Four graph partitioning algorithms

Fan Chung

University of California, San Diego

History of graph partitioning

NP-hard ⟹ approximation algorithms

- Spectral method, Fiedler 73, Folklore
- Multicommunity flow, Leighton+Rao 88
- Semidefinite programming, Arora+Rao+Vazirani 04
- Expander flow, Arora+Hazan+Kale 04
- Single commodity flows, Khandekar+Rao+Vazirani 06

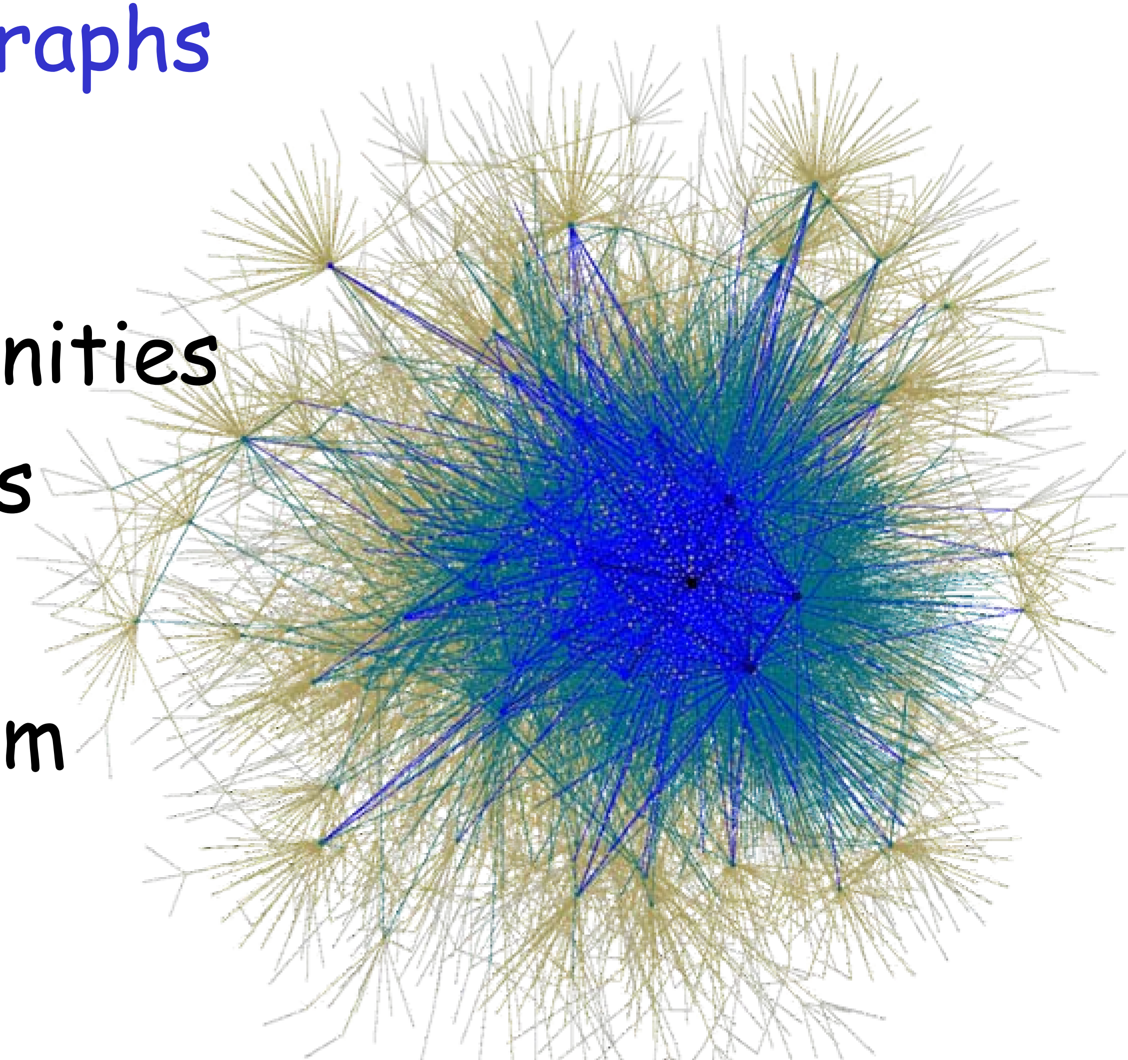
“traditional“ applications of graph partition algorithms:

Divide-and-conquer algorithms

- Circuit layout & designs
- Parallel computing
- Hierarchical clusterings
- Bioinformatics
- …

Applications of partitioning algorithms for massive graphs

- Web search
- identify communities
- locate hot spots
- trace targets
- combat link spam
- epidemics
- …



graph partitioning - Google Search - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.google.com/search?hl=en&q=graph+partitioning+&btnG=Search

Getting Started | Latest Headlines

Fan Chung Graham's Homepage | Fan Chung Graham's link page | graph partitioning - Google Search

Web Images Maps News Shopping Gmail more ▾ | Sign in

Google [graph partitioning] Search | Advanced Search | Preferences

Web Results **1** - **10** of about **460,000** for **graph partitioning**. (**0.09** seconds)

1.5.6 Graph Partition

Excerpt from The Algorithm Design Manual: **Graph partitioning** arises as a preprocessing step to divide-and-conquer algorithms, where it is often a good idea **...**

www.cs.sunysb.edu/~algorith/files/**graph-partition**.shtml - 19k - Cached - Similar pages

Algorithms and Software for Partitioning Graphs

Graph partitioning is an NP hard problem with numerous applications. **...** An Improved Spectral **Graph Partitioning** Algorithm for Mapping Parallel Computations **...**

www.sandia.gov/~bahendr/**partitioning**.html - 11k - Cached - Similar pages

Graph Partitioning

Then, the **graph partitioning** problem consists on dividing G into k disjoint partitions. The goal is minimize the number of cuts in the edges of the **...**

www.ace.ual.es/~cgil/grafos/**Graph_Partitioning**.html - 12k - Cached - Similar pages

Graph partition - Wikipedia, the free encyclopedia

The **graph partitioning** problem in mathematics consists of dividing a **graph** into pieces, such that the pieces are of about the same size and there are few **...**

en.wikipedia.org/wiki/**Graph_partitioning** - 16k - Cached - Similar pages

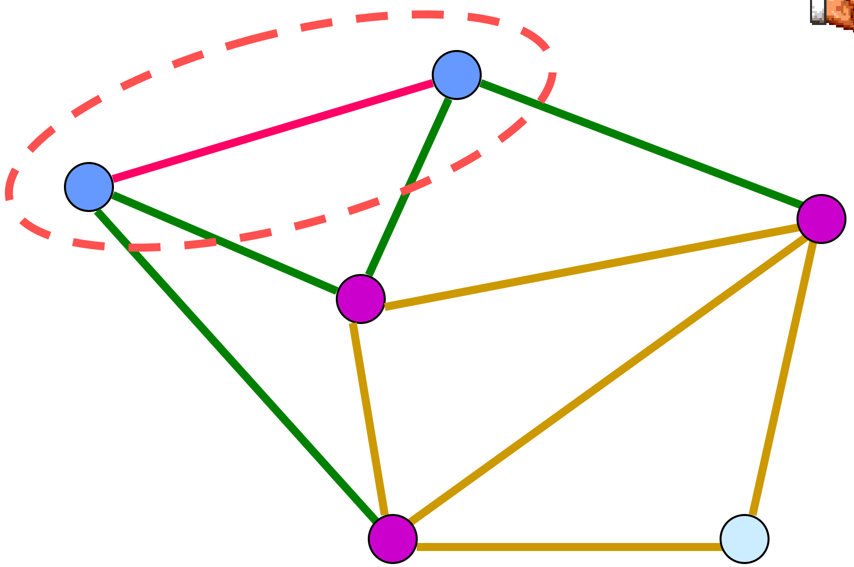
Done

Outline of the talk

- Motivations
- Conductance and Cheeger's inequality
- Four graph paritioning algorithms by using:
  - eigenvectors
  - random walks
  - PageRank
  - heat kernel
- Local graph algorithms
- Future directions

