: One step of the simplices method in the plane ($n = 2$). The old simplex S^t (solid) has centroid c^t; the oracle returns a, and K lies in the shaded part $S^t \cap \{x : e(x) > 0\}$, on the side of the dotted line through c^t opposite to a. The vertex $v_\ell = v_2$ maximizes e and stays fixed; v_0, with $e(v_0) > 0$, moves away from v_ℓ, and v_1, with $e(v_1) < 0$, moves toward it. The new simplex S^{t+1} (dashed) contains the shaded region.

2) *$d_i \geq 1 - 1/n^2 > 0$ for all i, and $v_\ell' = v_\ell$.*

3) *If $i \neq \ell$ and $e(v_i) > 0$, that is, v_i lies on the same side of the hyperplane $\{x : e(x) = 0\}$ as v_ℓ, then $d_i < 1$, so $1/d_i > 1$ and v_i moves away from v_ℓ. If $e(v_i) < 0$, that is, v_i lies on the other side, where K cannot be, then $d_i > 1$ and v_i moves closer to v_ℓ. If $e(v_i) = 0$, then $d_i = 1$ and $v_i' = v_i$.*

Proof. The map e is affine, so $\sum_{i=0}^{n} e(v_i) = (n+1)\, e(c^t) = 0$. Hence $\max_i e(v_i) \geq 0$, and equality would force $e(v_i) = 0$ for all i, that is, all the vertices would lie on the hyperplane $\{x : e(x) = 0\}$. This is impossible, because $a \neq 0$ and the vertices of a simplex of positive volume do not lie on a common hyperplane. This proves the first part, and S^{t+1} again has positive volume by Lemma 25.7 below. Since $e(v_i) \leq e(v_\ell)$, we get $d_i \geq 1 - 1/n^2 \geq 3/4$, and $d_\ell = 1 - 1/n^2$, so $v_\ell' = v_\ell$. The third part follows from $v_i' - v_\ell = (v_i - v_\ell)/d_i$. □

We want to show two facts: the new simplex contains the part of the old one on the side of K, so that $K \subseteq S^{t+1}$, and its volume is smaller by a factor $1 - \Omega(1/n^2)$, which is the parameter δ of the general approach.

Lemma 25.6. *Every point $x \in S^t$ with $e(x) > 0$ lies in S^{t+1}. Hence $K \subseteq S^t$ implies $K \subseteq S^{t+1}$.*

Proof. Let $x = \sum_i \lambda_i v_i$ with $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$, and suppose that $e(x) > 0$. Define

$$\lambda_i' := \begin{cases} \lambda_i d_i & \text{if } i \neq \ell, \\ \lambda_\ell d_\ell + \dfrac{e(x)}{n^2 e(v_\ell)} & \text{if } i = \ell. \end{cases}$$

We show that these are convex coefficients that express x in terms of the new vertices. First, $\lambda_i' \geq 0$, because all the terms are nonnegative (by Lemma 25.5, $d_i > 0$ and $e(v_\ell) > 0$). Second, since e is affine and $\sum_i \lambda_i = 1$, we have $\sum_i \lambda_i e(v_i) = e(x)$, so

$$\sum_i \lambda_i' = \sum_i \lambda_i d_i + \frac{e(x)}{n^2 e(v_\ell)} = \sum_i \lambda_i \Big(1 - \frac{e(v_i)}{n^2 e(v_\ell)}\Big) + \frac{e(x)}{n^2 e(v_\ell)} = \sum_i \lambda_i - \frac{\sum_i \lambda_i e(v_i) - e(x)}{n^2 e(v_\ell)} = 1.$$

Finally, using $v_i' = (1 - 1/d_i)v_\ell + v_i/d_i$ for every i (including $i = \ell$),

$$\begin{aligned}\sum_i \lambda_i' v_i' &= \sum_i \lambda_i d_i \Big(\Big(1 - \frac{1}{d_i}\Big)v_\ell + \frac{v_i}{d_i}\Big) + \frac{e(x)}{n^2 e(v_\ell)}\, v_\ell \\ &= v_\ell \sum_i \lambda_i (d_i - 1) + \sum_i \lambda_i v_i + \frac{e(x)}{n^2 e(v_\ell)}\, v_\ell.\end{aligned}$$

Here $\sum_i \lambda_i v_i = x$, and $\sum_i \lambda_i (d_i - 1) = -\sum_i \lambda_i e(v_i)/(n^2 e(v_\ell)) = -e(x)/(n^2 e(v_\ell))$ is the negative of the coefficient in the last term. Hence $\sum_i \lambda_i' v_i' = x$, and $x \in S^{t+1}$.

For the second statement, the oracle guarantees $K \subseteq \{x : e(x) > 0\}$, so $K \subseteq S^t$ gives $K \subseteq S^t \cap \{x : e(x) > 0\} \subseteq S^{t+1}$. □

Lemma 25.7. *In every step with answer NO,*

$$\operatorname{vol}(S^{t+1}) = \frac{\operatorname{vol}(S^t)}{\prod_{i \neq \ell} d_i} \leq e^{-1/(2n^2)+1/(6n^3)} \operatorname{vol}(S^t) \leq e^{-1/(3n^2)} \operatorname{vol}(S^t).$$

Proof. The volume of a simplex $\operatorname{conv}(v_0, v_1, \ldots, v_n)$ is

$$\frac{1}{n!}\left|\det \begin{pmatrix} v_1 - v_0 \\ v_2 - v_0 \\ \vdots \\ v_n - v_0 \end{pmatrix}\right|,$$

with the absolute value, since the determinant changes sign when two vertices are swapped. For simplicity, renumber the vertices so that $\ell = 0$. Then $v_0' = v_0$ and $v_i' = v_0 + (v_i - v_0)/d_i$, so $v_i' - v_0' = (v_i - v_0)/d_i$, and since all $d_i > 0$,

$$\operatorname{vol}(S^{t+1}) = \frac{1}{n!}\left|\det \begin{pmatrix} (v_1 - v_0)/d_1 \\ \vdots \\ (v_n - v_0)/d_n \end{pmatrix}\right| = \frac{\operatorname{vol}(S^t)}{\prod_{i \neq 0} d_i}.$$

It remains to bound $\prod_{i \neq 0} d_i$ from below. By construction,

$$d_i = 1 - \frac{e(v_i)}{n^2 e(v_0)} \geq 1 - \frac{1}{n^2},$$

since $e(v_0) \geq e(v_i)$. Moreover, since $\sum_{i=0}^n e(v_i) = (n+1)\, e(c^t) = 0$ by the definition of $e(x) = a^\top(c^t - x)$,

$$\sum_{i \neq 0} d_i = \sum_{i \neq 0}\Big(1 - \frac{e(v_i)}{n^2 e(v_0)}\Big) = n - \sum_{i=0}^{n} \frac{e(v_i)}{n^2 e(v_0)} + \frac{e(v_0)}{n^2 e(v_0)} = n + \frac{1}{n^2}.$$

