

CS137: Electronic Design Automation

Day 7: January 28, 2002
Partitioning 2
(spectral, network flow,
replication)



CALTECH CS137 Winter2002 -- DeHon

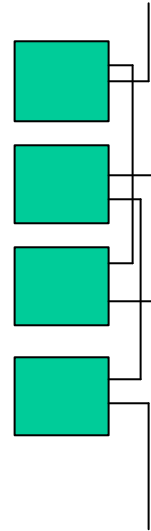
Today

- Alternate views of partitioning
- Two things we can solve optimally
 - (but don't exactly solve our original problem)
- Techniques
 - Linear Placement w/ squared wire lengths
 - Network flow MinCut

CALTECH CS137 Winter2002 -- DeHon

Optimization Target

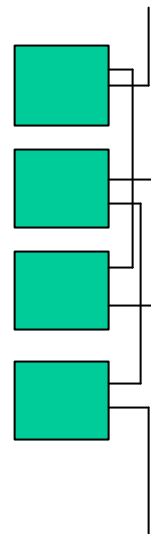
- Place cells
- In linear arrangement
- Wire length between connected cells:
 - distance = $X_i - X_j$
 - cost is sum of distance squared
- Want to pick X_i 's to minimize cost



CALTECH CS137 Winter2002 -- DeHon

Why this Target?

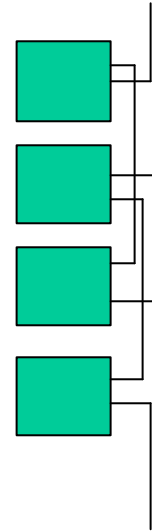
- Minimize sum of squared wire distances
- Prefer:
 - **Area:** minimize channel width
 - **Delay:** minimize critical path length



CALTECH CS137 Winter2002 -- DeHon

Why this Target?

- Our preferred targets are discontinuous and discrete
- Cannot formulate analytically
- Not clear how to drive toward solution
 - Does reducing the channel width at a non-bottleneck help or not?
 - Does reducing a non-critical path help or not?



CALTECH CS137 Winter2002 -- DeHon

Spectral Ordering

Minimize Squared Wire length -- 1D layout

- Start with connection array C ($c_{i,j}$)
- “Placement” Vector X for x_i placement
- State problem:
 - Minimize cost = $0.5 * \sum (\text{all } i,j) (x_i - x_j)^2 c_{i,j}$
 - cost sum is $X'BX$
 - $B = D - C$
 - $D = \text{diagonal matrix, } d_{i,i} = \sum (\text{over } j) c_{i,j}$

CALTECH CS137 Winter2002 -- DeHon

Spectral Ordering

- Constraint: $X'X=1$
 - prevent trivial solution all x_i 's =0
- Minimize cost= $X'BX$ w/ constraint
 - minimize $L=X'BX-\lambda(X'X-1)$
 - $\partial L / \partial X = 2BX - 2\lambda X = 0$
 - $(B-\lambda I)X=0$
 - $X \rightarrow$ Eigenvector of B
 - cost is Eigenvalue λ

CALTECH CS137 Winter2002 -- DeHon

Spectral Solution

- Smallest eigenvalue is zero
 - Corresponds to case where all x_i 's are the same Uninteresting
- Second smallest eigenvalue (eigenvector) is the solution we want

CALTECH CS137 Winter2002 -- DeHon

Spectral Ordering

- X (x_i 's) continuous
- use to order nodes
 - real problem wants to place at discrete locations
 - this is one case where can solve ILP from LP
 - Solve LP giving continuous x_i 's
 - then move back to closest discrete point

CALTECH CS137 Winter2002 -- DeHon

Spectral Ordering Option

- With iteration, can reweigh connections to change cost model being optimized
 - linear
 - (distance)^{1..X}
- Maybe: encourage “closeness”
 - by weighting connection between nodes
 - (have to allow some nodes to not be close)