Two types of cuts:

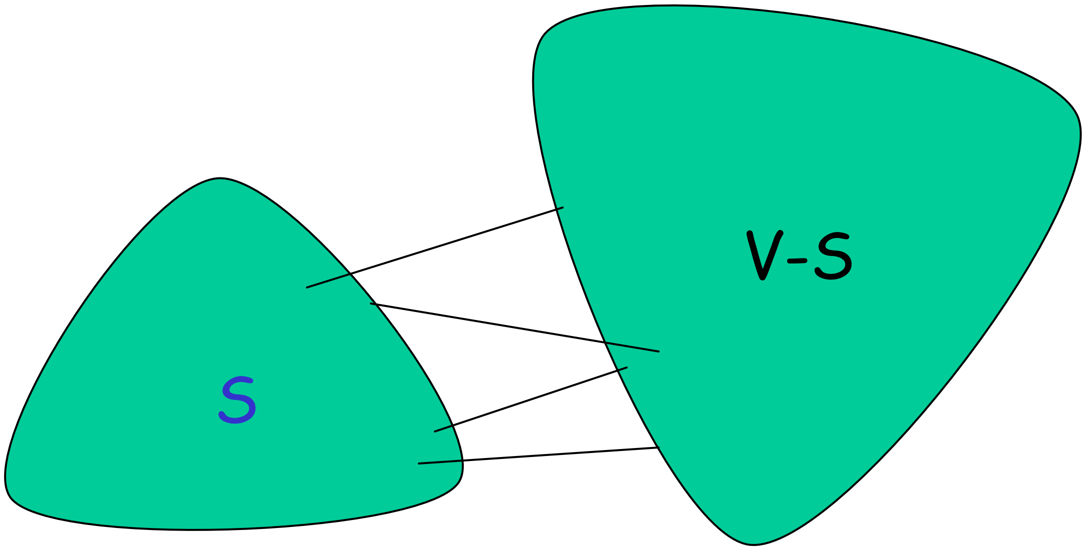
- Vertex cut
- edge cut



How “good” is the cut?

$$\frac{e(S, V-S)}{\mathrm{Vol}\ S} \Longleftrightarrow \frac{e(S, V-S)}{|S|}$$

$$\mathrm{Vol}\ S = \sum_{v \in S} \deg(v) \qquad |S| = \sum_{v \in S} 1$$



The Cheeger constant for graphs

The Cheeger constant

$$\Phi_G = \min_S \frac{e(S,\bar{S})}{\min(vol\ S, vol\ \bar{S})}$$

The volume of S is $$vol(S) = \sum_{x \in S} d_x$$

Φ_G and its variations are sometimes called "conductance", "isoperimetric number", …

The Cheeger inequality

The Cheeger constant

$$\Phi_{G} = \min_{S} \frac{e(S,\bar{S})}{\min(vol\ S, vol\ \bar{S})}$$

The Cheeger inequality

$$2\Phi_G \geq \lambda \geq \frac{\Phi_G^{\ 2}}{2}$$

λ : the first nontrivial eigenvalue of the (normalized) Laplacian.

The spectrum of a graph

- Adjacency matrix

Many ways to define the spectrum of a graph.

How are the eigenvalues related to properties of graphs?

The spectrum of a graph

- Adjacency matrix

- Combinatorial Laplacian

$$L = D - A$$

adjacency matrix (A)

diagonal degree matrix (D)

- Normalized Laplacian

Random walks

Rate of convergence

The spectrum of a graph

Discrete Laplace operator

$$\Delta f(x) = \frac{1}{d_x} \sum_{y \sim x} (f(x) - f(y))$$

$$L(x,y) = \begin{cases} 1 & if \quad x = y \\ -\frac{1}{d_x} & if \quad x \neq y \quad and \quad x \sim y \end{cases}$$

not symmetric in general

•Normalized Laplacian

symmetric
normalized

$$\mathrm{L}(x,y) = \begin{cases} 1 & if \quad x = y \\ -\frac{1}{\sqrt{d_x d_y}} & if \quad x \neq y \quad and \quad x \sim y \end{cases}$$

with eigenvalues

$$0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{n-1} \leq 2$$

$\lambda_1 = \lambda$

Can you hear the shape of a network?

λ dictates many properties of a graph.

- connectivity
- diameter
- isoperimetry (bottlenecks)
- … …

How "good" is the cut by using the eigenvalue λ ?

Finding a cut by a sweep

Using a sweep by the eigenvector,

can reduce the exponential number of

choices of subsets to a **linear** number.

Finding a cut by a sweep

Using a sweep by the eigenvector,
can reduce the exponential number of
choices of subsets to a linear number.

- Still, there is a lower bound guarantee
by using the Cheeger inequality.

$$2\Phi \geq \lambda \geq \frac{\Phi^2}{2}$$

Partitioning algorithm ⟺ The Cheeger inequality

Using eigenvector f,

the Cheeger inequality can be stated as

$$2\Phi \geq \lambda \geq \frac{\alpha^2}{2} \geq \frac{\Phi^2}{2}$$

where λ is the first non-trivial eigenvalue of the Laplacian and α is the minimum Cheeger ratio in a sweep using the eigenvector f.

Eigenvalue problem for n x n matrix:.

$n \approx$ 30 billion websites

Hard to compute eigenvalues

Even harder to compute eigenvectors

In the old days,

compute for a given (whole) graph.

In reality,

can only afford to compute “locally”.

(Access to a (huge) graph,

e.g., for a vertex v, find its neighbors.

Bounded number of access.)

Finding a cut by a sweep

Using a sweep by the eigenvector can reduce the exponential number of choices of subsets to a **linear** number.

Using a local sweep by random walks, PageRank and its variations can further reduce the a **linear** number of choices to a specified finite number of sizes.

