April 23, 2025 - More SAGT

Other polynomials	Math
Hypergraphs	Algorithms
Directed Graphs	Practice

Other Polynomials

$\mathrm{Eigs}(M)$ = roots of characteristic polynomial

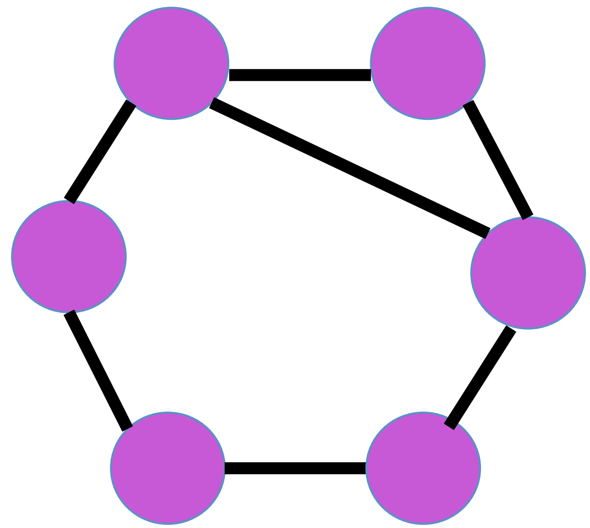
$= \mathrm{roots}(\det(xI - M))$

Other polynomials?

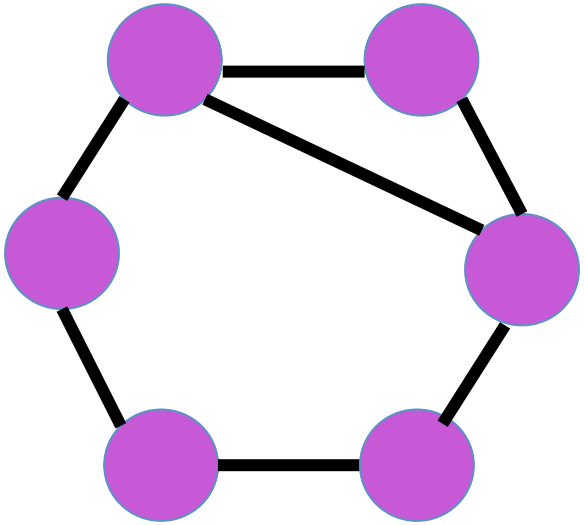
The matching polynomial (Heilmann-Leib '72)

$$\mu_G(x) = \sum_{i \geq 0} x^{n-2i} (-1)^i m_i$$

m_i = # matchings with i edges

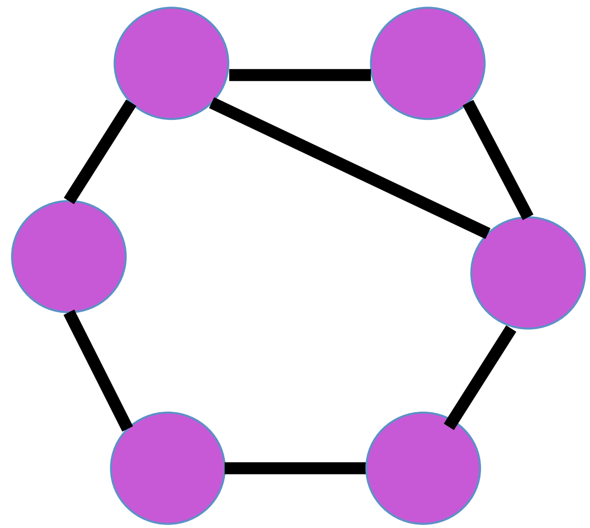


$$\mu_G(x) = x^6 - 7x^4 + 11x^2 - 2$$



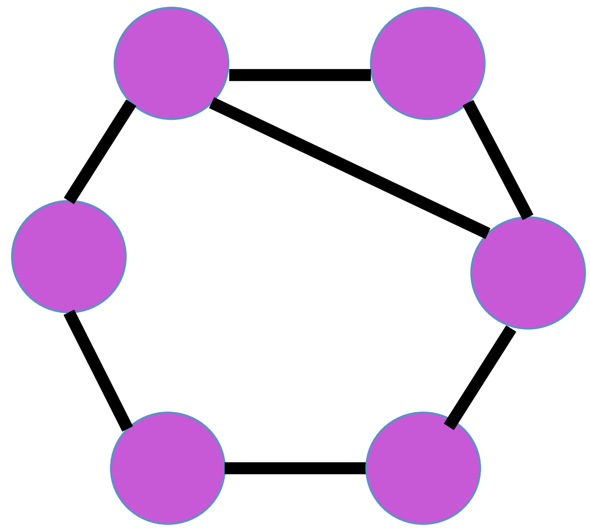
$$\mu_G(x) = x^6 - 7x^4 + 11x^2 - 2$$

one matching with 0 edges

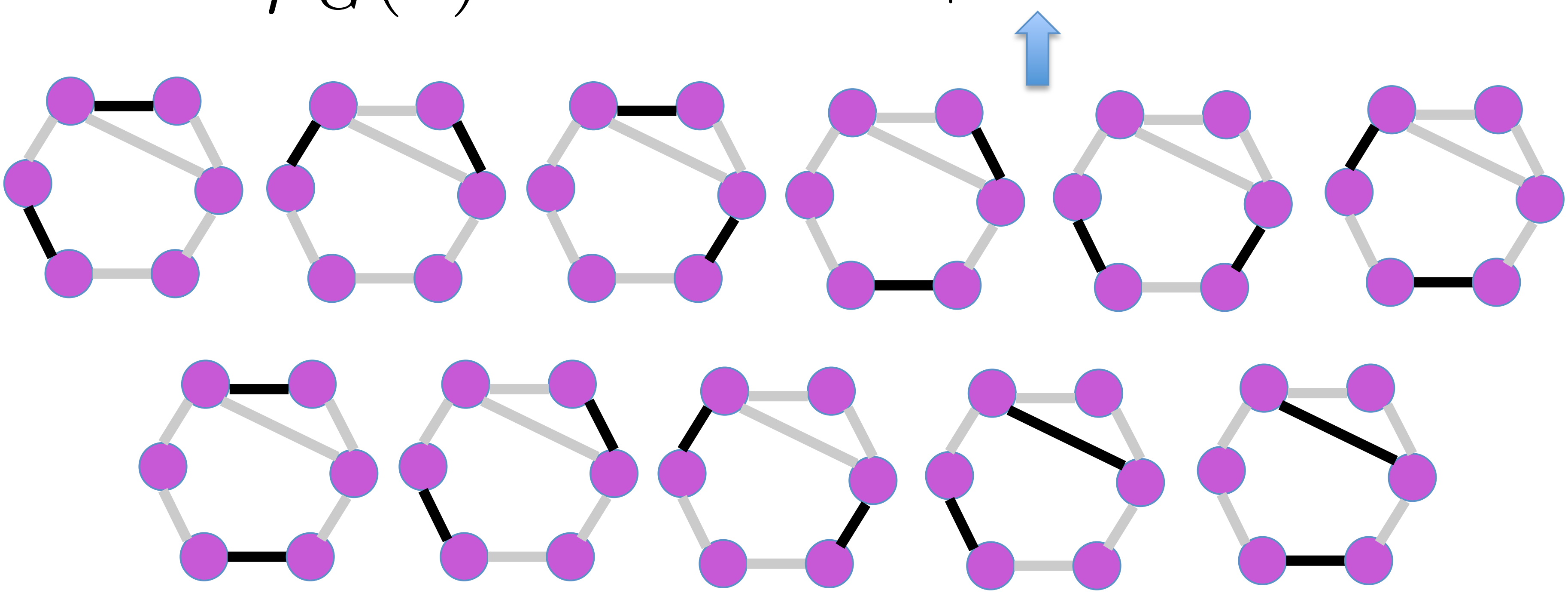


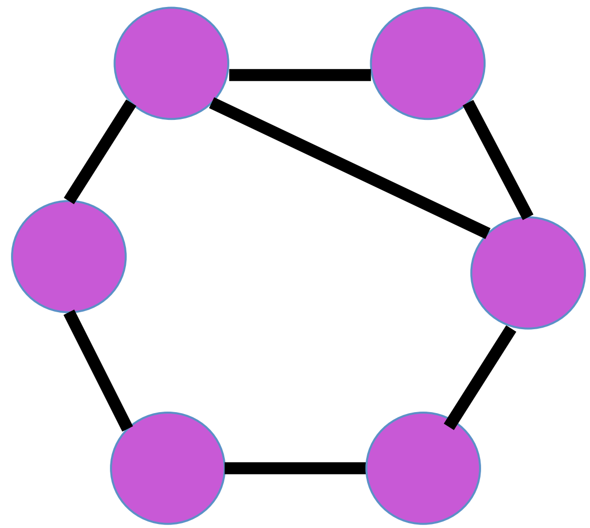
$$\mu_G(x) = x^6 - 7x^4 + 11x^2 - 2$$

7 matchings with 1 edge

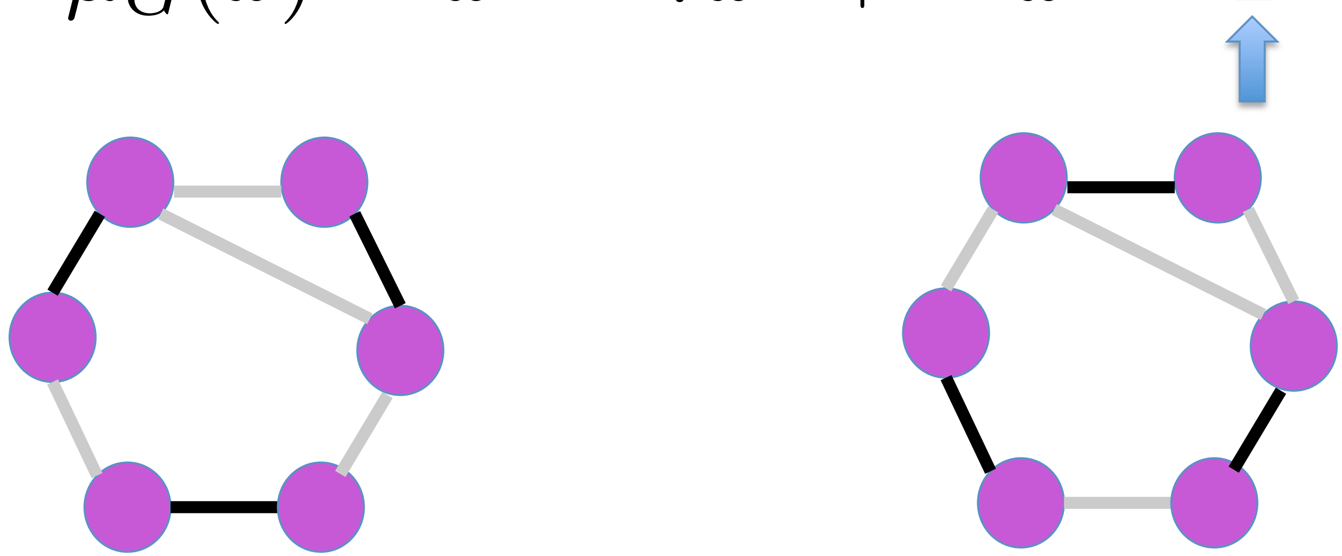


$$\mu_G(x) = x^6 - 7x^4 + 11x^2 - 2$$





$$\mu_G(x) = x^6 - 7x^4 + 11x^2 - 2$$



The matching polynomial (Heilmann-Lieb '72)

$$\mu_G(x) = \sum_{i \geq 0} x^{n-2i} (-1)^i m_i$$

Theorem (Heilmann-Lieb)
all the roots are real

The matching polynomial (Heilmann-Lieb '72)

$$\mu_G(x) = \sum_{i \geq 0} x^{n-2i} (-1)^i m_i$$

Theorem (Heilmann-Lieb)
all the roots are real
and have absolute value at most $2\sqrt{d-1}$

The matching polynomial (Heilmann-Lieb '72)

$$\mu_G(x) = \sum_{i \geq 0} x^{n-2i} (-1)^i m_i$$

Theorem (Heilmann-Lieb)
all the roots are real
and have absolute value at most $2\sqrt{d-1}$