So we want to minimize $\prod_{i \neq 0} d_i$ subject to $d_i \geq \alpha := 1 - 1/n^2$ and $\sum_{i \neq 0} d_i = n + 1/n^2$. The minimum is attained when all but one of the d_i equal α and the last one equals $n + 1/n^2 - (n-1)\alpha = 1 + 1/n$. Indeed, if two of the numbers, d_i and d_j, exceed α, then replacing them by α and $d_i + d_j - \alpha \geq \alpha$ keeps the sum and the constraints and does not increase the product, because

$d_i d_j - \alpha(d_i + d_j - \alpha) = (d_i - \alpha)(d_j - \alpha) \geq 0$; after at most $n-1$ such replacements, at most one number differs from α. Hence

$$\prod_{i \neq 0} d_i \geq \Big(1 - \frac{1}{n^2}\Big)^{n-1}\Big(1 + \frac{1}{n}\Big) = \frac{(1 - 1/n^2)^n}{1 - 1/n}.$$

Taking logarithms and expanding $\ln(1-y) = -\sum_{k \geq 1} y^k/k$ (valid since $n \geq 2$),

$$\begin{aligned}
\ln \frac{(1 - 1/n^2)^n}{1 - 1/n} &= -n \sum_{k \geq 1} \frac{1}{k n^{2k}} + \sum_{k \geq 1} \frac{1}{k n^k} \\
&= \Big(-\frac{n}{n^2} - \frac{n}{2n^4} - \frac{n}{3n^6} - \cdots\Big) + \Big(\frac{1}{n} + \frac{1}{2n^2} + \frac{1}{3n^3} + \cdots\Big) \\
&\geq \frac{1}{2n^2} - \frac{1}{6n^3}.
\end{aligned}$$

For the last step, the terms in $1/n$ cancel, the terms in $1/n^2$ and $1/n^3$ give $\frac{1}{2n^2} - \frac{1}{6n^3}$, and for $j \geq 2$ the terms in $1/n^{2j}$ and $1/n^{2j+1}$ together give

$$\frac{1}{2j\, n^{2j}} + \Big(\frac{1}{2j+1} - \frac{1}{j+1}\Big)\frac{1}{n^{2j+1}} = \frac{1}{n^{2j}}\Big(\frac{1}{2j} - \frac{j}{(2j+1)(j+1)\, n}\Big) \geq 0,$$

since $(2j+1)(j+1) \geq 2j^2$. Hence $\prod_{i \neq 0} d_i \geq e^{1/(2n^2) - 1/(6n^3)}$, which gives the first inequality of the lemma. The second follows from $\frac{1}{2n^2} - \frac{1}{6n^3} - \frac{1}{3n^2} = \frac{n-1}{6n^3} \geq 0$. □

So the volume shrinks by a factor of at most $e^{-1/(2n^2)+1/(6n^3)} = 1 - \Theta(1/n^2)$ in each step; this is the parameter $\delta = \Theta(1/n^2)$ of the simplices method.

Theorem 25.8. *Let $n \geq 2$, let $K \subseteq \mathbb{R}^n$ be a convex compact set given by a separation oracle, let S^0 be a simplex of positive volume with $K \subseteq S^0$, and let $\varepsilon > 0$. Let T be an integer with $T \geq 3n^2 \ln(\mathrm{vol}(S^0)/\varepsilon)$. Then the simplices method either outputs a point of K within its first T oracle calls, or the first T answers are NO and $\mathrm{vol}(K) \leq \varepsilon$.*

Proof. By Lemma 25.6 and induction on t, $K \subseteq S^t$ for every t reached. If the first T answers are NO, then by Lemma 25.7, $\mathrm{vol}(K) \leq \mathrm{vol}(S^T) \leq e^{-T/(3n^2)}\,\mathrm{vol}(S^0) \leq \varepsilon$. This is Lemma 25.3 with $\mathcal{E}^{(t)} = S^t$. □

Remark 25.9. The same kind of ideas hold for the ellipsoid algorithm, but there one works with positive definite matrices that define the ellipsoids.

Remark 25.10. The analysis above assumes exact arithmetic, and the numbers that describe the vertices v_i^t may blow up. Bartels [Bar00] gives a rounding technique that enlarges the simplex slightly and obtains vertices with bounded precision; see his paper for the details.

25.5 Solving linear programs

The simplices method solves the feasibility problem. To use it for the feasibility of a linear program, with K the polytope of feasible solutions, we use two facts for full-dimensional polytopes, which we state without proof (see [GLS88]); L is again the bit complexity of the input.

1) If the linear program has an optimal solution, then it has one with bounded bit precision, as in Lecture 24. So adding a bounding box of the form $|x_i| \le 2^{\mathrm{poly}(L)}$ for all i does no harm, and the initial simplex can be chosen with $\mathrm{vol}(S^0) \le 2^{\mathrm{poly}(n,L)}$. For instance, for $R > 0$ the simplex $S^0 = \mathrm{conv}(-R\mathbf{1}, -R\mathbf{1} + 2nR\,e_1, \dots, -R\mathbf{1} + 2nR\,e_n)$, where $e_1, \dots, e_n$ are the standard unit vectors, contains the box $[-R, R]^n$: for x in the box, the point $y = x + R\mathbf{1}$ has coordinates in $[0, 2R]$, so $\sum_i y_i \le 2nR$. Its volume is $(2nR)^n/n!$.

2) If $\mathrm{vol}(K) > 0$, then $\mathrm{vol}(K) > 2^{-\mathrm{poly}(nL)}$. So it is fine to set ε to this value: if all oracle calls return NO, then $\mathrm{vol}(K) \le \varepsilon$, so $\mathrm{vol}(K) = 0$.

With these choices, the number of iterations is

$$\frac{1}{\delta} \ln\Big(\frac{\mathrm{vol}(S^0)}{\varepsilon}\Big) = O(n^2) \cdot \mathrm{poly}(nL),$$

which is polynomial in the input size.

26 Eigenvalue Computations

Lecture of October 29, 2021, by Anupam Gupta. Based on the lecturer's handwritten notes.

