MIN-CUT PLACEMENT WITH TERMINAL PROPAGATION

- Mayura Oak
- Melonia Mendonca

Objectives:

- Brief description
- Overview of Implementation
- Implementation issues
- Results
- Future scope

Brief Description

- Algorithm for placement of cells which targets wirelength reduction.
- Uses partitioning for performing placement
- We implemented the KL Algorithm - swap based
- 3 Types of Partitioning:
    a. Depth First
    b. Breadth First without terminal propagation
    c. Breadth First with terminal Propagation

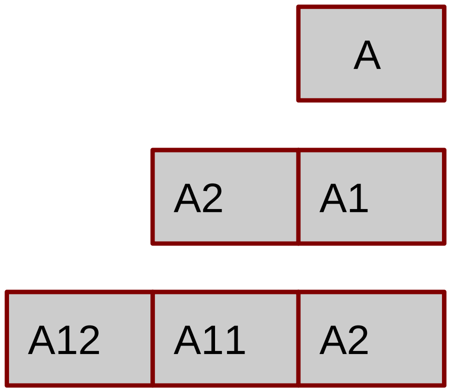
Overview of Implementation

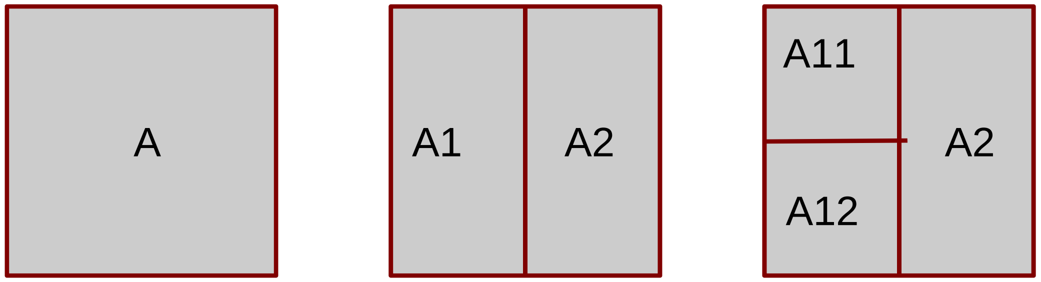
Recursion for Depth First Partitioning:

- recursive calls to self by each partition

Queue for Breadth First Partitioning:

- A queue is maintained to store data for each partition such as cells and number of cells in that partition and whether the partition was horizontal or vertical
- Each finished partition is pushed in from the back of the queue and new partition for subpartitioning is taken from the front of the queue. The previous partition is deleted from the queue.





Overview of Implementation

Adding Anchor nodes for Terminal Propagation:

