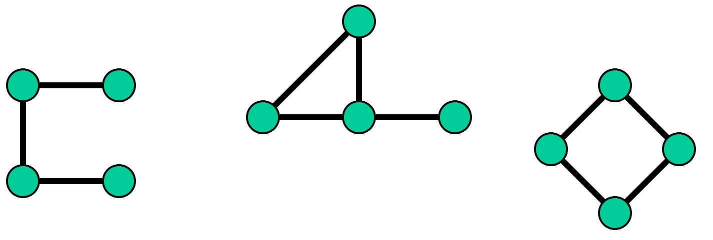
Structural Matching Via Optimal Basis Graphs



Fred DePiero and John Carlin

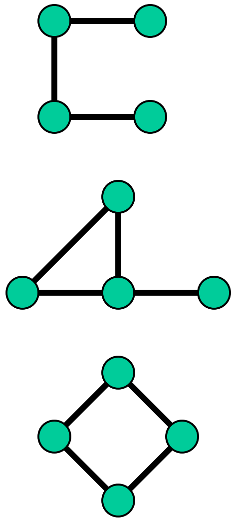
CalPoly State University,
San Luis Obispo, CA USA

Goals Target RT Measurements

- Sensing Conditions / Processing
  - Cluttered scenes and noisy sensor data. Deterministic Processing.
- Algorithm Test Conditions
  - Many excess (noise) nodes: up to 100%
  - Variety of types: random, strongly regular, banded
  - Low dynamic range of coloring: 0 or 2 discrete values
  - Approximation to maximum common subgraph: OK
- Application Example
  - Landmark-based registration, find corresponding points in a single step, then coordinate transform.
  - Next, we may pursue fingerprint correspondence

What Are 'Basis Graphs' And How Are They Used?

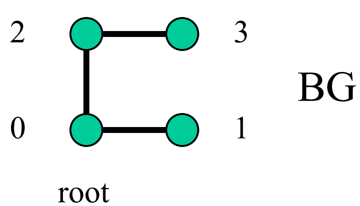
- Small (4-node) graphs used to characterize local structure

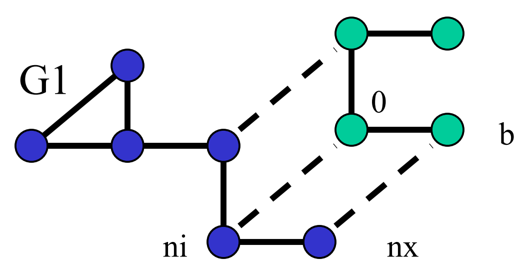


Basis Graphs

What Are ‘Basis Graphs’ And How Are They Used?

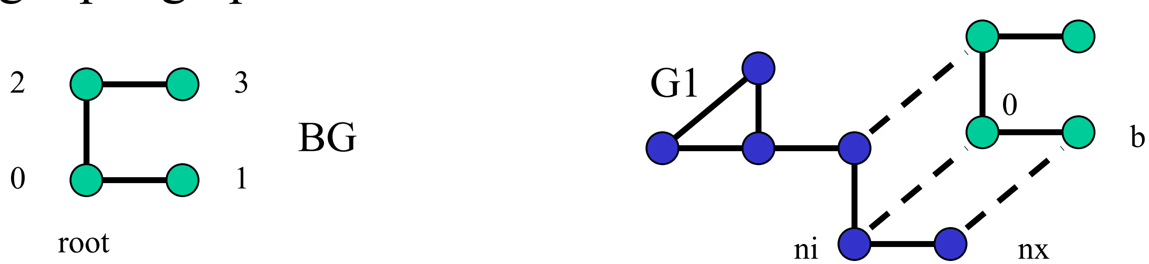
- ‘Throw a BG at an input G1, and see where it lands’
- Under conditions of a random mapping between BG and G1, Estimate pdf p1[ni][nx][b], describing how likely
  - Root node of BG ~ Node ni of G1, and
  - Node b of BG ~ Node nx of G1 (‘~’ means ‘associated with’)





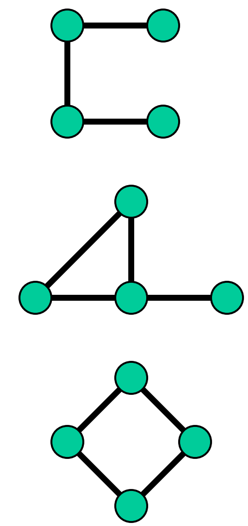
What Are 'Basis Graphs' And How Are They Used?