Given a symmetric matrix A, we want to find its top eigenpair (the eigenvalue of largest magnitude and an eigenvector), its top k eigenpairs, or all of them. Even when A has rational entries, its eigenvalues may be irrational, so algorithms compute approximations that converge to the right answer. This lecture presents the basic iterative methods: power iteration, inverse iteration with shifts, and Rayleigh quotient iteration for a single eigenpair, and orthogonal iteration and its reformulation as the QR algorithm for many eigenpairs at once. We prove the basic properties of these methods and state the deeper convergence results with references; the books of Trefethen and Bau [TB97] (very readable) and of Golub and Van Loan [GVL13] (more encyclopedic) contain the details.

26.1 The problem

Example 26.1. The matrix $A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ has rational entries, but its characteristic polynomial is $\lambda^2 - 2$, so its eigenvalues are $\pm\sqrt{2}$.

We focus on symmetric matrices, for which the eigenvalues are real and the eigenvectors can be chosen orthonormal.

Fact 26.2 (Spectral theorem)**.** *Every symmetric $A \in \mathbb{R}^{n\times n}$ can be written as $A = Q\Lambda Q^\top$, where Q is an orthogonal matrix ($QQ^\top = Q^\top Q = I$) whose columns are eigenvectors of A, and $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ holds the eigenvalues, which are real. (This is the Schur decomposition, which for a symmetric matrix is diagonal.)*

Throughout this lecture, $A \in \mathbb{R}^{n\times n}$ is symmetric, with eigenvalues $\lambda_1, \ldots, \lambda_n$ and orthonormal eigenvectors $q_1, \ldots, q_n$ as in Fact 26.2, numbered so that $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$.

26.2 Power iteration

Algorithm 26.3 (Power iteration)**.** Let v^0 be a random unit vector, say. Repeat $v^{t+1} \leftarrow Av^t$, renormalizing if desired to control the length.

Renormalizing changes only the lengths of the vectors v^t, not their directions, which are what we care about. Write $v^0 = \sum_{i=1}^n c_i q_i$ in the eigenbasis, with $c_i = \langle v^0, q_i\rangle$. Then, without renormalization,

$$v^t = A^t\Big(\sum_i c_i q_i\Big) = \sum_i c_i \lambda_i^t q_i = \lambda_1^t\Big(c_1 q_1 + \Big(\frac{\lambda_2}{\lambda_1}\Big)^t c_2 q_2 + \cdots + \Big(\frac{\lambda_n}{\lambda_1}\Big)^t c_n q_n\Big). \tag{26.1}$$

If $|\lambda_2| < |\lambda_1|$, then the terms except the first fade away (relatively), whereas the first term $c_1 q_1$ remains fixed up to the scaling by λ_1^t, so the direction of v^t converges to that of q_1, provided that $c_1 \neq 0$. This happens with probability 1 for a uniformly random unit vector v^0, since the unit vectors orthogonal to q_1 form a set of measure zero. We state this for a general symmetric matrix B, so that we can apply it to other matrices below. For nonzero vectors v, u, let $\theta(v, u) \in [0, \pi/2]$ be the angle between the lines spanned by v and u, so that $\cos\theta(v, u) = |\langle v, u\rangle|/(\|v\|_2\|u\|_2)$.

Proposition 26.4. *Let $B \in \mathbb{R}^{n\times n}$ be symmetric with eigenvalues $\beta_1, \dots, \beta_n$ and orthonormal eigenvectors $u_1, \dots, u_n$, where $|\beta_1| > |\beta_2| \geq \cdots \geq |\beta_n|$. Let $v^0 \in \mathbb{R}^n$ satisfy $\langle v^0, u_1\rangle \neq 0$, and let $v^{t+1} = Bv^t$, possibly rescaled by a nonzero factor. Then $v^t \neq 0$ for all t, and*

$$\tan\theta(v^t, u_1) \leq \Big|\frac{\beta_2}{\beta_1}\Big|^t \tan\theta(v^0, u_1).$$

In particular, $\theta(v^t, u_1) \to 0$, and $\min_{s=\pm1}\|v^t/\|v^t\|_2 - s\,u_1\|_2 \to 0$ as $t \to \infty$.

Proof. Rescaling does not change angles, so we may assume $v^t = B^t v^0$. Write $v^0 = \sum_i c_i u_i$. As in (26.1), $v^t = \sum_i c_i \beta_i^t u_i$; its u_1-component $c_1\beta_1^t$ is nonzero, because $c_1 \neq 0$ and $|\beta_1| > |\beta_2| \geq 0$. By orthonormality,

$$\tan\theta(v^t, u_1) = \frac{\big(\sum_{i\geq 2} c_i^2 \beta_i^{2t}\big)^{1/2}}{|c_1||\beta_1|^t} \leq \frac{|\beta_2|^t\big(\sum_{i\geq 2} c_i^2\big)^{1/2}}{|c_1||\beta_1|^t} = \Big|\frac{\beta_2}{\beta_1}\Big|^t \tan\theta(v^0, u_1).$$

For a unit vector w with $\theta = \theta(w, u_1)$, the closer of $\pm u_1$ is at distance $2\sin(\theta/2) \leq \theta \leq \tan\theta$, which gives the last claim. □

Since λ_1^t may alternate in sign (when $\lambda_1 < 0$), the normalized vectors $v^t/\|v^t\|_2$ converge to q_1 only up to sign.

26.3 Inverse iteration and shifts

Power iteration finds the eigenvalue of largest magnitude; inverting A finds the one of smallest magnitude. Suppose that A has full rank. Then $A^{-1} = Q\Lambda^{-1}Q^\top$ has the eigenvalues $1/\lambda_1, \dots, 1/\lambda_n$, with the same eigenvectors. So if λ_n is strictly the smallest in magnitude, that is, $|\lambda_1| \geq \cdots \geq |\lambda_{n-1}| > |\lambda_n| > 0$, then $1/\lambda_n$ is strictly the largest in magnitude, and power iteration is good for it.

Algorithm 26.5 (Inverse iteration)**.** Let v^0 be a random unit vector. At each step, set $v^{t+1} \leftarrow A^{-1}v^t$, or equivalently, solve the linear system $Av^{t+1} = v^t$, by Gaussian elimination (Lecture 24), for example.

For example, if $A = LU$ with L lower triangular and U upper triangular (computed once, by Gaussian elimination), then each step solves $Lz = v^t$ and then $Uv^{t+1} = z$, both by substitution. By Proposition 26.4 applied to $B = A^{-1}$, the direction of v^t converges to that of q_n (if $\langle v^0, q_n\rangle \neq 0$, which holds with probability 1), and the tangent of the angle shrinks by a factor $|\lambda_n/\lambda_{n-1}|$ per step. So if $|\lambda_n| \ll |\lambda_{n-1}|$, we get good convergence.