G a tree ⟶ $\mu_G(x) = \det(xI - M)$

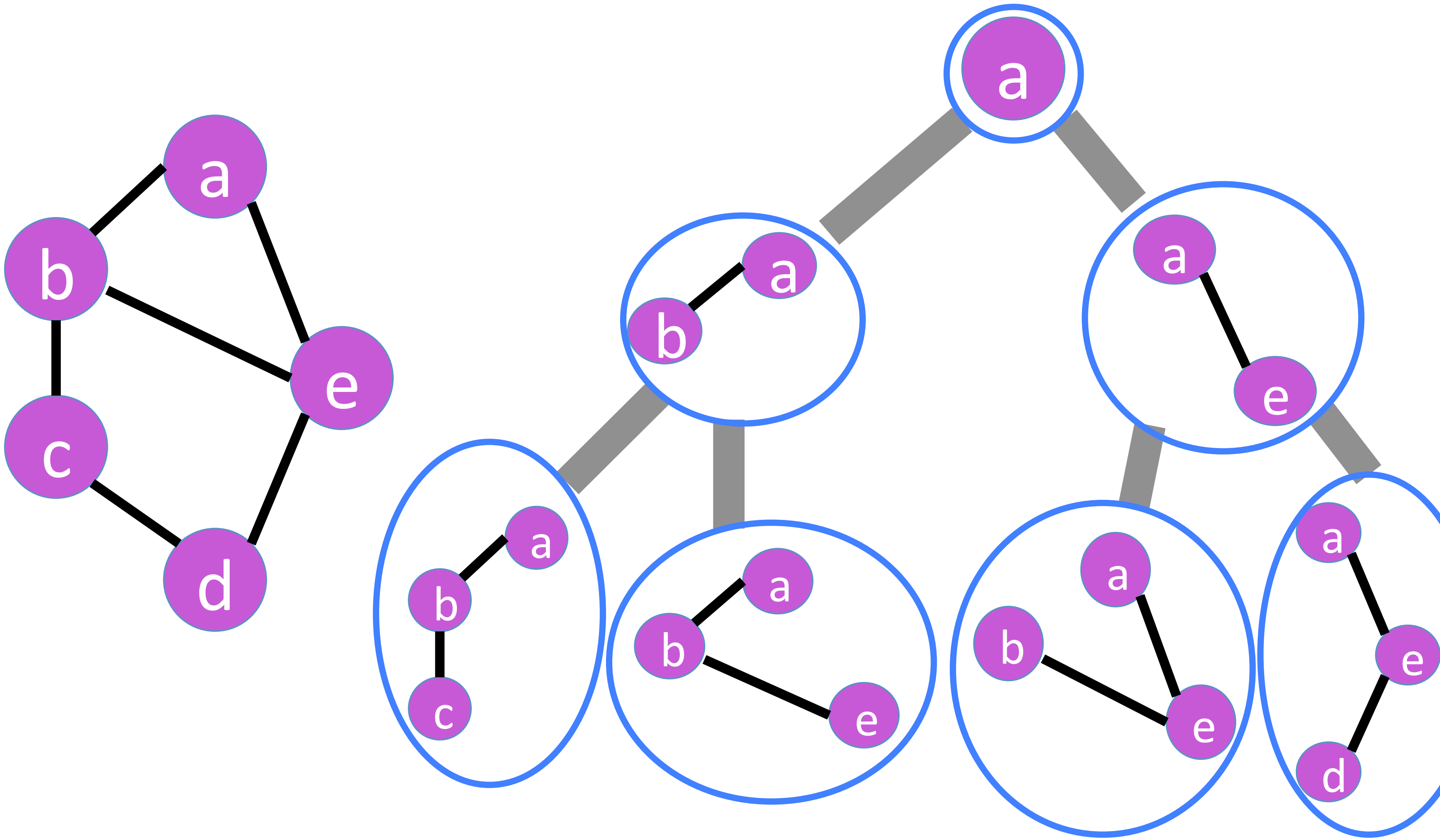
Godsil's Proof of Heilmann-Lieb

$T(G,v)$: the path tree of G at v

vertices are paths in G starting at v

edges to paths differing in one step

Godsil's Proof of Heilmann-Lieb



Godsil's Proof of Heilmann-Lieb

$T(G,v)$: the path tree of G at v

vertices are paths in G starting at v

edges to paths differing in one step

Theorem:

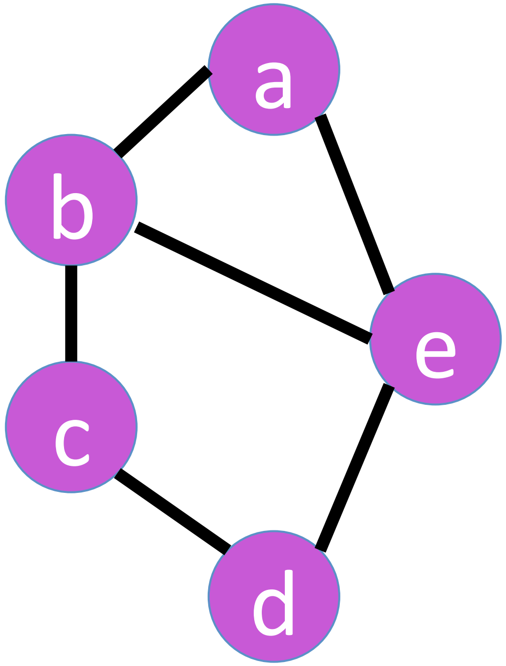
The matching polynomial divides the characteristic polynomial of $T(G,v)$

So, real rooted

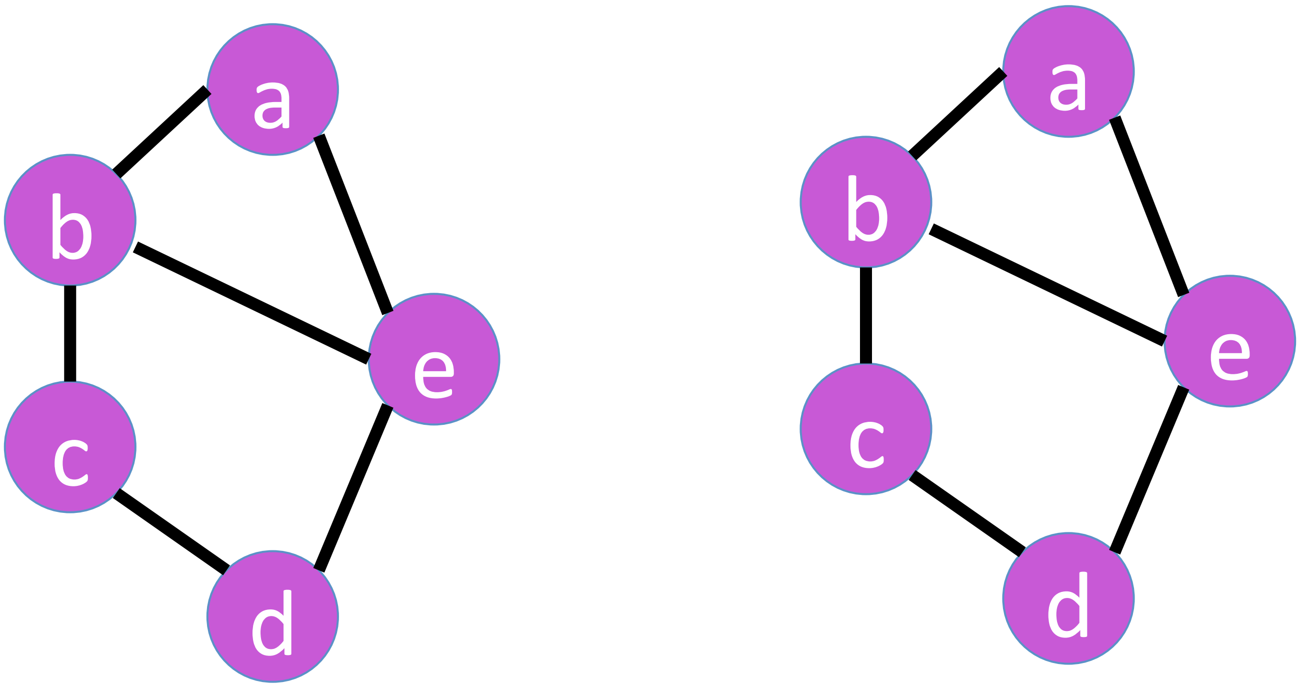
G a tree of max-degree $\leq d \rightarrow \|M\| \leq 2\sqrt{d-1}$

