Circuit Partitioning:
EIG+KL Hybrid Algorithm Accelerator

Algorithm - EIG

- Input - Netlist (net, nodes)
- Create clique-based graph model of netlist
- Using 2-clique model
- Compute Laplacian Matrix from Adjacency and Degree matrix (L = D-A)
- Compute 2nd smallest eigenvalue and corresponding eigenvector
- Obtain CutSize and RatioCut with area bound.
- Large matrices take too long to compute, use Spectra eigensolver on Sparse representation of matrix

Algorithm - KL

- Input - Netlist (net, nodes)
- Create clique-based graph model of netlist
- Using 2-clique model
- Compute CutSize and Gain by swapping nodes across partitions.
- Obtain best Cutsize across all swaps
- Large matrix prove far too time consuming. Therefore develop some heuristics to help reduce computation size and improve timing.
    - Results still unattainable for largest benchmark

Fract

Sparsity Plot of Benchma

Nets Numbers: 147

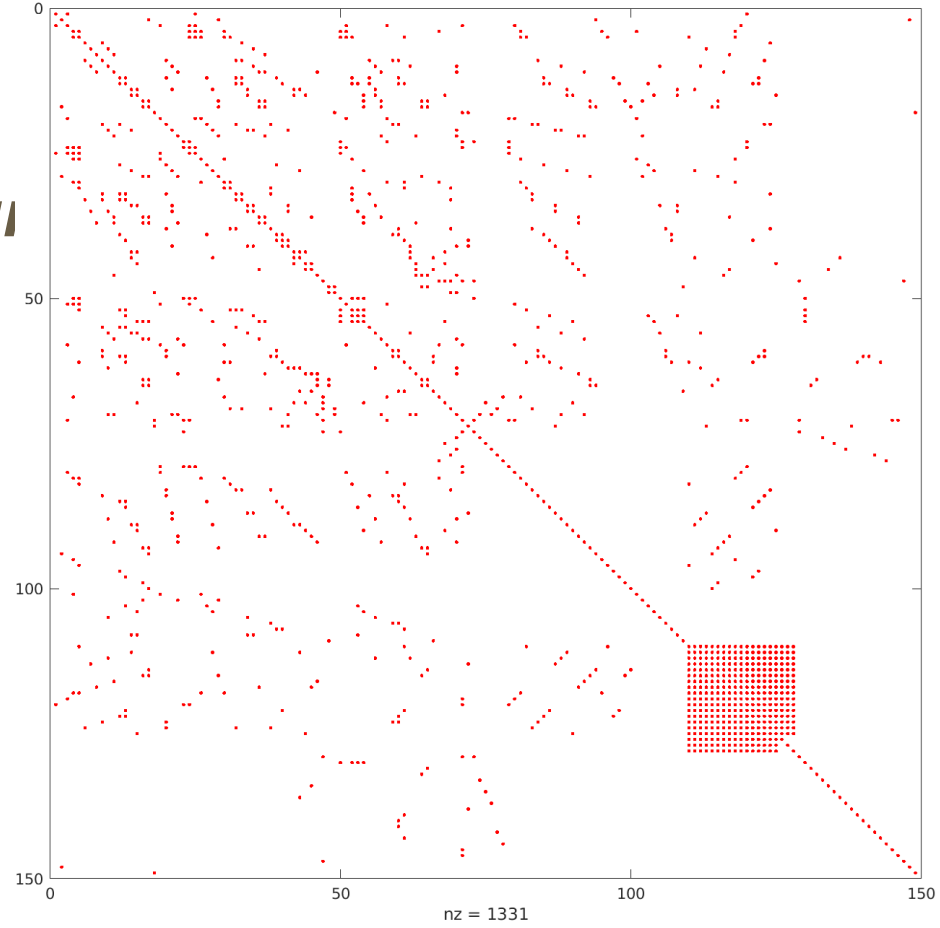
Nodes Numbers: 149

Size of matrix: 22201 (88.804 KB)

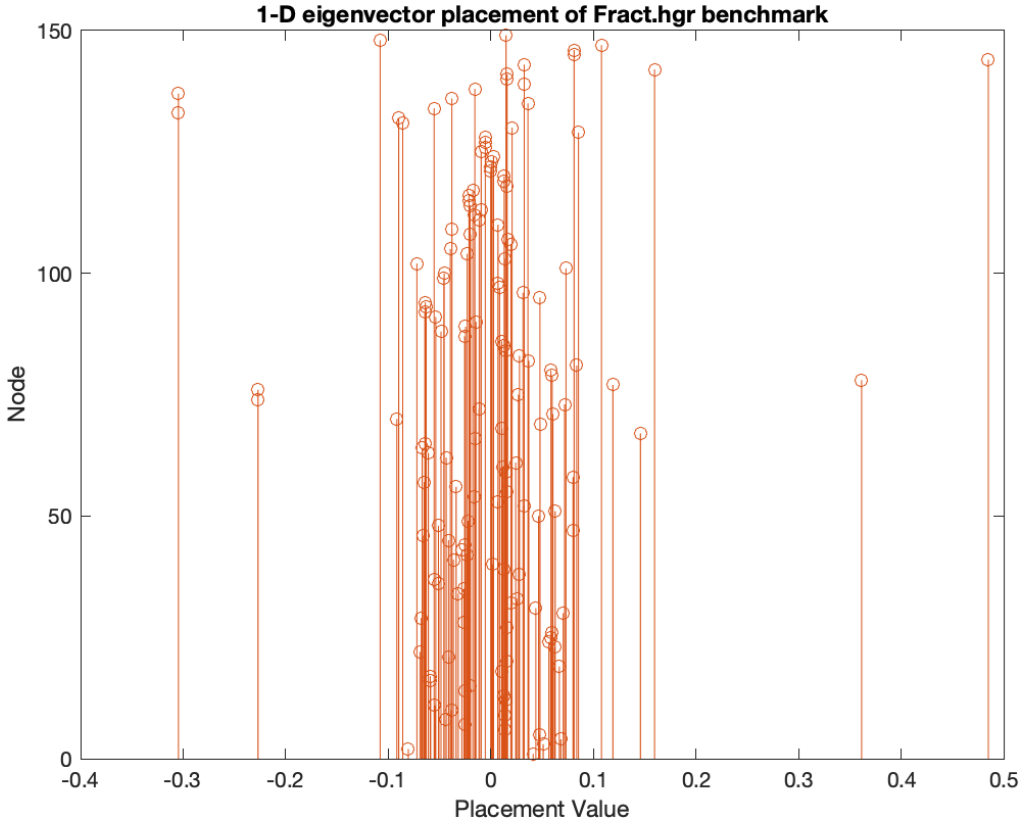
Non-Zero Elements: 1331 (10.648 KB)

