ESE535: Electronic Design Automation

Day 6: February 4, 2014
Partitioning 2
(spectral, network flow)

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 (time permit)

Optimization Target

• Place cells
• In linear arrangement
• Wire length between connected cells:
  – distance=Xᵢ - Xⱼ
  – cost is sum of distance squared
• Pick Xᵢ’s to minimize cost

Why this Target?

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

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?

Preclass: Initial Placement

• Metrics:
  – Wirelength
  – Squared wirelength
  – Channel width
  – Critical path length
Spectral Ordering

Minimize Squared Wire length -- 1D layout
• Start with connection array C (c_{i,j})
• "Placement" Vector X for x_i placement
• Problem:
  – Minimize cost = \( \frac{1}{2} \sum_{i} \sum_{j} c_{i,j} (x_i - x_j)^2 \)
  – cost sum is \( X^T B X \)
• \( B = D - C \)
• \( D \) = diagonal matrix, \( d_{i,i} = \sum_j c_{i,j} \)

Preclass Netlist

• Squared wire lengths:
  \( (X_A - X_G)^2 \)
  \( + (X_B - X_G)^2 \)
  \( + (X_B - X_H)^2 \)
  \( + (X_C - X_H)^2 \)
  \( + (X_G - X_O)^2 \)
  \( + (X_H - X_O)^2 \)

C Matrix

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>G</th>
<th>H</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

D Matrix

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>G</th>
<th>H</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>1</td>
<td>1</td>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

B=D-C Matrix

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>G</th>
<th>H</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td></td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>-1</td>
<td>-1</td>
<td>3</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>-1</td>
<td>-1</td>
<td>3</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O</td>
<td></td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BX

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>G</th>
<th>H</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>X_A</td>
<td></td>
<td></td>
<td></td>
<td>X_A</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>X_B</td>
<td></td>
<td></td>
<td>X_B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>X_C</td>
<td></td>
<td></td>
<td></td>
<td>X_C</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
<td>X_G</td>
<td></td>
<td>X_G</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td></td>
<td></td>
<td>X_H</td>
<td></td>
<td>X_H</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td>X_O</td>
<td></td>
<td>X_O</td>
</tr>
</tbody>
</table>

= 

\[
\begin{align*}
X_A & \cdot X_G \\
2X_B & \cdot X_G \cdot X_H \\
X_C & \cdot X_H \\
3X_G & \cdot X_A \cdot X_B \cdot X_O \\
3X_H & \cdot X_B \cdot X_C \cdot X_O \\
2X_O & \cdot X_G \cdot X_H 
\end{align*}
\]
Can See Will Converage To...

- Squared wire lengths:
  - $(X_A - X_G)^2$
  - $(X_B - X_G)^2$
  - $(X_C - X_G)^2$
  - $(X_D - X_G)^2$
  - $(X_E - X_G)^2$
  - $(X_F - X_G)^2$

$$((X_A - X_G)^2 + (X_B - X_G)^2 + (X_C - X_G)^2 + (X_D - X_G)^2 + (X_E - X_G)^2 + (X_F - X_G)^2)$$

Trying to Minimize

- Squared wire lengths:
  - $(X_A - X_G)^2$
  - $(X_B - X_G)^2$
  - $(X_C - X_G)^2$
  - $(X_D - X_G)^2$
  - $(X_E - X_G)^2$
  - $(X_F - X_G)^2$

- Which we know is also $X^T(BX)$

- Make all $X_i$'s same?

- ...but, we probably need to be in unique positions.
Spectral Ordering

• Add constraint: $X^T X = 1$
  – prevent trivial solution all $x_i$'s = 0
• Minimize cost=$X^T BX$ w/ constraint
  – minimize $L = X^T BX - \lambda (X^T X - 1)$
  – $\nabla L / \nabla X = 2BX - 2\lambda X = 0$
  – $(B - \lambda I)X = 0$
  – What does this tell us about $X, \lambda$?
  – $X \rightarrow$ Eigenvector of $B$
  – cost is Eigenvalue $\lambda$.