CALTECH CS137 Winter2002 -- DeHon

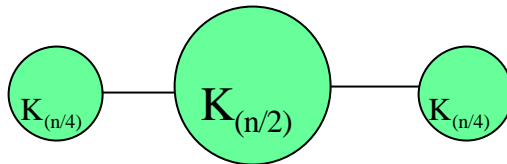
Spectral Partitioning

- Can form a basis for partitioning
- Attempts to cluster together connected components
- Form cut partition from ordering
 - *E.g.* Left half of ordering is one half, right half is the other

CALTECH CS137 Winter2002 -- DeHon

Spectral Ordering

- Midpoint bisect isn't necessarily best place to cut, consider:



CALTECH CS137 Winter2002 -- DeHon

Fanout

- How do we treat fanout?
- As described assumes point-to-point nets
- For partitioning, pay price when cut something once
 - *i.e.* the accounting did last time for KLFM
- Also a discrete optimization problem
 - Hard to model analytically

CALTECH CS137 Winter2002 -- DeHon

Spectral Fanout

- Typically:
 - Treat all nodes on a single net as fully connected
 - Model links between all of them
 - Weight connections so cutting in half counts as cutting the wire
 - Threshold out high fanout nodes
 - If connect to too many things give no information

CALTECH CS137 Winter2002 -- DeHon

Spectral Partitioning Options

- Can bisect by choosing midpoint
 - (not strictly optimizing for minimum bisect)
- Can relax cut criteria
 - min cut w/in some δ of balance
- Ratio Cut
 - minimize $(\text{cut}/|A||B|)$
 - idea tradeoff imbalance for smaller cut
 - more imbalance \rightarrow smaller $|A||B|$
 - so cut must be much smaller to accept

CALTECH CS137 Winter2002 -- DeHon

Spectral vs. FM

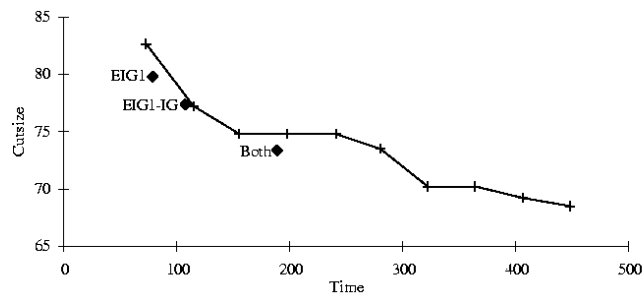


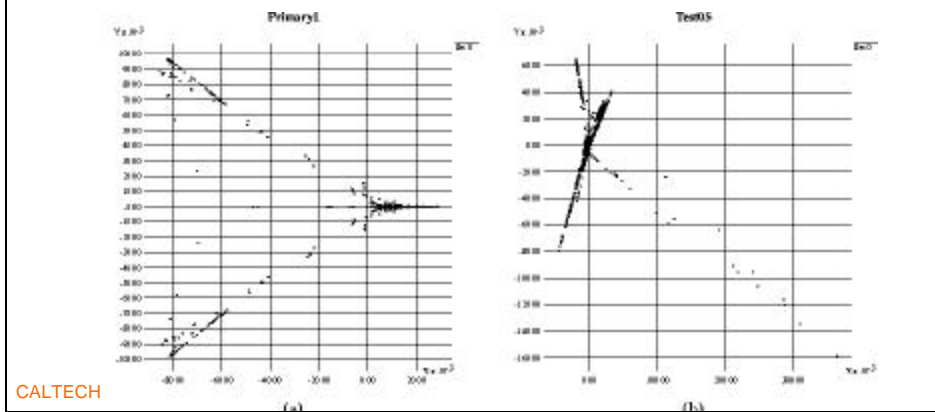
Figure 5. Graphs of cutsizes for different numbers of runs of our optimized version of KLFM versus the spectral initialization approaches. Values shown are the geometric means of the results for the 9 test circuits (all but industry3).