Ratio: 5.99523%

Node-to-node Edges: 869

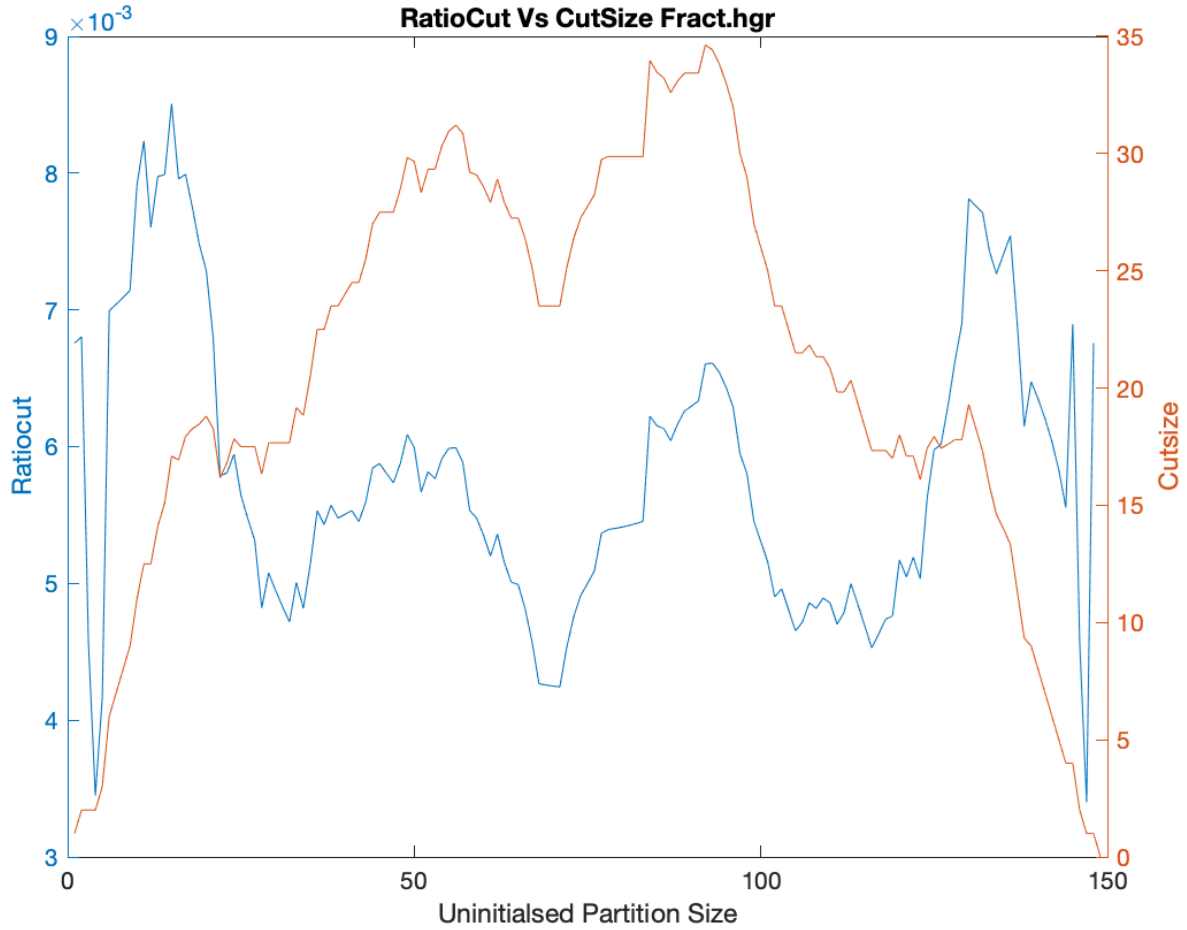


Fract - I-D Placement



Validity : $\sum (\text{placement value})^2 = 1$

Fract - RatioCut Vs Cutsize



Industry2

Sparsity Plot of Benchmark