Four one-sweep graph partitioning algorithms

- graph spectral method

spectral partition algorithm

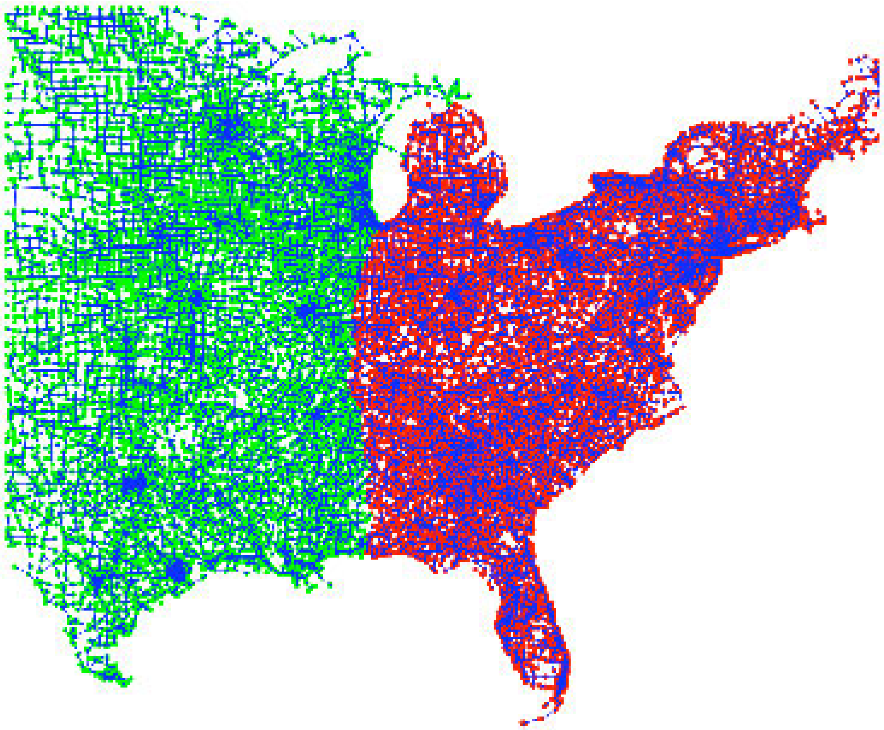
- random walks
- PageRank
- heat kernel

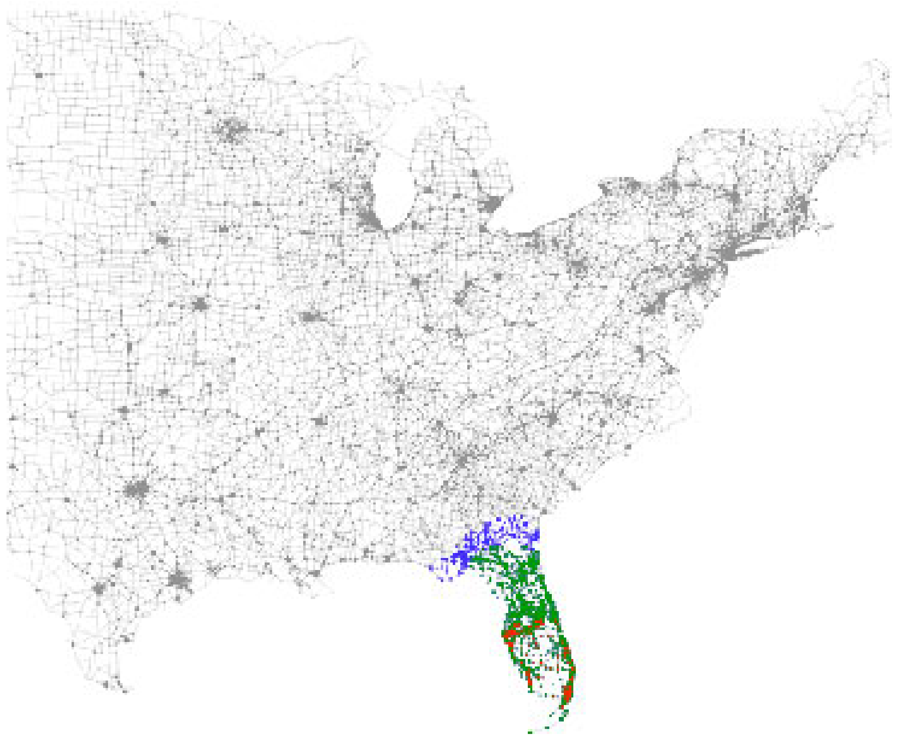
local partition algorithms

4 Partitioning algorithm ⟷ 4 Cheeger inequalities

- graph spectral method — Fiedler '73, Cheeger, 60's
- random walks — Mihail 89; Lovasz, Simonovits, 90, 93; Spielman, Teng, 04
- PageRank — Andersen, Chung, Lang, 06
- heat kernel — Chung, PNAS , 08.

Graph partitioning ⟷ Local graph partitioning

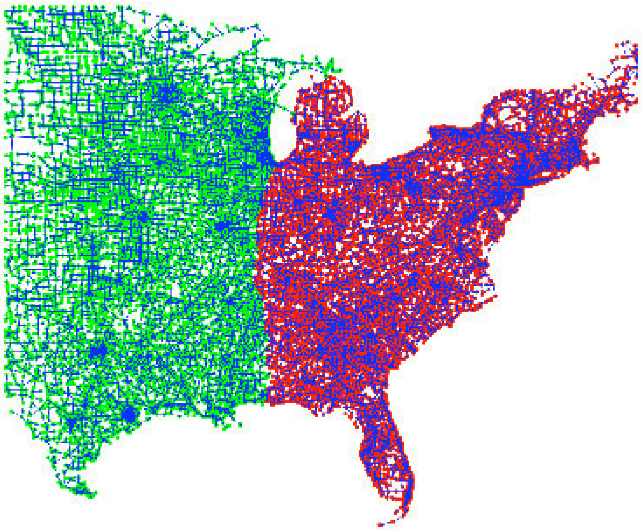


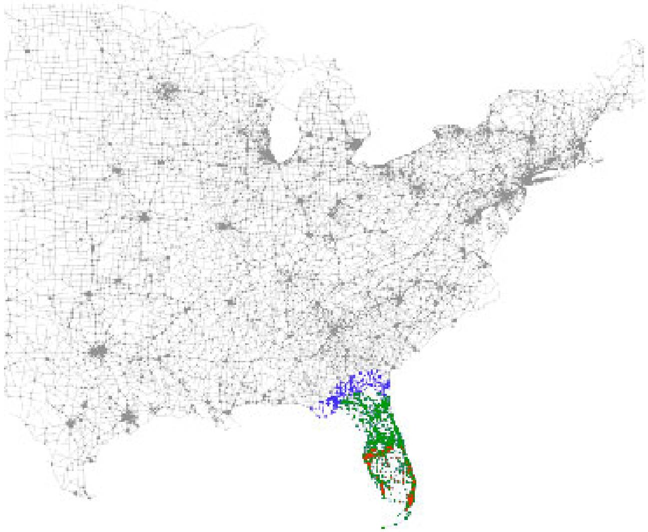


Courtesy of Reid Andersen

What is a local graph partitioning algorithm?

A local graph partitioning algorithm finds a small cut near the given seed(s) with running time depending only on the size of the output.