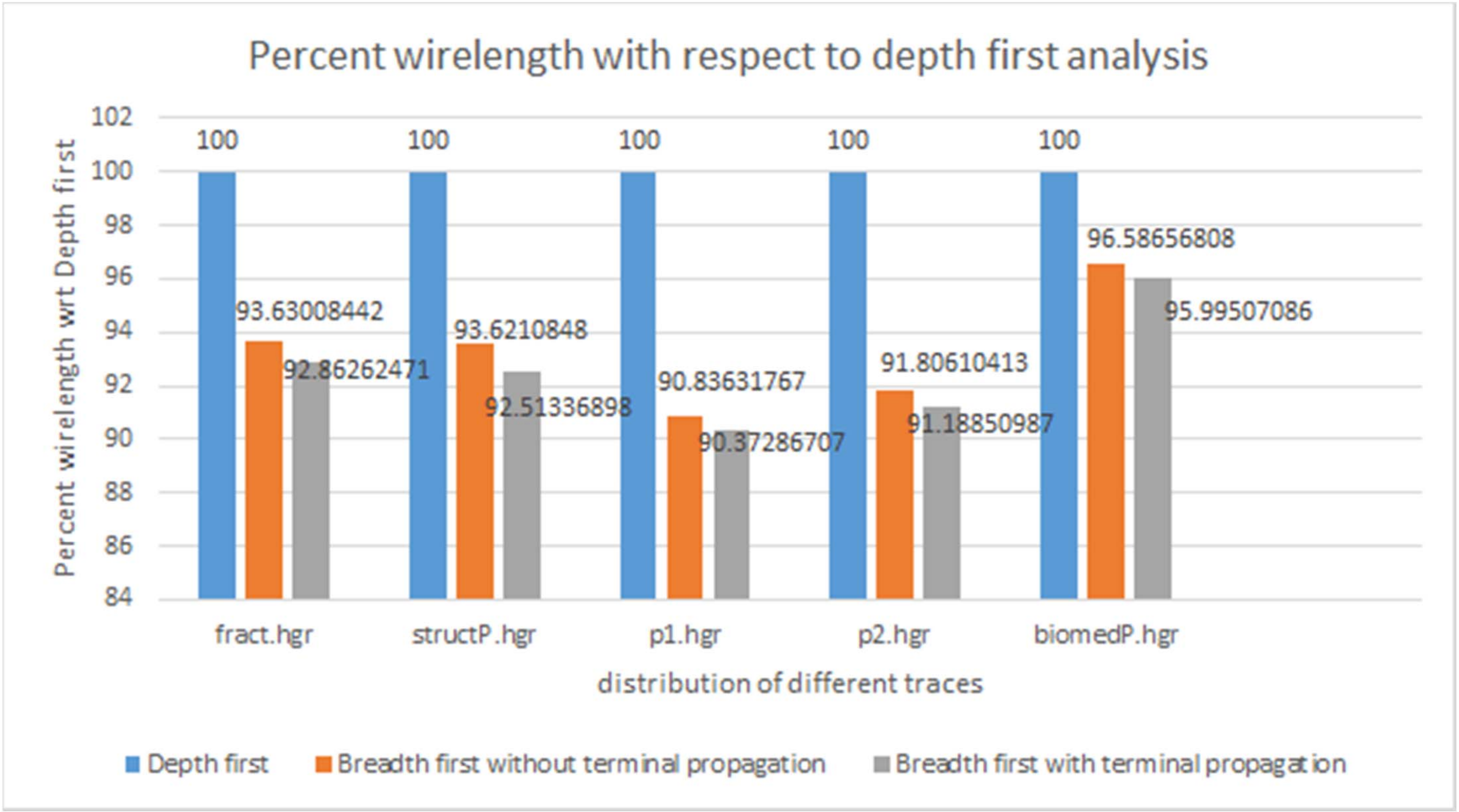
- The Anchor nodes were added in each partition which represented the external connection to the cells in that connection.
- If node 'a' has connection to external cells b, c, d which lie on the left side of the partition, then one anchor node will be added to left partition and weight of the edge connecting a and anchor node will be sum of the weights of the edges connecting a and b,c,d.
- These anchor nodes are always locked and thus can't be swapped into another partition by KL.

Implementation Issues

- We selected KL algorithm as the partitioning algorithm because of simplicity of implementation and as it gives better area balance. But the time complexity of KL is O(n^3), thus its run time increases exponentially as the number of cells and number of partitions. Thus unfortunately we could not perform experiments on bigger circuits such as ibm01.hgr

- Initially a binary tree was maintained to store all the partitions performed, and to check which of them fall outside the window, the tree was traversed till it reaches the leaves. This implementation was complex. Instead we made an array which stores x & y co-ordinates and checked it against window size constraint which was also in terms of x & y to determine whether that cell lies outside the window.