Nets Numbers: 13419

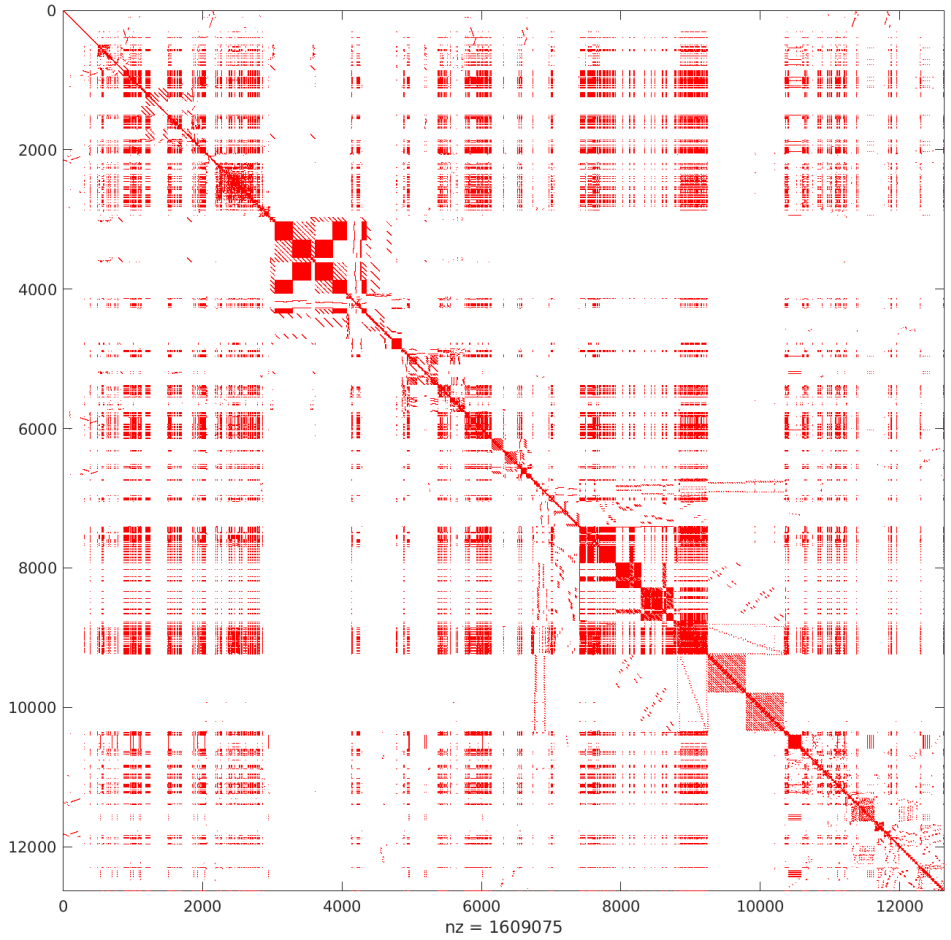
Nodes Numbers: 12637

Size of matrix: 159693769 (638.775 MB)

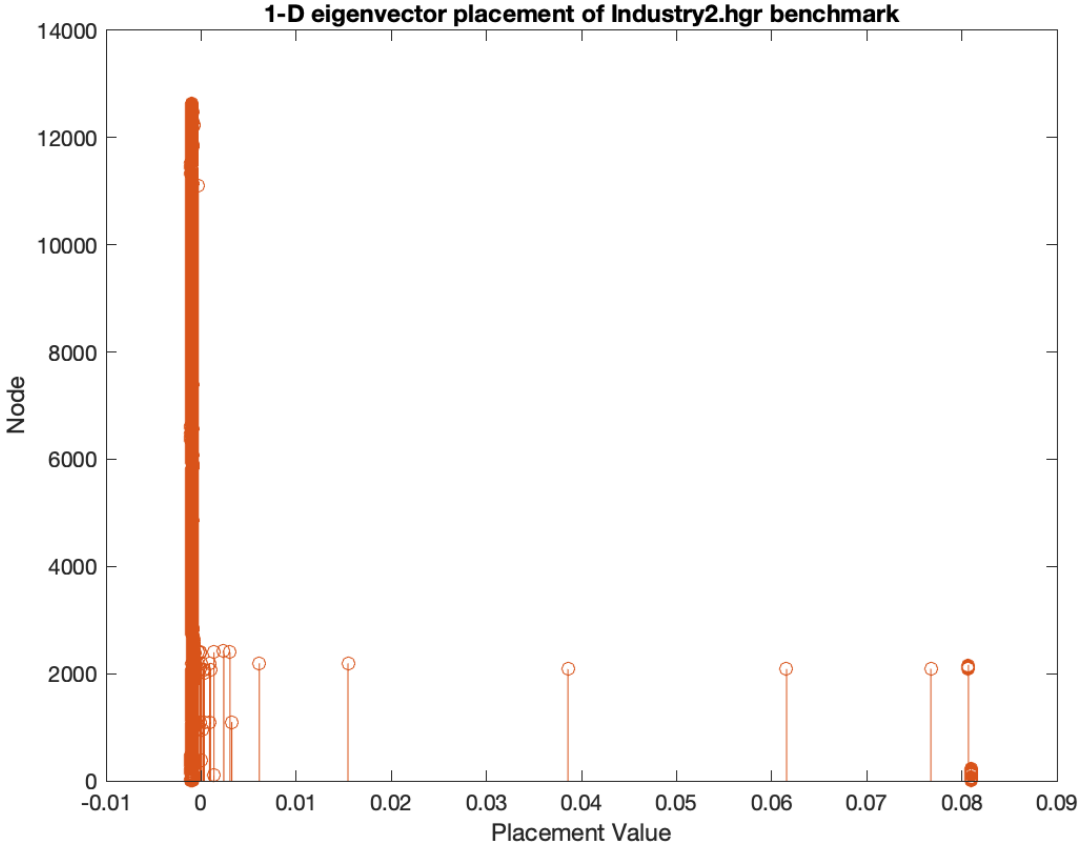
Non-Zero Elements: 1609075 (12.8726 MB)