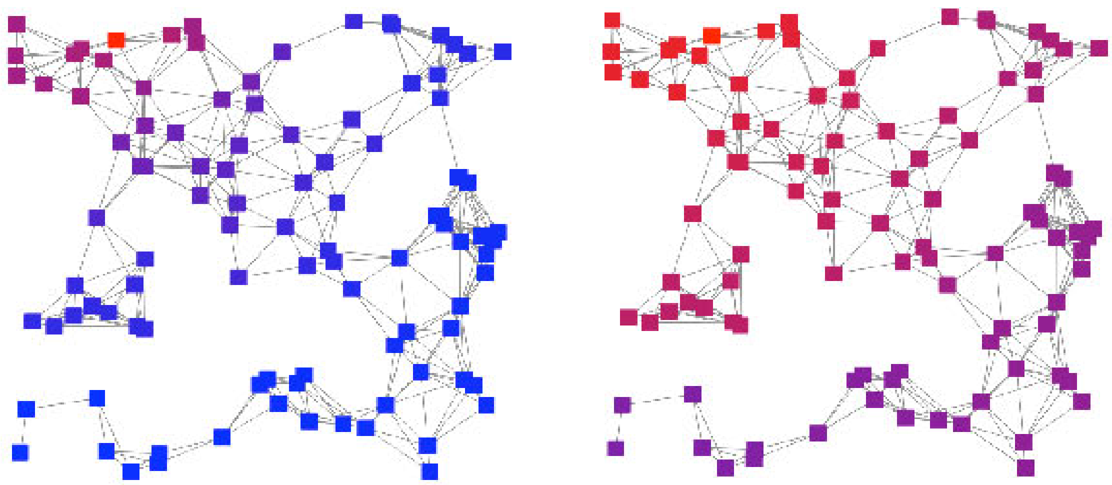




The definition of PageRank given by

Brin and Page is based on

random walks.



Partitioning ⟵ Computing PageRank

History of computing Pagerank

- Brin+Page 98
- Personalized PageRank, Haveliwala 03
- Computing personalized PageRank,
  Jeh+Widom 03
  Berkhin 06

Random walks in a graph.

G : a graph

P : transition probability matrix

$$P(u,v) = \begin{cases} \frac{1}{d_u} & if\ u \sim v, \\ 0 & otherwise. \end{cases}$$

d_u := the degree of u.

A lazy walk:

$$W = \frac{I+P}{2}$$

Original definition of PageRank

A (bored) surfer

- either surf a random webpage

  with probability α

- or surf a linked webpage

  with probability $1- \alpha$

α : the jumping constant

$$p = \alpha(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}) + (1-\alpha) pW$$

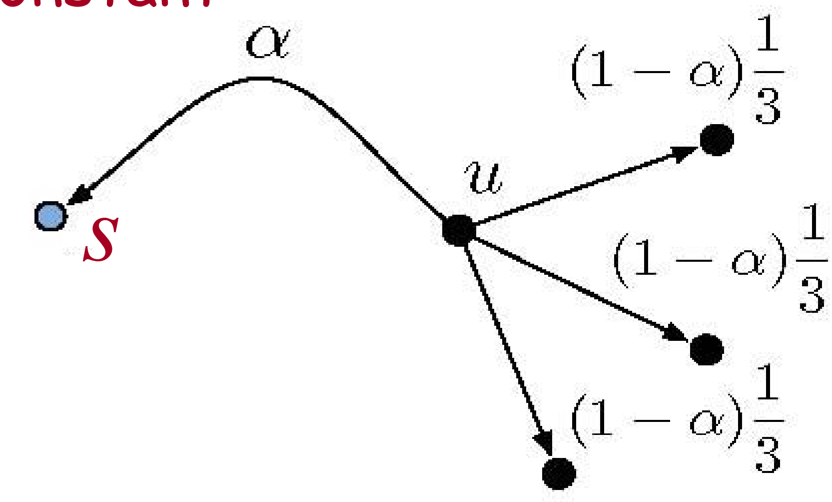
Definition of personalized PageRank

Two equivalent ways to define PageRank $pr(\alpha,s)$

(1) $$p = \alpha s + (1-\alpha) pW$$

s: the seed as a row vector

α : the jumping constant



Definition of PageRank

Two equivalent ways to define PageRank $p = pr(\alpha, s)$

(1) $$p = \alpha s + (1-\alpha) p W$$

(2) $$p = \alpha \sum_{t=0}^{\infty} (1-\alpha)^t (s W^t)$$

$s = (\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n})$ ⟹ the (original) PageRank

$s =$ some "seed", e.g., $(1, 0, \ldots, 0)$

⟹ personalized PageRank

(Organize the random walks by a scalar α.)

Partitioning algorithm using random walks

Mihail 89, Lovász+Simonovits, 90, 93

$$\left|W^k(u,S) - \pi(S)\right| \leq \sqrt{\frac{vol(S)}{d_u}}\left(1 - \frac{\beta_k^2}{8}\right)^k$$

Leads to a Cheeger inequality:

$$2\Phi \quad \geq \quad \lambda \quad \geq \quad \frac{\beta_G{}^2}{8\log n} \quad \geq \quad \frac{\Phi^2}{8\log n}$$

where β_G is the minimum Cheeger ratio over sweeps by using a lazy walk of k steps from every vertex for an appropriate range of k .

Algorithmic aspects of PageRank

- Fast approximation algorithm for personalized PageRank

  greedy type algorithm, linear complexity

- Can use the jumping constant to approximate PageRank with a support of the desired size.

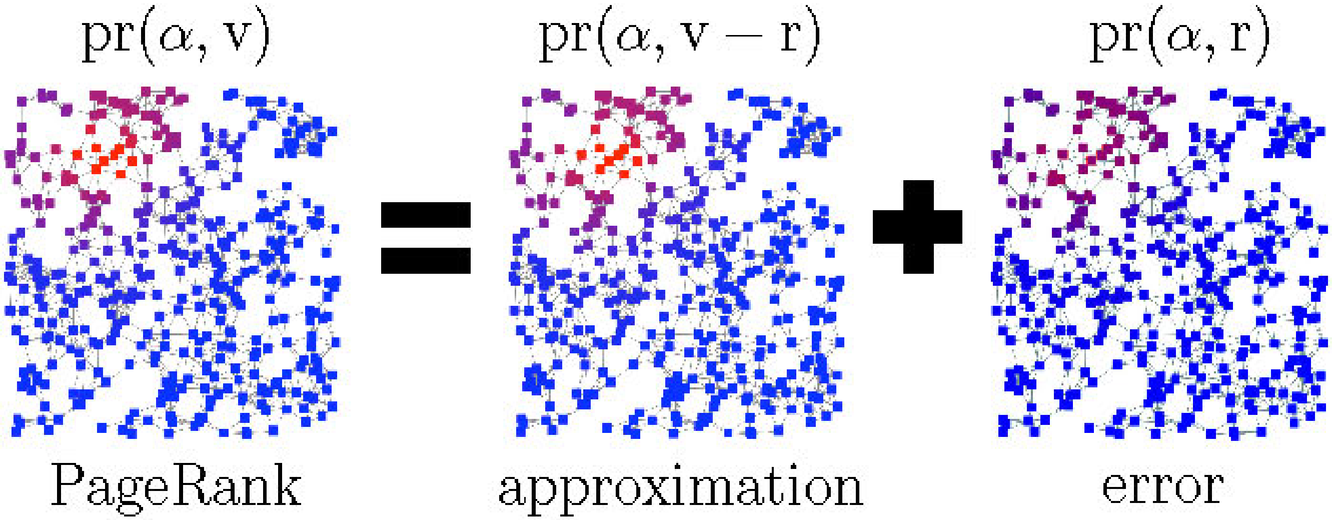
- Errors can be effectively bounded.