Proof that Ramanujan Graphs Exist

2-lifts of graphs

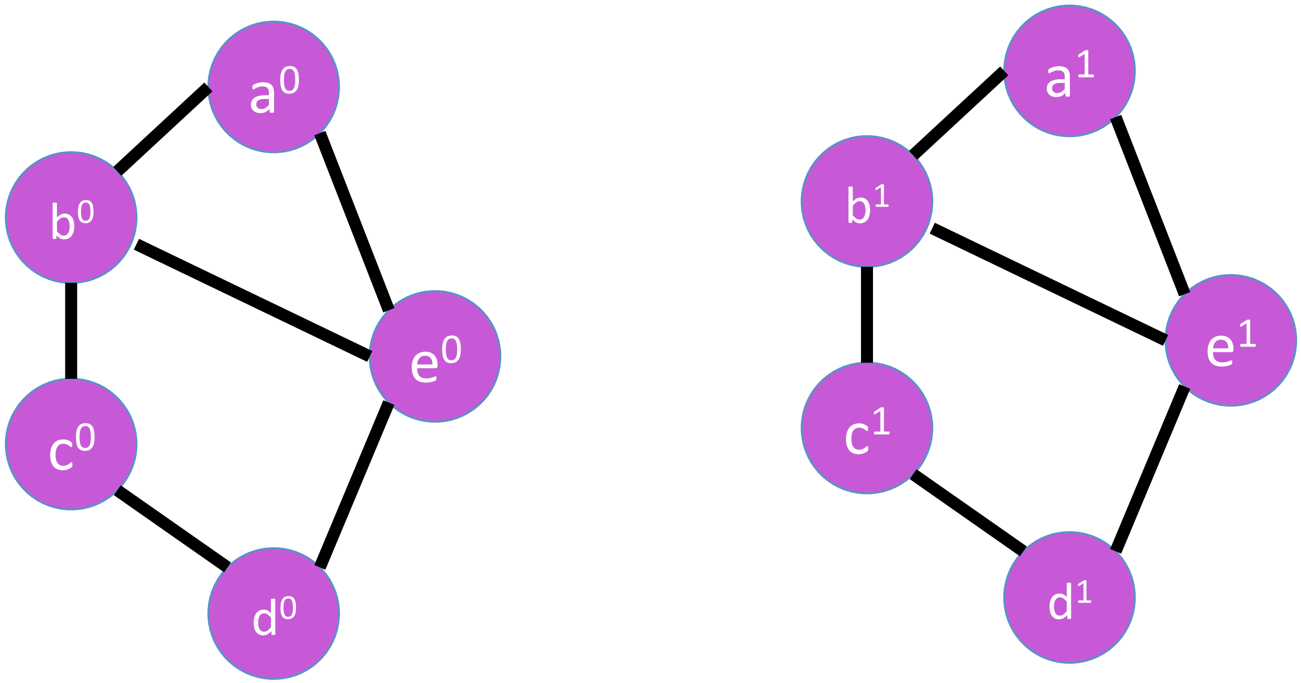


2-lifts of graphs



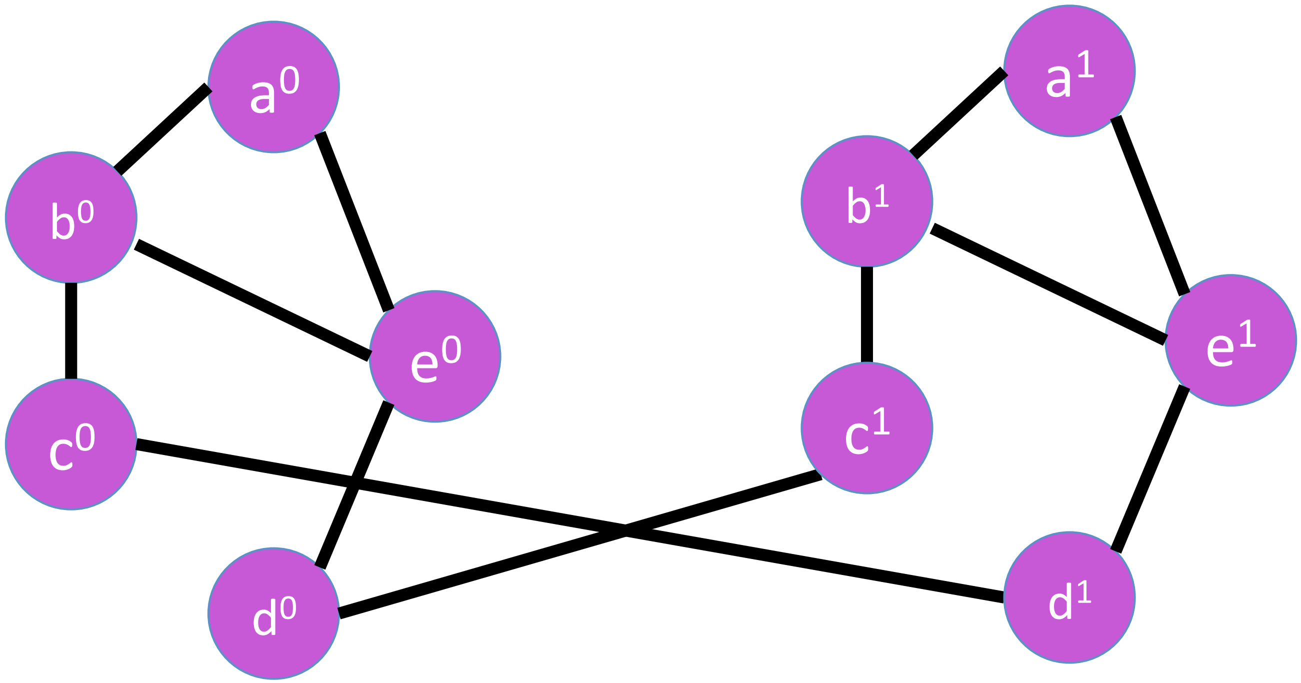
duplicate every vertex

2-lifts of graphs



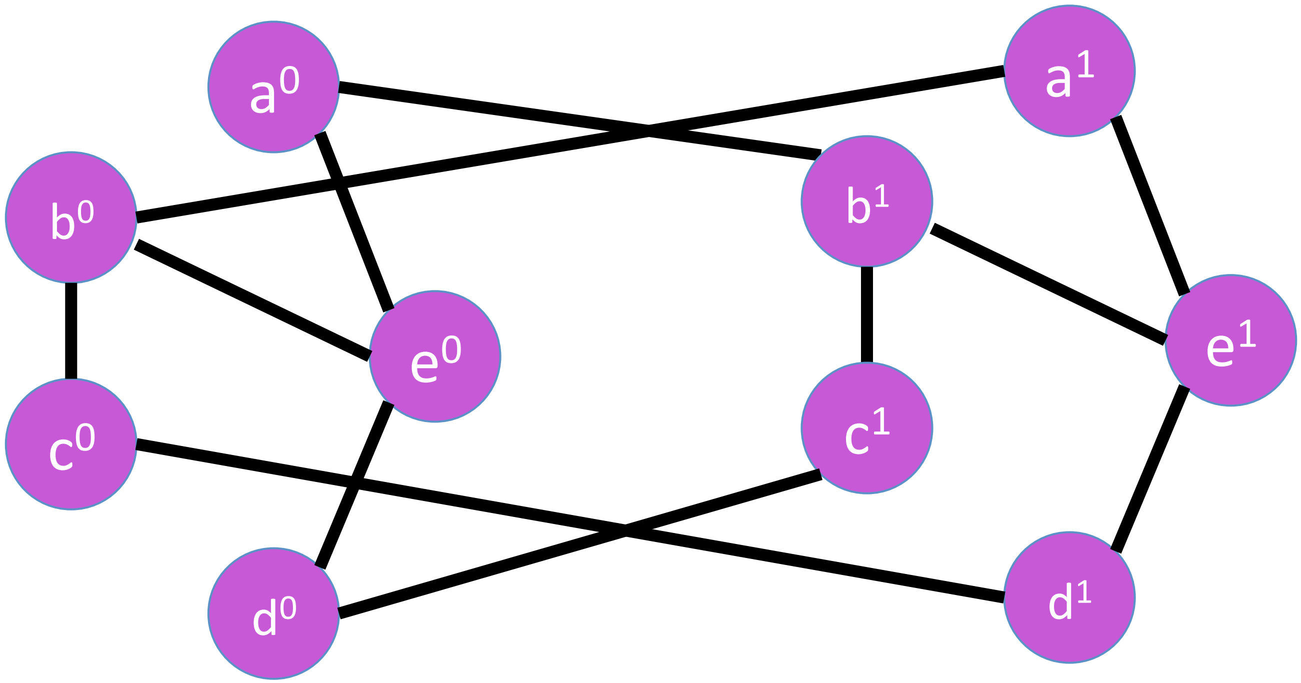
duplicate every vertex

2-lifts of graphs



for every pair of edges:
leave on either side (parallel),
or make both cross

2-lifts of graphs



for every pair of edges:
leave on either side (parallel),
or make both cross

2-lifts of graphs

0	1	0	0	1
1	0	1	0	1
0	1	0	1	0
0	0	1	0	1
1	1	0	1	0

2-lifts of graphs

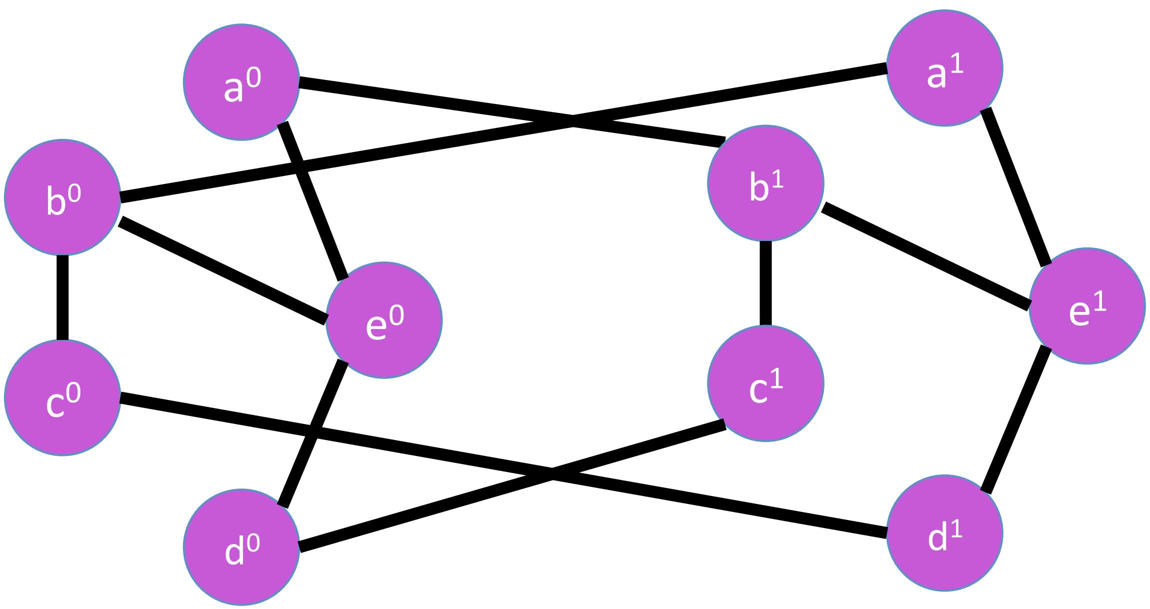
0	1	0	0	1	0	0	0	0	0
1	0	1	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	1	0	1	0	1
0	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	1	1	0	1	0

2-lifts of graphs

0	0	0	0	1	0	1	0	0	0
0	0	1	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0	1	0
0	0	0	0	1	0	0	1	0	0
1	1	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	1	0	0	0
0	0	1	0	0	0	0	0	0	1
0	0	0	0	0	1	1	0	1	0

Eigenvalues of 2-lifts (Bilu-Linial)

Given a 2-lift of G,
create a signed adjacency matrix A_s
with a -1 for crossing edges
and a 1 for parallel edges

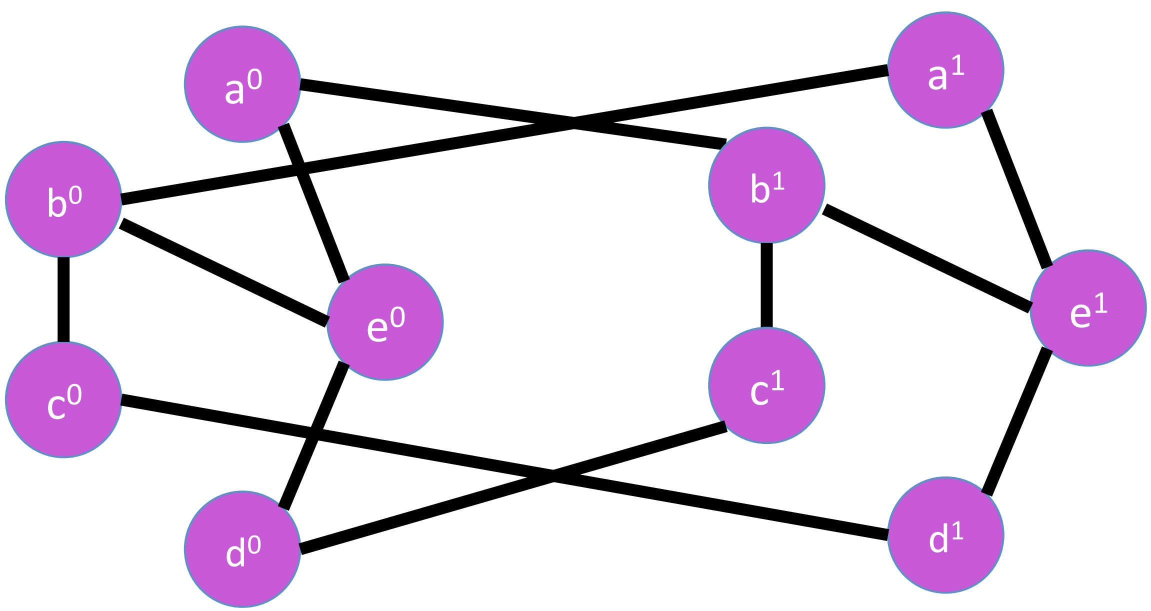


0	-1	0	0	1
-1	0	1	0	1
0	1	0	-1	0
0	0	-1	0	1
1	1	0	1	0

Eigenvalues of 2-lifts (Bilu-Linial)

Theorem:

The eigenvalues of the 2-lift are the union of the eigenvalues of A (old) and the eigenvalues of A_s (new)



0	-1	0	0	1
-1	0	1	0	1
0	1	0	-1	0
0	0	-1	0	1
1	1	0	1	0

Eigenvalues of 2-lifts (Bilu-Linial)

Theorem:
The eigenvalues of the 2-lift are the union of the eigenvalues of A (old) and the eigenvalues of A_s (new)

Conjecture:
Every d-regular graph has a 2-lift in which all the new eigenvalues have absolute value at most $2\sqrt{d-1}$

Eigenvalues of 2-lifts (Bilu-Linial)

Theorem:

Every d-regular bipartite graph has a 2-lift in which all the new eigenvalues have absolute value at most $2\sqrt{d-1}$

First idea: a random 2-lift

Specify a lift by $s \in \{\pm 1\}^m$

Pick s uniformly at random

But, this can fail

The expected polynomial

Consider $\mathop{\mathbb{E}}_{s}\left[\ \chi_{A_s}(x)\ \right]$

The expected polynomial

Theorem (Godsil-Gutman '81):

$$\mathop{\mathbb{E}}_{s} [\ \chi_{A_s}(x)\ ] = \mu_G(x)$$

the matching polynomial of G

The expected polynomial

Consider $\mathbb{E}_s\left[\ \chi_{A_s}(x)\ \right]$

Prove max-root $\left(\mathbb{E}_s\left[\ \chi_{A_s}(x)\ \right]\right) \leq 2\sqrt{d-1}$