Ratio: 1.0076%

Node-to-node Edges: 869256

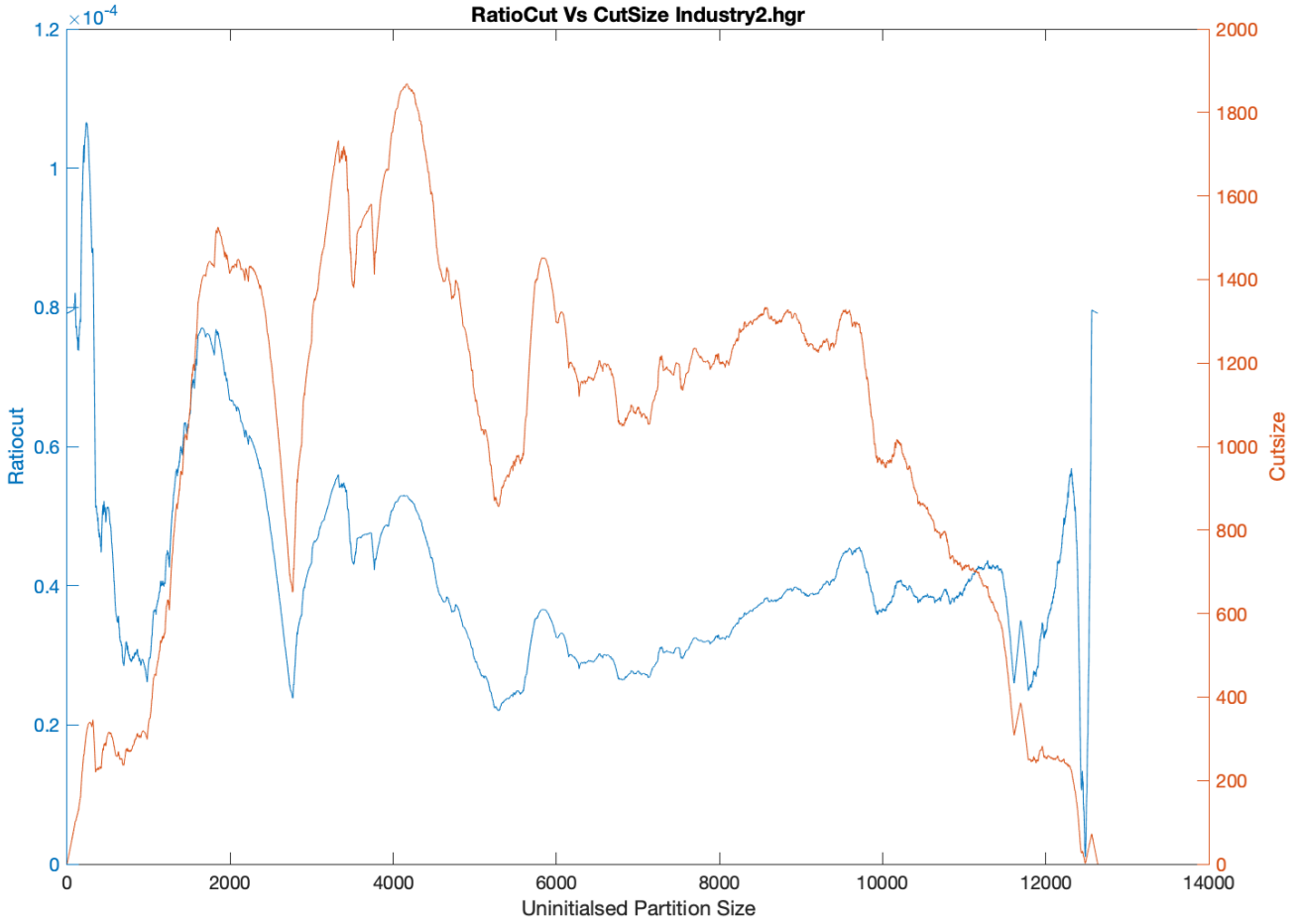


Industry2 - 1-D Placement



Validity : $\sum$ (placement value)^2 =1

Industry2- RatioCut Vs Cutsize



ibm01

Sparsity Plot of Benchmark