Approximate the pagerank vector :

$$pr(\alpha, s) = p + pr(\alpha, r)$$

Approximate pagerank

Residue vector



Partitioning algorithm using PageRank

Using the PageRank vector with seed as a subset S and $vol(S) \leq vol(G)/4$, a Cheeger inequality can be obtained :

$$\Phi_S \geq \frac{\gamma_u^2}{8\log s} \geq \frac{\Phi_u^2}{8\log s}$$

where γ_u is the minimum Cheeger ratio over sweeps by using personalized PageRank with a random seed in S. The volume of the set of such u is $> \mathrm{vol(S)/4}$.

A partitioning algorithm using PageRank

Algorithm(φ,s,b):

- Compute ε-approximate Pagerank $p=pr(\alpha,s)$ with $\alpha=0.1/(\varphi^2\ b)$, $\varepsilon=2^{-b}/b$.
- One sweep algorithm using p for finding cuts with conductance $< \varphi$.

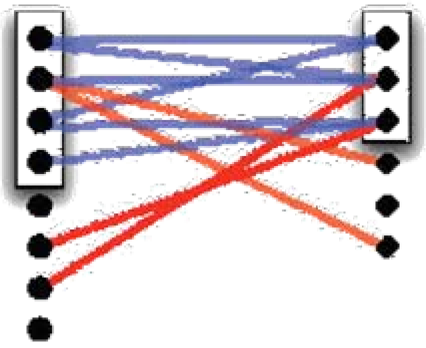
Performance analysis:

If s is in a set S with conductance $\Phi>\varphi^2 log\ s$, with constant probability, the algorithm outputs a cut C with condutance $< \varphi$, of size order s and $vol(C \cap S) > \frac{1}{4} vol(S)$.

(Improving previous bounds by a factor of φlog s.)

Finding submarkets in the sponsored search graph

Task. Find sets of advertisers and phrases that form isolated submarkets, with few edges leaving the submarket.

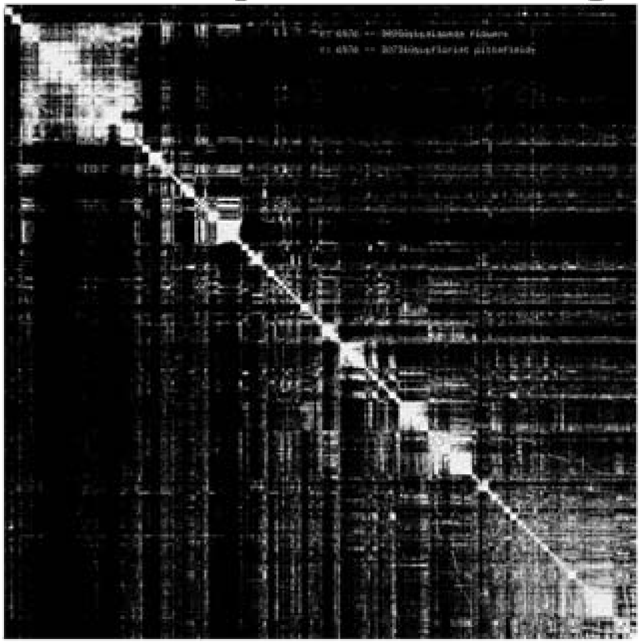


Applications

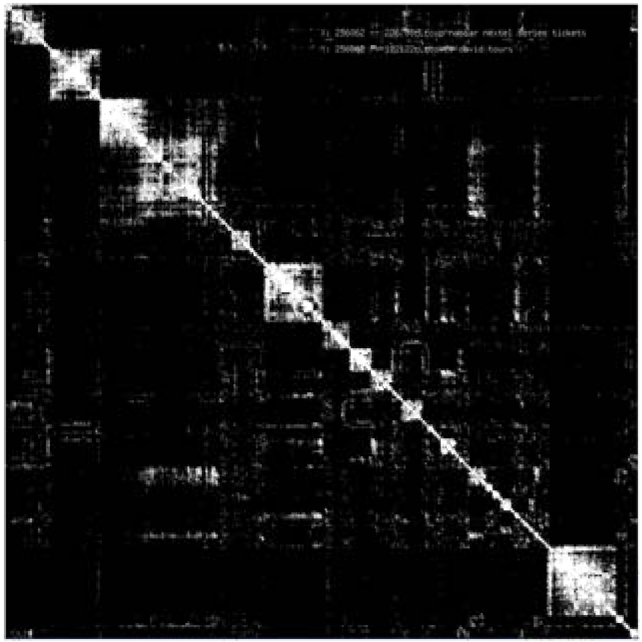
- Find groups of related phrases to suggest to advertisers.
- Find small submarkets for testing and experimentation.