- 'Throw a BG at an input G1, and see where it lands'
- Under conditions of a random mapping between BG and G1, Estimate pdf p1[ni][nx][b], describing how likely
  - Root node of BG ~ Node ni of G1, and
  - Node b of BG ~ Node nx of G1 ('~' means 'associated with')
- The pdf p1[ni][nx][b]
  - Describes local structure
  - Provides an 'auxiliary' means to create a description, other than just using input graphs G1 and G2



What Are 'Basis Graphs' And How Are They Used?

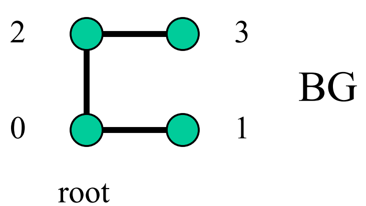
- Algorithm – Overview
  - Find (partial) occurrences of basis graphs in input graphs to estimate pdf p1[ni][nx][b], p2.
  - Find initial mapping probabilities by comparing basis graph occurrences and using a Gaussian model, (and any coloring).
  - Refine mapping probabilities via continuous relaxation.

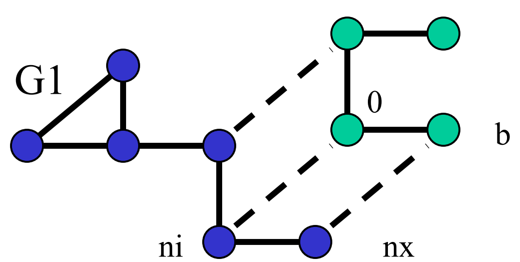


Basis Graphs

Basis Graphs Used To Find Initial Mapping Probabilities

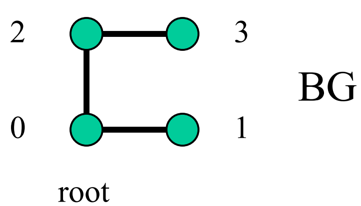
- For given input graph G1, and given BG
  - Count (partial) occurrences of BG appearing in G1. O{N^4}.
  - Estimate pdf p1[ni][nx][b] via histogram counts.

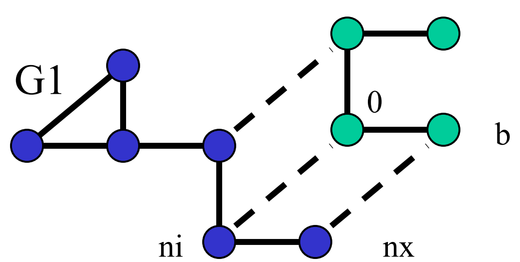




Basis Graphs Used To Find Initial Mapping Probabilities

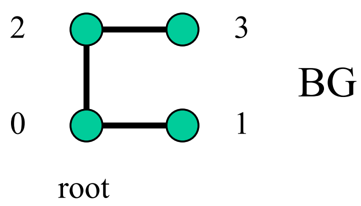
- For given input graph G1, and given BG
  - Count (partial) occurrences of BG appearing in G1. O{N^4}.
  - Estimate pdf p1[ni][nx][b] via histogram counts.
- Find p2[nj][ny][b] similarly for G2