Nets Numbers: 14111

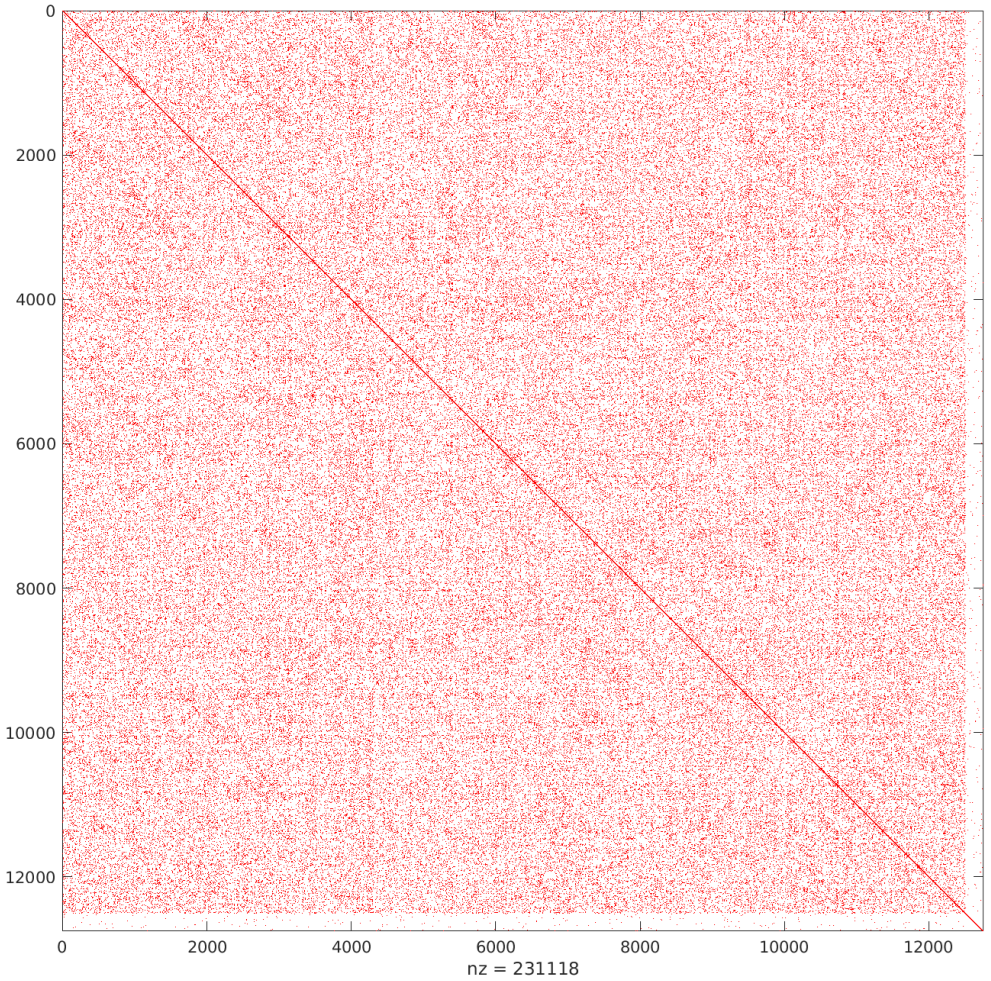
Nodes Numbers: 12752

Size of matrix: 162613504 (650.454 MB)

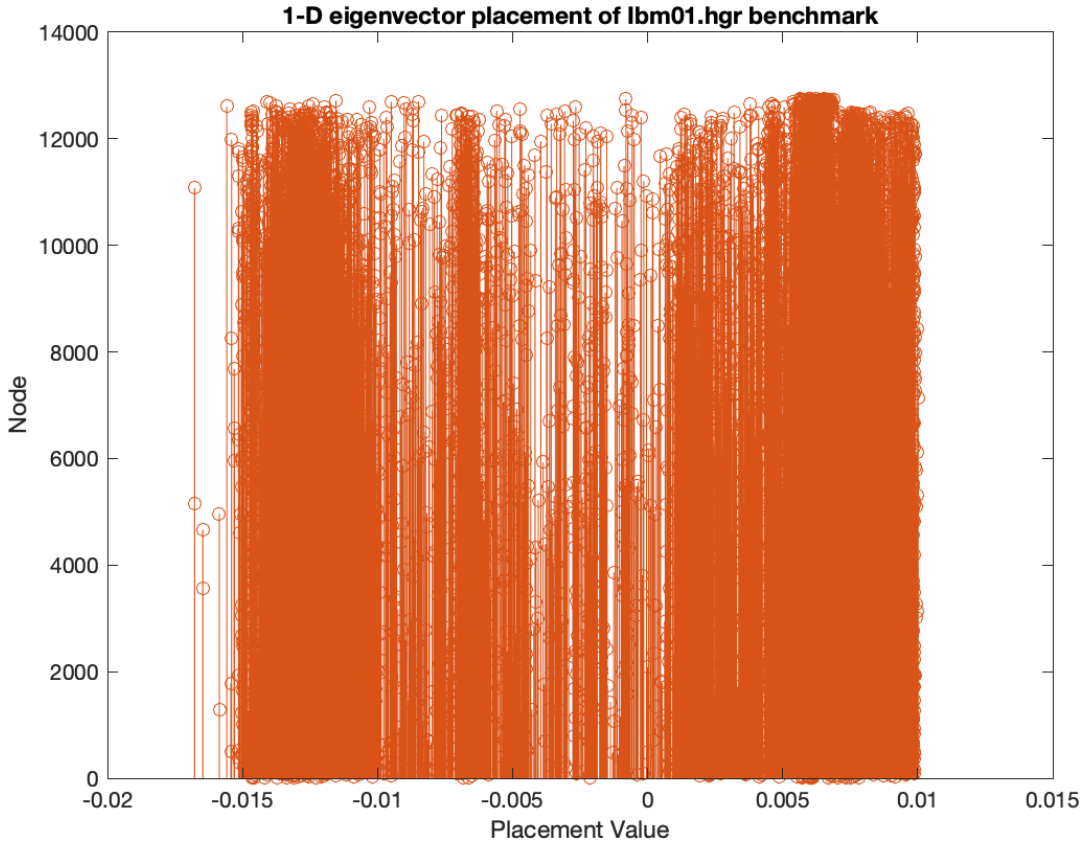
Non-Zero Elements: 231118 (1.848 MB)