There are thousands of submarkets

Full sponsored search graph

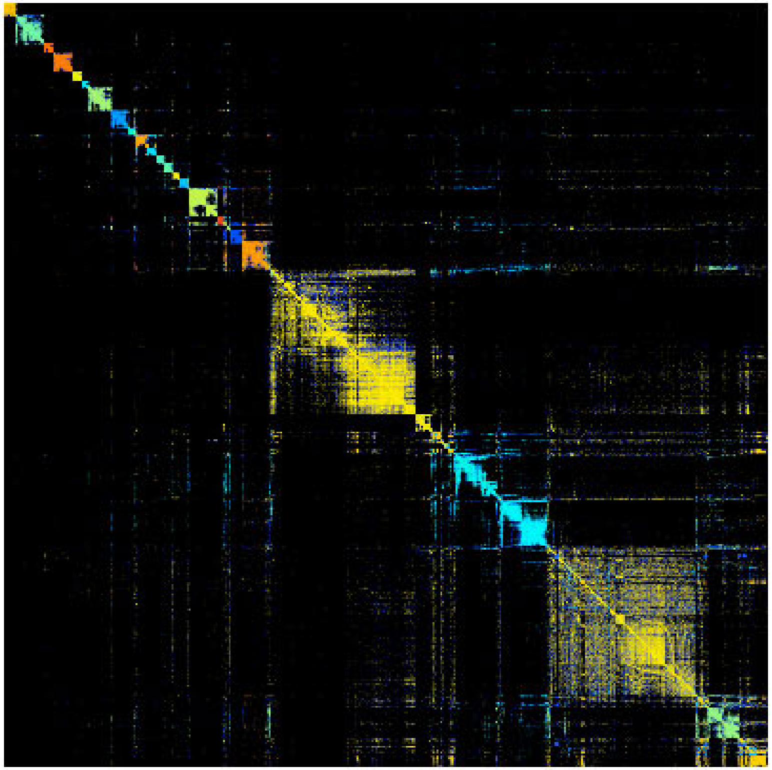


10x zoom

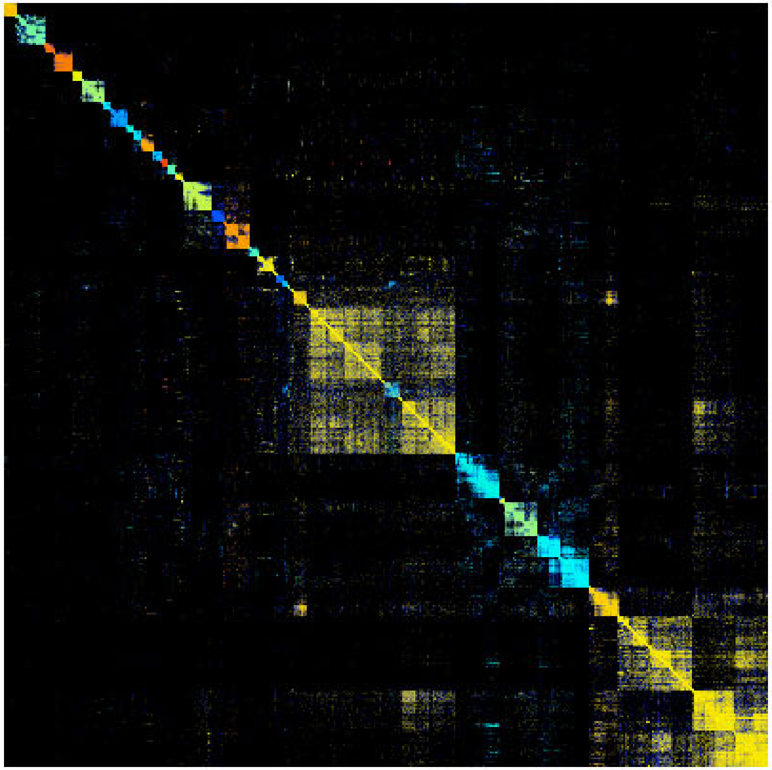


Courtesy of Reid Andersen

Internet Movie Database



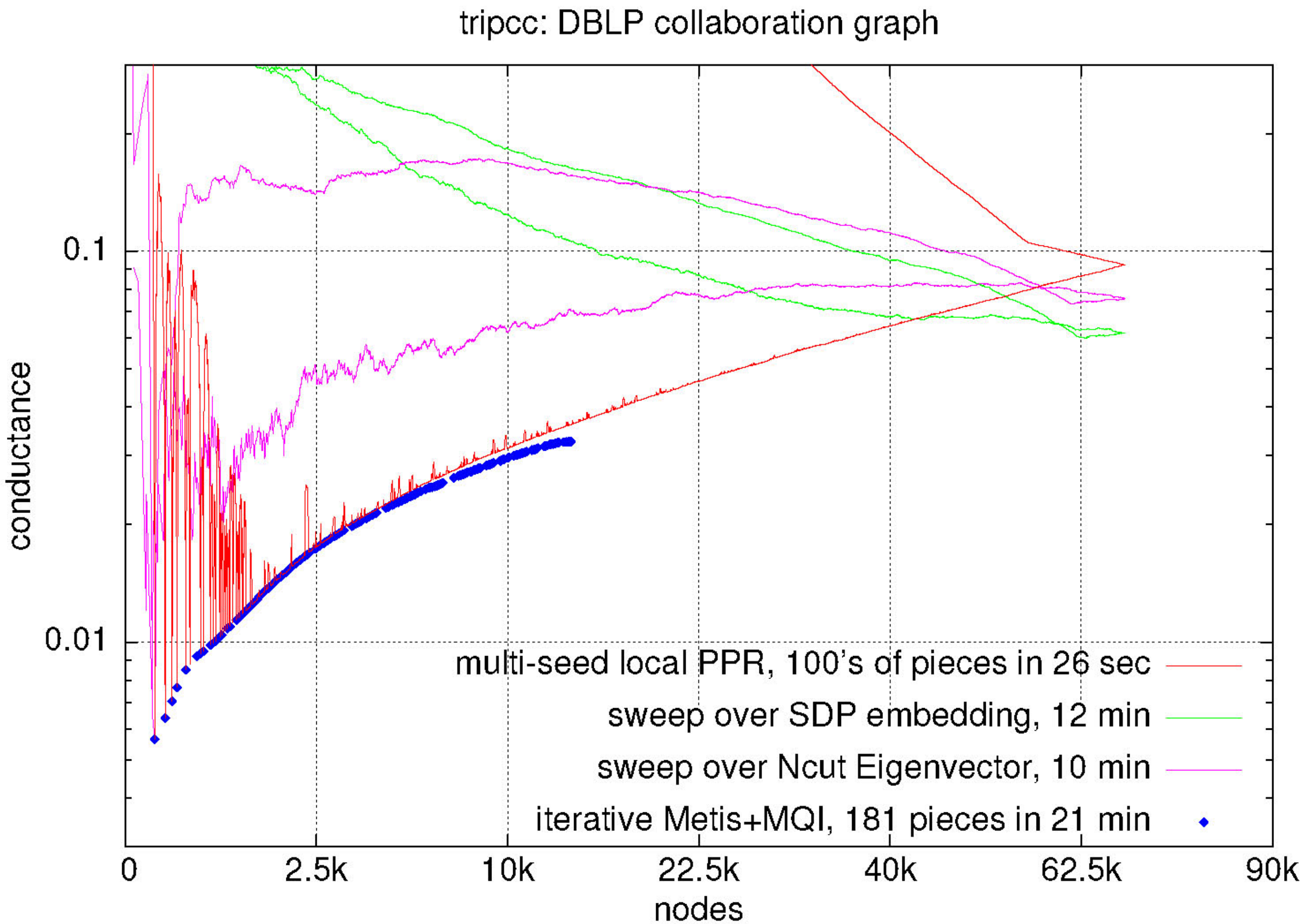
Local partitioning
(10 min)



Recursive spectral partitioning
(250 min)

Courtesy of Reid Andersen

Local PPR on DBLP graph



Kevin Lang 2007

4 Partitioning algorithm ⟷ 4 Cheeger inequalities

- graph spectral method — Fiedler '73, Cheeger, 60's
- random walks — Mihail 89; Lovasz, Simonovits, 90, 93; Spielman, Teng, 04
- PageRank — Andersen, Chung, Lang, 06
- heat kernel — Chung, PNAS , 08.