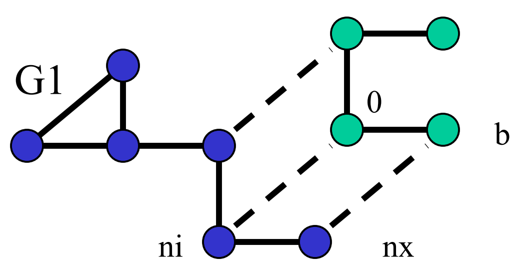




Basis Graphs Used To Find Initial Mapping Probabilities

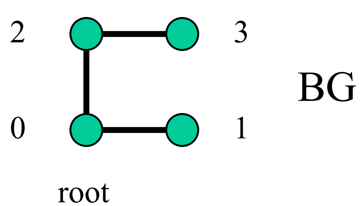
- For given input graph G1, and given BG
  - Count (partial) occurrences of BG appearing in G1. O{N^4}.
  - Estimate pdf p1[ni][nx][b] via histogram counts.
- Find p2[nj][ny][b] similarly for G2
- For pairs of nodes ni of G1 and nj of G2
  - **MIN**{p1[ni][nx][b] – p2[nj][ny][b]} by checking each nx, ny to find minimum

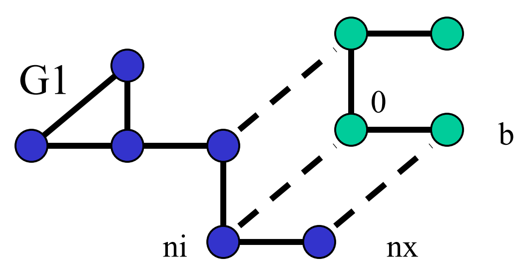




Basis Graphs Used To Find Initial Mapping Probabilities

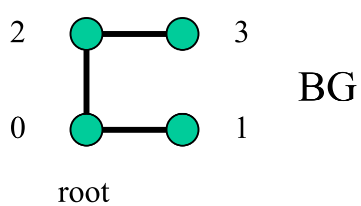
- For given input graph G1, and given BG
  - Count (partial) occurrences of BG appearing in G1. O{N^4}.
  - Estimate pdf p1[ni][nx][b] via histogram counts.
- Find p2[nj][ny][b] similarly for G2
- For pairs of nodes ni of G1 and nj of G2
  - Find P_w[ni][nj] = *N*{ **MIN**{p1[ni][nx][b] – p2[nj][ny][b]} } by checking each nx, ny to find minimum and applying a-priori Gaussian model, ***N***

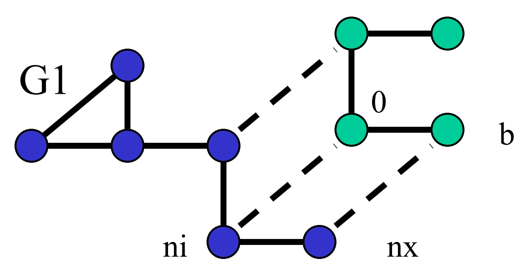




Basis Graphs Used To Find Initial Mapping Probabilities

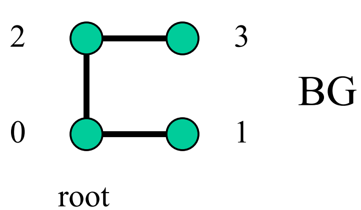
- For given input graph G1, and given BG
  - Count (partial) occurrences of BG appearing in G1. $O\{N^4\}$.
  - Estimate pdf p1[ni][nx][b] via histogram counts.
- Find p2[nj][ny][b] similarly for G2
- For pairs of nodes ni of G1 and nj of G2
  - Find P_w[ni][nj] = ***N***{ **MIN**{p1[ni][nx][b] – p2[nj][ny][b]} } by checking each nx, ny to find minimum and applying a-priori Gaussian model, ***N***
  - Find P[ni][nj] = **DS**{ P_w[ni][nj] } for all w. **DS** is the Dempster-Schafer method to combine evidence: $P3 = 1 - (1-P1)(1-P2)$

