2011 SIAM Conference on Computational Science and Engineering
MS43 Spectral Calculations on Complex Networks

Multilevel Aggregation of Small-World Graphs with Application to Random-Walk Ranking

Hans De Sterck

Reno, NV, USA, March 1st, 2011

Hans De Sterck

UNIVERSITY OF Waterloo

Geoffrey Sanders*

Van Emden Henson

Lawrence Livermore National Laboratory

Lawrence Livermore National Laboratory, P. O. Box 808, Livermore, CA 94551

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344

Outline

I. **Introduction**

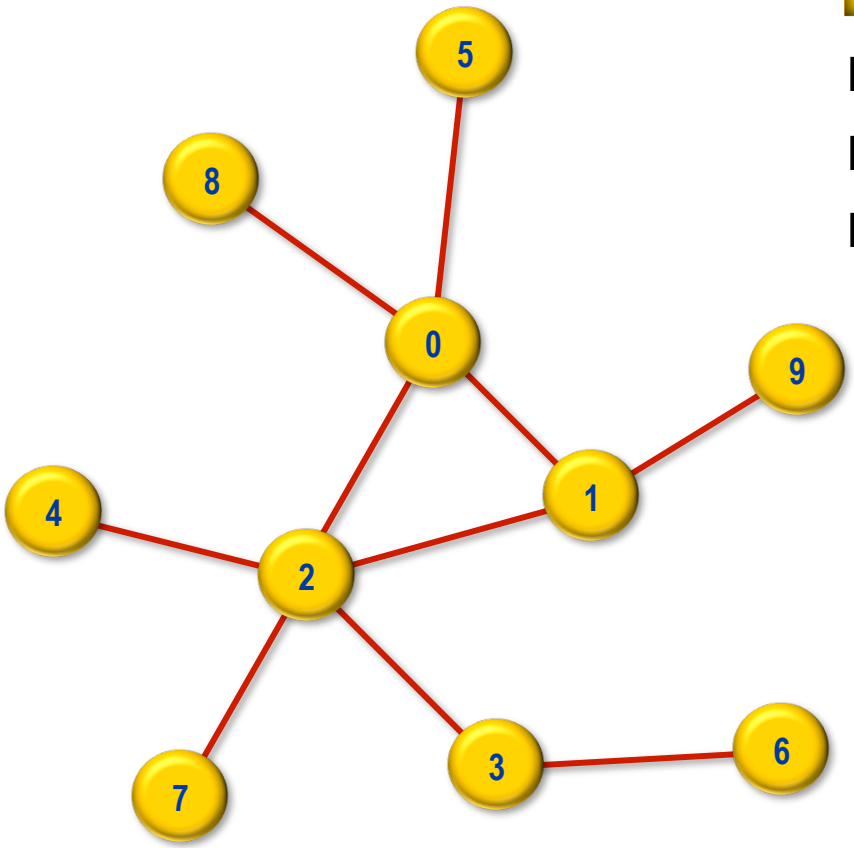
II. Multilevel Aggregation for Random Walks

III. Aggregation Methods

IV. Numerical Results

V. Conclusions and Further Work

Scale-Free Graph Problem: Barabasi-Alber Model

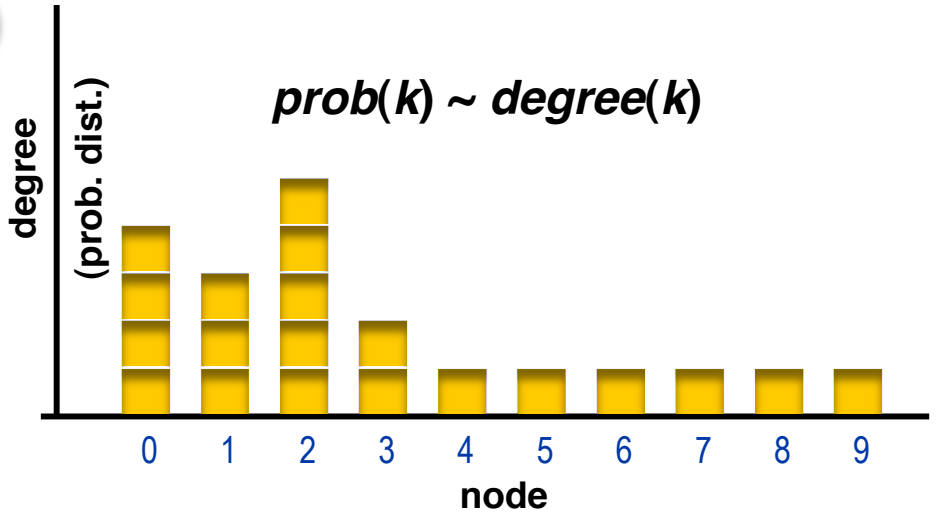


Preferential Attachment:

I. Start with a small ring.

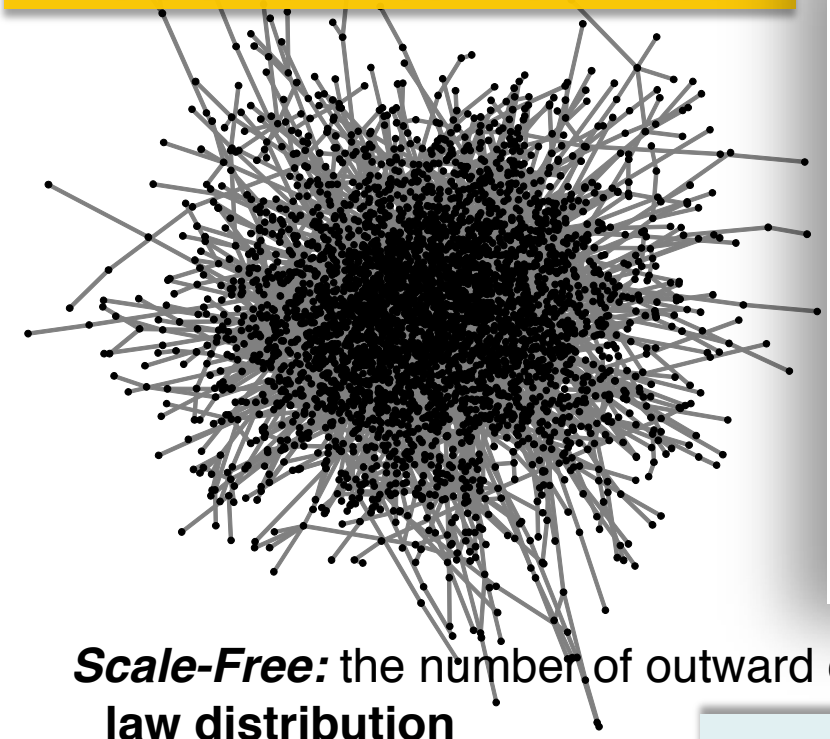
II. Add each new node with *p* edges (*p*=1).

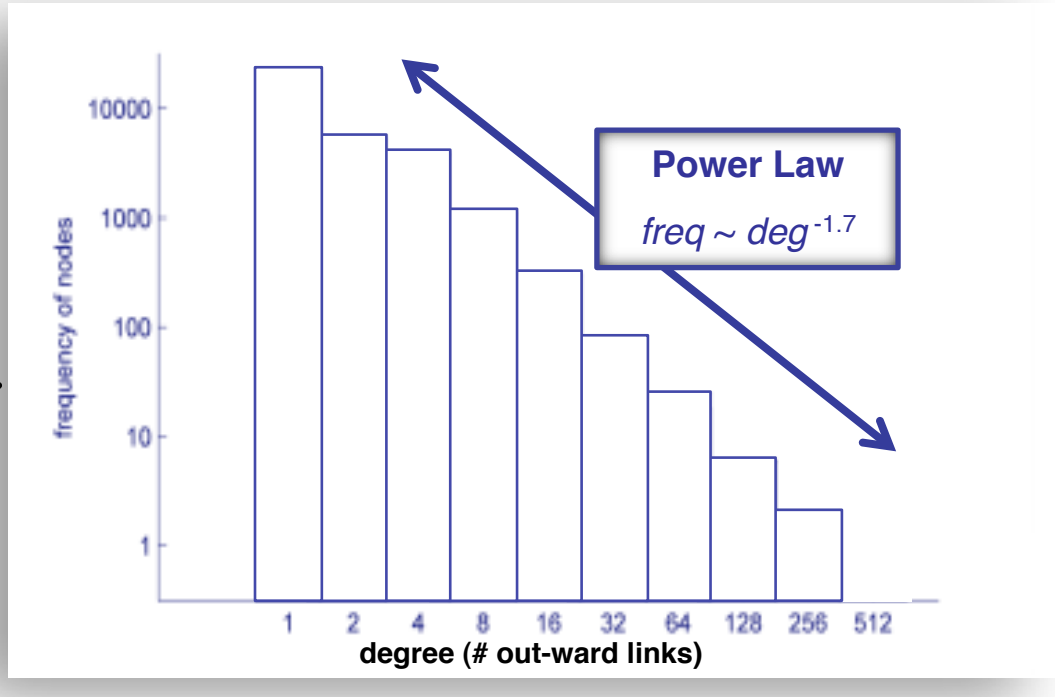
III. Edges connect with existing nodes preferentially:



Scale-Free Networks and Power-Law Distribution

Example. *Barabasi-Alber*
Preferential Attachment Model





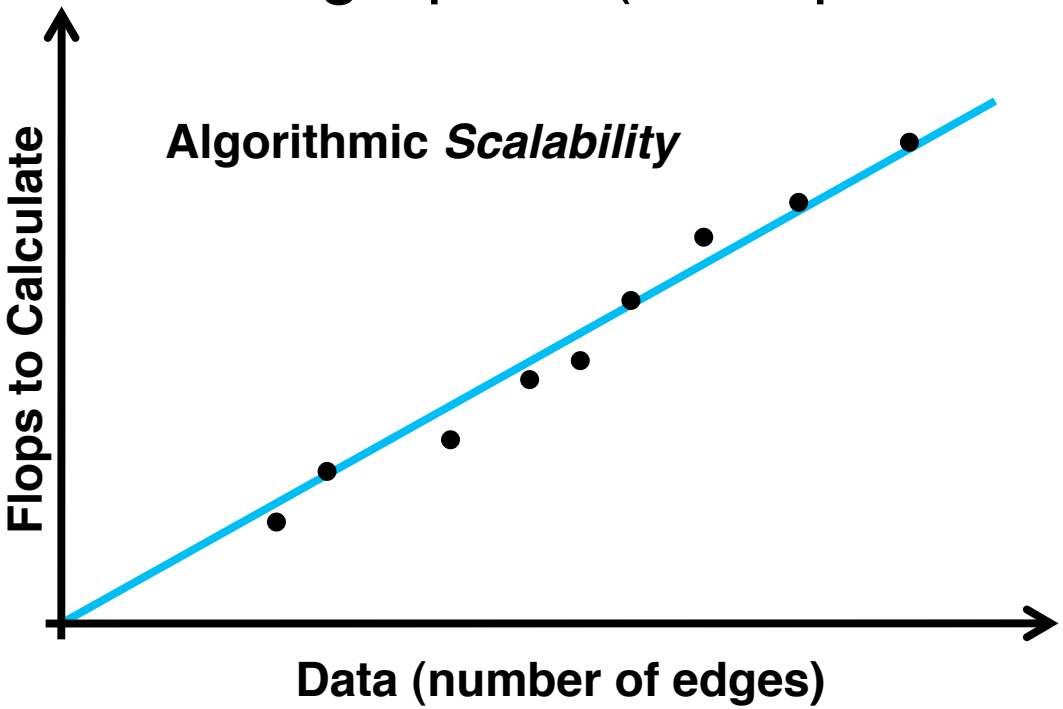
Scale-Free: the number of outward connections for each node has a **power-law distribution**

$$g(k) \sim k^{-\beta} \quad \text{with} \quad 1.5 < \beta < 4.5.$$

Small-World: such graphs tend to have **small diameter**, independent of size

Broad Question:

Can we develop *scalable* eigensolvers for such networks graphs? (example: random walk)



Parallel Scalability (Time)

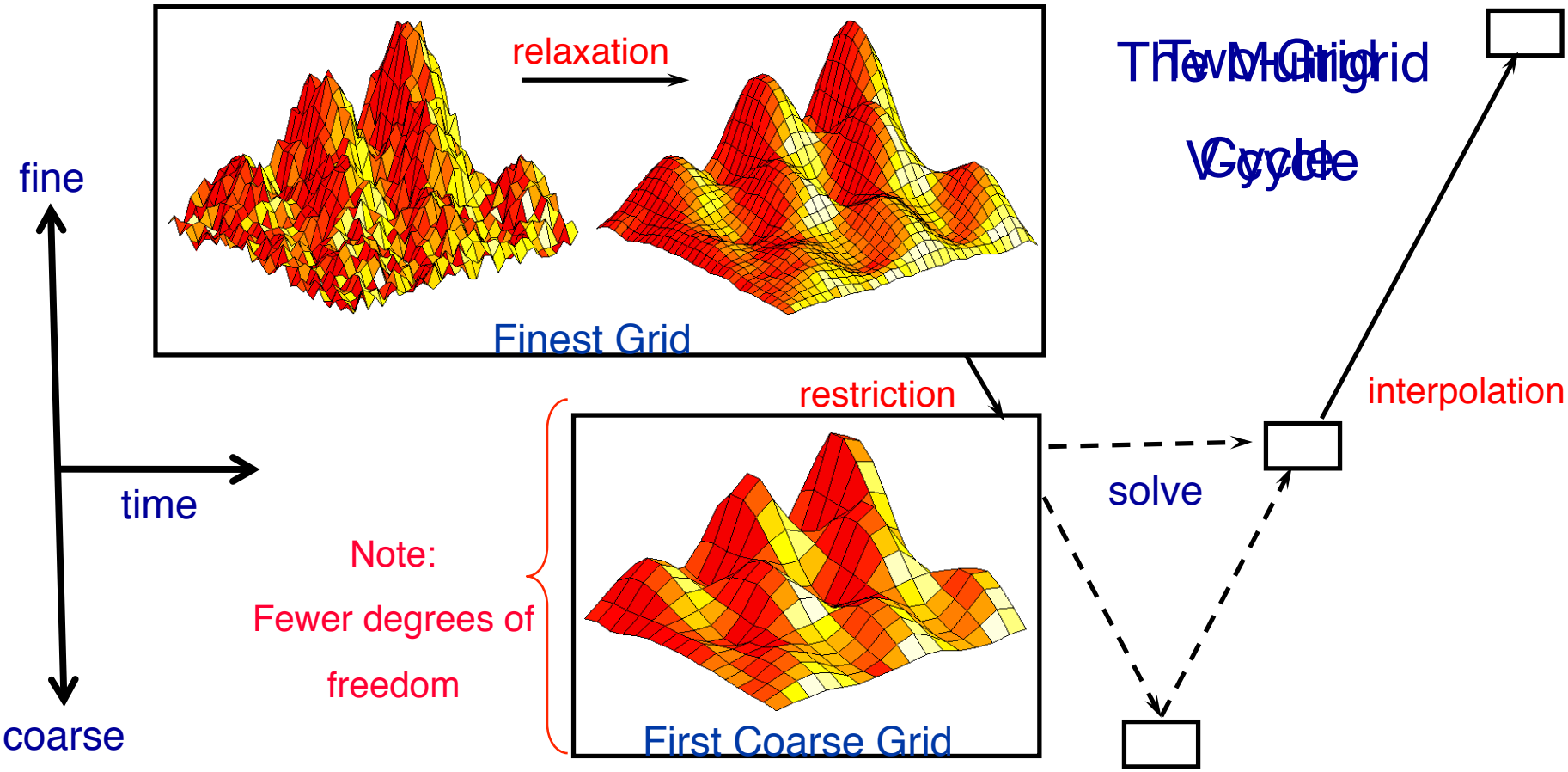
Targeted computing environments

- ***Parallel***
- ***Cloud***

Multilevel hierachies have been successfully employed to accelerate many similar computationally intensive tasks (linear solves for sparse PDE matrices)

Basic Multigrid for Poisson on a Square [1]

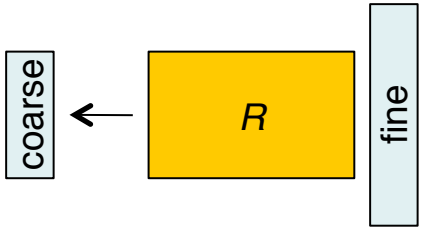
Consider solving $A\mathbf{x} = \mathbf{b}$ (Discrete Poisson Eq.)



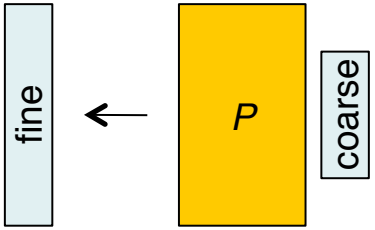
[1] *Slide courtesy of Rob Falgout, HYPRE project*

Restriction and Interpolation Goals

Between any two grids we must choose intergrid transfer operators:



Restriction, *R*, moves information from the fine level to the coarse level.



Interpolation, *P*, moves information from the coarse level to the fine level.

These operators should be sparse and accurately represent smooth error.

Are multilevel approaches useful for spectral calculations on scale-free graphs?

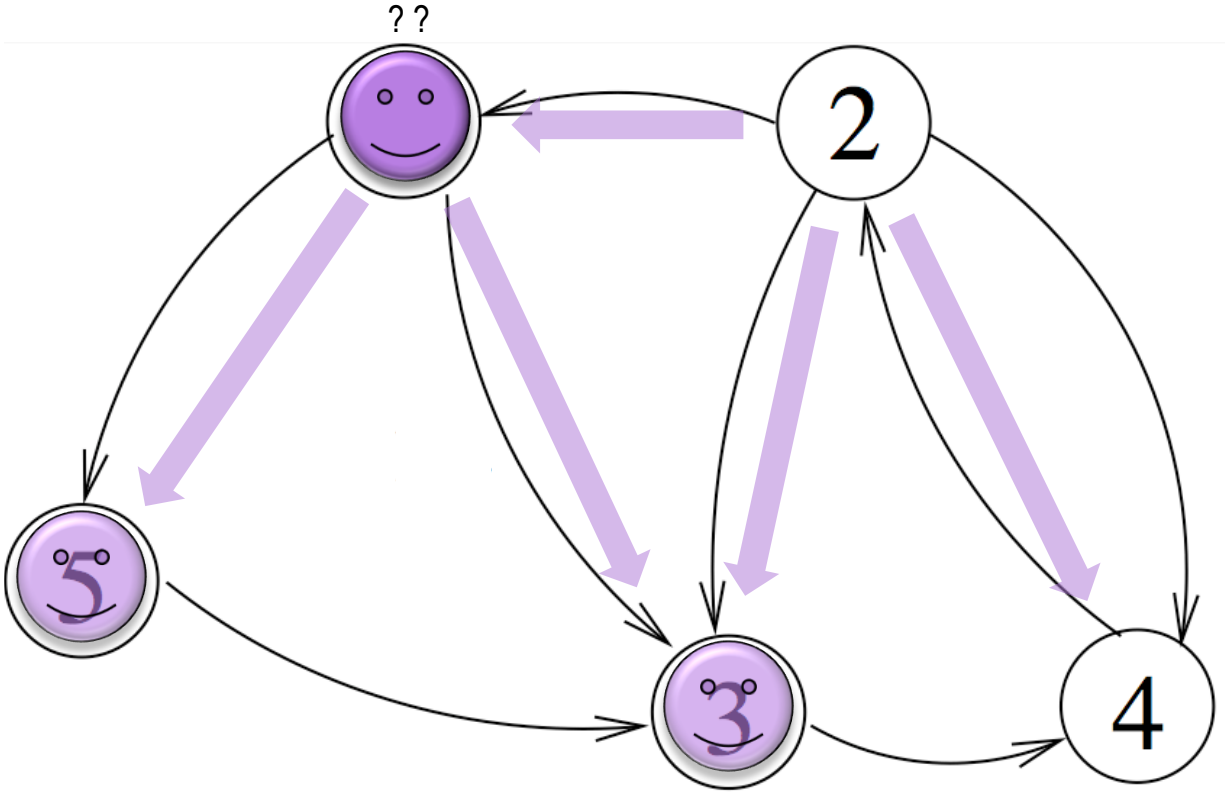
Main Example: Random-Walk Ranking

- **Example:** Random-walk ranking is a simple eigenproblem because it only involves calculating a single eigenvector.
- Define the *importance* of a node to be the likelihood that a long random walk through the graph would visit the node.
- **Task:** rank the nodes by their *importance*.
- For a network of webpages, this *importance* is used to sort query results in Google's search engine (PageRank, [2]).
- For more general networks, this *importance* can identify which nodes are central to the network.