PageRank versus heat kernel

$$p_{\alpha,s} = \alpha \sum_{k=0}^{\infty} (1-\alpha)^k (sW^k)$$

$$\rho_{t,s} = e^{-t} \sum_{k=0}^{\infty} s \frac{(tW)^k}{k!}$$

Geometric sum

Exponential sum

PageRank versus heat kernel

$$p_{\alpha,s} = \alpha \sum_{k=0}^{\infty} (1-\alpha)^k (sW^k)$$

Geometric sum

$$\rho_{t,s} = e^{-t} \sum_{k=0}^{\infty} s \frac{(tW)^k}{k!}$$

Exponential sum

$$p = \alpha + (1-\alpha) pW$$

recurrence

$$\frac{\partial \rho}{\partial t} = -\rho(I - W)$$

Heat equation

Definition of heat kernel

$$H_t = e^{-t}(I + tW + \frac{t^2}{2}W^2 + \ldots + \frac{t^k}{k!}W^k + \ldots)$$

$$= e^{-t(I-W)}$$

$$= e^{-tL}$$

$$= I - tL + \frac{t^2}{2}L^2 + \ldots + (-1)^k \frac{t^k}{k!}L^k + \ldots$$

$$\frac{\partial}{\partial t}H_t = -(I-W)H_t$$

$$\rho_{t,s} = sH_t$$

Partitioning algorithm using the heat kernel

Theorem:

$$\left|\rho_{t,u}(S) - \pi(S)\right| \leq \sqrt{\frac{vol(S)}{d_u}} e^{-t\kappa_{t,u}^2/4}$$

where $\kappa_{t,u}$ is the minimum Cheeger ratio over sweeps by using heat kernel pagerank over all u in S.

Partitioning algorithm using the heat kernel

Theorem:

$$\left|\rho_{t,u}(S) - \pi(S)\right| \le \sqrt{\frac{vol(S)}{d_u}}\, e^{-t\kappa_{t,u}^2/4}$$

where $\kappa_{t,u}$ is the minimum Cheeger ratio over sweeps by using heat kernel pagerank over all u in S.

Theorem: For $vol(S) \le vol(G)^{2/3}$,

$$\left|\rho_{t,S}(S) - \pi(S)\right| \ge e^{-th_S}.$$

(Improving the previous PageRank lower bound $1\text{-}t\, h_S$.)

Theorem:

$$\left|\rho_{t,S}(S)-\pi(S)\right| \geq (1-\pi(S))e^{-h_S t/(1-\pi(S))}$$

Sketch of a proof:

Consider $F(t) = -\log(\rho_{t,S}(S)-\pi(S))$

Show $\frac{\partial^2}{\partial t^2}F(t) \leq 0$

Then $\frac{\partial}{\partial t}F(t) \leq \frac{\partial}{\partial t}F(0) = \frac{\Phi_S}{1-\pi(S)}$

Solve and get $\left|\rho_{t,S}(S)-\pi(S)\right| \geq (1-\pi(S))e^{-h_S t/(1-\pi(S))}$

Random walks versus heat kernel

How fast is the convergence to the stationary distribution?

For what k, can one have

$$f W^k \to \pi \quad ?$$

Choose t to satisfy the required property.

Partitioning algorithm using the heat kernel

Using the upper and lower bounds,

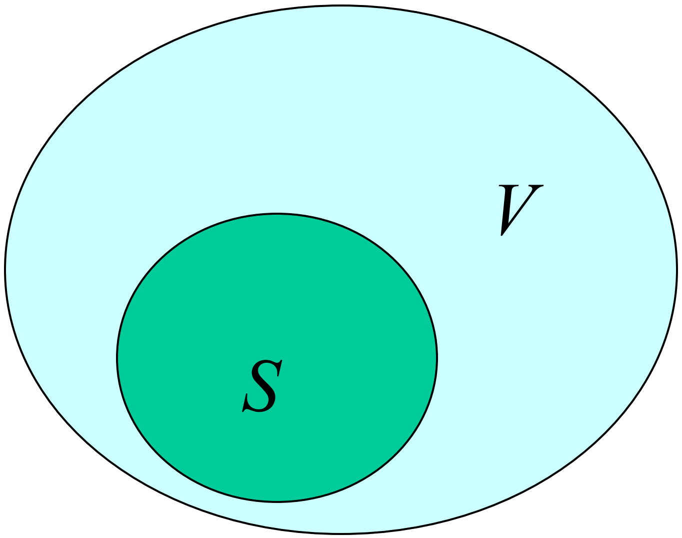
a Cheeger inequality can be obtained :

$$\Phi_S \geq \lambda_S \geq \frac{\kappa_S{}^2}{8} \geq \frac{\Phi_S{}^2}{8}$$

where λ_S is the Dirichlet eigenvalue of the Laplacian, and κ_S is the minimum Cheeger ratio over sweeps by using heat kernel with seeds S for appropriate t.

Dirichlet eigenvalues for a subset $S \subseteq V$

$$\lambda_S = \inf_f \frac{\sum_{u \sim v} (f(u) - f(v))^2}{\sum_w f(w)^2 d_w}$$



over all f satisfying the Dirichlet boundary condition:

$$f(v) = 0 \quad \text{for all } v \notin S.$$

Partitioning algorithm using the heat kernel

Using the upper and lower bounds,

a Cheeger inequality can be obtained :

$$\Phi_S \geq \lambda_S \geq \frac{\kappa_S^{\ 2}}{8} \geq \frac{\Phi_S^{\ 2}}{8}$$

where λ_S is the Dirichlet eigenvalue of the Laplacian, and κ_S is the minimum Cheeger ratio over sweeps by using heat kernel with seeds S for appropriate t.

Partitioning algorithm using the heat kernel

Using the upper and lower bounds,

a Cheeger inequality can be obtained :

$$\Phi_S \geq \lambda_S \geq \frac{\kappa_S^{\;2}}{8} \geq \frac{\Phi_S^{\;2}}{8}$$

where λ_S is the Dirichlet eigenvalue of the Laplacian, and κ_S is the minimum Cheeger ratio over sweeps by using heat kernel with seeds S for appropriate t.