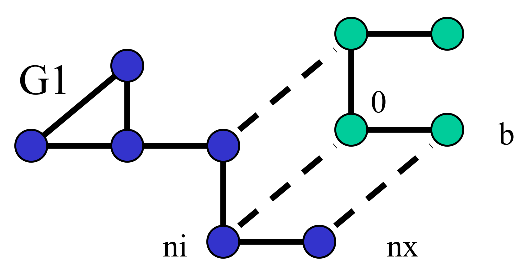




Basis Graphs Used To Find Initial Mapping Probabilities

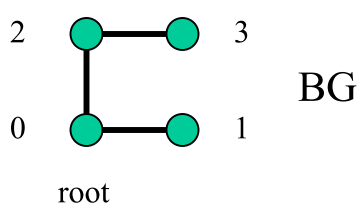
- For given input graph G1, and given BG
  - Count (partial) occurrences of BG appearing in G1. O{N^4}.
  - Estimate pdf p1[ni][nx][b] via histogram counts.
- Find p2[nj][ny][b] similarly for G2
- For pairs of nodes ni of G1 and nj of G2
  - Find P_w[ni][nj] = *N*{ **MIN**{p1[ni][nx][b] – p2[nj][ny][b]} } by checking each nx, ny to find minimum and applying a-priori Gaussian model, ***N***
  - Find P[ni][nj] = **DS**{ P_w[ni][nj] } for all w. **DS** is the Dempster-Schafer method to combine evidence: *P3 = 1 – (1-P1)(1-P2)*
- Refine P[ni][nj] via continuous relaxation. Use fixed # of iterations.

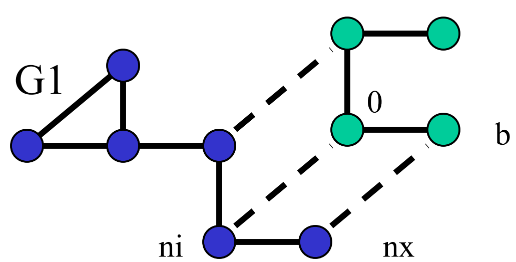




Basis Graphs Used To Find Initial Mapping Probabilities

- For given input graph G1, and given BG
  - Count (partial) occurrences of BG appearing in G1. O{N^4}.
  - Estimate pdf p1[ni][nx][b] via histogram counts.
- Find p2[nj][ny][b] similarly for G2
- For pairs of nodes ni of G1 and nj of G2
  - Find P_w[ni][nj] = *N*{ **MIN**{p1[ni][nx][b] – p2[nj][ny][b]} } by checking each nx, ny to find minimum and applying a-priori Gaussian model, *N*
  - Find P[ni][nj] = **DS**{ P_w[ni][nj] } for all w. **DS** is the Dempster-Schafer method to combine evidence: *P3 = 1 – (1-P1)(1-P2)*
- Refine P[ni][nj] via continuous relaxation. Use fixed # of iterations.
- Matching: Use P[ni][nj], selecting most likely assignment – subject to structural (and color) constraints.



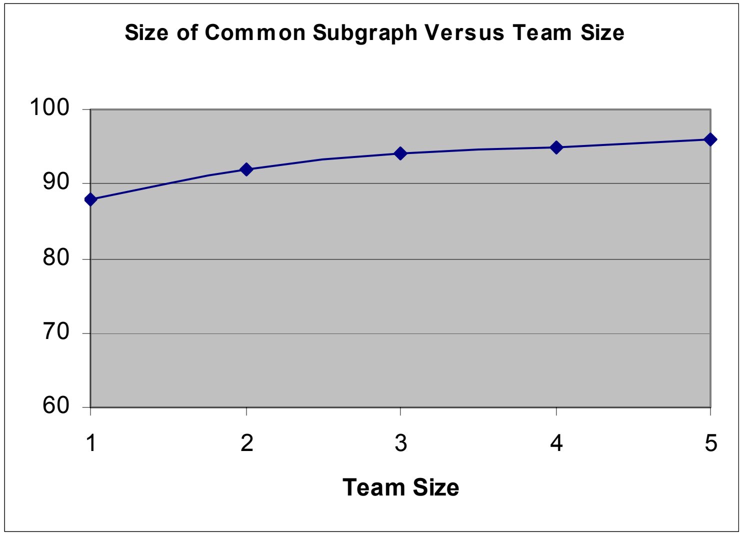


A Taxonomy: How is Local Structure Described?