[2] *Google PageRank,* Larry Page and Sergey Brin, 1995

Main Example: Random-Walk Ranking

Consider taking long, **random walks** on a **directed graph**:



Goal: Predict the likelihood of a random-walker's presence at each node.

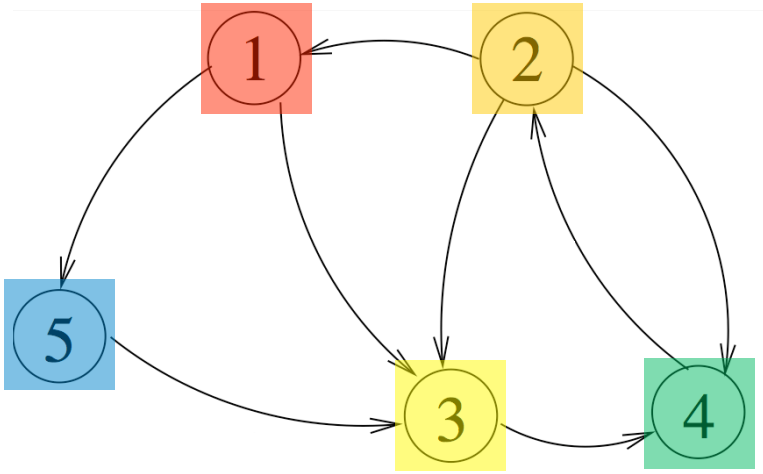
Main Example: Random-Walk Ranking

Stochastic approach, Markov Chains.

$$B = \begin{bmatrix} 0 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1/2 & 1/3 & 0 & 0 & 1 \\ 0 & 1/3 & 1 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Column stochastic

$$0 \leq b_{ij} \leq 1 \; \forall i, j \qquad \mathbf{1}^t B = \mathbf{1}^t$$



B_{ij} is probability of moving from *j* to *i*

Probability distribution, initially: $\mathbf{x}_0^T = [\,1,\; 0,\; 0,\; 0,\; 0\,]$

Distribution for *k*-length walk: $\mathbf{x}_k = B\mathbf{x}_{k-1} \qquad \mathbf{x}_k = B^k\mathbf{x}_0$

Steady-state distribution satisfies: $B\mathbf{x} = \mathbf{x},$

Steady-state distribution: $\mathbf{x}^T = [2/19 \;\; 6/19 \;\; 4/19 \;\; 6/19 \;\; 1/19]$

Main Example: Random-Walk Ranking

Ranking: node *i* is considered *more important* than node *j* if $x_i > x_j$ where

$$Bx = x, \qquad \|x\|_1 = 1, \qquad x \geq 0,$$

General Properties of This System

- ***Sparse***
- ***Nonsymmetric***
- ***Eigenproblem with known eigenvalue,* 1.**

$$1^T B = 1^T$$

Further Assumptions

- **Graph is *irreducible.***
- **Graph is *aperiodic*.**

Results:

I. ***Largest eigenvalue of B is uniquely* 1.**

II. ***B has a unique and positive eigenvector associated with eigenvalue* 1.**

Classical Iterative Methods and Scalability

$$\mathbf{x}_k = B\mathbf{x}_{k-1}$$

P**ower method** solves this problem, convergence rate is the **subdominant eigenvalue,**

$$|\lambda_2| := \max|\lambda| \quad \text{for} \quad \lambda \in \Sigma(B) \setminus \{1\}.$$

If $|\lambda_2| \approx 1$, the power method is *slowly mixing.*

If $\mathrm{Re}\ \lambda \approx 1$, other related iterative methods have *slow convergence rates*.

For many classes of network problems, $\lambda_2 \to 1$ as *n* grows large.

Multilevel methods can be used to overcome these slowness limitations (see our papers on Markov chains for lattice problems, queueing problems, etc.).

Will they work for scale-free graphs?

Outline

I. ~~Introduction~~

II. **Multilevel Aggregation for Random Walks**

III. Aggregation Methods

IV. Numerical Results

V. Conclusions and Further Work

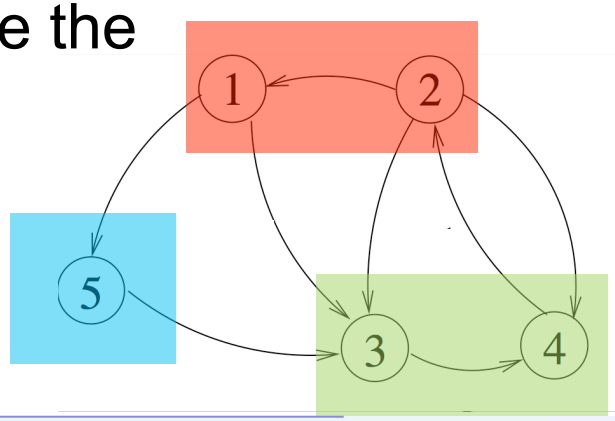
Pure Aggregation for Markov Chains [3]

$$B\mathbf{x} = \mathbf{x}, \qquad \|\mathbf{x}\|_1 = 1, \qquad \mathbf{x} \geq 0,$$

$$A = I - B \qquad A\mathbf{x} = 0$$

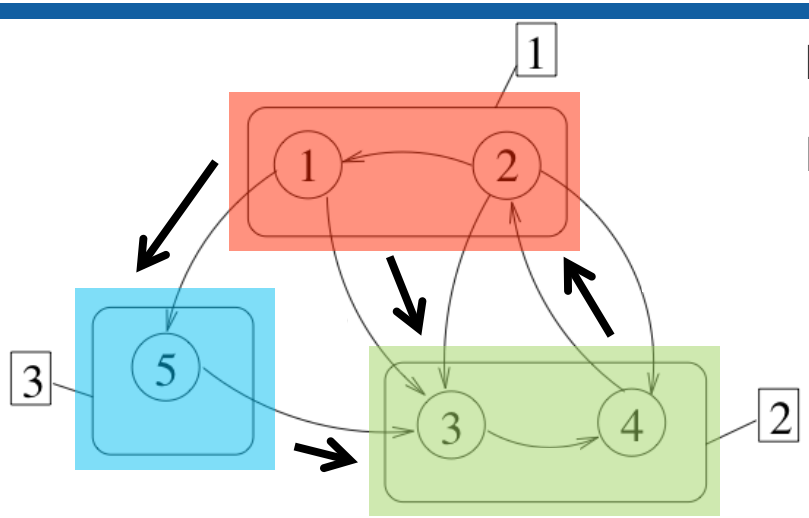
- Use a simple **relaxation** technique to efficiently resolve the *local character* of the steady-state vector.
  - Weighted Jacobi (essentially shifted power method)

- Use a **coarse-grid update** to accelerate the convergence to the *global character*.
  - Aggregation



[3] Takahashi, 1975, Horton et al, 1994, Krieger, 1995

Pure Aggregation for Markov Chains



$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

aggregation matrix

q_{iJ} = 1 iff node *i* is in *J*-th aggregate

Restriction: $R = Q^T$

Interpolation: $P = \mathrm{diag}(\mathbf{x}_k) Q \mathrm{diag}(Q^T \mathbf{x}_k)^{-1}$

$$RBP\mathbf{x}_c = RP\mathbf{x}_c \qquad B_c = RBP$$

Coarse-level problem: $B_c \mathbf{x}_c = \mathbf{x}_c$

$b_{c,IJ}$ probability of moving from *J* to *I*

Column Stochastic: $\mathbf{1}_c^T B_c = \mathbf{1}_c^T$

Solve coarse-level problem for $\mathbf{x}_c$

Update: $\mathbf{x}_{k+1} \leftarrow P\mathbf{x}_c$

Pure Aggregation for Markov Chains

$$A = I - B \qquad A\,\mathbf{x} = 0, \qquad \|\mathbf{x}\|_1 = 1, \qquad \mathbf{x} \geq 0,$$

Coarse-level matrix: $A_c = R\,A\,P$

Coarse-level problem: $A_c \mathbf{x}_c = \mathbf{0}$

General Multilevel Aggregation Algorithm

Algorithm 1: Multilevel Aggregation
$\mathbf{x} \longleftarrow \mathbf{MA}(A, \mathbf{x}, \nu_1, \nu_2, \mu)$:

if *not on coarsest level* **then**
  $\mathbf{x} \leftarrow \text{Relax}(A, \mathbf{x})$ ν_1 times.
  Build Q.
  $R \leftarrow Q^T$ and $P \leftarrow \text{diag}(\mathbf{x})\, Q$.
  $A_c \leftarrow R\, A\, P$.
  /* first coarse-level solve.
  $\mathbf{x}_c \leftarrow \mathbf{MA}(A_c\, \text{diag}(Q^T \mathbf{x})^{-1}, Q^T \mathbf{x}, \nu_1, \nu_2)$
  /* secondary coarse-level solves.
  for $k = 2, \ldots \mu$ **do**
    $\mathbf{x}_c \leftarrow \mathbf{MA}(A_c\, \text{diag}(Q^T \mathbf{x})^{-1}, \mathbf{x}_c, \nu_1, \nu_2)$
  end
  /* coarse-level correction.
  $\mathbf{x} \leftarrow P\, (\text{diag}(Q^T \mathbf{x}))^{-1}\, \mathbf{x}_c$
  $\mathbf{x} \leftarrow \text{Relax}(A, \mathbf{x})$ ν_2 times.
else
  $\mathbf{x} \leftarrow$ direct solve of $A\, \mathbf{x} = 0$, $\|\mathbf{x}\|_1 = 1$.
end

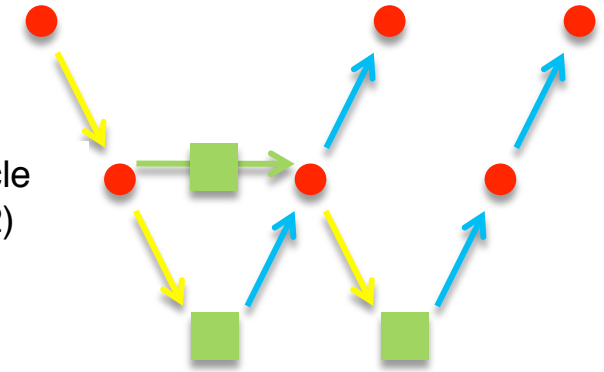
Relaxation: perform a low number of simple iterations so the local character of the approximation looks like the steady-state.