Shifting. Another idea is that the eigenvalues of $A - \mu I$ are $\lambda_1 - \mu, \dots, \lambda_n - \mu$, with the same eigenvectors. So if we have a good guess μ for some eigenvalue λ_i, but μ is not an eigenvalue itself, we run power iteration on $(A - \mu I)^{-1}$, whose eigenvalues are $1/(\lambda_j - \mu)$. Since $|\lambda_i - \mu|$ is small, the gap between it and the other values $|\lambda_j - \mu|$ is large, and the power method converges rapidly.

Corollary 26.6. *Let $\mu \in \mathbb{R}$ and $i \in [n]$ satisfy $|\lambda_i - \mu| < |\lambda_j - \mu|$ for all $j \neq i$, with μ not an eigenvalue of A, and let $\langle v^0, q_i\rangle \neq 0$. If $v^{t+1} = (A - \mu I)^{-1}v^t$, possibly rescaled, then the direction of v^t converges to that of q_i, and*

$$\tan\theta(v^t, q_i) \leq \Big(\frac{|\lambda_i - \mu|}{\min_{j\neq i}|\lambda_j - \mu|}\Big)^t \tan\theta(v^0, q_i).$$

Proof. Apply Proposition 26.4 to $B = (A - \mu I)^{-1}$, whose eigenvalue of largest magnitude, $1/(\lambda_i - \mu)$, is strictly the largest. □

Shifted inverse iteration is widely used, and it is numerically stable as well; see [TB97, GVL13].

26.4 Rayleigh quotient iteration

To get an estimate μ of an eigenvalue, we use the inverse iteration itself, together with the Rayleigh quotient (see also Lecture 8).

Definition 26.7. The *Rayleigh quotient* of a nonzero vector $x \in \mathbb{R}^n$ is $R(x) := \dfrac{x^\top A x}{x^\top x}$.

Lemma 26.8. *For every nonzero $x \in \mathbb{R}^n$, the function $\lambda \mapsto \|(A - \lambda I)x\|_2$ is minimized exactly at $\lambda = R(x)$.*

Proof. We have $\|(A - \lambda I)x\|_2^2 = \|Ax\|_2^2 - 2\lambda\, x^\top A x + \lambda^2\, x^\top x$, a quadratic function of λ with positive leading coefficient, minimized exactly at $\lambda = x^\top A x / x^\top x$. □

So if x were an eigenvector, $R(x)$ would be its eigenvalue and the norm in Lemma 26.8 would be zero. If x is not an eigenvector, the norm may be nonzero, but we can still think of $R(x)$ as an extension of the eigenvalue to all nonzero x: it gives a way to get a scalar from x.

Algorithm 26.9 (Rayleigh quotient iteration)**.** Let $v^{(0)}$ be a random unit vector and $\lambda^{(0)} = R(v^{(0)})$. For $k = 1, 2, \ldots$, repeat:

1) solve $(A - \lambda^{(k-1)} I)\, x = v^{(k-1)}$ for x (a step of inverse iteration with shift $\lambda^{(k-1)}$);

2) normalize: $v^{(k)} = x / \|x\|_2$;

3) set $\lambda^{(k)} = R(v^{(k)})$.

If $\lambda^{(k-1)}$ is exactly an eigenvalue, the system in step 1 is singular, and we stop with this eigenvalue.

Fact 26.10 ([TB97, Lecture 27], [Par98])**.** *For symmetric A, Rayleigh quotient iteration converges to an eigenpair for all starting vectors except a set of measure zero. When it converges, the convergence is ultimately cubic: if λ_J is a simple eigenvalue and $v^{(0)}$ is sufficiently close to q_J, then $\min_{s=\pm 1}\|v^{(k+1)} - s q_J\|_2 = O(\min_{s=\pm 1}\|v^{(k)} - s q_J\|_2^3)$ and $|\lambda^{(k+1)} - \lambda_J| = O(|\lambda^{(k)} - \lambda_J|^3)$.*

We do not prove Fact 26.10; cubic convergence means that the error drops from ε to about ε^3 at every step. The following lemma makes one step of the heuristic argument below precise.

Lemma 26.11. *Let q^* be a unit eigenvector of A with eigenvalue λ^*. Then every unit vector x satisfies*

$$|R(x) - \lambda^*| \le \|A - \lambda^* I\|_2\, \|x - q^*\|_2^2.$$

Proof. Let $B = A - \lambda^* I$, which is symmetric with $Bq^* = 0$. Expanding, $(x - q^*)^\top B (x - q^*) = x^\top B x - 2(q^*)^\top B x + (q^*)^\top B q^* = x^\top B x$, because $(q^*)^\top B = (Bq^*)^\top = 0$. Since $\|x\|_2 = 1$, we get $R(x) - \lambda^* = x^\top B x = (x - q^*)^\top B (x - q^*)$, whose absolute value is at most $\|B\|_2 \|x - q^*\|_2^2$. □

Remark 26.12 (A heuristic estimate). Here is a heuristic argument for cubic convergence. Suppose that we are at $v^{(k)}$, that q^* is a unit eigenvector, and that $\|v^{(k)} - q^*\|_2 \le \varepsilon$.

1) R is a smooth function on the unit sphere S^{n-1}, and a calculation gives $\nabla R(x) = \frac{2}{x^\top x}(Ax - R(x)x)$. So $\nabla R(q^*) = 0$, which suggests $R(x) - R(q^*) = O(\|x - q^*\|_2^2)$; Lemma 26.11 shows this precisely.

2) Let λ^* be the corresponding eigenvalue, and assume that it is simple (no repeated eigenvalues here). Then $|\lambda^{(k)} - \lambda^*| = O(\varepsilon^2)$.

3) Then $\|v^{(k+1)} - q^*\|_2 \le O(|\lambda^{(k)} - \lambda^*| \cdot \|v^{(k)} - q^*\|_2) \le O(\varepsilon^2 \cdot \varepsilon) = O(\varepsilon^3)$, up to the sign of q^*, because multiplying by $(A - \lambda^{(k)} I)^{-1}$ scales the component along q^* up by $1/|\lambda^{(k)} - \lambda^*|$, much more than the other components, and rescaling therefore shrinks the other components by a factor $O(|\lambda^{(k)} - \lambda^*|)$. Corollary 26.6 with $t = 1$ and $\mu = \lambda^{(k)}$ states this step precisely for tangents of angles.