- Messmer
  - Identifies small graphs that were commonly occurring in expected scenes.
  - *BG*: No a-priori knowledge of particular inputs, generic set of BG
- Superclique
  - Local neighborhood formed via a fixed pattern (adjacent nodes).
  - *BG*: Use varied size & shape of neighborhood, multiple structures.
- Paths of Varying Length
  - Random walks, also Length-r Paths (LeRP)
  - *BG*: (vs. RW) Deterministic, tried to optimize shape – not a random shape
- (In Contrast) Eigenvalue-Based Approaches
  - Eigenvalues are a global property of adjacency matrix
  - *BG*: Local structure is characterized
- (Note) Relaxation, Used to refine mapping probabilities
  - *BG:* Preprocessing effort used to find initial probability mapping.

Choice of Basis Graphs Optimized

- (*When using a set of BG, repeat processing for each and select best result - largest common subgraph*.)
- Size of BG = 4 nodes (due to throughput needs). Team of smaller BG better than single larger BG.
- Team size = 3, based on speed/performance tradeoff.



- *Which team members?*

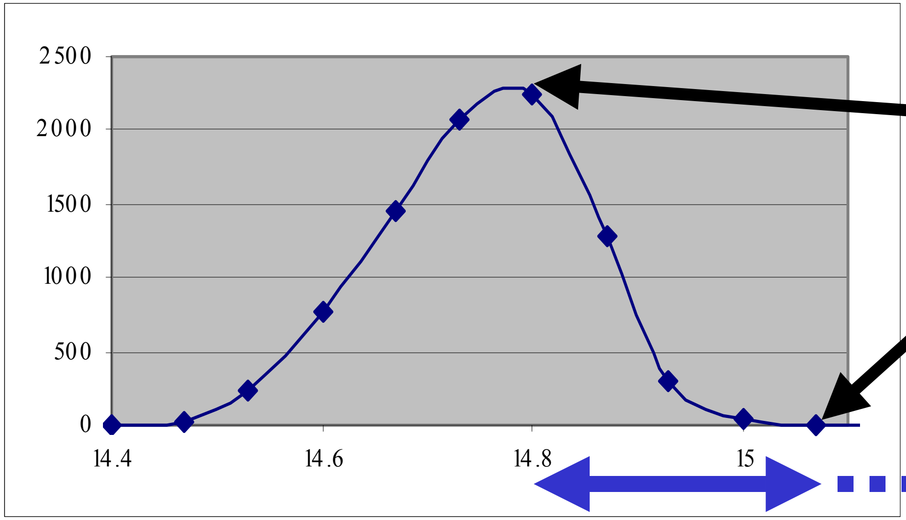
  Check all permutations of

  4-node connected graphs (38)

Many Teams Can Perform Well

Number of Teams

$C_3^{38} = 8436$



Many other good teams

The very best team

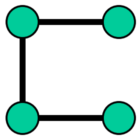
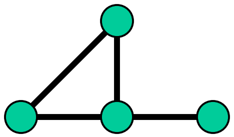
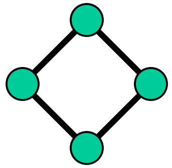
Not much improvement!

Mean Size of Common Subgraph
(For a 16 node input, nominal)

- Original Question: *Which is the best team choice?*
- New Question: *What properties are common in the better teams?*