Results



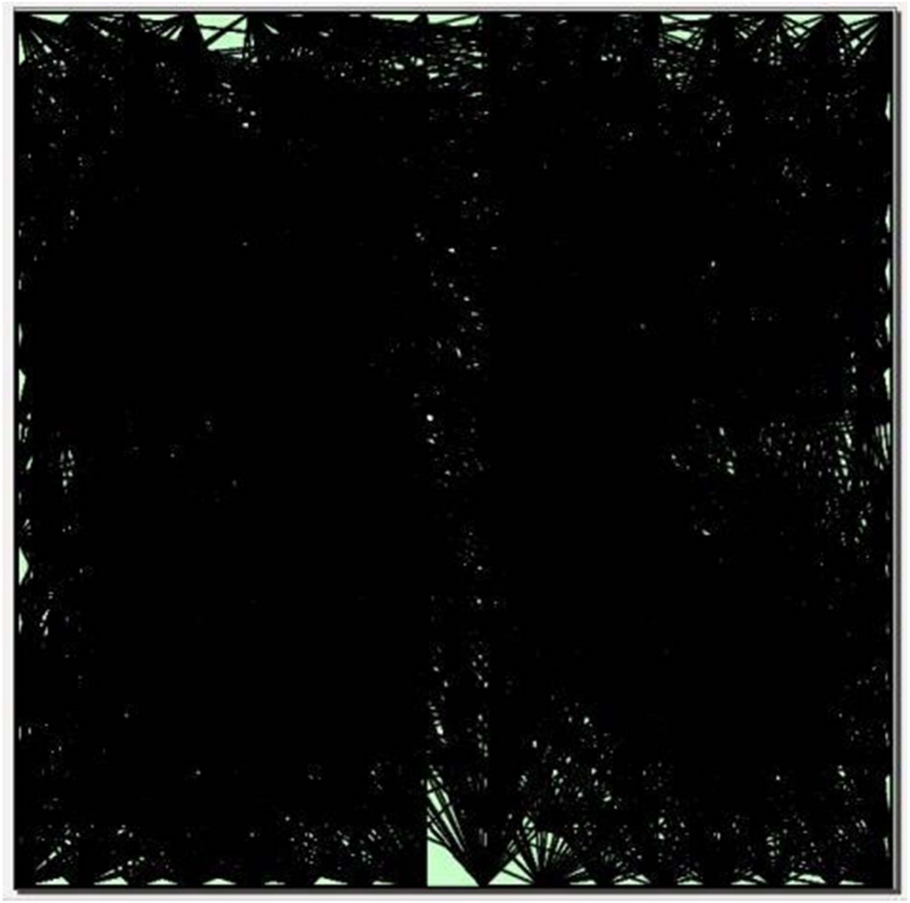
Percent wirelength with respect to depth first analysis
Percent wirelength wrt Depth first
102
100
98
96
94
92
90
88
86
84
100
100
100
100
100
93.63008442
92.86262471
93.6210848
92.51336898
90.83631767
90.37286707
91.80610413
91.18850987
96.58656808
95.99507086
fract.hgr
structP.hgr
p1.hgr
p2.hgr
biomedP.hgr
distribution of different traces
Depth first
Breadth first without terminal propagation
Breadth first with terminal propagation