Spectral Solution

• Smallest eigenvalue is zero
  – Corresponds to case where all $x_i$’s are the same $\rightarrow$ uninteresting
• Second smallest eigenvalue (eigenvector) is the solution we want.

For this $B$ Matrix

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>G</th>
<th>H</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>-1</td>
<td>3</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>-1</td>
<td>-1</td>
<td>3</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
</tbody>
</table>

Eigenvector is:

$$X = \begin{bmatrix} X_A \\ X_B \\ X_C \\ X_D \\ X_G \\ X_H \\ X_O \end{bmatrix} = \begin{bmatrix} 0.6533 \\ 1.116E-14 \\ -0.6533 \\ 0.2706 \\ -0.2706 \\ 1.934E-14 \end{bmatrix}$$

Order from Eigenvector

$$Eigenvector = \begin{bmatrix} X_A \\ X_B \\ X_C \\ X_D \\ X_G \\ X_H \\ X_O \end{bmatrix} = \begin{bmatrix} 0.6533 \\ 1.116E-14 \\ -0.6533 \\ 0.2706 \\ -0.2706 \\ 1.934E-14 \end{bmatrix}$$

Order: $A, G, O, B, H, C$

Quality of this solution? all metrics

Anyone get a solution with a better metric?
Spectral Ordering Option

- Can encourage “closeness”
  - Making some $c_{ij}$ larger
  - Must allow some to be not close
- Could use $c_{ij}$ for power opt
  - $c_{ij} = P_{\text{switch}}$

\[
\begin{array}{cccccc}
A & B & C & G & H & O \\
A & 1 & -1 & & & \\
B & 2 & -1 & -1 & & \\
C & 1 & -1 & & & \\
G & -1 & -1 & 3 & -1 & \\
H & -1 & -1 & 3 & -1 & \\
O & -1 & -1 & 2 & & \\
\end{array}
\]

Spectral Ordering Option

- With iteration, can reweigh connections to change cost model being optimized
  - linear
  - (distance)\(^1\times\)

\[
c_{ij} = \frac{1}{\sqrt{|X_i - X_j|}}
\]

Spectral Partitioning

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

Spectral Partitioning Options

- Can bisect by choosing midpoint
  - (not strictly optimizing for minimum bisect)
- Can relax cut criteria
  - min cut w/in some $\delta$ of balance
- Ratio Cut
  - Minimize (cut/|A||B|)
    - idea tradeoff imbalance for smaller cut
    - more imbalance $\rightarrow$ smaller |A||B|
    - so cut must be much smaller to accept
  - Easy to explore once have spectral ordering
    - Compute at each cut point in $O(N)$ time

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
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 – e.g. 1/(nodes-1)
  - Threshold out high fanout nodes
    - If connect too many things give no information

Spectral Fanout Cut Approximation

- Weight edges: 1/(4-1)=1/3

Spectral vs. FM

- More Eigenvalues
  - look at clusters in n-d space
  - But: 2 eigenvectors is not opt. solution to 2D placement
  - Partition cut is plane in this higher-dimensional 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
- After doing spectral
  - Can often improve incrementally using KLFM pass
  - Remember spectral optimizing squared wirelength, not directly cut width

If Time Permits
Max Flow

MinCut

Max Flow

MinCut Goal

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

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
- $O(|E|^2 \cdot \text{cutsize})$
- [Our problem simpler than general case CLR]

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?

Directionality

For logic net: cutting a net is the same regardless of which way the signal flows

Directionality Construct
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

Observation

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

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 a perfect/ideal bisection
    - If pick randomly, probability different halves: 50%
    - Few random selections likely to yield s,t in different halves
    - would also look at all nodes to collapse into smaller
    - could formulate as branching search

Picking Nodes

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

Big Ideas

- Divide-and-Conquer
- Techniques
  - flow based
  - numerical/linear-programming based
  - Transformation constructs
- Exploit problems we can solve optimally
  - Min-cut
  - Linear ordering
Admin

• Assign 3 due on Thursday
• Reading for Monday online
• Assignment 4 exercise out
  – Should be small part
  – Most effort on partitioning project