Best Teams
Share a Shape Property

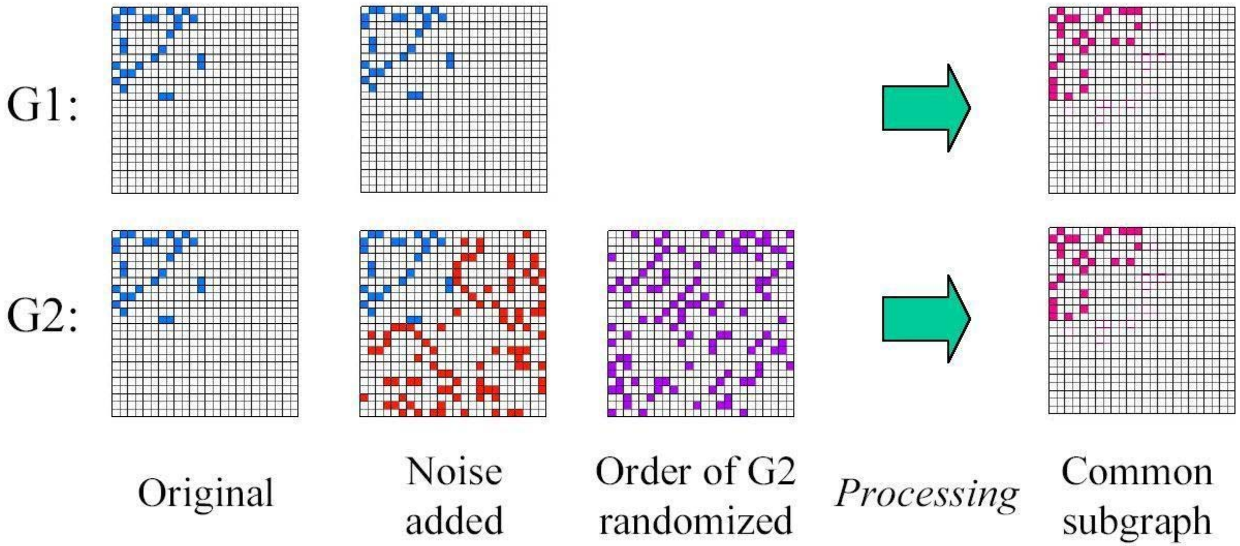
- Number and size of loops
- A large percentage of the better teams share property.
- Noticed some similar trends for best teams with more nodes.
- Intuitively pleasing result!

	# Loops	Length
	0	-
	1	3
	1	4

Testing Intended to be Challenging

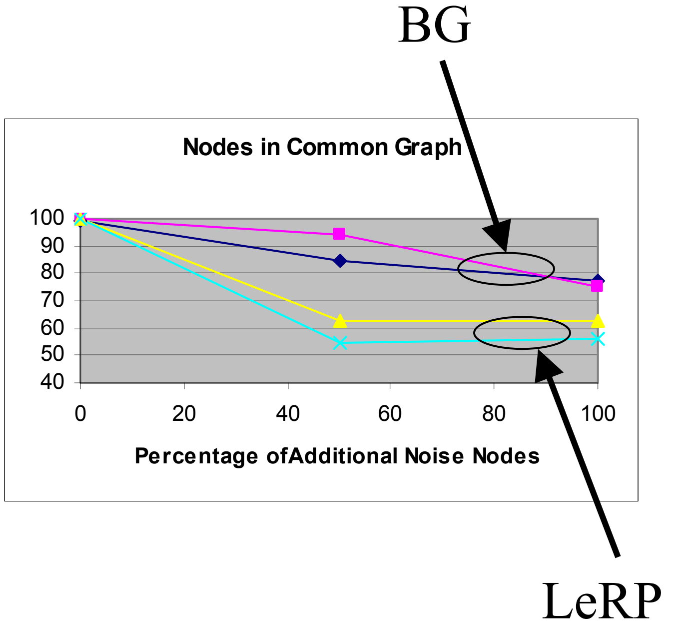
- Monte Carlo trials (3000)
- High clutter, up to 100% additional nodes: 16->32
- Limited coloring: either none, or 2 colors
- Strongly regular, random, banded adjacency matrix

Banded graphs approximate many natural and man-made structures.



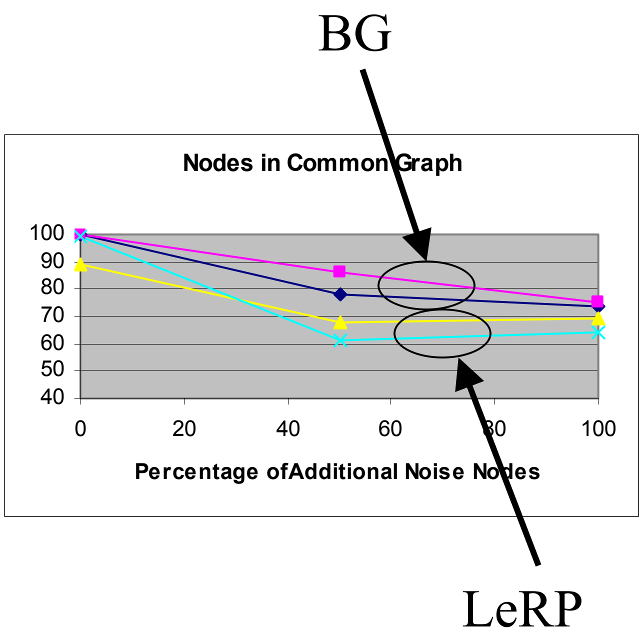
BG Performs Better Than LeRP For Smaller Graphs