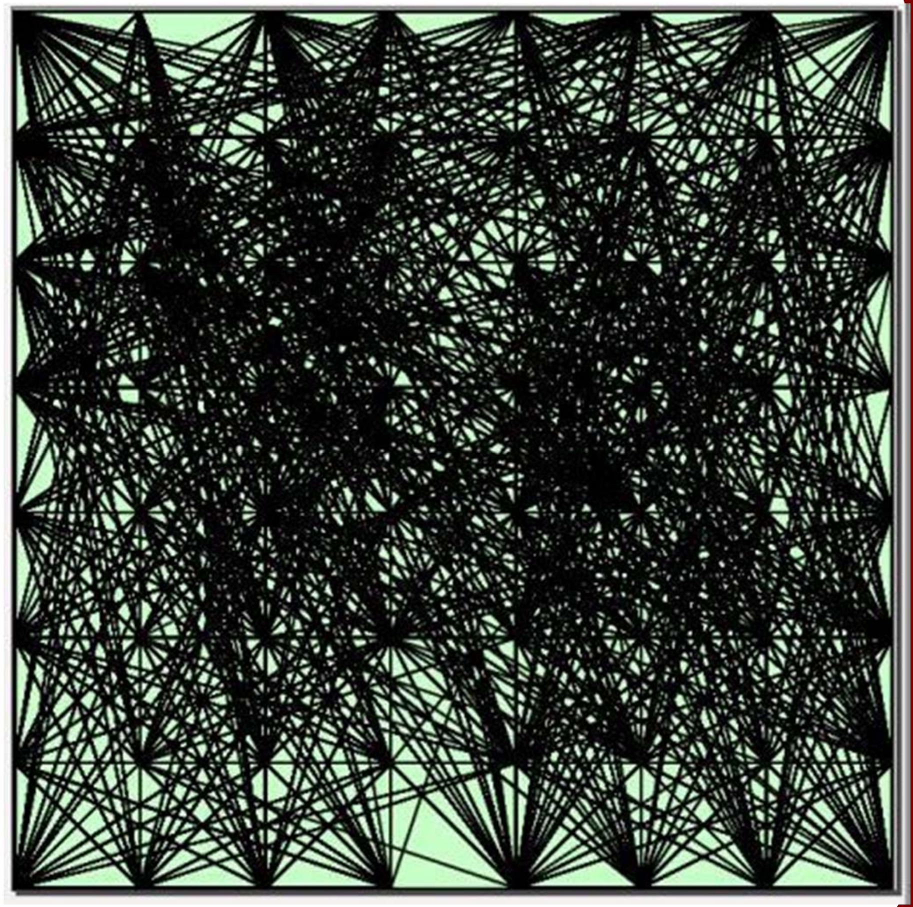
Results

libboard: A vector Graphics C++ Library used for plotting. Results for p2.hgr for 8x8 grid & 50% window size:

Without Terminal Propagation (WL = 13925)