Prove $\chi_{A_s}(x)$ is an interlacing family * in book

Conclude there is an s so that

$$\text{max-root}\left(\chi_{A_s}(x)\right) \leq 2\sqrt{d-1}$$

Random Matrices – Another way to Ramanujan

Given symmetric matrices A and B

and a random orthogonal Q,

understand distribution of eigenvalues of

$$A + QBQ^T$$

Random Matrices

Given symmetric matrices *A* and *B*

and a random orthogonal *Q*,

understand distribution of eigenvalues of

$$A + QBQ^T$$

It only depends on the eigenvalues of *A* and *B*

Free Probability for Random Matrices

Given symmetric matrices *A* and *B*

and a random orthogonal *Q*,

understand distribution of eigenvalues of

$$A + QBQ^T$$

in the limit as the dimension grows and

the eigenvalue distributions of *A* and *B* are given

Finite Free Probability

Given symmetric matrices *A* and *B*

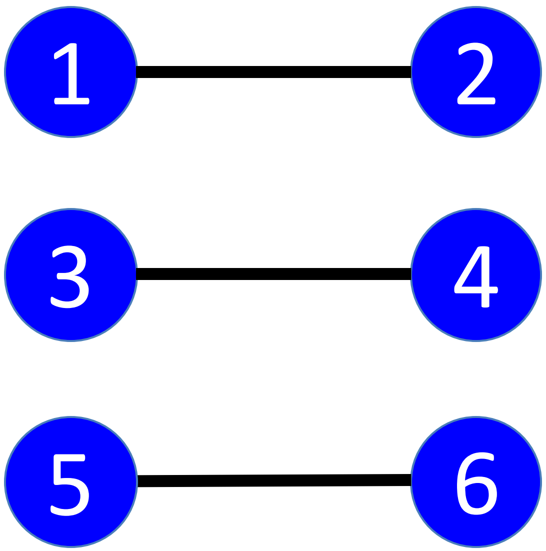
and a random orthogonal *Q*,

understand distribution of eigenvalues of

$$\mathbb{E}\left[\ \mathrm{poly}(A + QBQ^T)\ \right]$$

$$= \mathop{\mathbb{E}}_{\Pi \in \text{perms}}\left[\ \mathrm{poly}(A + \Pi B \Pi^T)\ \right]$$

A matching



0	1	0	0	0	0
1	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	0	0	0	1
0	0	0	0	1	0

$$\mathrm{poly}(M) = (x-1)^{n/2}(x+1)^{n/2}$$

the characteristic polynomial of the adjacency matrix

Proof outline:

Let $M_1, ..., M_d$ be random matchings on n vertices.

Consider $p(x) \stackrel{\mathrm{def}}{=} \mathbb{E}[\ \mathrm{poly}(M_1 + \cdots + M_d)\ ]$

1. Prove $p(x)$ is real rooted.

2. Prove $\lambda_2(p(x)) \leq 2\sqrt{d-1}$

3. Prove "interlacing family",
   so exists $M_1, ..., M_d$ with
   $\lambda_2(M_1 + \cdots + M_d) \leq 2\sqrt{d-1}$

To prove $\lambda_2(p) \leq 2\sqrt{d-1}$

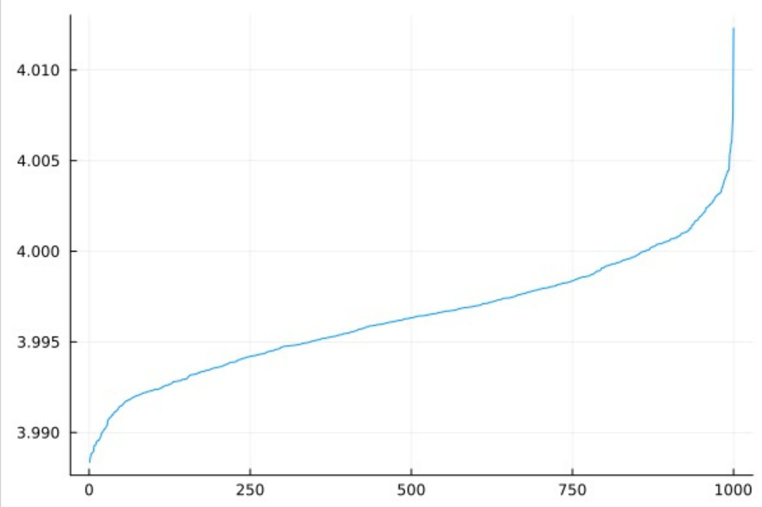
Analyze λ_2 by Upper Barrier Function

Cauchy/Stieltjes/Hilbert Transform

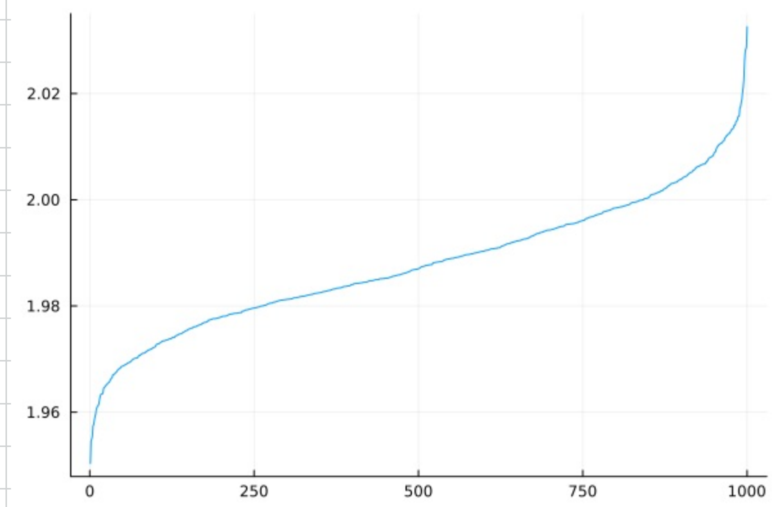
$$\alpha\text{−}\max(\lambda_1, \ldots, \lambda_n) = \max\left\{x : \tfrac{1}{n}\sum \tfrac{1}{x-\lambda_i} = \alpha\right\}$$

$$\mathcal{G}_p(x) = \frac{1}{n}\frac{p'(x)}{p(x)} = \frac{1}{n}\sum_i \frac{1}{x-\lambda_i}$$

69% of Random Regular graphs are Ramanujan



Distribution of M_2 for 1000 5-regular graphs



Tracy-Widom

largest eigval of symmetric matrix of iid Gaussians

Mathematics > Probability

[Submitted on 28 Dec 2024 (v1), last revised 2 Feb 2025 (this version, v2)]

Ramanujan Property and Edge Universality of Random Regular Graphs

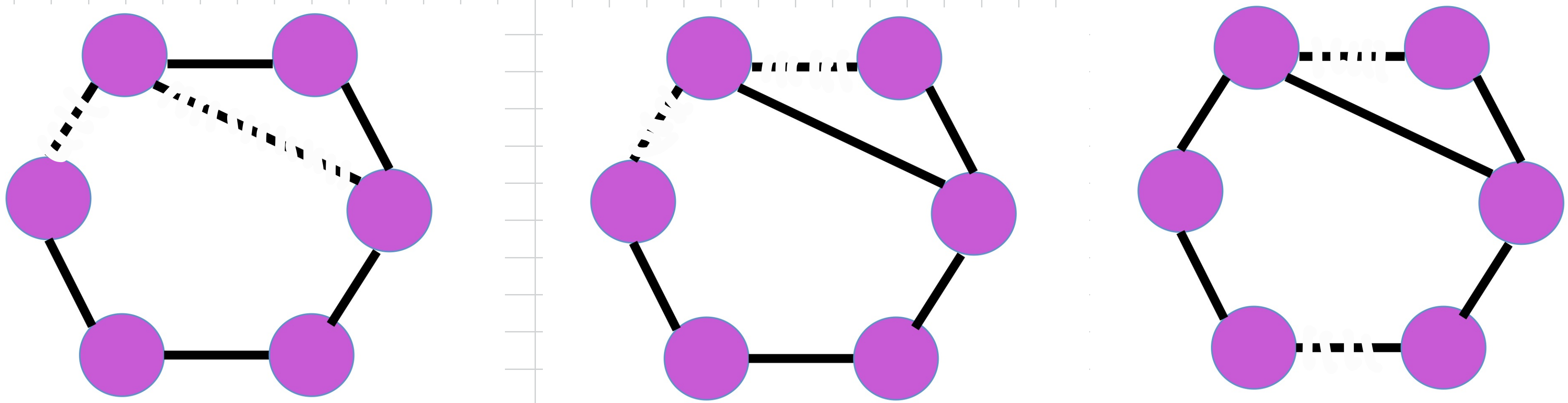
Jiaoyang Huang, Theo McKenzie, Horng-Tzer Yau

We consider the normalized adjacency matrix of a random d-regular graph on N vertices with any fixed degree $d \geq 3$ and denote its eigenvalues as $\lambda_1 = d/\sqrt{d-1} \geq \lambda_2 \geq \lambda_3 \cdots \geq \lambda_N$. We establish the following two results as $N \to \infty$. (i) With high probability, all eigenvalues are optimally rigid, up to an additional $N^{o(1)}$ factor. Specifically, the fluctuations of bulk eigenvalues are bounded by $N^{-1+o(1)}$, and the fluctuations of edge eigenvalues are bounded by $N^{-2/3+o(1)}$. (ii) Edge universality holds for random d-regular graphs. That is, the distributions of λ_2 and $-\lambda_N$ converge to the Tracy-Widom$_1$ distribution associated with the Gaussian Orthogonal Ensemble. As a consequence, for sufficiently large N, approximately 69% of d-regular graphs on N vertices are Ramanujan, meaning $\max\{\lambda_2, |\lambda_N|\} \leq 2$.

Application: show random walks mix quickly.

Needed for sampling. Physics, Finance,

Redistricting, Spanning Trees



Need bounds on λ_2

Computer Science > Data Structures and Algorithms

[Submitted on 2 Jan 2020 (v1), last revised 17 Sep 2020 (this version, v3)]

Spectral Independence in High-Dimensional Expanders and Applications to the Hardcore Model

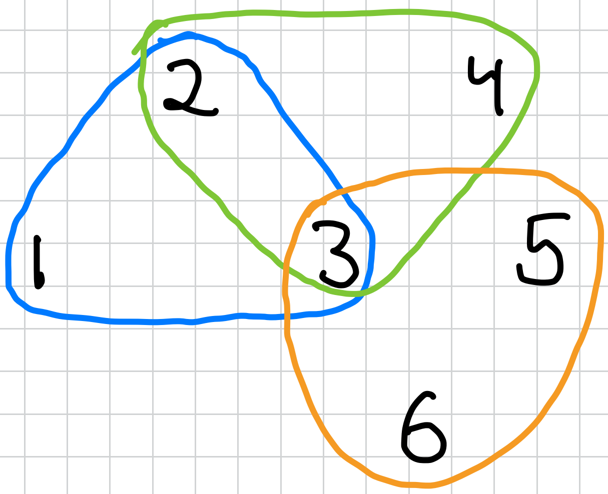
Nima Anari, Kuikui Liu, Shayan Oveis Gharan

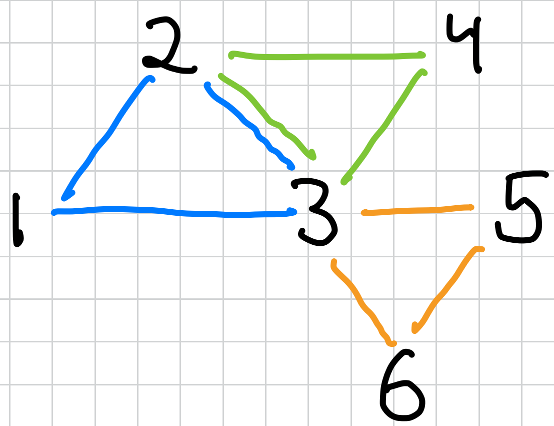
We say a probability distribution μ is spectrally independent if an associated correlation matrix has a bounded largest eigenvalue for the distribution and all of its conditional distributions. We prove that if μ is spectrally independent, then the corresponding high dimensional simplicial complex is a local spectral expander. Using a line of recent works on mixing time of high dimensional walks on simplicial complexes \cite{KM17,DK17,KO18,AL19}, this implies that the corresponding Glauber dynamics mixes rapidly and generates (approximate) samples from μ.