- Conditions:
  - 16-node inputs. No color.
  - Random graphs, edge probability 0.2, 0.3
  - 0, 50, 100% additional noise nodes
  - Optimal basis used for BG



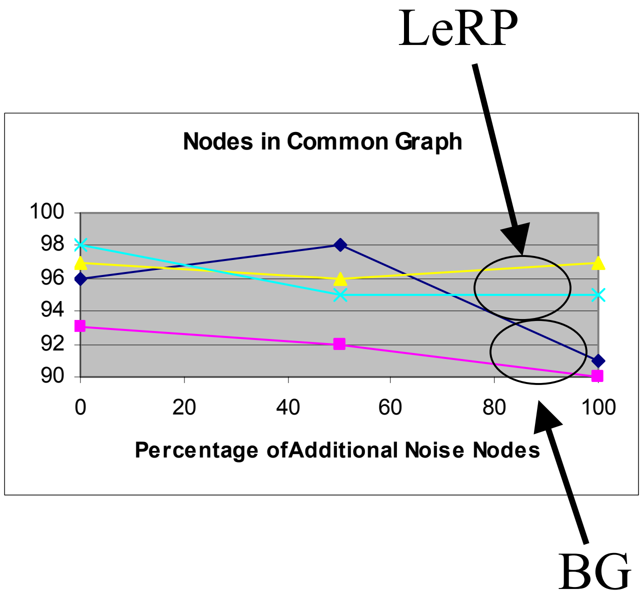
Strongly Regular Most Challenging

- Conditions:
  - 16-node inputs. No color.
  - Strongly regular, randomly generated graphs. Degree 3,4
  - 0, 50, 100% additional noise nodes
  - Optimal basis used for BG



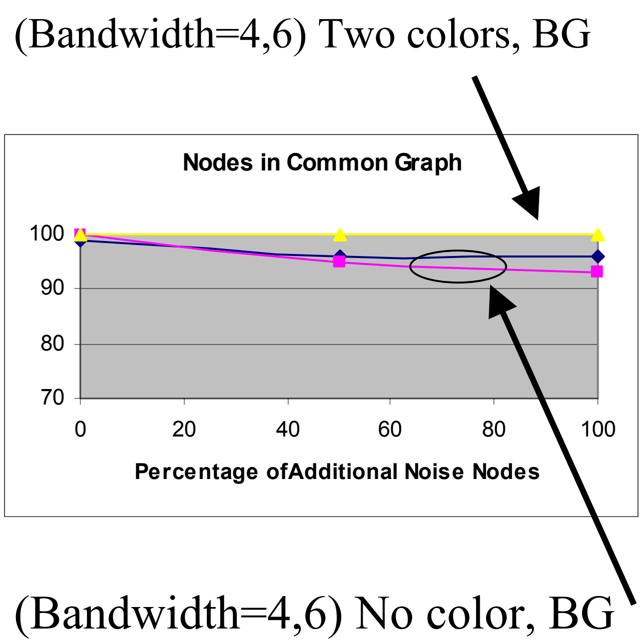
LeRP Performs Better Than BG For Larger Graphs

- Conditions:
  - 32-node inputs. No color.
  - Random graphs with banded adjacency matrices
  - 0, 50, 100% additional noise nodes
  - Optimal basis used for BG



Modest Coloring Yields Near-Ideal Results

- Conditions:
  - 16-node inputs.
  - Random graphs with banded adjacency matrices
  - 0, 50, 100% additional noise nodes
  - Optimal basis used for BG
- All cases with 16+16 or less, under 1 sec



Conclusions

- BG can perform well with high noise (100%) and zero coloring.
- Improved means to describe local structure benefits matching performance.
- Optimal choice of BG reported

	BG	LeRP
Smaller Inputs	+ +	
Larger Inputs		+

Performance Vs. Size