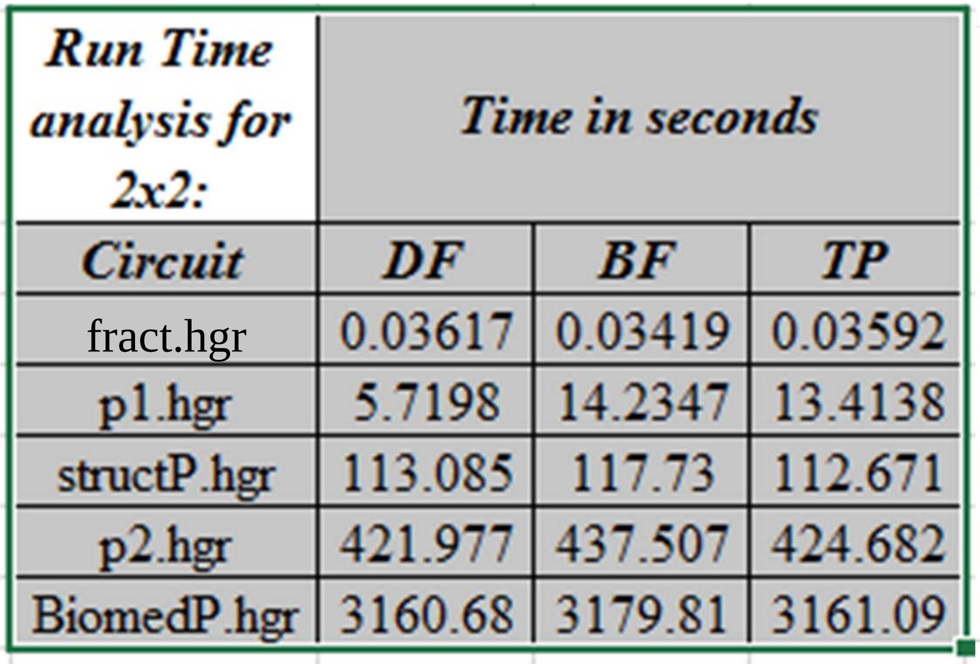
With Terminal Propagation(WL



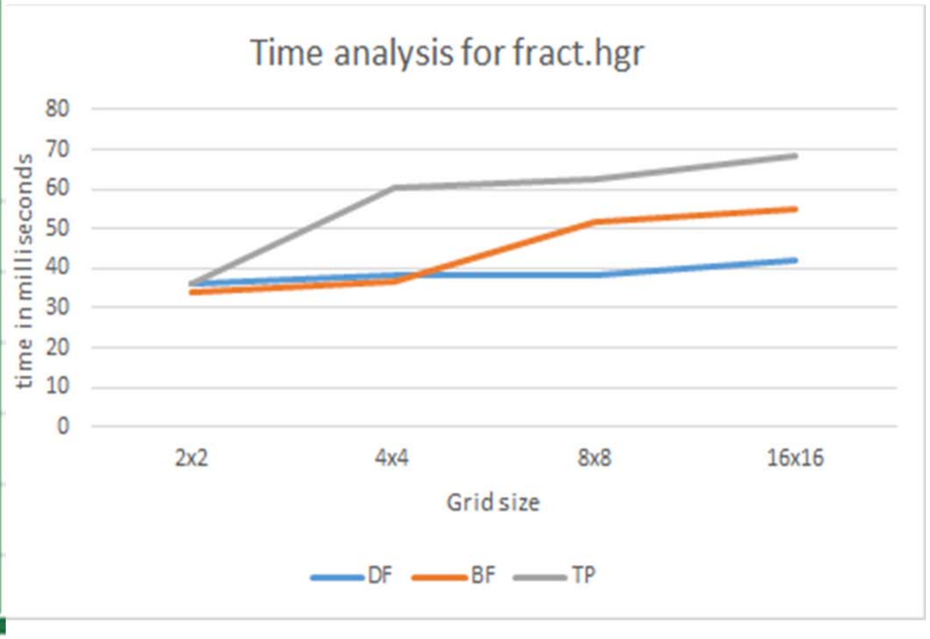


Results

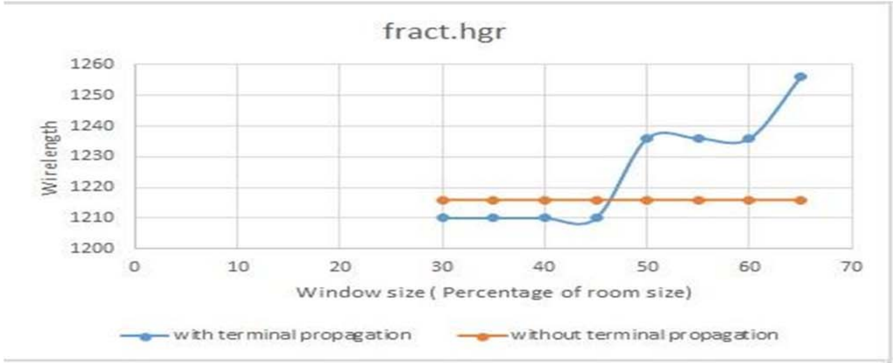
Run Time Analysis:

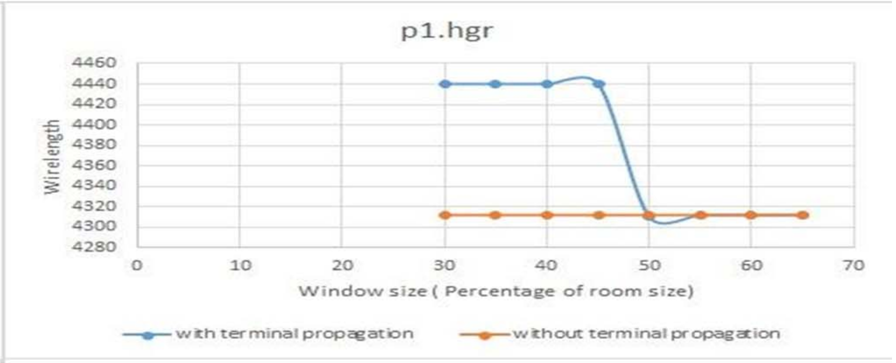


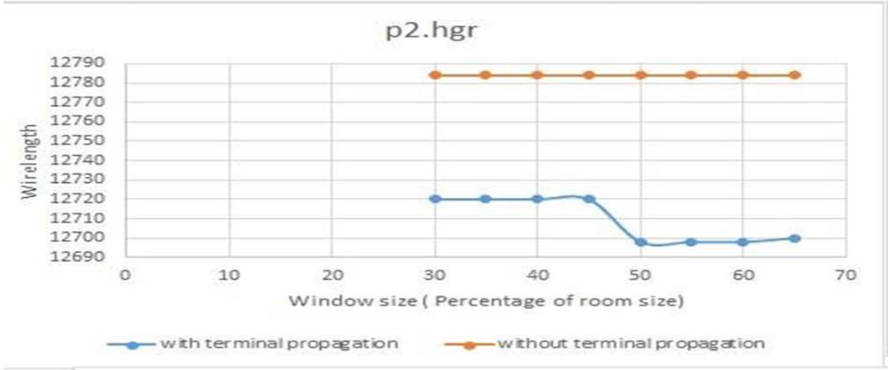
Run Time analysis for 2x2:	***Time in seconds***		
Circuit	***DF***	***BF***	***TP***
fract.hgr	0.03617	0.03419	0.03592
p1.hgr	5.7198	14.2347	13.4138
structP.hgr	113.085	117.73	112.671
p2.hgr	421.977	437.507	424.682
BiomedP.hgr	3160.68	3179.81	3161.09

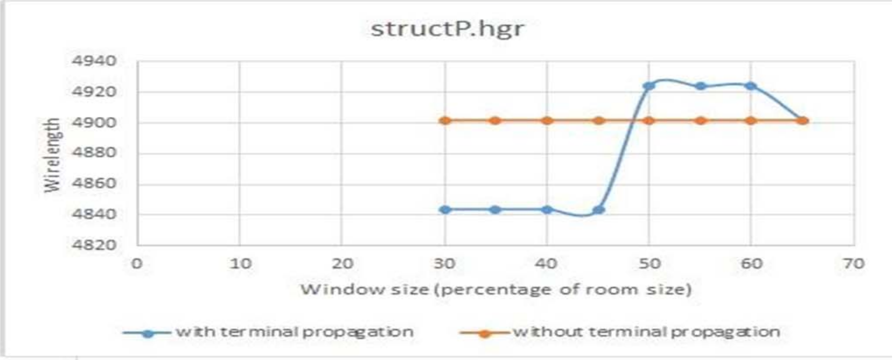


Analysis of varying window size as a percentage of room size.