Choose an **aggregation**, use it and the current approximation to form **restriction** and **interpolation** operators.

Form a **coarse-level problem** and *solve* it using recursive application of this method.

Update fine-level approximation by **interpolating** coarse-level approximation.

W-cycle
(μ=2)



To Improve Convergence: Smoothed Aggregation

Apply simple sparse **smoother** to the **rows of restriction** and the **columns of interpolation:**

$$R_s = RS^T \qquad P_s = SP$$

Coarse-level matrix:

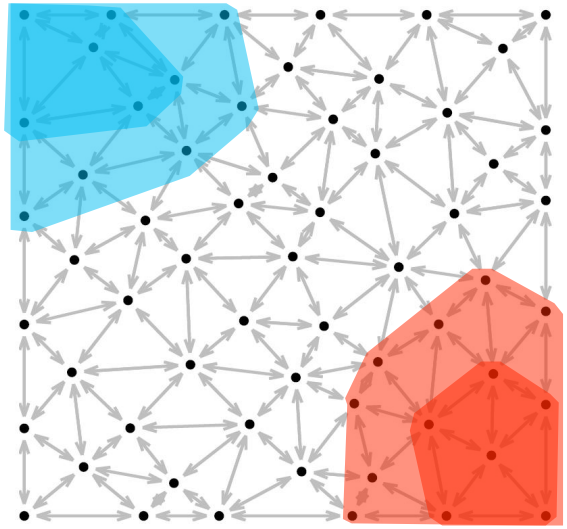
$$A_c = R_s A P_s$$

Coarse-level problem:

$$A_c \mathbf{x}_c = \mathbf{0}$$

Improves representation of original problem, but <u>increases number of coarse edges.</u>

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad SQ = \begin{bmatrix} \times & 0 & 0 \\ \times & \times & 0 \\ \times & \times & 0 \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix}$$



Outline

I. ~~Introduction~~

II. ~~Multilevel Aggregation for Random Walks~~

III. **Aggregation Methods**

IV. Numerical Results

V. Conclusions and Further Work

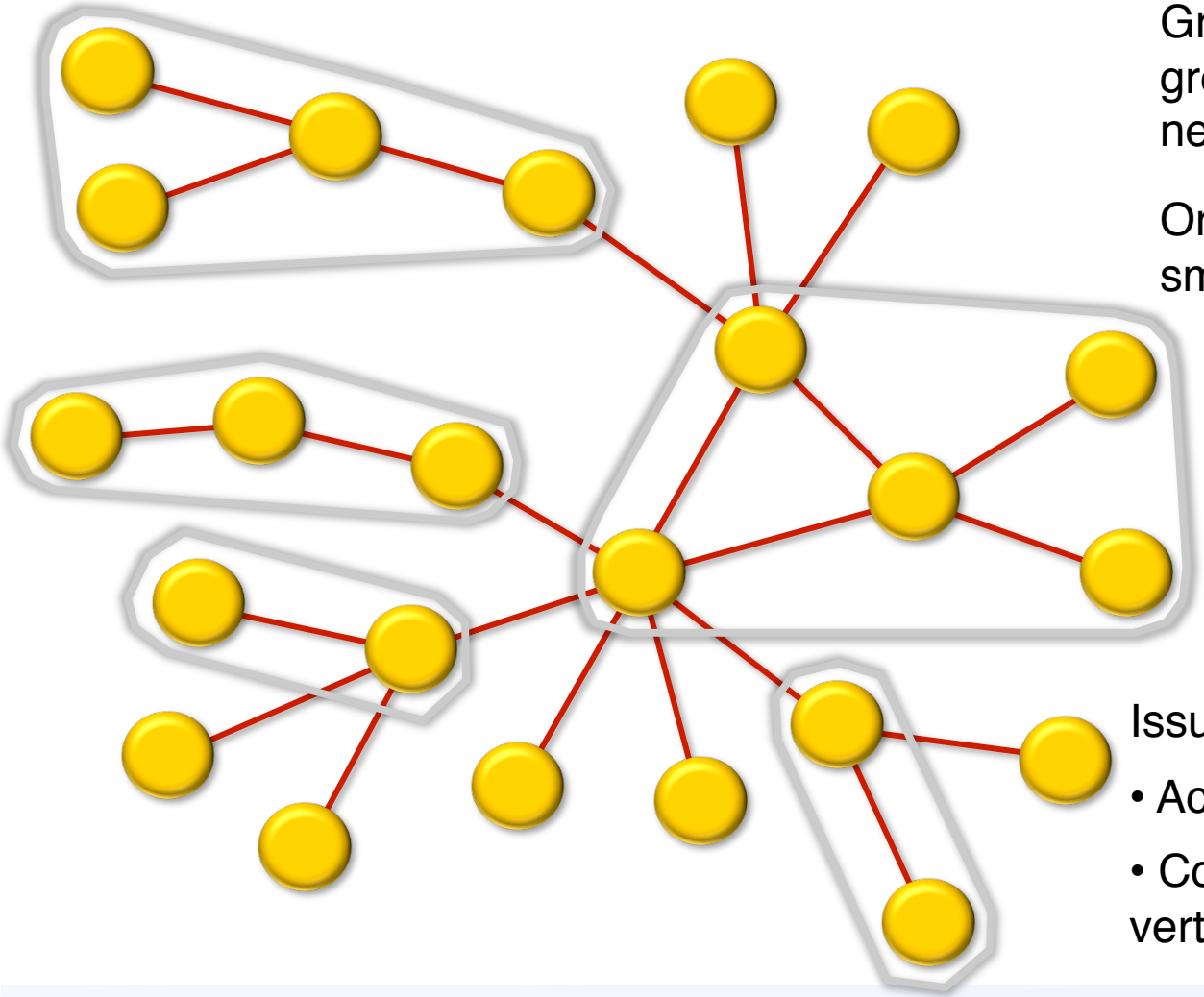
This Talk: Compare Three Aggregation Methods

1. pure (unsmoothed) neighborhood aggregation

2. smoothed neighborhood aggregation

3. new leaf-based aggregation (pure) with special interpolation formula (our 2011 paper in Computing and Informatics)

1. Neighborhood-Based Aggregation



Group nodes to ensure each group contains a proper neighborhood.

Originally designed for operator smoothing on mesh-like graphs.

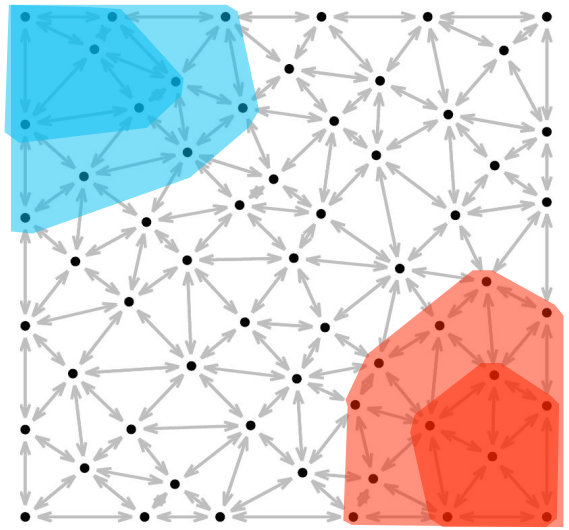
Issues with scale-free graphs:

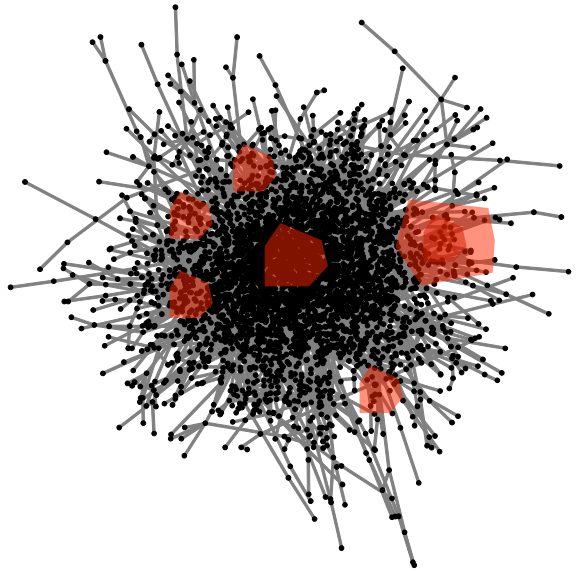
- Accuracy (large aggregates)
- Complexity (high-degree vertices are grouped)