26.5 The QR factorization and similar matrices

All these methods compute one eigenpair. To compute the entire decomposition, we use a generalization of the same idea, but first we recall another tool, the QR factorization. Matrix factorizations, and the QR algorithm of Section 26.7, appear on a well-known list of the top ten algorithms of the twentieth century [DS00].

Definition 26.13. A *QR factorization* of a matrix $A \in \mathbb{R}^{n \times r}$, with $n \ge r$, is a factorization $A = QR$ in which $Q \in \mathbb{R}^{n \times r}$ has orthonormal columns $q_1, \dots, q_r$ and $R = (r_{ji}) \in \mathbb{R}^{r \times r}$ is upper triangular:

$$\begin{pmatrix} | & & | \\ A_1 & \cdots & A_r \\ | & & | \end{pmatrix} = \begin{pmatrix} | & | & & | \\ q_1 & q_2 & \cdots & q_r \\ | & | & & | \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1r} \\ & r_{22} & \cdots & r_{2r} \\ & & \ddots & \vdots \\ 0 & & & r_{rr} \end{pmatrix}.$$

So the column $A_i = \sum_{j \le i} r_{ji} q_j$ is written as a combination of the first i columns of Q.

Fact 26.14. *Every $A \in \mathbb{R}^{n \times r}$ with $n \ge r$ has a QR factorization. If the columns of A are linearly independent, there is exactly one QR factorization in which R has positive diagonal entries.*

The well-known way to compute it is Gram–Schmidt orthogonalization, for a matrix with linearly independent columns: for $i = 1, \dots, r$, let

$$q_i \coloneqq A_i - \sum_{j<i} \langle A_i, q_j \rangle q_j, \quad \text{renormalized to be a unit vector.}$$

Then the vectors q_i are even orthonormal, and $r_{ji} = \langle A_i, q_j \rangle$ for $j < i$, while $r_{ii} > 0$ is the norm before renormalization. People do not use it in practice because of numerical issues; they use modified versions, or Householder triangularization (see the books [TB97, GVL13]).

One other fact is very useful.

Lemma 26.15. *Let $A \in \mathbb{R}^{n \times n}$ and let $S \in \mathbb{R}^{n \times n}$ be invertible. Then A and $S^{-1}AS$ have the same eigenvalues.*

Proof. For every x and λ, $Ax = \lambda x$ holds if and only if $(S^{-1}AS)(S^{-1}x) = \lambda(S^{-1}x)$, and $x \ne 0$ if and only if $S^{-1}x \ne 0$. □

The matrices A and $S^{-1}AS$ are called *similar*. So it is fine to replace A by a similar matrix when computing eigenvalues; if S is orthogonal, the matrix $S^{-1}AS = S^\top AS$ is moreover symmetric.

26.6 Orthogonal iteration

Back to computing several eigenvalues and eigenvectors at once, say r of them.

Algorithm 26.16 (Orthogonal iteration)**.** Let $Q^{(0)} \in \mathbb{R}^{n\times r}$ have orthonormal columns. For $k = 1, 2, \dots$, set $Z = AQ^{(k-1)}$ and compute a QR factorization $Q^{(k)}R^{(k)} = Z$.

If $r = 1$, this is power iteration with renormalization. In fact, consider the first columns $Q^{(0)}e_1, Q^{(1)}e_1, \dots, Q^{(k)}e_1$, even when $r > 1$: since $R^{(k)}$ is upper triangular, $AQ^{(k-1)}e_1 = Ze_1 = Q^{(k)}R^{(k)}e_1 = r_{11}^{(k)}Q^{(k)}e_1$, so this sequence is a run of the power method on $Q^{(0)}e_1$. The second column $q_2^{(k)}$ is the vector from which we remove the component along $q_1^{(k)}$ each time, the third column has its components along $q_1^{(k)}$ and $q_2^{(k)}$ removed, and so on. Subtracting out the higher eigenvectors prevents all columns from converging to q_1, the top eigenvector.

Theorem 26.17 ([TB97, Lecture 28])**.** *Assume that $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_r| > |\lambda_{r+1}|$ (the last condition is void if $r = n$, in which case assume $|\lambda_n| > 0$), and that every leading principal minor of the $r \times r$ matrix $\hat{Q}^\top Q^{(0)}$ is nonzero, where $\hat{Q} = (q_1 \cdots q_r)$. Then orthogonal iteration converges to the top r eigenvectors: for $j = 1, \dots, r$, the j-th column $q_j^{(k)}$ of $Q^{(k)}$ satisfies $\min_{s=\pm 1}\|q_j^{(k)} - s\,q_j\|_2 \to 0$ as $k \to \infty$. More precisely, this distance is $O(C^k)$, where $C = \max_{1\le j\le r}|\lambda_{j+1}|/|\lambda_j| < 1$ (with $\lambda_{n+1} := 0$).*

We state Theorem 26.17 without proof. The condition on $Q^{(0)}$ holds with probability 1 for a random start; without it, the conclusion can fail, for instance if the first column of $Q^{(0)}$ is orthogonal to q_1, as Proposition 26.4 suggests.

26.7 The QR algorithm

Now take $r = n$ and $Q^{(0)} = I$ in orthogonal iteration. To compare two procedures, we rename the matrices of orthogonal iteration: start with $\underline{Q}^{(0)} = I$, and for $k \ge 1$ set $Z = A\underline{Q}^{(k-1)}$ and factor $\underline{Q}^{(k)}\underline{R}^{(k)} = Z$. This procedure can be rewritten as the *QR algorithm*, as opposed to the QR factorization.

Algorithm 26.18 (QR algorithm)**.** Set $A^{(0)} = A$. For $k \ge 1$, compute a QR factorization $Q^{(k)}R^{(k)} = A^{(k-1)}$ and set $A^{(k)} = R^{(k)}Q^{(k)}$.

So we take the QR factorization of $A^{(k-1)}$, flip the two factors, and multiply them. Two questions are obvious. The first is whether the matrix $A^{(k)}$ is even similar to A; it is.

Lemma 26.19. *For every $k \ge 1$, $A^{(k)} = (Q^{(k)})^\top A^{(k-1)}Q^{(k)}$. Hence $A^{(k)}$ is symmetric and similar to A, and has the same eigenvalues.*

Proof. Since $Q^{(k)}$ is orthogonal,

$$A^{(k)} = R^{(k)}Q^{(k)} = (Q^{(k)})^{-1}Q^{(k)}R^{(k)}Q^{(k)} = (Q^{(k)})^\top A^{(k-1)}Q^{(k)},$$

so $A^{(k)}$ is similar to $A^{(k-1)}$, and by induction to $A^{(0)} = A$; Lemma 26.15 gives the eigenvalues. □

The second is why the QR algorithm computes the eigenvalues, and why the products $\overline{Q}^{(k)} := Q^{(1)}Q^{(2)}\cdots Q^{(k)}$ (with $\overline{Q}^{(0)} = I$) converge to the eigenvectors. The reason is that the QR algorithm is orthogonal iteration in disguise.