High Dimensional Expanders (HDX) / Hypergraphs

Graph: $E \subseteq \binom{V}{2}$

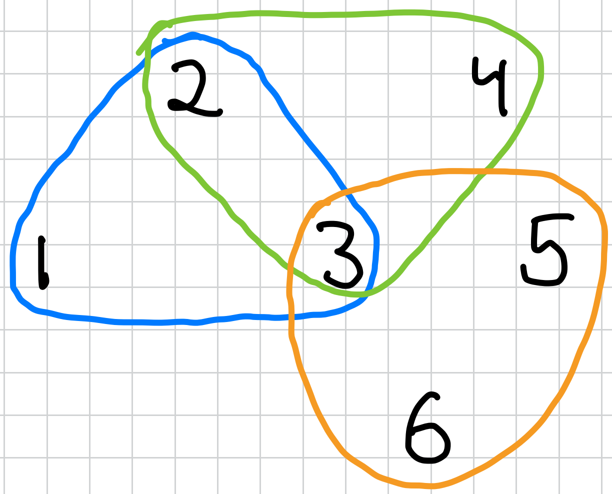
d-regular Hypergraph $E \subseteq \binom{V}{d}$



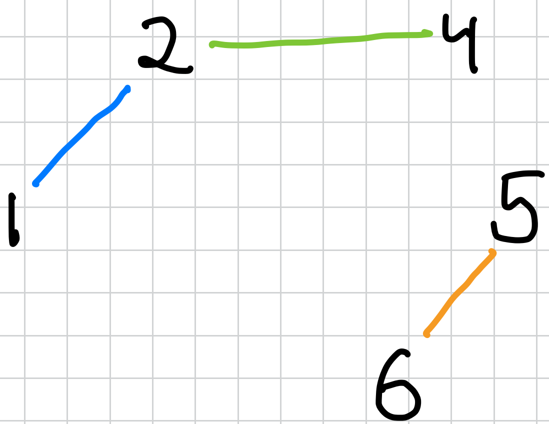


Expands if all links expand

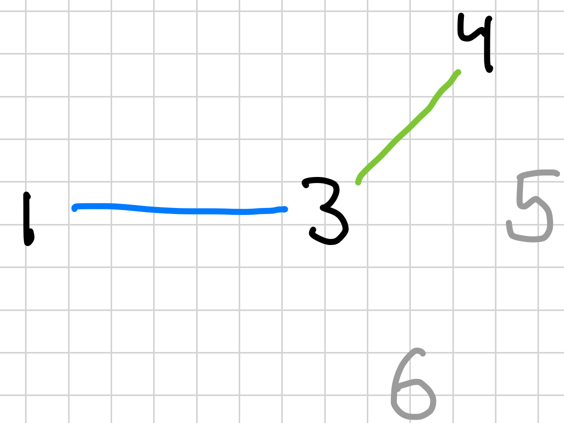
$$\text{link}(s) = \{ e - s : s \subseteq e \}$$



link(3)



link(2)



Great Results from HDX

Revision #1 to TR21-151 | 16th December 2021 09:50

Contact Add Comment

Locally Testable Codes with constant rate, distance, and locality

Revision #1

Authors: Irit Dinur, Shai Evra, Ron Livne, Alexander Lubotzky, Shahar Mozes

Accepted on: 16th December 2021 09:50
Downloads: 1097
Keywords:

Abstract:

Computer Science > Information Theory

[Submitted on 5 Nov 2021 (v1), last revised 21 Jan 2022 (this version, v2)]

Asymptotically Good Quantum and Locally Testable Classical LDPC Codes

Pavel Panteleev, Gleb Kalachev

We study classical and quantum LDPC codes of constant rate obtained by the lifted product construction over non-abelian groups. We show that the obtained families of quantum LDPC codes are asymptotically good, which proves the qLDPC conjecture. Moreover, we show that the produced classical LDPC codes are also asymptotically good and locally testable with constant query and soundness parameters, which proves a well-known conjecture in the field of locally testable codes.

Mathematics > Combinatorics

[Submitted on 21 Apr 2025]

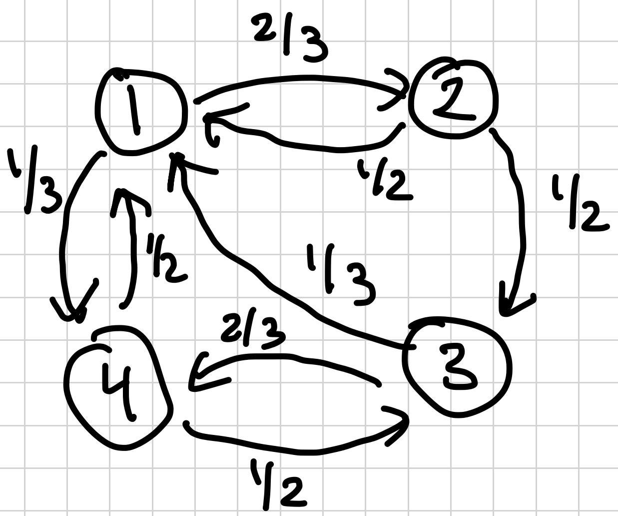
Explicit Lossless Vertex Expanders

Jun-Ting Hsieh, Alexander Lubotzky, Sidhanth Mohanty, Assaf Reiner, Rachel Yun Zhang

We give the first construction of explicit constant-degree lossless vertex expanders. Specifically, for any $\varepsilon > 0$ and sufficiently large d, we give an explicit construction of an infinite family of d-regular graphs where every small set S of vertices has $(1-\varepsilon)d|S|$ neighbors (which implies $(1-2\varepsilon)d|S|$ unique-neighbors). Our results also extend naturally to construct biregular bipartite graphs of any constant imbalance, where small sets on each side have strong expansion guarantees. The graphs we construct admit a free group action, and hence realize new families of quantum LDPC codes of Lin and M. Hsieh with a linear time decoding algorithm.
Our construction is based on taking an appropriate product of a constant-sized lossless expander with a base graph constructed from Ramanujan Cayley cubical complexes.

$$|N(S)| \geq (1-\varepsilon)d|S| \quad \text{for} \quad |S| \leq \alpha n$$

Random walks on directed graphs



$$p_{t+1} = W p_t$$

Stable distribution is p_∞

How compute?

Simulate or solve $Wx = x \leftrightarrow (I-W)x = 0$

A directed Laplacian

Key concept: Eulerian = in-degree = out-degree

Directed Laplacian Solvers, Sparsifiers, etc.

Computer Science > Data Structures and Algorithms

[Submitted on 2 Nov 2016]

Almost-Linear-Time Algorithms for Markov Chains and New Spectral Primitives for Directed Graphs

Michael B. Cohen, Jonathan Kelner, John Peebles, Richard Peng, Anup Rao, Aaron Sidford, Adrian Vladu

In this paper we introduce a notion of spectral approximation for directed graphs. While there are many potential ways one might define approximation for directed graphs, most of them are too strong to allow sparse approximations in general. In contrast, we prove that for our notion of approximation, such sparsifiers do exist, and we show how to compute them in almost linear time.
Using this notion of approximation, we provide a general framework for solving asymmetric linear systems that is broadly inspired by the work of [Peng-Spielman, STOC`14]. Applying this framework in conjunction with our sparsification algorithm, we obtain an almost linear time algorithm for solving directed Laplacian systems associated with Eulerian Graphs. Using this solver in the recent framework of [Cohen-Kelner-Peebles-Peng-Sidford-Vladu, FOCS`16], we obtain almost linear time algorithms for solving a directed Laplacian linear system, computing the stationary distribution of a Markov chain, computing expected commute times in a directed graph, and more.

Computer Science > Data Structures and Algorithms

[Submitted on 26 Nov 2018]

Solving Directed Laplacian Systems in Nearly-Linear Time through Sparse LU Factorizations

Michael B. Cohen, Jonathan Kelner, Rasmus Kyng, John Peebles, Richard Peng, Anup B. Rao, Aaron Sidford

We show how to solve directed Laplacian systems in nearly-linear time. Given a linear system in an $n \times n$ Eulerian directed Laplacian with m nonzero entries, we show how to compute an ϵ-approximate solution in time $O(m \log^{O(1)}(n) \log(1/\epsilon))$. Through reductions from [Cohen et al. FOCS'16] , this gives the first nearly-linear time algorithms for computing ϵ-approximate solutions to row or column diagonally dominant linear systems (including arbitrary directed Laplacians) and computing ϵ-approximations to various properties of random walks on directed graphs, including stationary distributions, personalized PageRank vectors, hitting times, and escape probabilities. These bounds improve upon the recent almost-linear algorithms of [Cohen et al. STOC'17], which gave an algorithm to solve Eulerian Laplacian systems in time $O((m + n2^{O(\sqrt{\log n \log \log n})}) \log^{O(1)}(n\epsilon^{-1}))$.

Computer Science > Data Structures and Algorithms

[Submitted on 4 Oct 2018]

Perron-Frobenius Theory in Nearly Linear Time: Positive Eigenvectors, M-matrices, Graph Kernels, and Other Applications

AmirMahdi Ahmadinejad, Arun Jambulapati, Amin Saberi, Aaron Sidford

In this paper we provide nearly linear time algorithms for several problems closely associated with the classic Perron-Frobenius theorem, including computing Perron vectors, i.e. entrywise non-negative eigenvectors of non-negative matrices, and solving linear systems in asymmetric M-matrices, a generalization of Laplacian systems. The running times of our algorithms depend nearly linearly on the input size and polylogarithmically on the desired accuracy and problem condition number.

Computer Science > Data Structures and Algorithms

[Submitted on 19 Aug 2024]

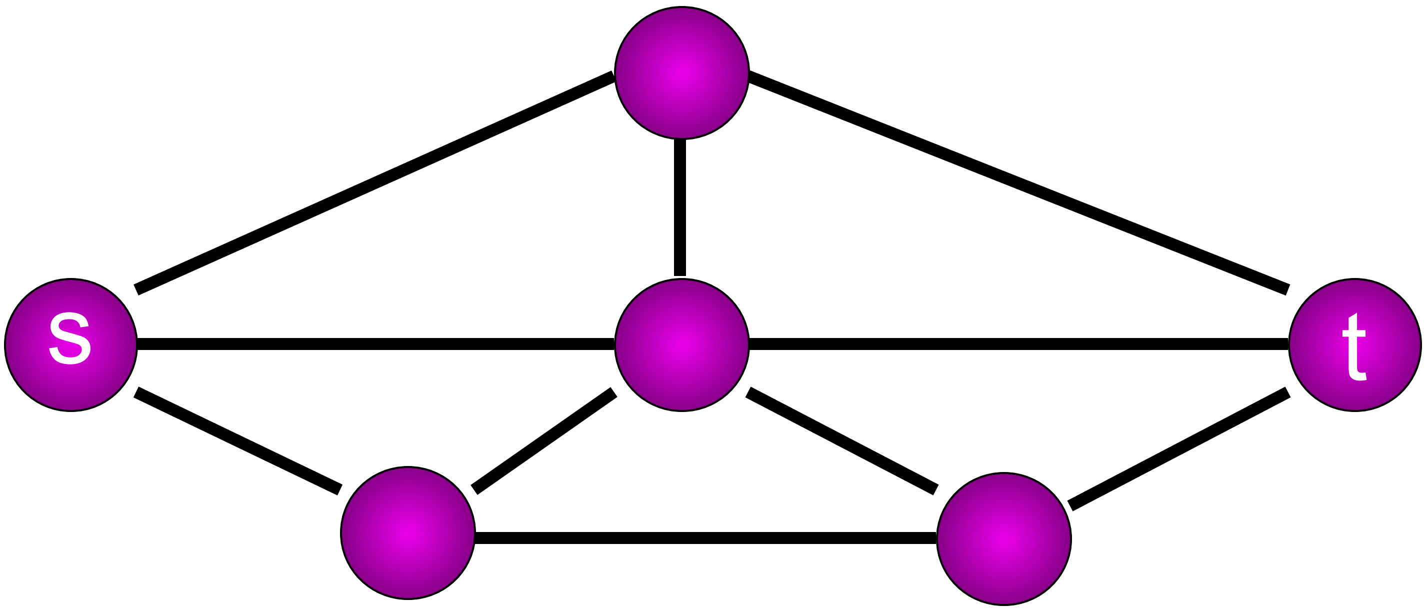
Eulerian Graph Sparsification by Effective Resistance Decomposition

Arun Jambulapati, Sushant Sachdeva, Aaron Sidford, Kevin Tian, Yibin Zhao

We provide an algorithm that, given an n-vertex m-edge Eulerian graph with polynomially bounded weights, computes an $\breve{O}(n \log^2 n \cdot \varepsilon^{-2})$-edge ε-approximate Eulerian sparsifier with high probability in $\breve{O}(m \log^3 n)$ time (where $\breve{O}(\cdot)$ hides polyloglog(n) factors). Due to a reduction from [Peng-Song, STOC '22], this yields an $\breve{O}(m \log^3 n + n \log^6 n)$-time algorithm for solving n-vertex m-edge Eulerian Laplacian systems with polynomially-bounded weights with high probability, improving upon the previous state-of-the-art runtime of $\Omega(m \log^8 n + n \log^{23} n)$. We also give a polynomial-time algorithm