2. Smoothed Neighborhood Aggregation

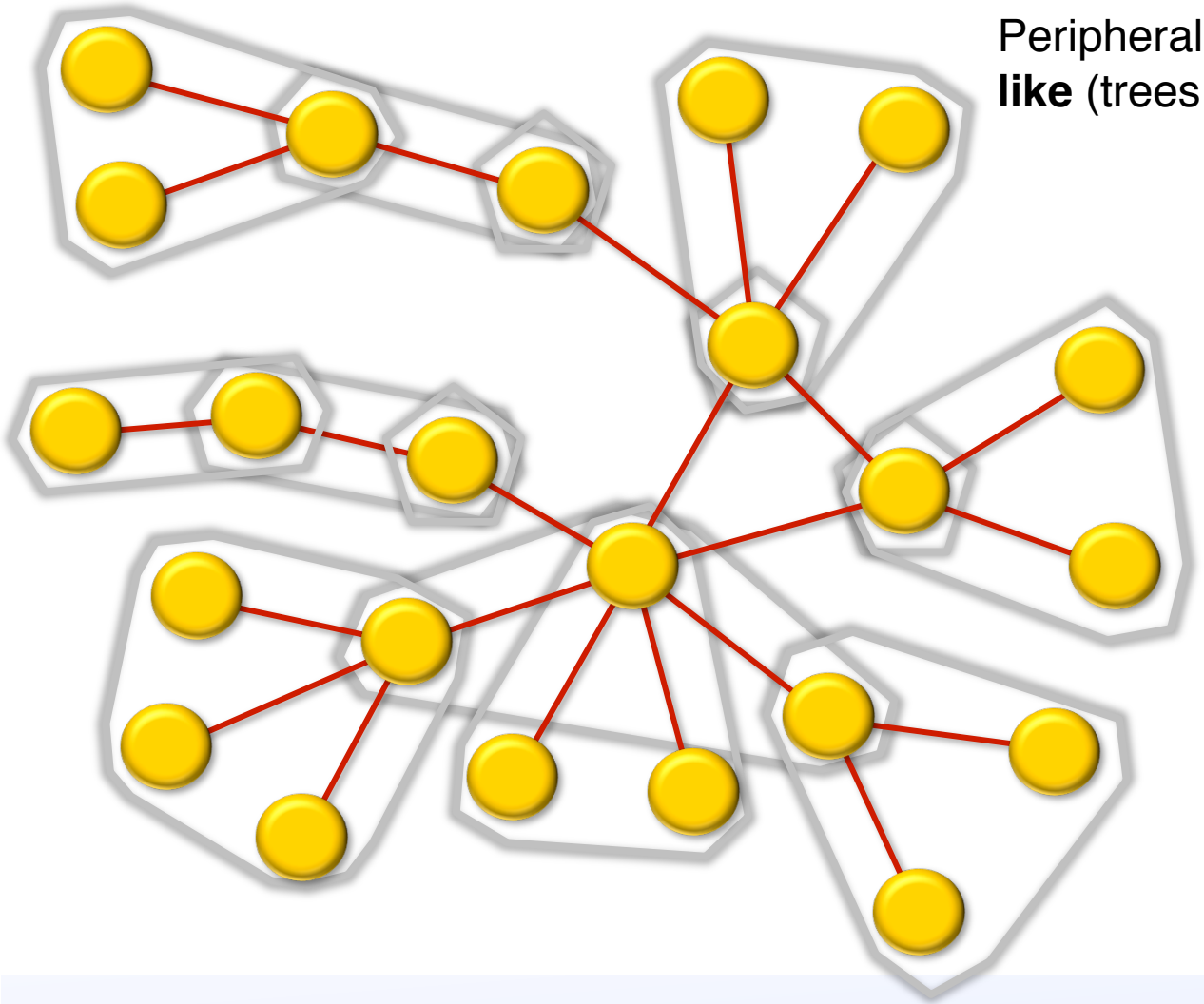
Improves representation of original problem,
but increases number of coarse edges.

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad Q_s = \begin{bmatrix} \times & 0 & 0 \\ \times & \times & 0 \\ \times & \times & 0 \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix}$$





3. Leaf-Based Aggregation (for Tree-Like Structure)



Peripheral structure often **highly Tree-like** (trees hang from generating nodes)

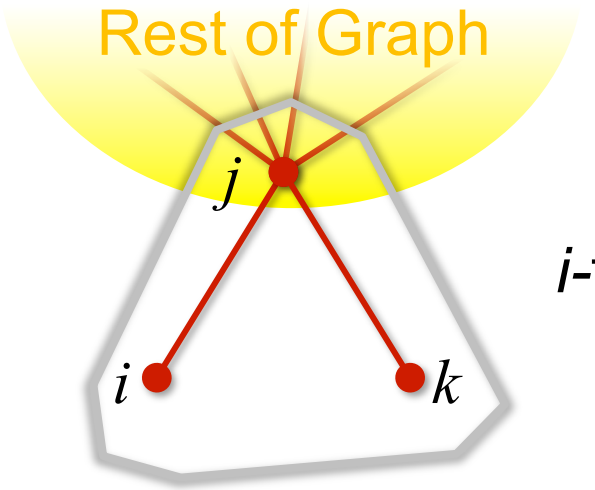
Leaves depend on one node.

Aggregate each leaf with its parent. For parents with no leaf as children, group alone.

Coarsened graph is again highly tree-like.

Apply technique recursively.

Leaf-Based Aggregation for Highly Tree-like Graphs



Leaves: vertices of degree 1.

$$(A - \lambda I)\mathbf{x} = \mathbf{0}$$

i-th eq'n: $(a_{ii} - \lambda)x_i + a_{ij}x_j = 0$

$$x_i = \frac{-a_{ij}}{a_{ii} - \lambda} x_j$$

Interpolation Formula:

$$\lambda = 0 \quad \Longrightarrow \quad x_i = \frac{-a_{ij}}{a_{ii}} x_j$$

Group the leaves with their parents and set the values of interpolation based on the entries of *A*.

Outline

I. ~~Introduction~~

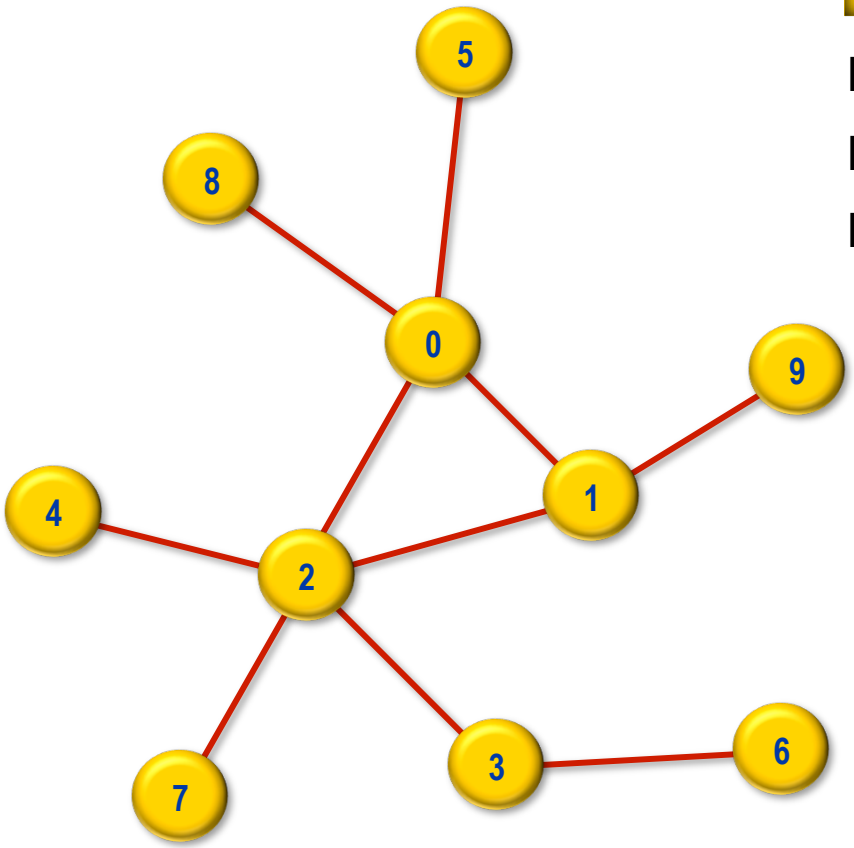
II. ~~Multilevel Aggregation for Random Walks~~

III. ~~Aggregation Methods~~

IV. **Numerical Results**

V. Conclusions and Further Work

Example: Barabasi-Alber Model (p=1, p=2)

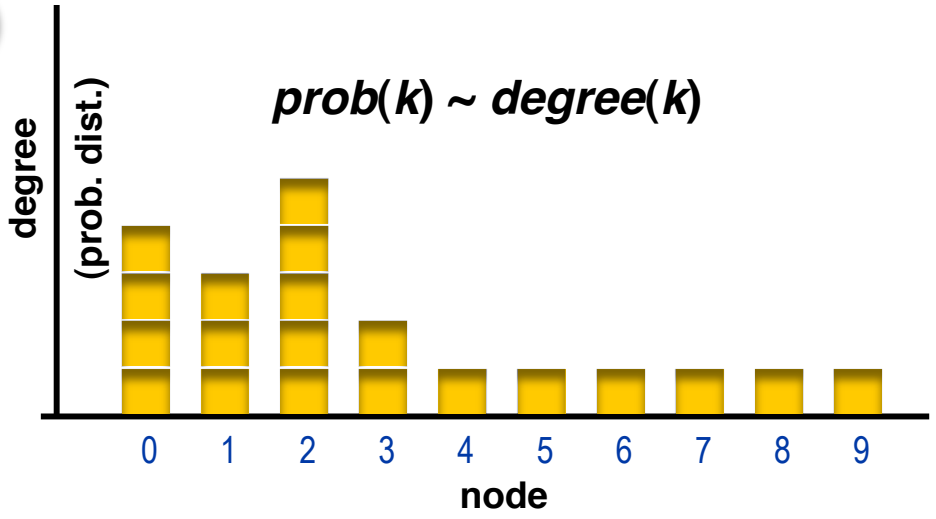


Preferential Attachment:

I. Start with a small ring.

III. Add each new node with *p* edges.

IV. Edges connect with existing nodes preferentially:



Outline

I. ~~Introduction~~

II. ~~Multilevel Aggregation for Random Walks~~

III. ~~Aggregation Methods~~

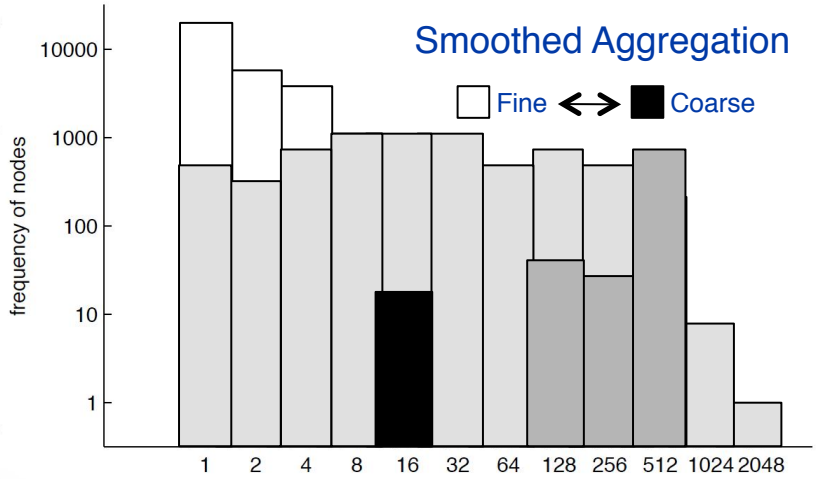
IV. **Numerical Results**

- **Neighborhood Aggregation**

V. Conclusions and Further Work

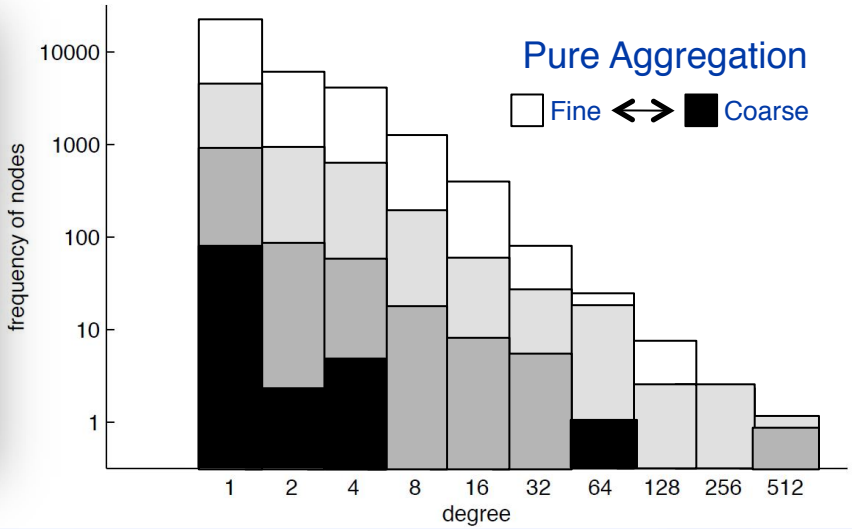
Numerical Results: Neighborhood Aggregation

	Smoothed Aggregation		Pure Aggregation	
n	$p=1$	$p=2$	$p=1$	$p=2$
1024	3.23	3.89	1.23	1.34
2048	3.37	5.17	1.22	1.36
4096	4.34	7.03	1.23	1.38
8192	5.63	9.14	1.24	1.41
16384	7.48	11.42	1.24	1.42
32768	9.54	14.62	1.24	1.44



Smoothed Aggregation cannot produce a hierarchy with bounded complexity.

Pure Aggregation does produce a hierarchy with bounded complexity and retains power-law distribution.



Numerical Results: Neighborhood Aggregation

nodes

Interpolation type

$$A_c = R_s A P_s$$

New connections per new node

	Smoothed Aggregation		Pure Aggregation	
n	$p=1$	$p=2$	$p=1$	$p=2$
1024	3.23	3.89	1.23	1.34
2048	3.37	5.17	1.22	1.36
4096	4.34	7.03	1.23	1.38
8192	5.63	9.14	1.24	1.41
16384	7.48	11.42	1.24	1.42
32768	9.54	14.62	1.24	1.44

Operator Complexity: $C_{op} = \dfrac{\textit{Total Nonzeros in Operator Hierarchy}}{\textit{Nonzeros in Original Problem Matrix}}$

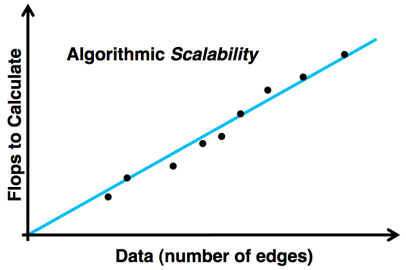
Numerical Results

- **Operator Complexity** of multilevel hierarchies:

  C_{op} = (*total nonzeros*)/(*nonzeros on fine-grid*)

- **Number of iterations to converge:**

  $\| B\mathbf{x}_k - \mathbf{x}_k \|_1 < 10^{-6} \| B\mathbf{x}_0 - \mathbf{x}_0 \|_1$



Algorithmically scalable means C_{op} and k are bounded, independent of the problem size, n.

The number C_{op} is an indicator of the cost of a single multilevel cycle.
The number kC_{op} is an indicator of the cost to calculate $\boldsymbol{x}$.

Ranking Problem, Neighborhood Aggregation

Smoothed Aggregation V-cycles

			Iteration Counts	
n	C_{op}	levs	SAM	SAM+
1024	3.26	4	166	28
2048	3.53	4	224	31
4096	4.82	4	303	38
8192	5.63	4	430	63
16384	7.48	4	670	79
32768	9.54	5	862	83

Pure Aggregation V-cycles

			Iteration Counts	
n	C_{op}	levs	V	V+
1024	1.23	4	355	59
2048	1.23	4	366	58
4096	1.22	4	696	69
8192	1.24	5	745	82
16384	1.24	5	>999	127
32768	1.23	5	>999	142

Pure Aggregation W-cycles

16384	1.55	5	>999	96
32768	1.54	5	>999	109

Large, growing iteration counts (not scalable).

W-cycles are slightly better than V-cycles.

Pure Aggregation outperforms smooth aggregation, due to operator complexities.

Algorithmic Scalability is not achieved.

Outline

I. ~~Introduction~~

II. ~~Multilevel Aggregation for Random Walks~~

III. ~~Aggregation Methods~~

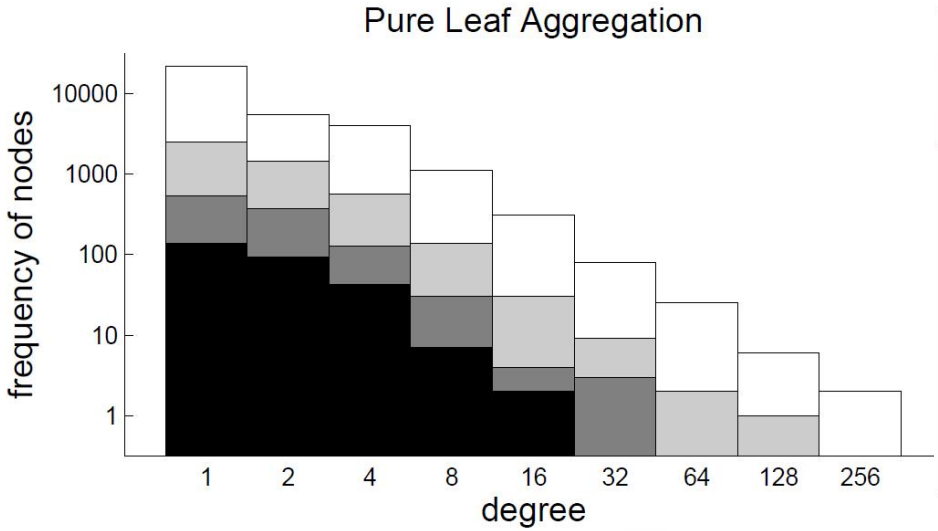
IV. **Numerical Results**

- **Leaf-based Aggregation**

V. Conclusions and Further Work

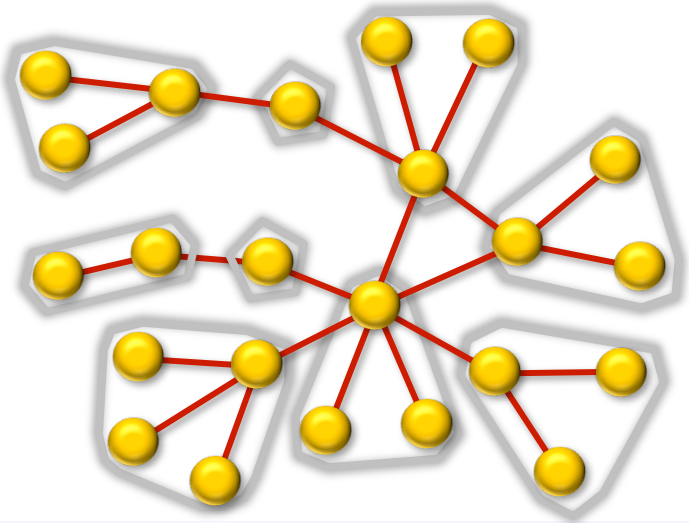
Ranking Problem, Leaf-Based Aggregation

Histograms for levels 1, 3, 5, and 7



Pure Aggregation V-cycles

			Iteration Counts	
n	C_{op}	levs	V	V+
1024	1.55	4	12	8
2048	1.57	5	12	8
4096	1.59	6	12	8
8192	1.59	6	12	8
16384	1.61	8	12	8
32768	1.61	9	12	8



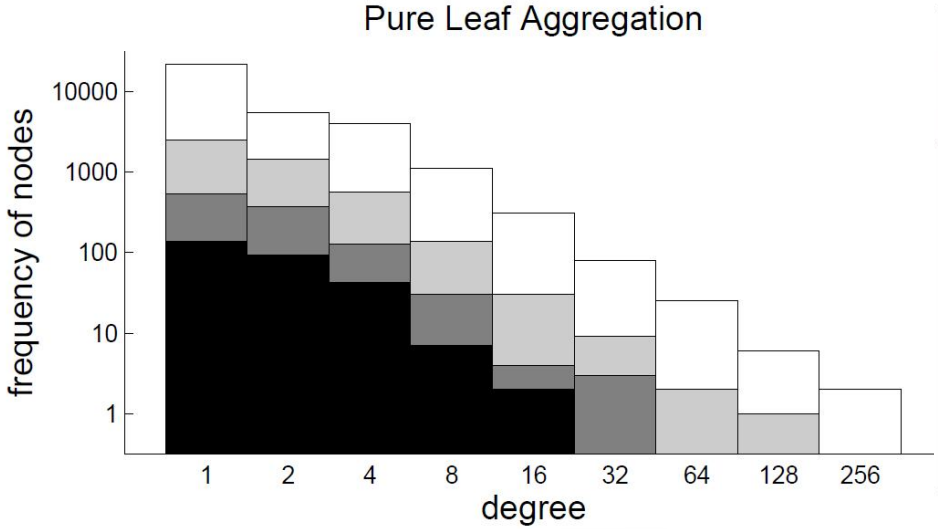
Pure, Leaf-Based Aggregation

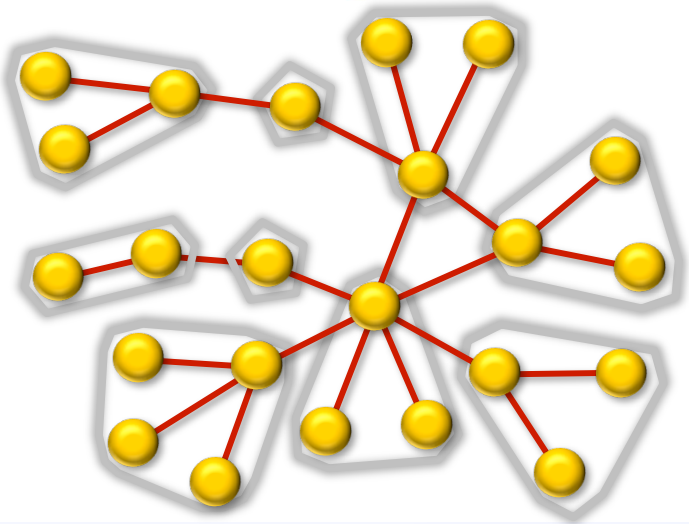
-preserves power-laws on all levels.