Theorem 26.20 (Equivalence of orthogonal iteration and the QR algorithm)**.** *Assume that A is nonsingular and that every QR factorization in both procedures is normalized so that its upper triangular factor has positive diagonal entries. Then for every $k \geq 0$,*

$$\underline{Q}^{(k)} = \overline{Q}^{(k)} \quad \text{and} \quad A^{(k)} = (\underline{Q}^{(k)})^{\top} A \, \underline{Q}^{(k)},$$

and $\underline{R}^{(k)} = R^{(k)}$ for every $k \geq 1$.

Proof. We use induction on k. For $k = 0$, both $\underline{Q}^{(0)}$ and $\overline{Q}^{(0)}$ equal I, and $A^{(0)} = A$. Let $k \geq 1$ and assume the claims for $k-1$. Since $\overline{Q}^{(k-1)}$ is orthogonal, the induction hypothesis gives $A\overline{Q}^{(k-1)} = \overline{Q}^{(k-1)} A^{(k-1)}$, and therefore

$$A\underline{Q}^{(k-1)} = \overline{Q}^{(k-1)} A^{(k-1)} = \overline{Q}^{(k-1)} Q^{(k)} R^{(k)} = \overline{Q}^{(k)} R^{(k)}.$$

This is a QR factorization of $Z = A\underline{Q}^{(k-1)}$ with positive diagonal, since $\overline{Q}^{(k)}$ is orthogonal. The matrix Z is nonsingular, so by Fact 26.14 this factorization is the only one with positive diagonal; hence $\underline{Q}^{(k)} = \overline{Q}^{(k)}$ and $\underline{R}^{(k)} = R^{(k)}$. Finally, by Lemma 26.19 and the induction hypothesis, $A^{(k)} = (Q^{(k)})^{\top} (\overline{Q}^{(k-1)})^{\top} A \, \overline{Q}^{(k-1)} Q^{(k)} = (\overline{Q}^{(k)})^{\top} A \, \overline{Q}^{(k)}$. (The factorizations in Algorithm 26.18 are unique too, because every $A^{(k-1)}$ is similar to A and hence nonsingular.) □

Corollary 26.21. *Assume that $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0$, that every leading principal minor of the eigenvector matrix $Q^{\top} = (q_1 \cdots q_n)^{\top}$ is nonzero, and that the QR factorizations are normalized as in Theorem 26.20. Then the columns of $\overline{Q}^{(k)}$ converge to $q_1, \ldots, q_n$ up to sign, and $A^{(k)}$ converges to $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$.*

Proof. By Theorem 26.20, $\overline{Q}^{(k)} = \underline{Q}^{(k)}$ is orthogonal iteration with $r = n$ and $Q^{(0)} = I$, so Theorem 26.17 applies, with $\hat{Q}^{\top} Q^{(0)} = Q^{\top}$: there are diagonal sign matrices D_k with $\overline{Q}^{(k)} - QD_k \to 0$. Since $D_k Q^{\top} A Q D_k = D_k \Lambda D_k = \Lambda$ and all these matrices are bounded, $A^{(k)} = (\overline{Q}^{(k)})^{\top} A \, \overline{Q}^{(k)}$ converges to Λ. □

The QR algorithm (John Francis, 1961 [Fra61]) is called by Trefethen and Bau [TB97] one of the jewels of numerical analysis, and it was nominated as one of the top ten algorithms of the twentieth century [DS00]. For more on why it works, see Watkins [Wat82]. To get good guarantees, however, one needs other ideas, such as shifts, and Householder reflections instead of Gram–Schmidt; without them the algorithm can be slow, or may not converge at all.

Example 26.22. For $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, the QR factorization with positive diagonal is $Q^{(1)} = A$, $R^{(1)} = I$, so $A^{(1)} = R^{(1)} Q^{(1)} = A$, and the QR algorithm never moves, although the eigenvalues are ± 1. Here $|\lambda_1| = |\lambda_2|$, so Corollary 26.21 does not apply.

Still, the QR algorithm gives good results in many cases, and an eigenvalue algorithm does not get simpler than this to state. Much more, with details, is in the books [TB97, GVL13]. We have given few proofs of convergence, and almost no discussion of numerical stability or of bit precision; these raise many interesting and deep questions, for which Higham [Hig02] is a standard reference.

References

[ABKPM09] Eric Allender, Peter Bürgisser, Johan Kjeldgaard-Pedersen, and Peter Bro Miltersen. On the complexity of numerical analysis. *SIAM Journal on Computing*, 38(5):1987–2006, 2009.

[ABS15] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. *Journal of the ACM*, 62(5):1–25, 2015. Preliminary version in FOCS 2010.

[AGK+04] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. *SIAM Journal on Computing*, 33(3):544–562, 2004.

[AK00] Paola Alimonti and Viggo Kann. Some APX-completeness results for cubic graphs. *Theoretical Computer Science*, 237(1-2):123–134, 2000.

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[ALN08] Sanjeev Arora, James R. Lee, and Assaf Naor. Euclidean distortion and the sparsest cut. *Journal of the American Mathematical Society*, 21(1):1–21, 2008.

[AM85] N. Alon and V. D. Milman. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory, Series B*, 38(1):73–88, 1985.

[AP90] Baruch Awerbuch and David Peleg. Sparse partitions. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 503–513, 1990.

[Aro98] Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.

[ARV09] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM*, 56(2):5:1–5:37, 2009.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[AS04] A. A. Ageev and M. I. Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8(3):307–328, 2004.

[Bar68] Erwin H. Bareiss. Sylvester's identity and multistep integer-preserving Gaussian elimination. *Mathematics of Computation*, 22(103):565–578, 1968.

[Bar96] Yair Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 184–193, 1996.

[Bar00] Sven G. Bartels. The complexity of Yamnitsky and Levin's simplices algorithm. In Gil Kalai and Günter M. Ziegler, editors, *Polytopes – Combinatorics and Computation*, volume 29 of *DMV Seminar*, pages 199–225. Birkhäuser, Basel, 2000.

[BF81] József Beck and Tibor Fiala. "Integer-Making" theorems. *Discrete Applied Mathematics*, 3(1):1–8, 1981.

[Blö91] Johannes Blömer. Computing sums of radicals in polynomial time. In *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 670–677. IEEE Computer Society Press, 1991.