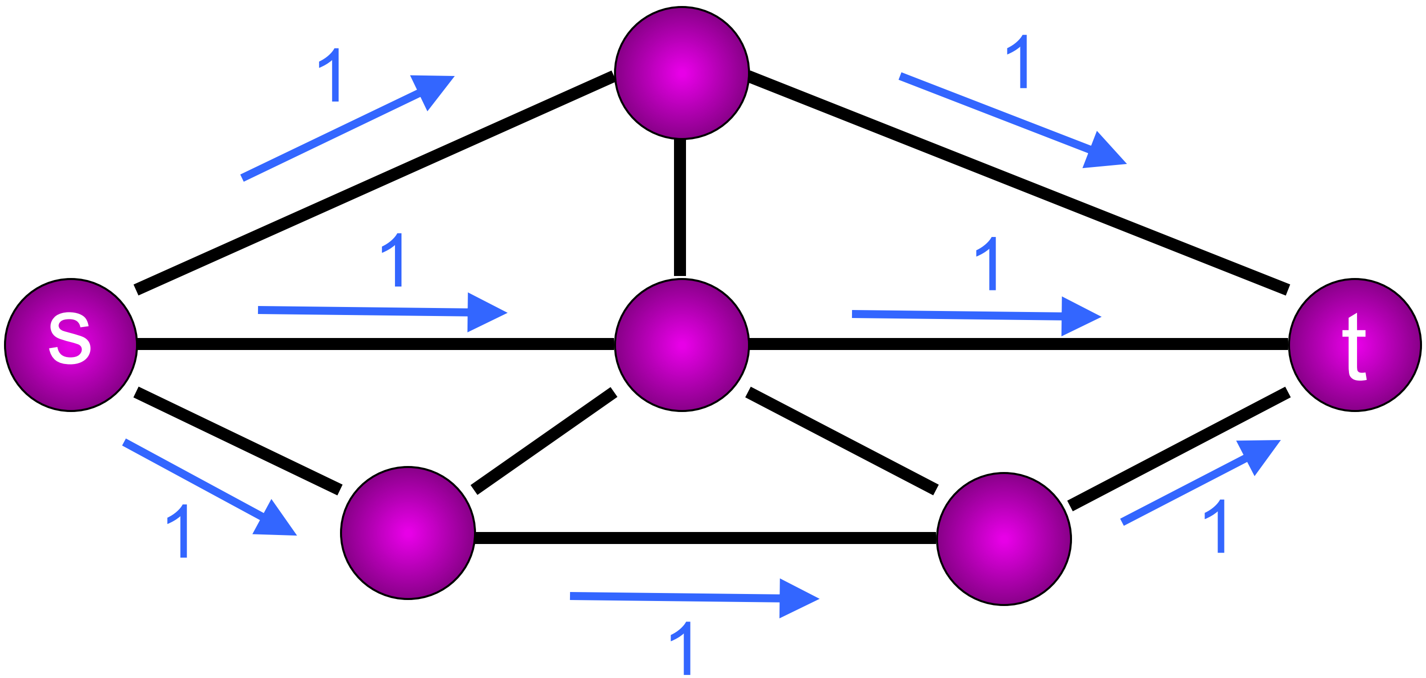
Maximum flow problem

Send as much stuff as possible from s to t.
At most one unit can go through each edge.



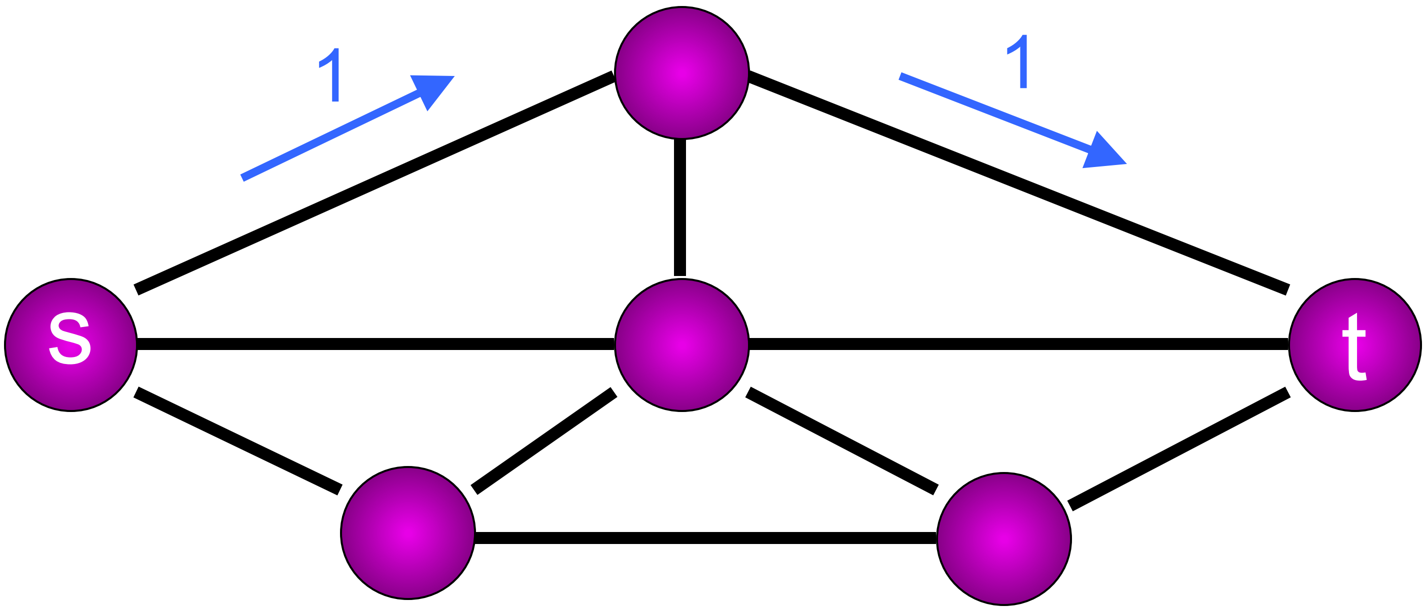
Maximum flow problem

Send as much stuff as possible from s to t.
At most one unit can go through each edge.



Maximum flow problem

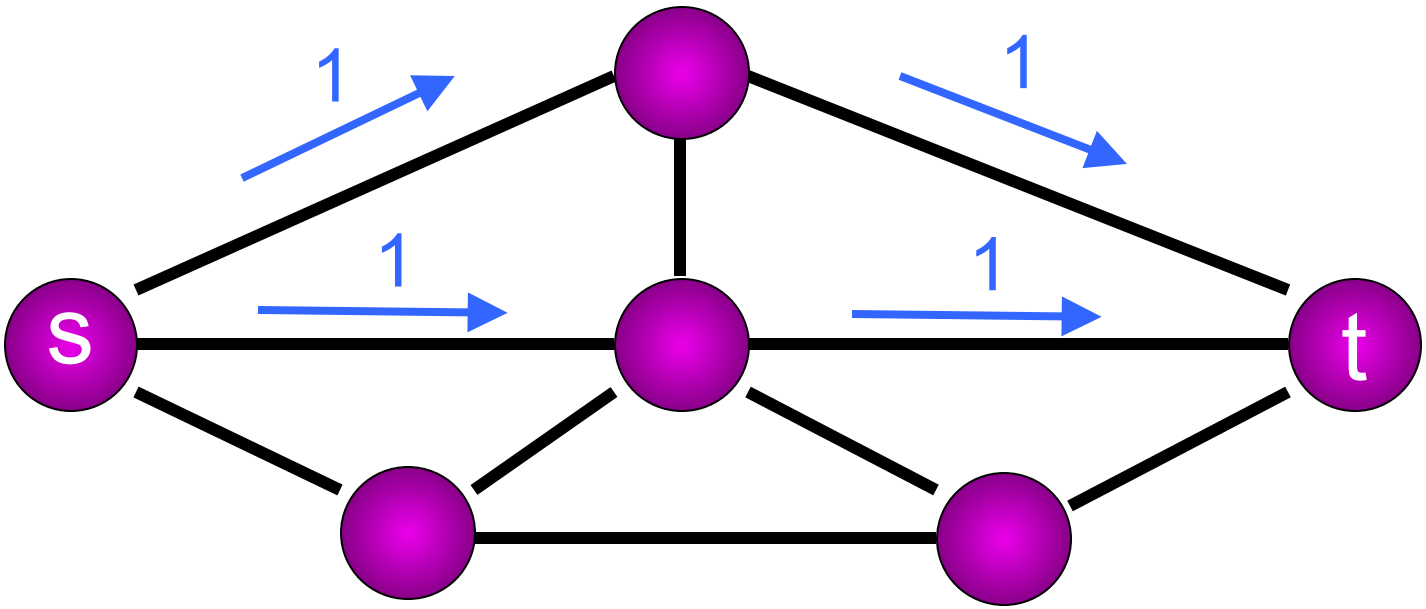
Send as much stuff as possible from s to t.
At most one unit can go through each edge.



Standard approach (Ford-Fulkerson):
incrementally add flow paths

Maximum flow problem

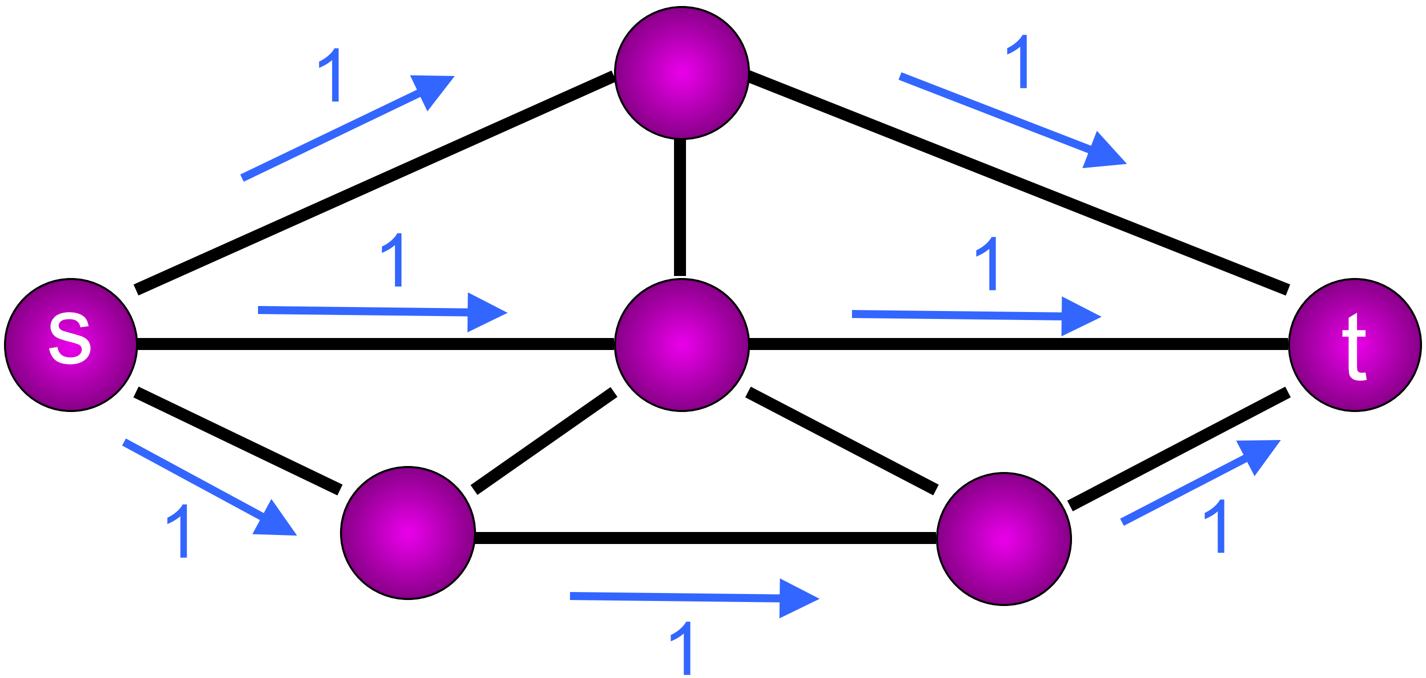
Send as much stuff as possible from s to t.
At most one unit can go through each edge.