-achieves algorithmic scalability.

Ranking Problem, Leaf-Based Aggregation

Histograms for levels 1, 3, 5, and 7





Pure Aggregation V-cycles

			Iteration Counts	
n	C_{op}	levs	V	V+
1024	1.55	4	12	8
2048	1.57	5	12	8
4096	1.59	6	12	8
8192	1.59	6	12	8
16384	1.61	8	12	8
32768	1.61	9	12	8

Neighborhood Aggregation W-cycles

			Iteration Counts	
n	C_{op}	levs	W	W+
1024	1.52	4	233	37
2048	1.50	4	319	47
4096	1.53	4	326	51
8192	1.55	5	492	63
16384	1.55	5	>999	96
32768	1.54	5	>999	109

Outline

I. ~~Introduction~~

II. ~~Multilevel Aggregation for Random Walks~~

III. ~~Aggregation Methods~~

IV. ~~Numerical Results~~

V. **Conclusions and Further Work**

Conclusions and Further Work

- Successful multilevel approaches for mesh-like graphs fail for small-world / scale-free graphs:

  ~~Neighborhood-Based Aggregation~~

  ~~Smoothed Neighborhood-Based Aggregation~~

- Using pure, leaf-based aggregation yields scalable ranking calculations for the model problem (p=1) (can also be used for solving other eigenproblems)
- Ongoing work on dedicated coarsening mechanisms for non-tree-like scale-free graphs

(also: related multigrid method described on poster:)

A Self-learning Algebraic Multigrid Method for Singular Value Decomposition

Hans De Sterck

Department of Applied Mathematics

University of Waterloo, Canada

- **goal:** for $A \in \mathbb{R}^{m \times n}$, compute a few of the largest or smallest singular values and associated singular vectors:

$$A = U \Sigma V^t$$

$$A v = \sigma u,$$

$$A^t u = \sigma v.$$

- **approach:** bootstrap algebraic multigrid in two multilevel phases
- also works for eigenpairs of SPD matrices (graph Laplacians...)

[0] *Picture from Barrett Lyon's Opte project, 2003*

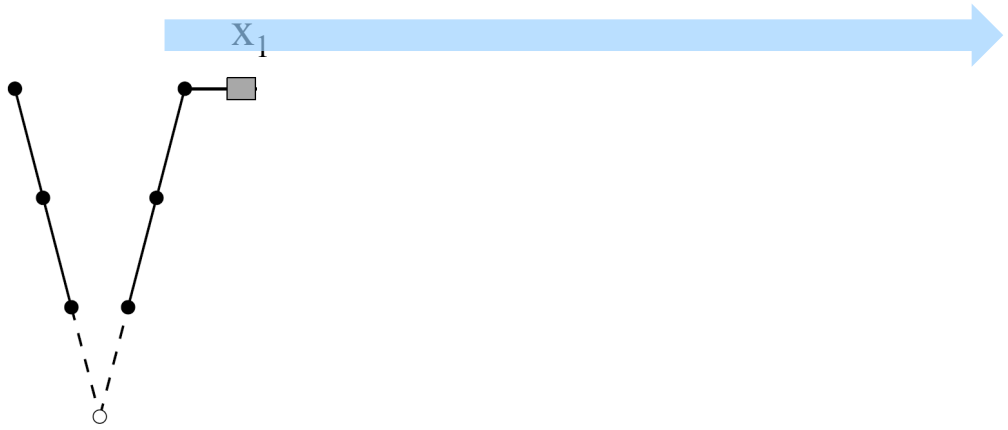
Top-Level Acceleration

Flexible Krylov Method

- Combine the last several iterations to form a *better* approximation to the steady-state vector.

- Very similar to Conjugate Gradient or GMRES applied to linear systems.

Top-Level Acceleration



Collect last *m* iterations:

$$X = [\mathbf{x}_k,\ \mathbf{x}_{k-1},\ \ldots,\ \mathbf{x}_{k-m+2}, \mathbf{x}_{k-m+1}]$$

$$\mathbf{x}_i > 0 \quad \text{and} \quad \|\mathbf{x}_i\|_1 = 1$$

Choose the *best* linear combination:

$$\mathbf{w} = X\mathbf{z}$$

$$\mathcal{P} := \{\mathbf{w} \in \mathbb{R}^n \quad \text{such that} \quad \|\mathbf{w}\|_1 = 1, \ \text{and} \quad \mathbf{w} \geq 0\}$$

Probability vector

Result

Let $A = I - B$, then the steady-state distribution vector we seek is the unique probability vector that minimizes $\mathcal{F}(\mathbf{w}) = \langle A\mathbf{w}, A\mathbf{w} \rangle$.

Top-Level Acceleration

$$\begin{array}{lll} \textbf{(C1)} & \text{(Normalization Constraint)} & \|\mathbf{w}\|_1 = 1 \\ \textbf{(C2)} & \text{(Nonnegativity Constraints)} & \mathbf{w} \geq 0 \\ \textbf{(C3)} & \text{(Subspace Constraint)} & \mathbf{w} \in \mathcal{R}(X) \end{array}$$

Nonlinear Constraint

(rewritten as linear)

$$\mathbf{x}_i > 0 \quad \text{and} \quad \|\mathbf{x}_i\|_1 = 1$$

$$\|\mathbf{w}\|_1 = \sum_i^n w_i = \sum_{i=1}^n \sum_{j=1}^m X_{ij} z_j = \sum_{j=1}^m z_j \sum_{i=1}^n X_{ij} = \sum_{j=1}^m z_j$$

Minimize $\mathcal{F}(\mathbf{w}) = \langle A\mathbf{w}, A\mathbf{w} \rangle$ within $\mathcal{R}(X) \cap \mathcal{P} = \left\{ \mathbf{w} = X\mathbf{z} : \sum_{i=1}^m z_i = 1 \,,\, X\mathbf{z} \geq \mathbf{0} \right\}$

Top-Level Acceleration

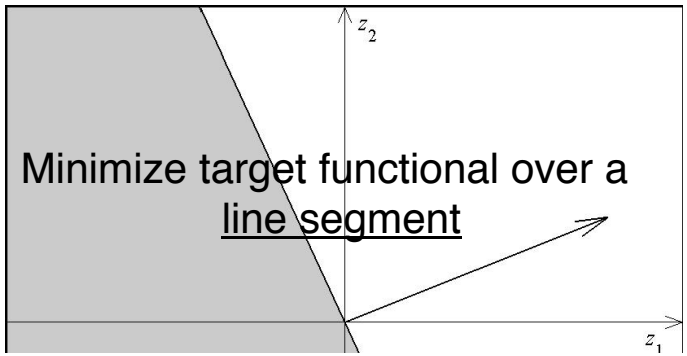
Constrained minimization problem of size *m*:

$$\begin{aligned} \text{minimize:} \quad & \mathbf{z}^t (X^t A^t A X) \mathbf{z}, \\ \text{subject to:} \quad & \mathbf{1}^t \mathbf{z} = 1, \quad \text{and} \\ & X\mathbf{z} \geq \mathbf{0}. \end{aligned}$$

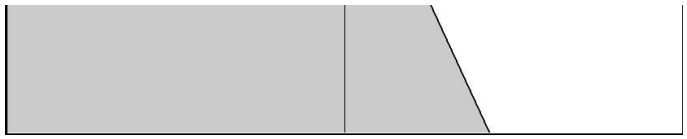
Single Equality Constraint

n Inequality Constraints

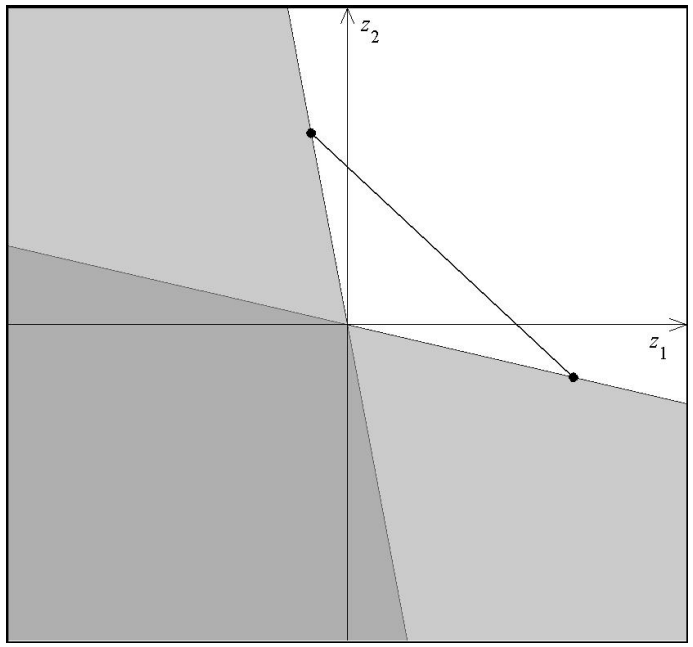
Efficient explicit solution for recombination of *two* iterates:



$$z_1^* = \frac{\langle A\mathbf{x}_2, A\mathbf{x}_2 \rangle - \langle A\mathbf{x}_1, A\mathbf{x}_2 \rangle}{\langle A\mathbf{x}_1, A\mathbf{x}_1 \rangle - 2\langle A\mathbf{x}_1, A\mathbf{x}_2 \rangle + \langle A\mathbf{x}_2, A\mathbf{x}_2 \rangle}$$