Partitioning algorithm using the heat kernel

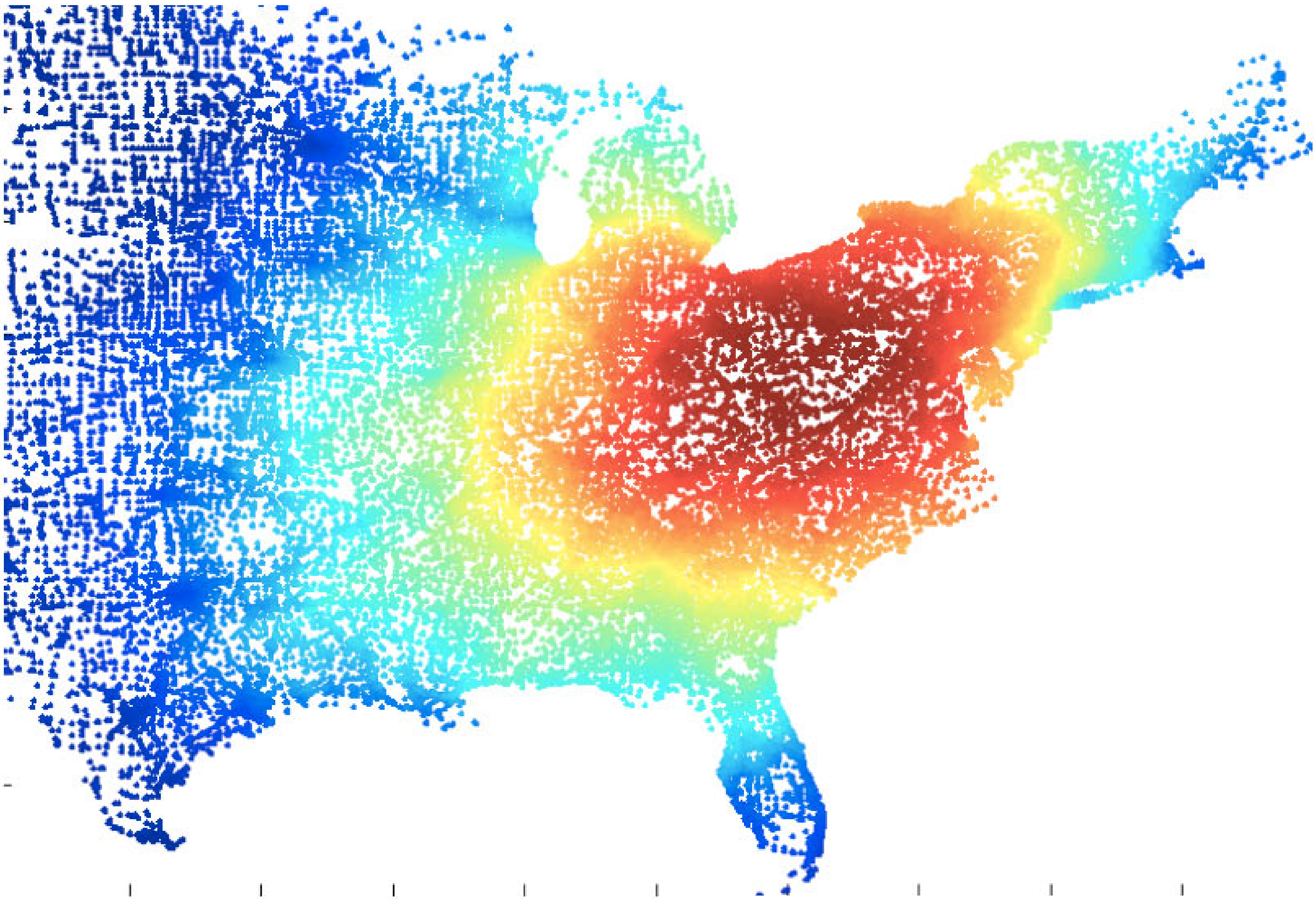
Using the upper and lower bounds,

a Cheeger inequality can be obtained :

$$\Phi_S \geq \lambda_S \geq \frac{\kappa_u^{\,2}}{8\log s} \geq \frac{\Phi_u^{\,2}}{8\log s}$$

where λ_S is the Dirichlet eigenvalue of the Laplacian, and κ_u is the minimum Cheeger ratio over sweeps by using heat kernel with a random seed in S. The volume of the set of such u *is* $>$ vol(S)/4.

What the sweep should look like



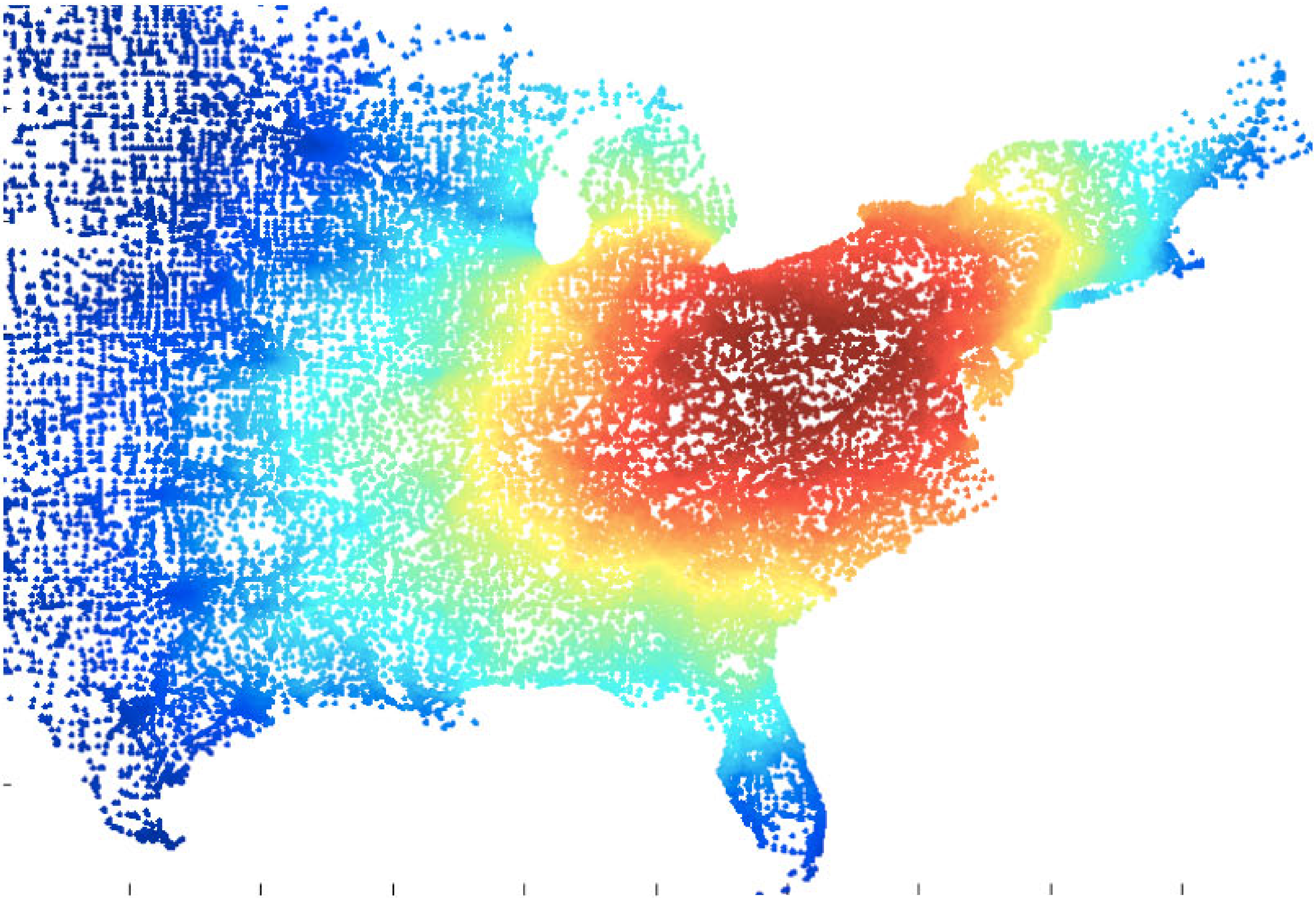
Courtesy of Reid Andersen

Future directions:

Use PageRank and the heat kernel pagerank to shed light on:

- The geometry of graphs?
- Solving combinatorial problems, such as covering, packing, matching, etc.
- Graph drawing, visualization
- Metric embedding …

What the sweep should look like



Courtesy of Reid Andersen