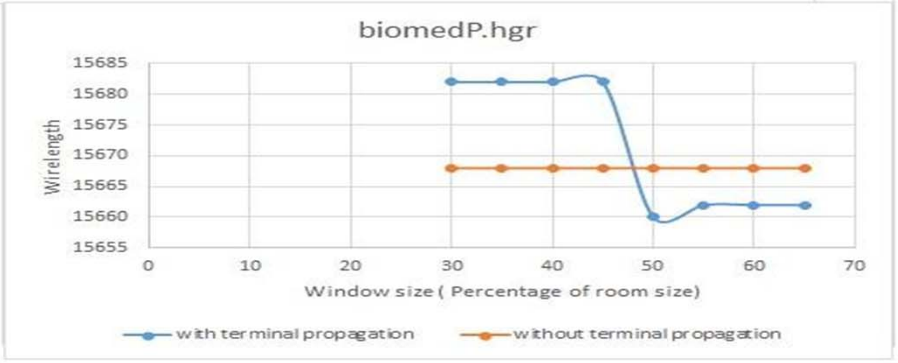
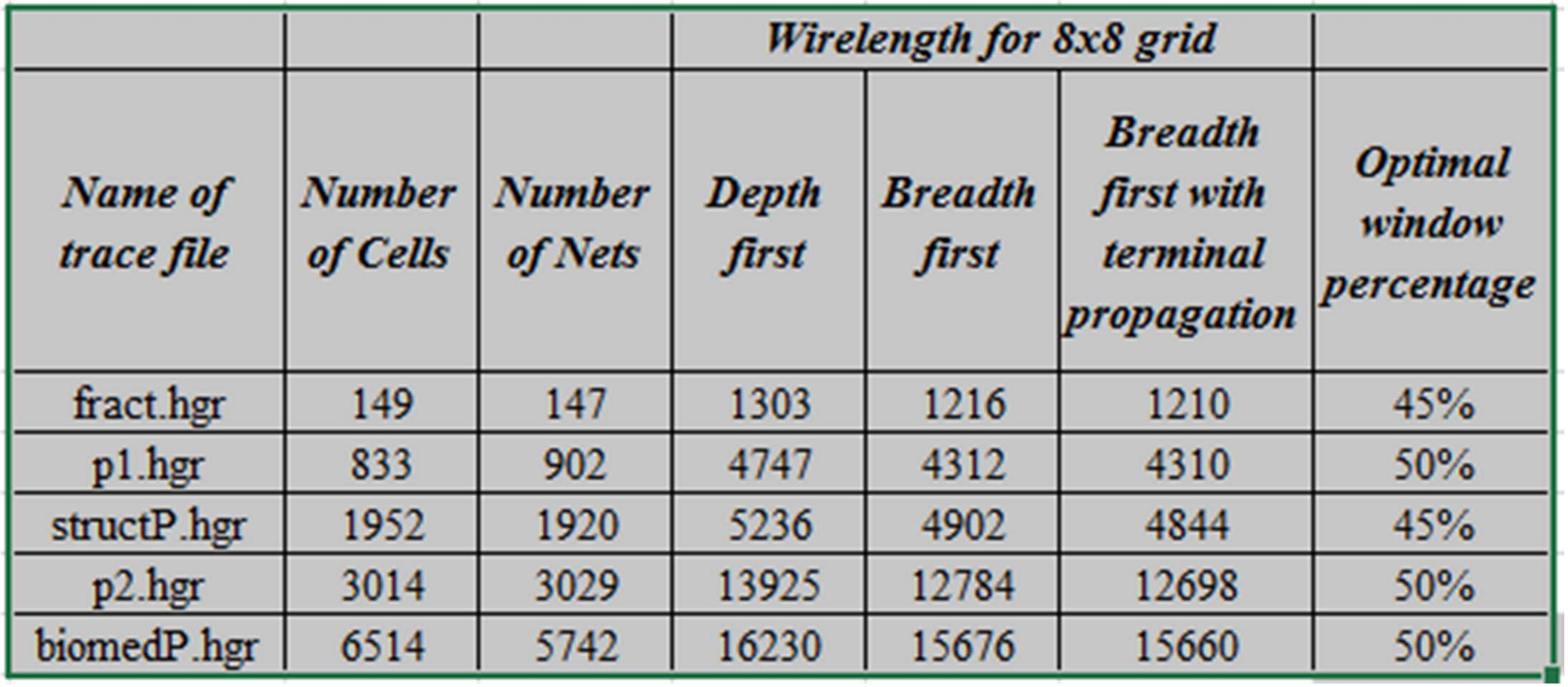


Figure 1

Results

Summary:



			Wirelength for 8x8 grid			
Name of trace file	*Number of Cells*	*Number of Nets*	*Depth first*	*Breadth first*	*Breadth first with terminal propagation*	*Optimal window percentage*
fract.hgr	149	147	1303	1216	1210	45%
p1.hgr	833	902	4747	4312	4310	50%
structP.hgr	1952	1920	5236	4902	4844	45%
p2.hgr	3014	3029	13925	12784	12698	50%
biomedP.hgr	6514	5742	16230	15676	15660	50%

Future Scope

- Reduce run time by further optimizing the code
- Keep bound on number of external connections that are considered for terminal propagation so that they don't overpower internal connections

Thank you!

Questions?