An Inequality Constraint



Multiple Constraints

Pure Aggregation for Markov Chains

$$A_c = R\,A\,P \qquad A_c(\mathrm{diag}(Q^T\,\mathbf{x}_i))^{-1} = I - B_c$$

$$B_c = Q^T\,B\,\mathrm{diag}(\mathbf{x}_i)\,Q\,(\mathrm{diag}(Q^T\,\mathbf{x}_i))^{-1}$$

$$\mathbf{1}_c^T\,B_c = \mathbf{1}_c^T$$

$$Q^T\,A\,\mathrm{diag}(\mathbf{x}_i)\,Q\,\mathbf{e}_c = 0$$

$$P^T\,\mathbf{1} \;=\; R\,\mathbf{x}_i \;=\; Q^T\,\mathbf{x}_i$$

$$\mathbf{x}_{i+1} = P\,(\mathrm{diag}(Q^T\,\mathbf{x}_i))^{-1}\,\mathbf{x}_c = P\,\mathbf{e}_c$$

$$Q = \left[\begin{array}{c|c|c|c} 1 & 0 & 0 & \cdots \\ 1 & 0 & 0 & \cdots \\ \hline 0 & 1 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ \hline 0 & 0 & 1 & \cdots \\ 0 & 0 & 1 & \cdots \\ \hline 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{array}\right]$$

Pure Aggregation

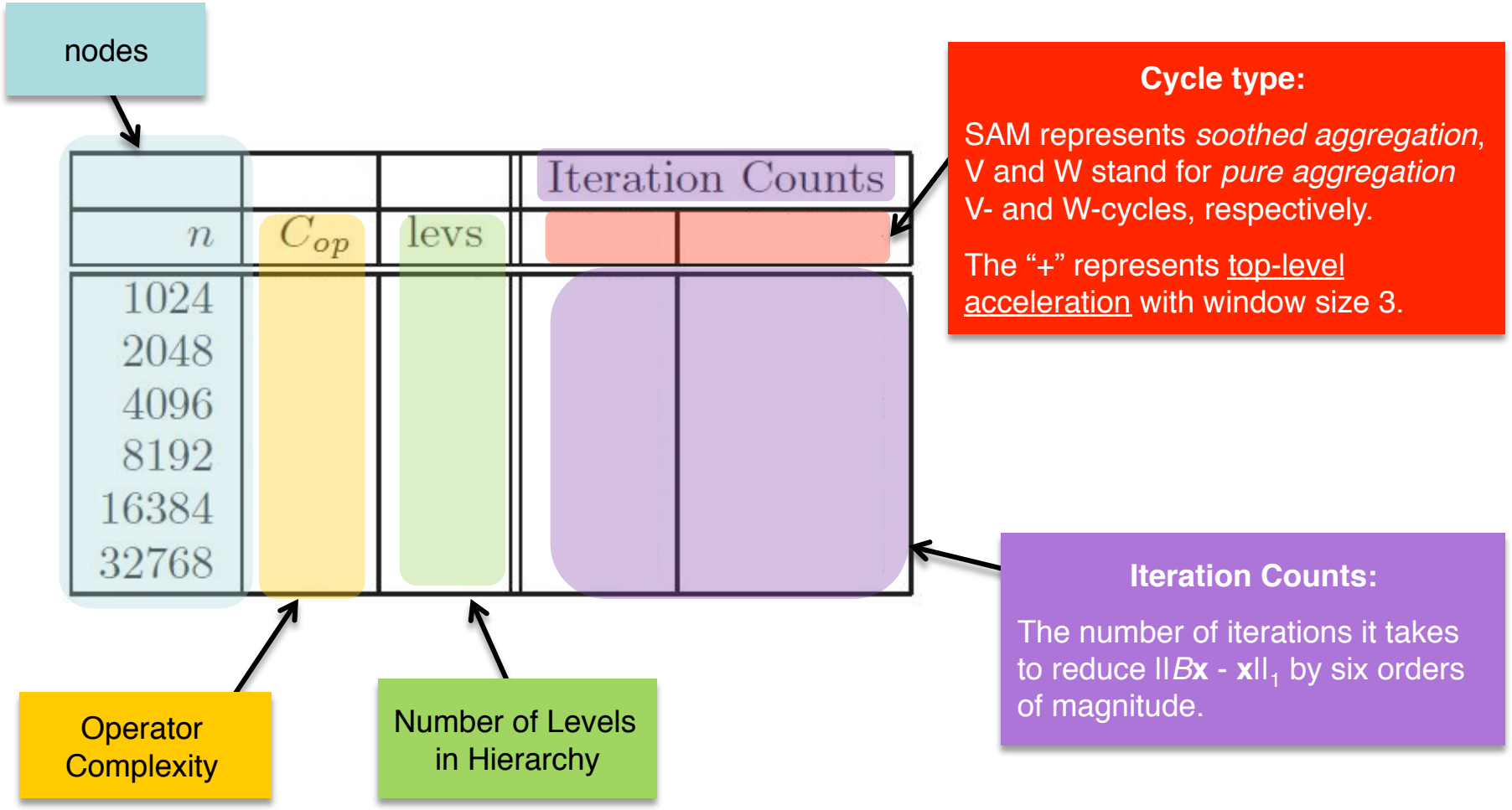
$$A_c \left(\mathrm{diag}(Q^T \mathbf{x}_i)\right)^{-1} \mathbf{x}_c = 0$$

$$\mathbf{x}_c = \mathrm{diag}(Q^T \mathbf{x}_i)\, \mathbf{e}_c$$

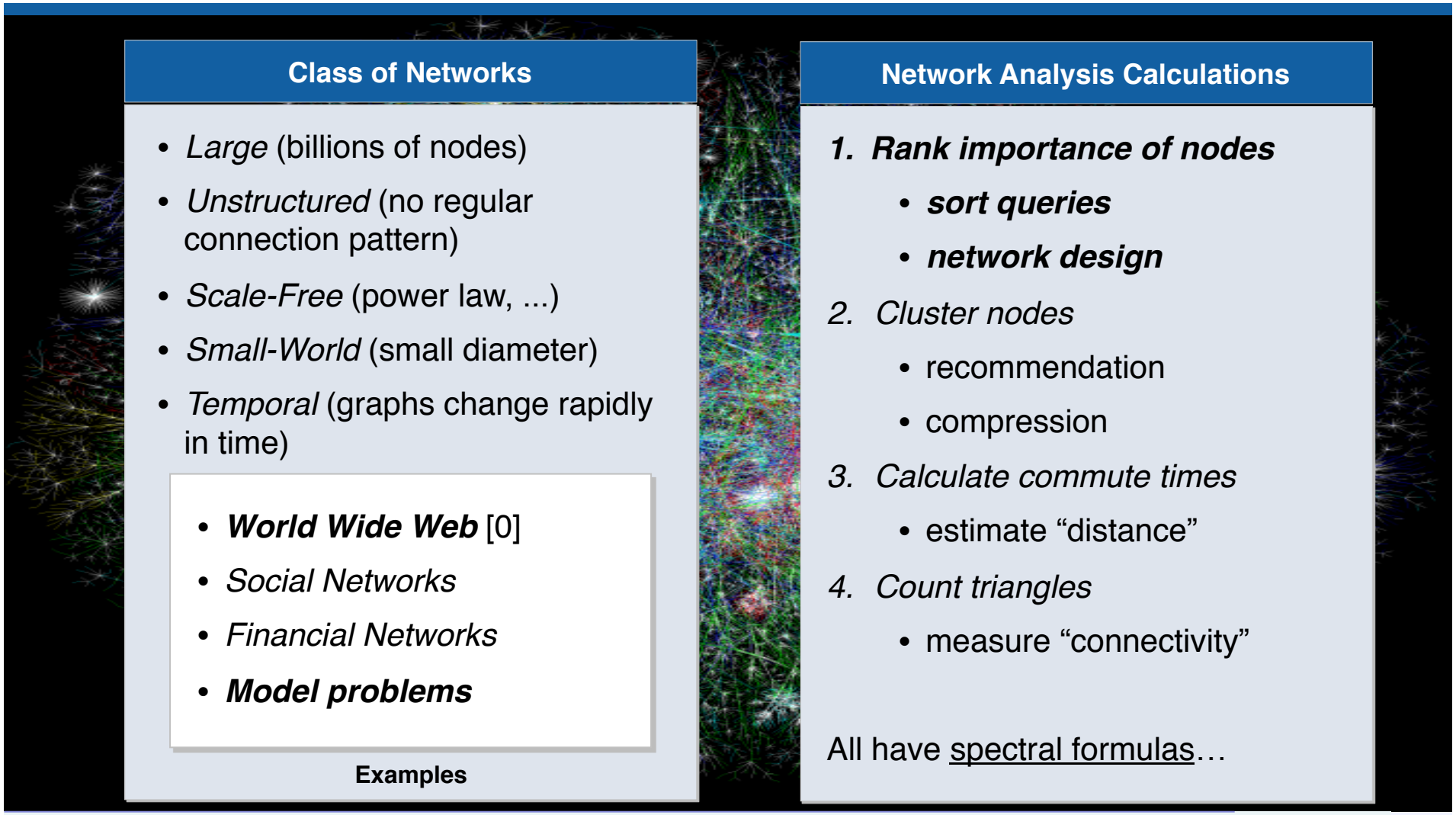
$$R = Q^T \text{ and } P = \mathrm{diag}(\mathbf{x}_i)\, Q$$

$$R\, A\, P\, \mathbf{e}_c = 0$$

Numerical Results: Ranking Problem



Large Network Graph Properties and Examples



Class of Networks

- *Large* (billions of nodes)
- *Unstructured* (no regular connection pattern)
- *Scale-Free* (power law, ...)
- *Small-World* (small diameter)
- *Temporal* (graphs change rapidly in time)

- ***World Wide Web*** [0]
- *Social Networks*
- *Financial Networks*
- ***Model problems***

Examples

Network Analysis Calculations

1. ***Rank importance of nodes***
   - ***sort queries***
   - ***network design***
2. *Cluster nodes*
   - recommendation
   - compression
3. *Calculate commute times*
   - estimate “distance”
4. *Count triangles*
   - measure “connectivity”

All have spectral formulas…

[0] *Picture from Barrett Lyon’s Opte project, 2003*

Top-Level Acceleration

Flexible Krylov Method

- Combine the last several iterations to form a *better* approximation to the steady-state vector.

- Very similar to Conjugate Gradient or GMRES applied to linear systems.