Standard approach (Ford-Fulkerson):
incrementally add flow paths

Maximum flow problem

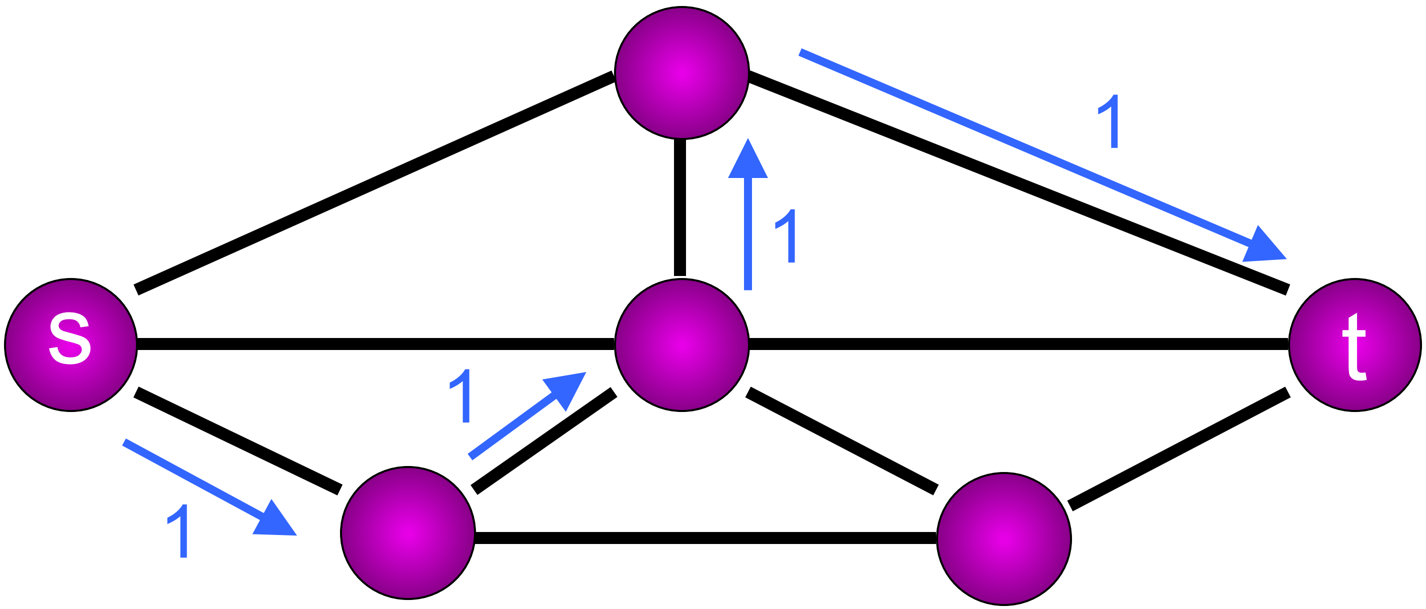
Send as much stuff as possible from s to t.
At most one unit can go through each edge.



Standard approach (Ford-Fulkerson):
incrementally add flow paths

Maximum flow problem

Send as much stuff as possible from s to t.
At most one unit can go through each edge.

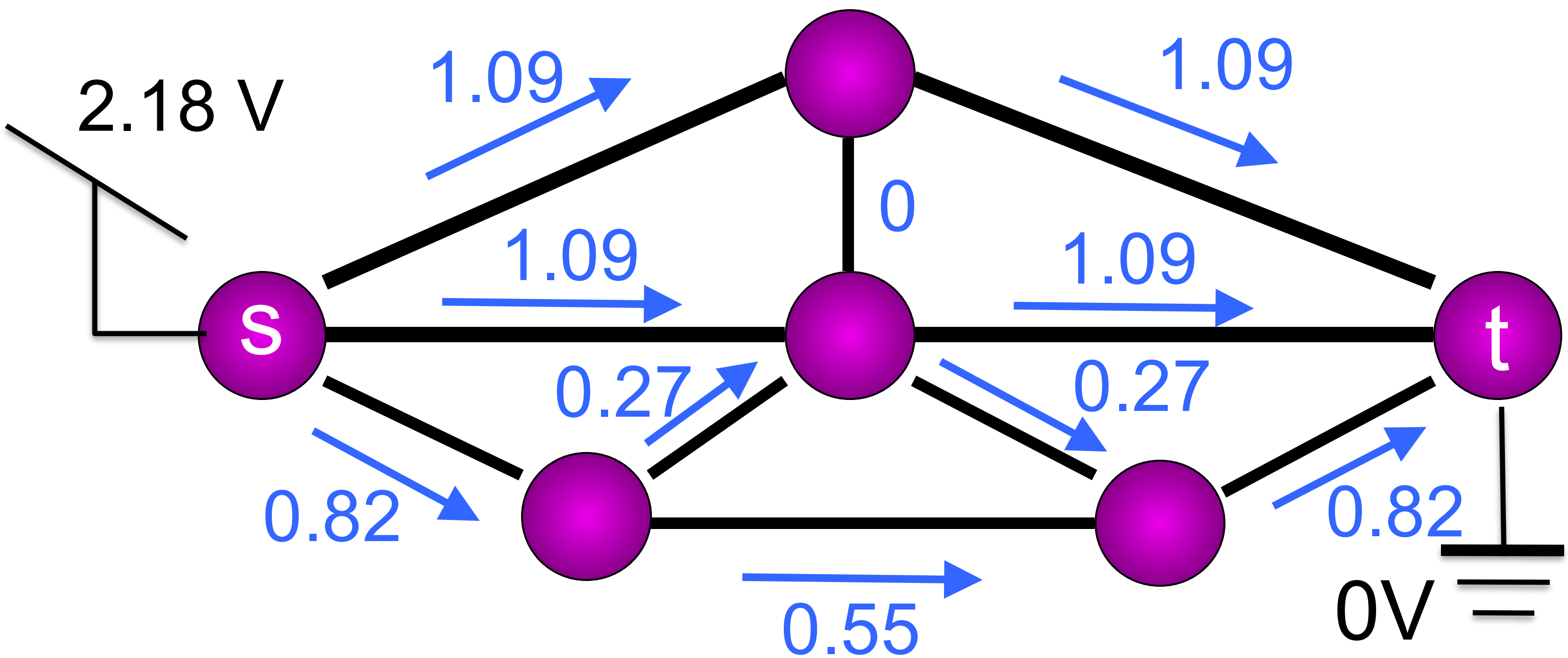


Issue: sometimes requires backtracking

Maximum flow problem, electrical approach

[Christiano-Kelner-Madry-S-Teng '11]

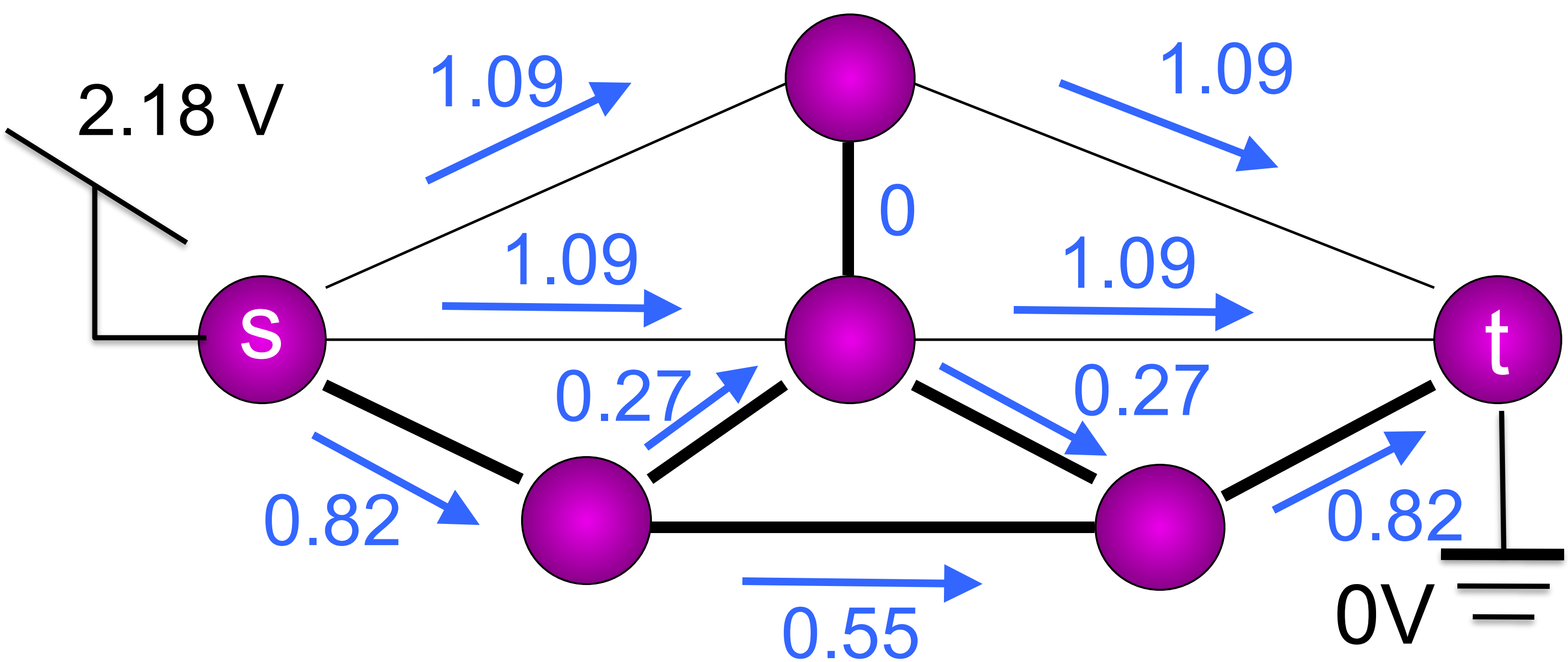
1. Try the electrical flow.



Maximum flow problem, electrical approach

[Christiano-Kelner-Madry-S-Teng '11]

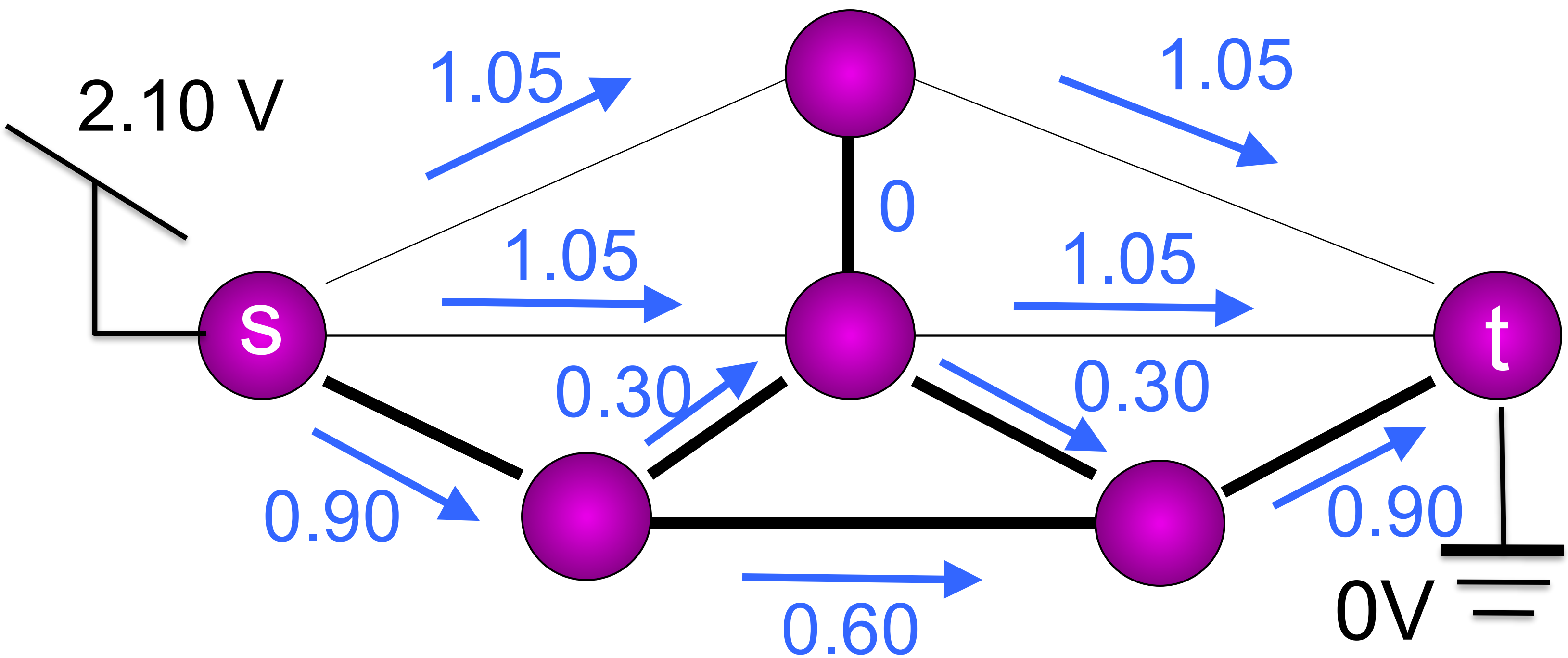
1. Try the electrical flow.
2. Increase resistance when too much flow



Maximum flow problem, electrical approach

[Christiano-Kelner-Madry-S-Teng '11]

1. Try the electrical flow.
2. Increase resistance when too much flow



Computer Science > Data Structures and Algorithms

[Submitted on 1 Mar 2022 (v1), last revised 22 Apr 2022 (this version, v2)]

Maximum Flow and Minimum-Cost Flow in Almost-Linear Time

Li Chen, Rasmus Kyng, Yang P. Liu, Richard Peng, Maximilian Probst Gutenberg, Sushant Sachdeva

We give an algorithm that computes exact maximum flows and minimum-cost flows on directed graphs with m edges and polynomially bounded integral demands, costs, and capacities in $m^{1+o(1)}$ time. Our algorithm builds the flow through a sequence of $m^{1+o(1)}$ approximate undirected minimum-ratio cycles, each of which is computed and processed in amortized $m^{o(1)}$ time using a new dynamic graph data structure.