Ratio: 0.142127%

Node-to-node Edges: 144148

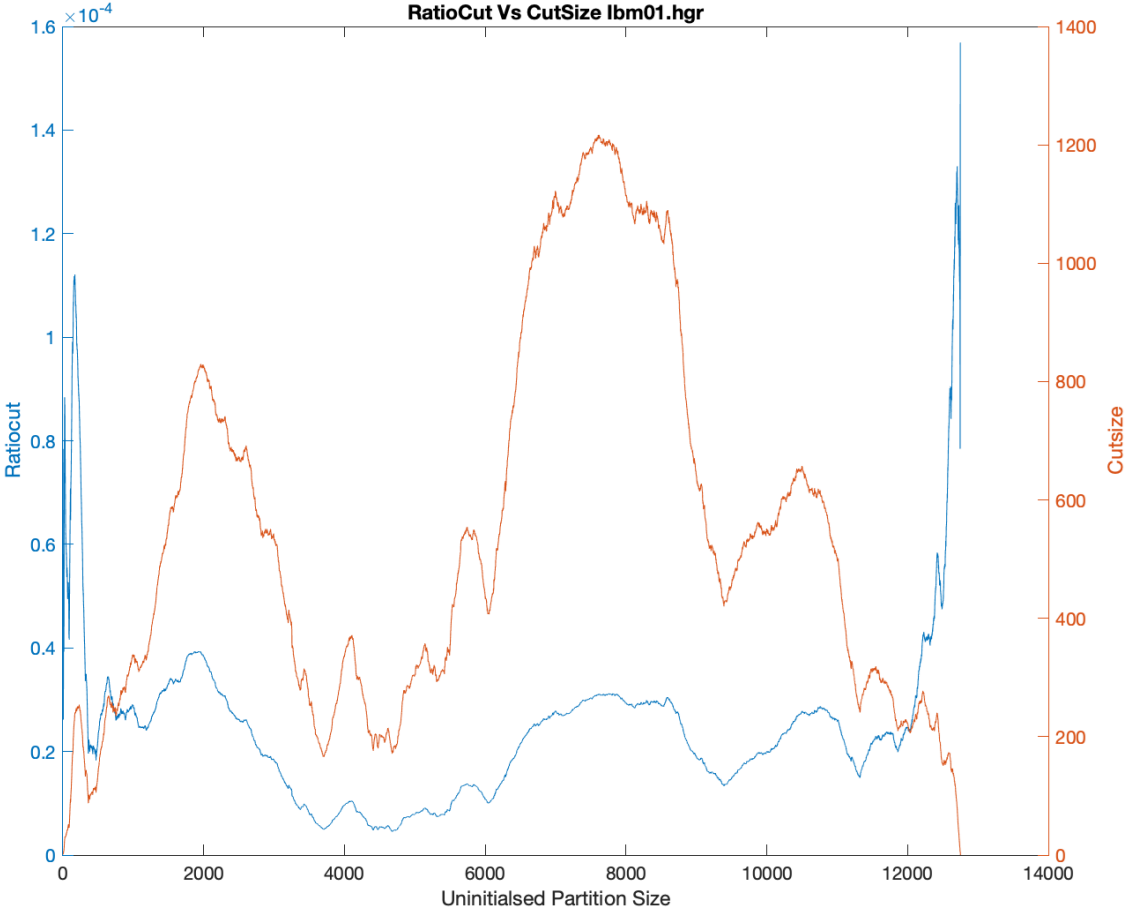


Ibm01- 1-D placement



Validity : $\sum (\text{placement value})^2 = 1$

Ibm01- RatioCut Vs Cutsize



ibm10

Sparsity Plot of Benchmark

Nets Numbers: 75196

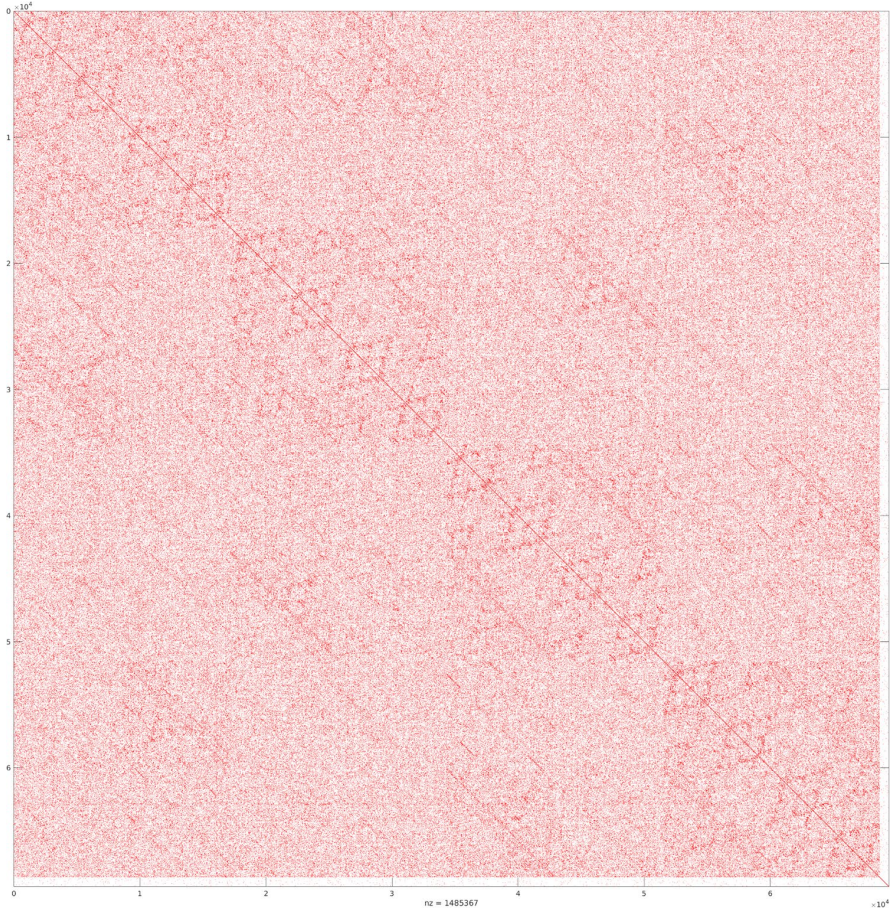
Nodes Numbers: 69429

Size of matrix: 4820386041 (19.2815 GB)