[Bor75] Christer Borell. The Brunn–Minkowski inequality in Gauss space. *Inventiones Mathematicae*, 30(2):207–216, 1975.

[Bor85] Christer Borell. Geometric bounds on the Ornstein–Uhlenbeck velocity process. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 70(1):1–13, 1985.

[Bou85] J. Bourgain. On Lipschitz embedding of finite metric spaces in Hilbert space. *Israel Journal of Mathematics*, 52(1-2):46–52, 1985.

[BS16] Boaz Barak and David Steurer. Proofs, beliefs, and algorithms through the lens of sum-of-squares. Lecture notes, https://www.sumofsquares.org, 2016.

[Buk16] Boris Bukh. An improvement of the Beck–Fiala theorem. *Combinatorics, Probability and Computing*, 25(3):380–398, 2016.

[BYE81] R. Bar-Yehuda and S. Even. A linear-time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms*, 2(2):198–203, 1981.

[Che70] Jeff Cheeger. A lower bound for the smallest eigenvalue of the Laplacian. In *Problems in Analysis: A Symposium in Honor of Salomon Bochner*, pages 195–200. Princeton University Press, 1970.

[Chv79] V. Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

[CKR05] Gruia Călinescu, Howard Karloff, and Yuval Rabani. Approximation algorithms for the 0-extension problem. *SIAM Journal on Computing*, 34(2):358–372, 2005.

[CMM06] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for unique games. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 205–214, 2006.

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 151–158, 1971.

[CS03] Fabián A. Chudak and David B. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM Journal on Computing*, 33(1):1–25, 2003.

[Din07] Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM*, 54(3):12, 2007.

[DKK+18a] Irit Dinur, Subhash Khot, Guy Kindler, Dor Minzer, and Muli Safra. On non-optimally expanding sets in Grassmann graphs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 940–951, 2018.

[DKK+18b] Irit Dinur, Subhash Khot, Guy Kindler, Dor Minzer, and Muli Safra. Towards a proof of the 2-to-1 games conjecture? In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 376–389, 2018.

[DS00] Jack Dongarra and Francis Sullivan. Guest editors' introduction to the top 10 algorithms. *Computing in Science & Engineering*, 2(1):22–23, 2000.

[DS05] Irit Dinur and Samuel Safra. On the hardness of approximating vertex cover. *Annals of Mathematics*, 162(1):439–485, 2005.

[DS14] Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC)*, pages 624–633, 2014.

[Edm67] Jack Edmonds. Systems of distinct representatives and linear algebra. *Journal of Research of the National Bureau of Standards, Section B: Mathematics and Mathematical Physics*, 71B(4):241–245, 1967.

[EK72] Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19(2):248–264, 1972.

[Fei98] Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[FF56] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[Fie73] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298–305, 1973.

[FKL02] Uriel Feige, Marek Karpinski, and Michael Langberg. Improved approximation of Max-Cut on graphs of bounded degree. *Journal of Algorithms*, 43(2):201–219, 2002.

[Fra61] J. G. F. Francis. The QR transformation: A unitary analogue to the LR transformation—part 1. *The Computer Journal*, 4(3):265–271, 1961.

[FRT04] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer and System Sciences*, 69(3):485–497, 2004.

[FS02] Uriel Feige and Gideon Schechtman. On the optimality of the random hyperplane rounding technique for MAX CUT. *Random Structures & Algorithms*, 20(3):403–440, 2002.

[GG61] P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6):849–859, 1961.

[GG81] Ofer Gabber and Zvi Galil. Explicit constructions of linear-sized superconcentrators. *Journal of Computer and System Sciences*, 22(3):407–420, 1981.

[GJS76] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.

[GK99] Sudipto Guha and Samir Khuller. Greedy strikes back: Improved facility location algorithms. *Journal of Algorithms*, 31(1):228–248, 1999.

[GLLRK79] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. In *Discrete Optimization II*, volume 5 of *Annals of Discrete Mathematics*, pages 287–326. Elsevier, 1979.

[GLS88] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, Berlin, Heidelberg, 1988.

[GO05] Venkatesan Guruswami and Ryan O'Donnell. CSE 533: The PCP theorem and hardness of approximation, lecture notes. https://courses.cs.washington.edu/courses/cse533/05au/, 2005.

[Grü60] Branko Grünbaum. Partitions of mass-distributions and of convex bodies by hyperplanes. *Pacific Journal of Mathematics*, 10(4):1257–1261, 1960.

[GT08] Anupam Gupta and Kanat Tangwongsan. Simpler analyses of local search algorithms for facility location. arXiv:0809.2554, 2008.

[GVL13] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, fourth edition, 2013.

[GW95] 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.

[Hås99] Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182(1):105–142, 1999.

[Hås01] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[Hig02] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2002.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[HSSW97] Leslie A. Hall, Andreas S. Schulz, David B. Shmoys, and Joel Wein. Scheduling to minimize average completion time: Off-line and on-line approximation algorithms. *Mathematics of Operations Research*, 22(3):513–544, 1997.

[Hu63] T. C. Hu. Multi-commodity network flows. *Operations Research*, 11(3):344–360, 1963.

[Jai01] Kamal Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[Joh74] David S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256–278, 1974.

[JV01] Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM*, 48(2):274–296, 2001.

[Kar72] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer US, 1972.

[Kar99] Howard Karloff. How good is the Goemans–Williamson MAX CUT algorithm? *SIAM Journal on Computing*, 29(1):336–350, 1999.

[Kha80] Leonid G. Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72, 1980.

[Kho02] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 767–775, 2002.

[KKMO07] 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.

[KLL+13] Tsz Chiu Kwok, Lap Chi Lau, Yin Tat Lee, Shayan Oveis Gharan, and Luca Trevisan. Improved Cheeger's inequality: Analysis of spectral partitioning algorithms through higher order spectral gap. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, pages 11–20, 2013.

[KMN11] Anna R. Karlin, Claire Mathieu, and C. Thach Nguyen. Integrality gaps of linear and semi-definite programming relaxations for Knapsack. In Oktay Günlük and Gerhard J. Woeginger, editors, *Integer Programming and Combinatorial Optimization (IPCO 2011)*, Lecture Notes in Computer Science, pages 301–314. Springer, 2011.

[KMR17] Pravesh K. Kothari, Raghu Meka, and Prasad Raghavendra. Approximating rectangles by juntas and weakly-exponential lower bounds for LP relaxations of CSPs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 590–603, 2017.

[KMS17] Subhash Khot, Dor Minzer, and Muli Safra. On independent sets, 2-to-2 games, and Grassmann graphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 576–589, 2017.