Our framework extends to algorithms running in $m^{1+o(1)}$ time for computing flows that minimize general edge-separable convex functions to high accuracy. This gives almost-linear time algorithms for several problems including entropy-regularized optimal transport, matrix scaling, p-norm flows, and p-norm isotonic regression on arbitrary directed acyclic graphs.

Quanta magazine | Physics | Mathematics | Biology | Computer Science | Topics | Archive

NETWORKS

Researchers Achieve 'Absurdly Fast' Algorithm for Network Flow

12

Computer scientists can now solve a decades-old problem in practically the time it takes to write it down.

Beyond Spectral Relaxations

$$\phi_G = \min_S \phi(S) = \min_S \frac{1_S^T L_G 1_S}{1_S^T L_{K_n} 1_S} \geq \min_x \frac{x^T L_G x}{x^T L_{K_n} x} = \lambda_2$$

Linear Programs (LP)

$$\min_x \; c^T x \quad \text{s.t.} \quad a_i^T x \leq b_i \quad 1 \leq i \leq m$$

Semi-Definite Programs (SDP)

$$\min_{X \succeq 0} \; Tr(C X) \quad \text{s.t.} \quad Tr(A_i X) \leq b_i \quad 1 \leq i \leq m$$

Improved estimates of ϕ. Leighton-Rao

Arora-Rao-Vazirani

Graph Neural Networks

Apply operations like $x_{t+1} = M x_t$

But non-linear and learned

Less Theory. More Practice.

Additive view of Gaussian Elimination

$$\begin{pmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 14 & 0 \\ -4 & -1 & 0 & 7 \end{pmatrix}$$

Find the rank-1 matrix that agrees on the first row and column.

$$\begin{pmatrix} 16 & -4 & -8 & -4 \\ -4 & 1 & 2 & 1 \\ -8 & 2 & 4 & 2 \\ -4 & 1 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 4 \\ -1 \\ -2 \\ -1 \end{pmatrix} \begin{pmatrix} 4 \\ -1 \\ -2 \\ -1 \end{pmatrix}^\top$$

Additive view of Gaussian Elimination

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 10 & -2 \\ 0 & -2 & -2 & 6 \end{pmatrix}$$

Find the rank-1 matrix that agrees on the **next** row and column.

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 1 & 1 \\ 0 & -2 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ -1 \\ -1 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \\ -1 \\ -1 \end{pmatrix}^\top$$

Additive view of Gaussian Elimination

$$\begin{pmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 14 & 0 \\ -4 & -1 & 0 & 7 \end{pmatrix}$$

$$= \begin{pmatrix} 4 \\ -1 \\ -2 \\ -1 \end{pmatrix} \begin{pmatrix} 4 \\ -1 \\ -2 \\ -1 \end{pmatrix}^\top + \begin{pmatrix} 0 \\ 2 \\ -1 \\ -1 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \\ -1 \\ -1 \end{pmatrix}^\top + \begin{pmatrix} 0 \\ 0 \\ 3 \\ -1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 3 \\ -1 \end{pmatrix}^\top + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2 \end{pmatrix}^\top$$

$$= \begin{pmatrix} 4 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 \\ -2 & -1 & 3 & 0 \\ -1 & -1 & -1 & 2 \end{pmatrix} \begin{pmatrix} 4 & -1 & -2 & -1 \\ 0 & 2 & -1 & -1 \\ 0 & 0 & 3 & -1 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

Gaussian Elimination of Laplacians

If this is a Laplacian, then so is this

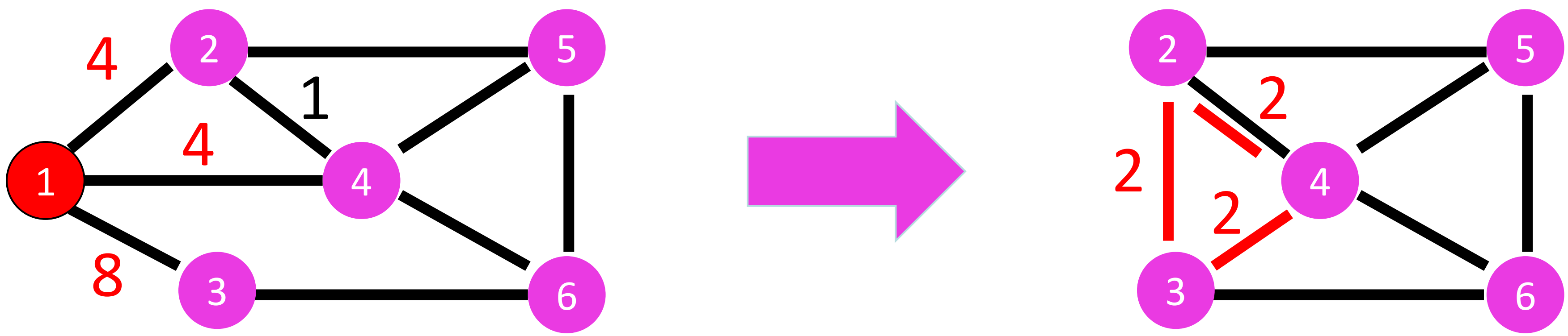
$$\begin{pmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 14 & 0 \\ -4 & -1 & 0 & 7 \end{pmatrix} \qquad \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 10 & -2 \\ 0 & -2 & -2 & 6 \end{pmatrix}$$

Elimination in Laplacians

If this is a Laplacian, then so is this

$$\begin{pmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 14 & 0 \\ -4 & -1 & 0 & 7 \end{pmatrix} \qquad \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 10 & -2 \\ 0 & -2 & -2 & 6 \end{pmatrix}$$

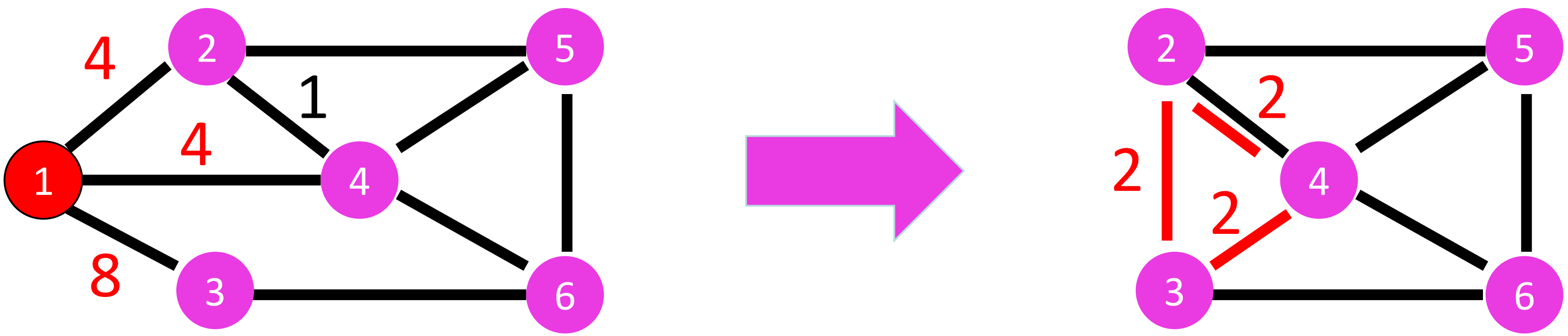
When eliminate a node, add a clique on its neighbors



Approximate Elimination

(Kyng & Sachdeva '16)

1. when eliminate a node, add a clique on its neighbors



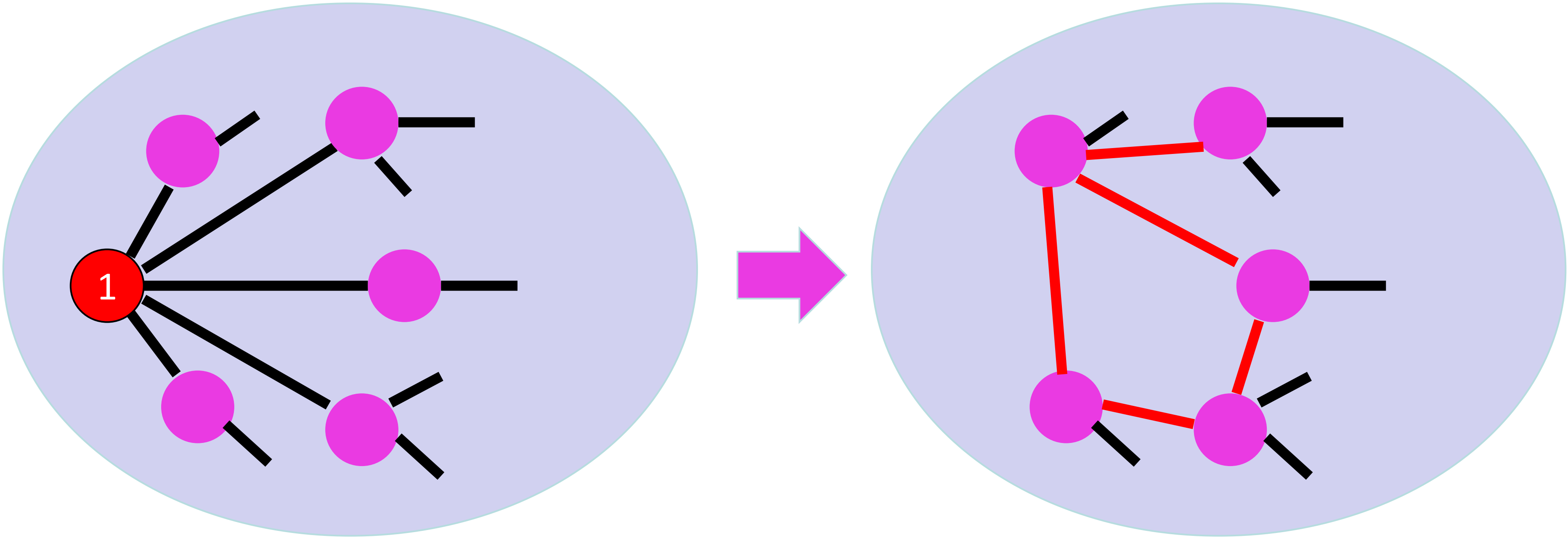
2. Approximate that clique by a sparse random graph, without ever constructing it

Approximate Elimination

(Kyng & Sachdeva '16)

When eliminate a node of degree *d*,

add *d* edges at random between its neighbors, sampled with probability proportional to the weight of the edge to the eliminated node



Approximate Gaussian Elimination

(Kyng & Sachdeva '16)

1. Initialize by making $O(\log^2 n)$ copies of every edge

2. and randomly ordering the vertices,

Total time is $O(m \log^3 n)$

Approximate Elimination is basis of fastest Laplacian solver

Laplacians.jl

How I do research

See also DASopt.jl, 3-minutes at HLF '23