From Hauck/Boriello '96

CALTECH CS137 Winter2002 -- DeHon

Improving Spectral

- More Eigenvalues
 - look at clusters in n-d space
 - 5--70% improvement over EIG1



Spectral Note

- Unlike KLFM, attacks global connectivity characteristics
- Good for finding “natural” clusters
 - hence use as clustering heuristic for multilevel algorithms (both types)

Spectral Theory

- There are conditions under which spectral is optimal [Boppana/FOCS28 (1987)]
 - $B=A+\text{diag}(d)$
 - $g(G,d)=[\text{sum}(B)-n*\lambda(B_S)]/4$
 - B_S mapping Bx to closest point on S
 - $h(G) = \max d g(G,d)$
- $h(G)$ lower bound on cut size

CALTECH CS137 Winter2002 -- DeHon

Spectral Theory

- Boppana paper gives a probabilistic model for graphs
 - model favors graphs with small cuts
 - necessary since *truly* random graph has cut size $O(n)$
 - shows high likelihood of bisection being lower bound
- In practice
 - known to be **very** weak for graphs with large cuts

CALTECH CS137 Winter2002 -- DeHon

Max Flow

MinCut

CALTECH CS137 Winter2002 -- DeHon

MinCut Goal

- Cut find maximum flow (mincut) between a source and a sink
 - no balance guarantee

CALTECH CS137 Winter2002 -- DeHon

MaxFlow

- Set all edge flows to zero
 - $F[u,v]=0$
- While there is a path from s,t
 - (breadth-first-search)
 - for each edge in path $f[u,v]=f[u,v]+1$
 - $f[v,u]=-f[u,v]$
 - When $c[v,u]=f[v,u]$ remove edge from search
- $O(|E| \cdot \text{cutsizesize})$
- [Our problem simpler than general case CLR]

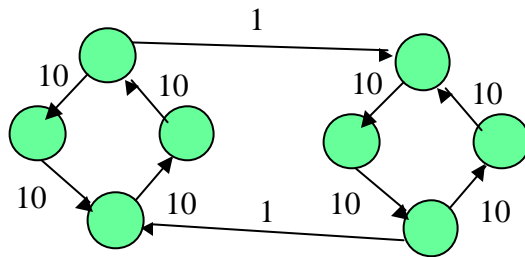
CALTECH CS137 Winter2002 -- DeHon

Technical Details

- For min-cut in graphs,
 - Don't really care about directionality of cut
 - Just want to minimize wire crossings
- Fanout
 - Want to charge discretely ...cut or not cut
- Pick start and end nodes?