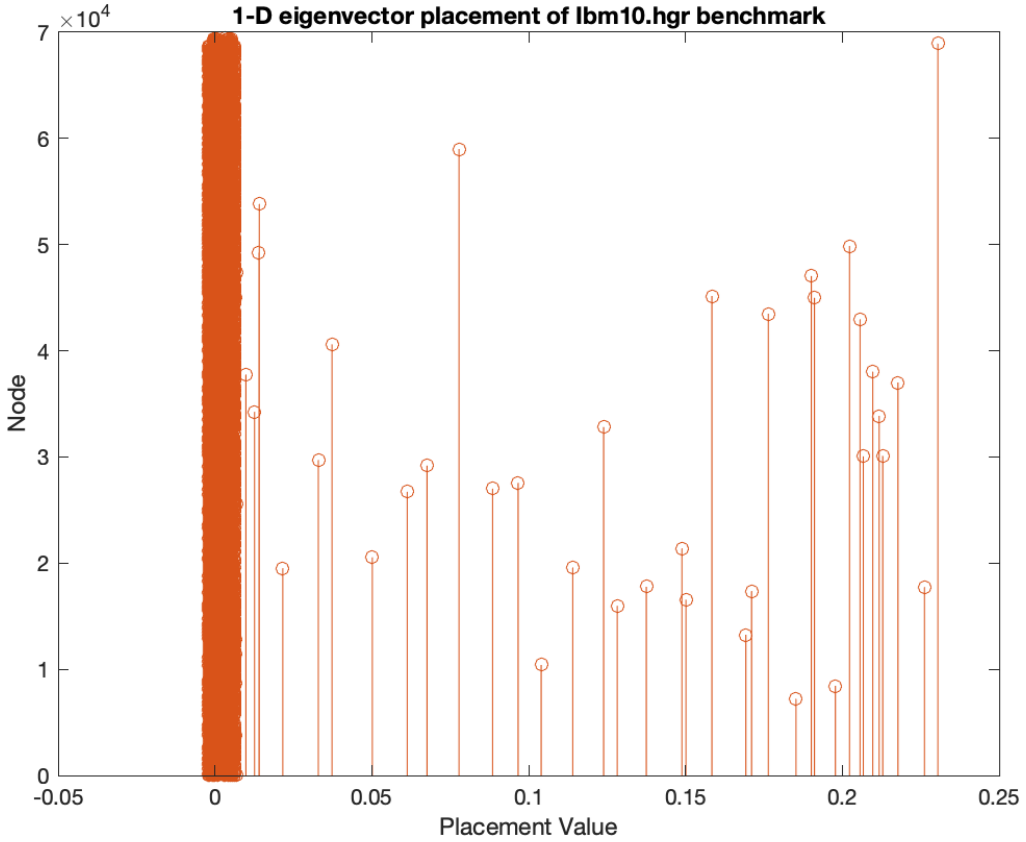
Non-Zero Elements: 1485367 (11.8829 MB)

Ratio: 0.0308143%

Node-to-node Edges: 916464



Ibm10 - I-D placement



Validity : $\sum (\text{placement value})^2 = 1$

ibm18

Sparsity Plot of Benchmark

NetsNum: 201920

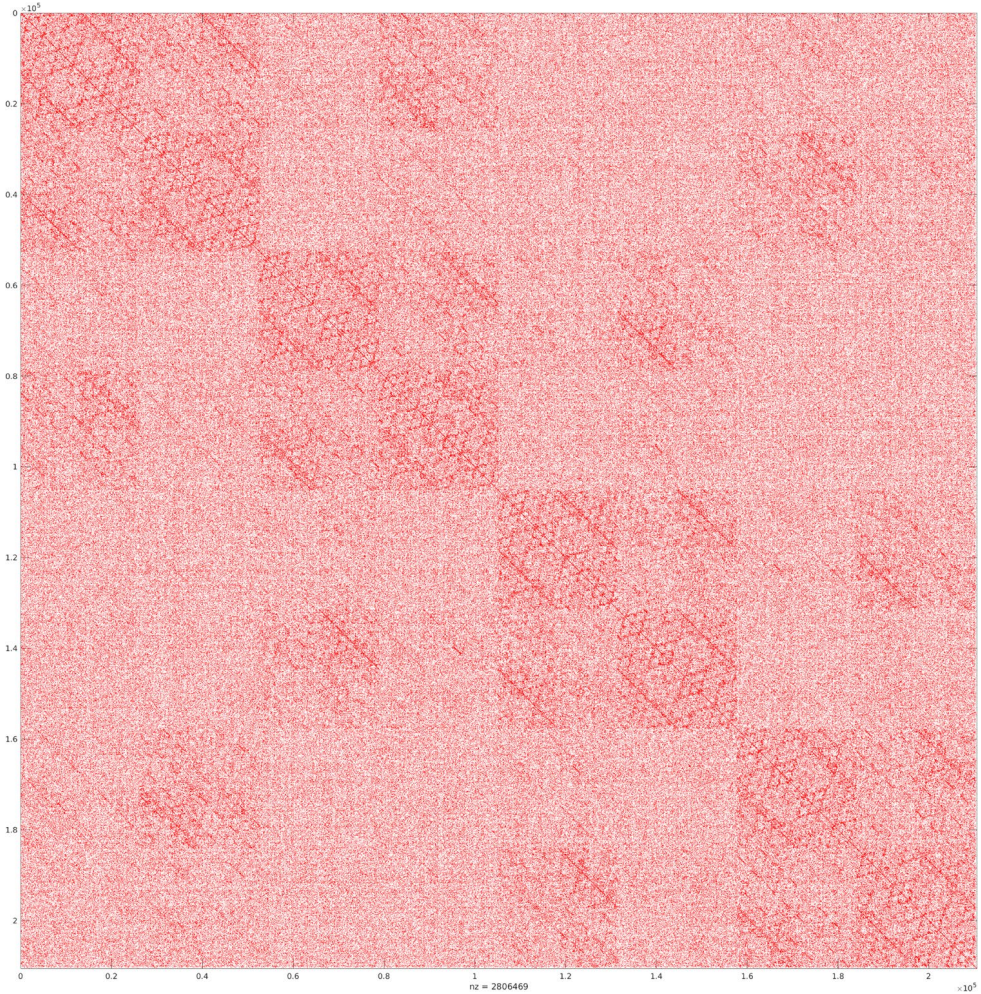
NodesNum: 210613

Size of matrix: 44357835769 (177.43 GB)