[KMS18] Subhash Khot, Dor Minzer, and Muli Safra. Pseudorandom sets in Grassmann graph have near-perfect expansion. In *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 592–601, 2018.

[KR08] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within $2-\varepsilon$. *Journal of Computer and System Sciences*, 74(3):335–349, 2008.

[KSW99] David Karger, Cliff Stein, and Joel Wein. Scheduling algorithms. In Mikhail J. Atallah, editor, *Algorithms and Theory of Computation Handbook*. CRC Press, Boca Raton, FL, 1999.

[Las01] Jean B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[LG18] László Lovász and Péter Gács. Gaussian elimination without rounding. Lecture notes, from L. Lovász's lecture notes, by P. Gács. http://www.cs.bu.edu/faculty/gacs/courses/cs530/lectures/exact-Gauss.pdf, 2018.

[Li13] Shi Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. *Information and Computation*, 222:45–58, 2013.

[LLR95] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

[LPS88] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[LR99] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, 1999.

[LRS11] Lap Chi Lau, R. Ravi, and Mohit Singh. *Iterative Methods in Combinatorial Optimization*. Cambridge University Press, 2011.

[LS93] Nathan Linial and Michael Saks. Low diameter graph decompositions. *Combinatorica*, 13(4):441–454, 1993.

[LST90] Jan Karel Lenstra, David B. Shmoys, and Éva Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming*, 46(1–3):259–271, 1990.

[LY94] Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.

[Mat13] Jiří Matoušek. Lecture notes on metric embeddings. https://kam.mff.cuni.cz/~matousek/ba-a4.pdf, 2013.

[MG07] Jiří Matoušek and Bernd Gärtner. *Understanding and Using Linear Programming*. Universitext. Springer, 2007.

[MOO10] Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences: Invariance and optimality. *Annals of Mathematics*, 171(1):295–341, 2010.

[Mos12] Dana Moshkovitz. The tale of the PCP theorem. *XRDS: Crossroads, The ACM Magazine for Students*, 18(3):23–26, 2012.

[MPX13] Gary L. Miller, Richard Peng, and Shen Chen Xu. Parallel graph decompositions using random shifts. In *Proceedings of the 25th Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 196–203, 2013.

[MS90] David W. Matula and Farhad Shahrokhi. Sparsest cuts and bottlenecks in graphs. *Discrete Applied Mathematics*, 27(1-2):113–123, 1990.

[New65] Donald J. Newman. Location of the maximum on unimodal surfaces. *Journal of the ACM*, 12(3):395–398, 1965.

[NY18] Assaf Naor and Robert Young. Vertical perimeter versus horizontal perimeter. *Annals of Mathematics*, 188(1):171–279, 2018.

[O'D07] Ryan O'Donnell. Lecture 21: Berry–Esseen theorem. Lecture notes, Analysis of Boolean Functions (CMU 18-859S, Spring 2007), scribed by Suresh Purini, https://www.cs.cmu.edu/~odonnell/boolean-analysis/lecture21.pdf, 2007.

[O'D14] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

[OW08] Ryan O'Donnell and Yi Wu. An optimal SDP algorithm for Max-Cut, and equally optimal Long Code tests. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 335–344, 2008.

[Par98] Beresford N. Parlett. *The Symmetric Eigenvalue Problem*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1998.

[Par03] Pablo A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96(2):293–320, 2003.

[PSW98] Cynthia Phillips, Clifford Stein, and Joel Wein. Minimizing average completion time in the presence of release dates. *Mathematical Programming*, 82(1–2):199–223, 1998.

[Que93] Maurice Queyranne. Structure of a simple scheduling polyhedron. *Mathematical Programming*, 58(1–3):263–285, 1993.

[Rag08] Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 245–254, 2008.

[Raz98] Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

[Rot16] Thomas Rothvoss. Lecture notes on the ARV algorithm for sparsest cut. arXiv:1607.00854, 2016.

[RVW02] Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders. *Annals of Mathematics*, 155(1):157–187, 2002.

[RW66] B. Rothschild and A. Whinston. On two commodity network flows. *Operations Research*, 14(3):377–387, 1966.

[Sch68] Linus Schrage. A proof of the optimality of the shortest remaining processing time discipline. *Operations Research*, 16(3):687–690, 1968.

[Sch86] Alexander Schrijver. *Theory of Linear and Integer Programming*. Wiley, New York, 1986.

[Sey80] P. D. Seymour. Four-terminus flows. *Networks*, 10(1):79–86, 1980.

[She99] W. F. Sheppard. On the application of the theory of error to cases of normal distribution and normal correlation. *Philosophical Transactions of the Royal Society of London. Series A*, 192:101–167, 1899.

[Sho70] Naum Z. Shor. Utilization of the operation of space dilatation in the minimization of convex functions. *Cybernetics*, 6(1):7–15, 1970.

[SM00] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[Sot15] José A. Soto. Improved analysis of a max-cut algorithm based on spectral partitioning. *SIAM Journal on Discrete Mathematics*, 29(1):259–268, 2015.

[ST78] V. N. Sudakov and B. S. Tsirel'son. Extremal properties of half-spaces for spherically invariant measures. *Journal of Soviet Mathematics*, 9(1):9–18, 1978.

[ST93] David B. Shmoys and Éva Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical Programming*, 62(1–3):461–474, 1993.

[Ste74] Gilbert Stengle. A nullstellensatz and a positivstellensatz in semialgebraic geometry. *Mathematische Annalen*, 207(2):87–97, 1974.

[TB97] Lloyd N. Trefethen and David Bau, III. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.

[Tre12] Luca Trevisan. Max cut and the smallest eigenvalue. *SIAM Journal on Computing*, 41(6):1769–1786, 2012.

[Tre16] Luca Trevisan. Lecture notes on graph partitioning, expanders and spectral methods. https://lucatrevisan.github.io/books/expanders-2016.pdf, 2016.

[TSSW00] Luca Trevisan, Gregory B. Sorkin, Madhu Sudan, and David P. Williamson. Gadgets, approximation, and linear programming. *SIAM Journal on Computing*, 29(6):2074–2097, 2000.

[Wat82] David S. Watkins. Understanding the QR algorithm. *SIAM Review*, 24(4):427–440, 1982.

[WS11] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.

[YL82] Boris Yamnitsky and Leonid A. Levin. An old linear programming algorithm runs in polynomial time. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 327–328, 1982.

[You16] Neal E. Young. Greedy set-cover algorithms. In *Encyclopedia of Algorithms*, pages 886–889. Springer New York, 2016.

[Zuc07] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(1):103–128, 2007.