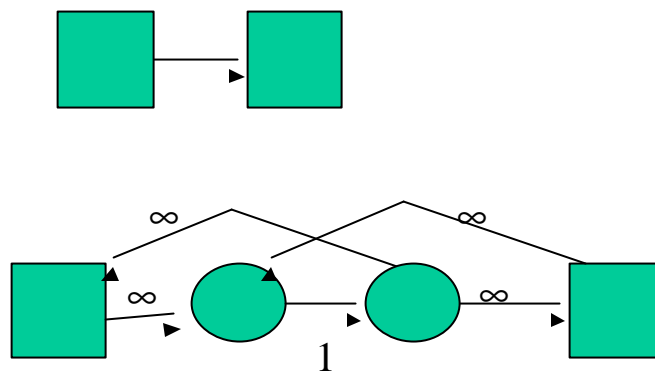
CALTECH CS137 Winter2002 -- DeHon

Directionality



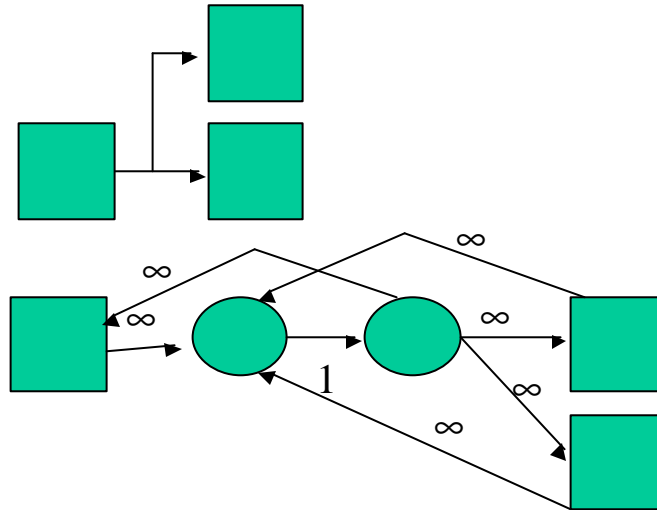
CALTECH CS137 Winter2002 -- DeHon

Directionality Construct



CALTECH CS137 Winter2002 -- DeHon

Fanout Construct



CALTECH CS137 Winter2002 -- DeHon

Extend to Balanced Cut

- Pick a start node and a finish node
- Compute min-cut start to finish
- If halves sufficiently balanced, done
- else
 - collapse all nodes in smaller half into one node
 - pick a node adjacent to smaller half
 - collapse that node into smaller half
 - repeat from min-cut computation

FBB -- Yang/Wong ICCAD'94

CALTECH CS137 Winter2002 -- DeHon

Observation

- Can use residual flow from previous cut when computing next cuts
- Consequently, work of multiple network flows is only $O(|E| \cdot \text{final_cut_cost})$

CALTECH CS137 Winter2002 -- DeHon

Picking Nodes

- Optimal:
 - would look at all s, t pairs
 - Just for first cut is merely $N-1$ “others”
 - ... $N/2$ to guarantee something in second half
 - Anything you pick **must** be in separate halves
 - Assuming there is perfect/ideal bisection
 - If pick randomly, probability in different halves is 50%
 - Few random selections likely to yield s, t in different halves
 - also would look at all nodes to collapse into smaller
 - could formulate as branching search

CALTECH CS137 Winter2002 -- DeHon

Picking Nodes

- Probably be more clever:
 - *i.e.* everything in first group around s (or t) would be there regardless of which s or t picked as kernel
 - probably similar observations would reduce branching factor on picking node to collapse into smaller cluster

CALTECH CS137 Winter2002 -- DeHon

Picking Nodes

- Randomly pick
 - (maybe try several starting points)
- With small number of adjacent nodes,
 - could afford to branch on all

CALTECH CS137 Winter2002 -- DeHon

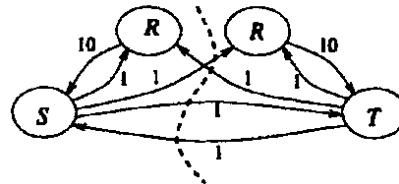
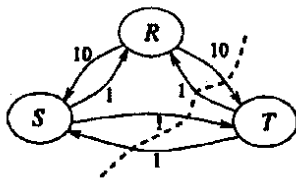
Approximation

- Can find 1/3 balanced cuts within $O(\log(n))$ of best cut in polynomial time
 - algorithm due to Leighton and Rao
 - exposition in Hochbaum's *Approx. Alg.*
- Bound cut size
 - Boppana -- lower bound $h(G)$
 - Boppana/Spectral cut -- one upper bound
 - Leighton/Rao -- upper bound

CALTECH CS137 Winter2002 -- DeHon

Min Cut Replication

- Noted last time could use replication to reduce cut size
 - Observed could use FM to replicate
- Can solve unbounded replication optimally with min-cut



CALTECH CS137 Winter2002 -- DeHon

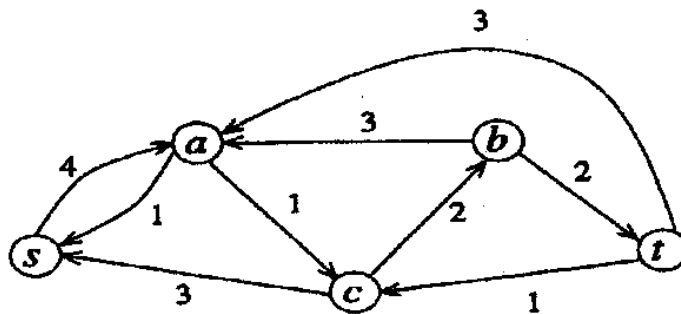
[Liu, Kuo, Cheng TRCAD v14n5p623]