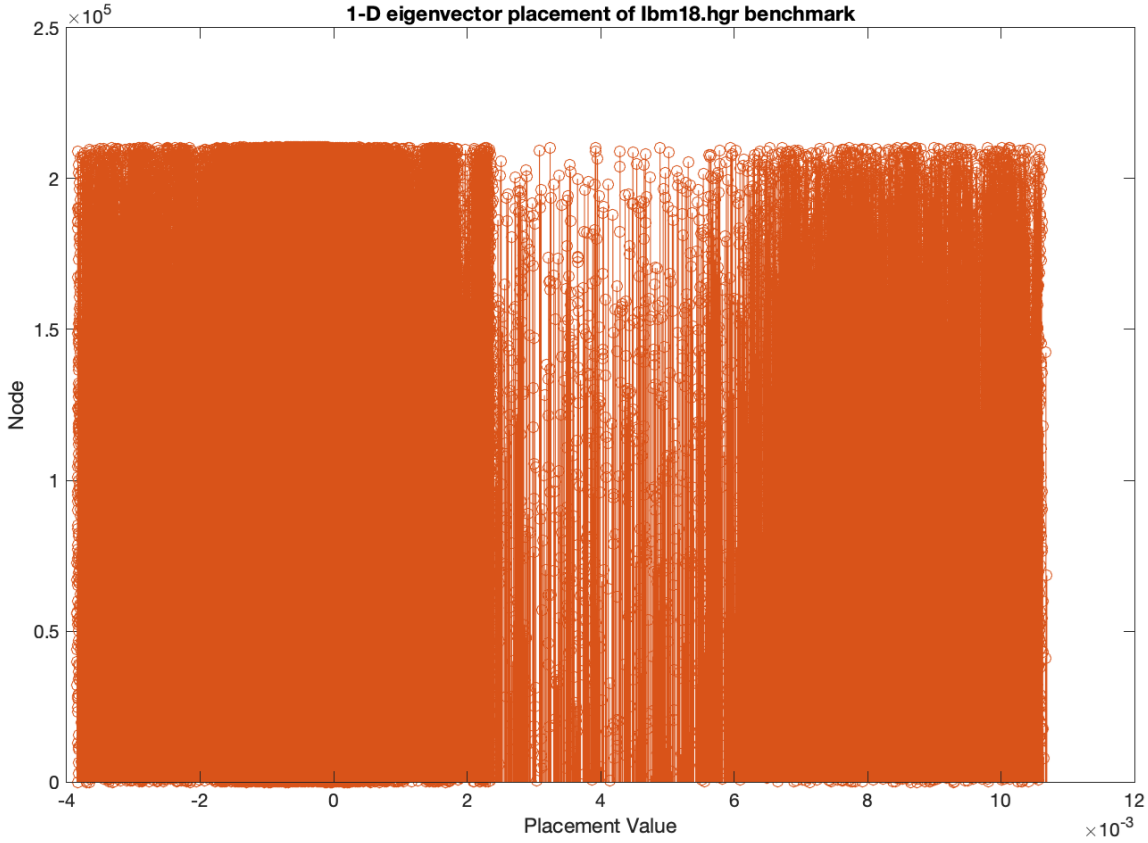
Non-Zero Elements: 4654333 (37.2346 MB)

Ratio: 0.0104927%

Node-to-node Edges: 2839800



ibm18



Validity : $\sum (\text{placement value})^2 = 1$

EIG Results

Benchmark	Total Cells	Left Partition Size	Right Partition Size	CutSize	RatioCut	Runtime (s)
Fract	149	75	74	27.75	5e-3	0.5
Industry2	12637	6319	6318	1153.63	2.89e-5	432
Ibm01	12752	6376	6376	740.948	1.82e-5	31
Ibm10	69429	34715	34714	5229.01	4.34e-6	3227
Ibm18	210613	105307	105306	4038.83	3.6e-7	37758

EIG+KL Hybrid Results

Benchmark	Total Cells	Left Partition Size	Right Partition Size	CutSize	RatioCut	Runtime (s)
Fract	149	75	74	**21.5**	5e-3	0.01
Industry2	12637	6319	6318	**957.714**	2.89e-5	24.159
Ibm01	12752	6376	6376	**501.365**	1.82e-5	11.625
Ibm10	69429	34715	34714	**4048.24**	4.34e-6	617.88
Ibm18	210613	105307	105306	**3245.44**	3.6e-7	3178.55

Summary:

- Heuristics With KL still improved timing but not noticeably for largest benchmark
- Result of Hybrid computation of both benchmarks proved significant for timing and saw Cutsize results improve my more than 20%
- Partitioning solution plot for 2 largest benchmarks proved to be time consuming to attain.
- Sparse Matrix prove to enhance space complexity as matrix size became larger and denser.
- View ReadMe on how to run code :)