Min-Cut replication

- Key Idea:
 - Create two copies of net
 - One copy connected to source, other to sink
 - Provide “free” links between them
 - Take mincut
 - Allows nodes to be associated with both ends

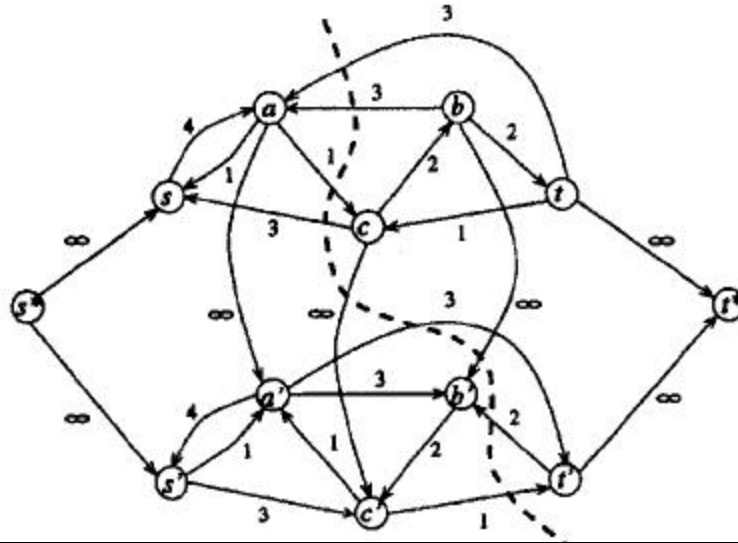
CALTECH CS137 Winter2002 -- DeHon

Min-cut Example



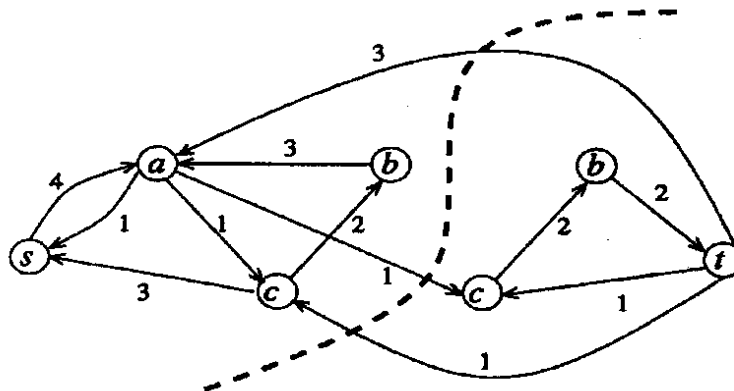
CALTECH CS137 Winter2002 -- DeHon

Min-cut Ex. Replication Graph



CA

Min-cut with Replication



CALTECH CS137 Winter2002 -- DeHon

Replication Note

- Cut of minimum width is not unique
 - Similar to phenomenon saw in LUT covering with network flow
- Want to identify **minimum** size replication set for given flow
- Can do by reweighing graph and another min-cut
 - **Idea** weight on replication connections
 - Minimize cut → minimize replication set
 - See Mak/Wong TRCAD v16n10p1221

CALTECH CS137 Winter2002 -- DeHon

Reading

- Added reading for next Monday
 - PDF link in reading
 - Longish paper on Conjugate Gradient
 - Try to scan, at least

CALTECH CS137 Winter2002 -- DeHon

Big Ideas

- Divide-and-Conquer
- Techniques
 - flow based
 - numerical/linear-programming based
- Exploit problems we can solve optimally