Non-tree Routing for Reliability and Yield Improvement

Ion Mandoiu

CSE Department, UC San Diego

Joint work with A.B. Kahng and B. Liu
Outline

• Manufacturing defect models
• Motivation for non-tree routing
• Problem formulation and basic properties
• Exact solution by integer programming
• Greedy heuristic
• Experimental results and conclusions
Manufacturing Reliability Trends

• Manufacturing defects increasingly difficult to control in nanometer processes
  – Cannot expect continued decrease in defect density as minimum feature size gets close to atomic dimensions

• Defects occur at
  – Front end of the line (FEOL), i.e., devices
  – Back end of the line (BEOL), i.e., interconnect and vias
  – BEOL defects are increasingly dominant

• Main BEOL failure mode: spot defects
Spot Defect Classification

- **Intra-layer extra material**
  - short-circuit faults

- **Intra-layer missing material**
  - open circuit faults

- **Inter-layer extra material**
  - open faults (blocked vias)

- **Inter-layer missing material**
  - short-circuit faults (oxide pinholes)
Failure Probability Models

- Chip failure probability given by

$$\text{POF} = P \int_{x_{\text{min}}}^{x_{\text{max}}} \frac{A(X)}{X^3} dX$$

- Where
  - $A(X) = \text{critical area for defect size } X = \text{area in which the center of a spot defect of size } X \text{ must fall to cause a fault}$
  - $P = \text{defect density (uniformly distributed defects)}$
  - $\text{Defect size distribution is given by } X_0^2 / X^3, \text{ where } X_0 \text{ is the peak defect size (Stapper 1984)}$
Failure Probability of a Net

- **Net parameters**
  - \( W \) = wire width
  - \( S \) = spacing
  - \( L \) = net wirelength
  - \( B \) = length of adjacent wires from other nets

- **Inter-layer extra material**
  - Defect size \( X < S \) \( \Rightarrow \) \( C(X) = 0 \) (no failure)
  - Defect size \( X \) between \( S \) and \( 2S + W \) \( \Rightarrow \) \( C(X) = (X-S)B \)
  - Defect size \( X > 2S + W \) \( \Rightarrow \) \( C(X) = (S+W)B \)

\[ \Rightarrow \text{POF} = \alpha B, \text{ where } \alpha = \frac{P X_0^2}{2} \left( \frac{1}{S} - \frac{1}{2S + W} \right) \]
Failure Probability of a Net (Cont.)

- Similarly, POF for inter-layer missing material = $\beta_L$, (see Huijbregts, Xue & Jess 1995)
- Overall, POF for a given net $\cong \alpha B + \beta L$, where $\alpha$ and $\beta$ are determined by the process and design rules
Outline

• Manufacturing defect models
• Motivation for non-tree routing
• Problem formulation and basic properties
• Exact solution by integer programming
• Greedy heuristic
• Experimental results and conclusions
Previous Work

• **Focused on reduction of short critical area**
  – Conservative design rules
  – Decompaction
  – Routing for reliable manufacturing
    • DTR = Defect Tolerant Routing (Pitaksanokul et al. 1985)
    • YOR = Yield Optimizing Routing (Kuo 1993)
    • Reliability-aware routing costs (Huijbregts, Xue & Jess 1995)

• **Open faults become increasingly dominant**
  – Changes in manufacturing processes
    • Aluminum interconnects etched → defect modality = short faults
    • Copper interconnects deposited → defect modality = open faults
Opens vs. Shorts - POF

- Open faults are significantly (3x) more likely to occur

(Source: de Gyvez, SLIP01)
Opens vs. Shorts - Critical Area (CA)

Open fault CA larger than short fault CA

(Source: de Gyvez, SLIP01)
Techniques for Open CA Reduction

• **Wire doubling**
  – Simple, easy to integrate in current design flows
  – Can be applied to all nets

• **Non-tree routing**
  – Still easy to integrate in current flows (post-processing approach)
  – Appropriate for non timing-critical nets
  – Potentially more effective
    • How effective?
Outline

• Manufacturing defect models
• Previous work on routing for reliable manufacturing
• Motivation for non-tree routing
• Problem formulation and basic properties
• Exact solution by integer programming
• Greedy heuristic
• Experimental results and conclusions
Problem Formulation

Manhattan Routed Tree Augmentation (MRTA) Problem

Given:
- Tree T routed in the Manhattan plane
- Feasible routing region FRR
- Wirelength increase budget W

Find:
- Augmenting paths A within FRR

Such that:
- Total length of augmenting paths is less than W
- Total length of biconnected edges in $T \cup A$ is maximum

- Wirelength increase budget used to balance open CA decrease with short CA increase
Types of Allowed Augmenting Paths

(A) Paths parallel to tree edges

(B) Paths connecting tree nodes (including corners)

(C) Paths connecting tree nodes or “projections” of tree nodes onto adjacent tree edges

(D) Arbitrary paths on the Hanan grid defined by tree nodes and FRR corners
Theorem: MRTA has an optimum solution on the Hanan grid defined by tree nodes and FRR corners.

Hanan Grid Theorem

Sliding in at least one direction is not decreasing biconnectivity

Re-embedding along Hanan grid does not decrease biconnectivity
Hanan Grid Theorem

Theorem: MRTA has an optimum solution on the Hanan grid defined by tree nodes and FRR corners.

- Sliding in at least one direction is not decreasing biconnectivity
- Re-embedding along Hanan grid does not decrease biconnectivity
Outline

• Manufacturing defect models
• Previous work on routing for reliable manufacturing
• Motivation for non-tree routing
• Problem formulation and basic properties
• Exact solution by integer programming
• Greedy heuristic
• Experimental results and conclusions
Integer Linear Program (Type A-C Paths)

Maximize \( \sum_{e \in T} y_e l(e) + \left( W - \sum_{p \in P} x_p l(p) \right) \) \text{ Total biconnected length}

Subject to:

\[ \sum_{p \in P} x_p l(p) \leq W \] \text{ Wirelength budget}

\[ \sum_{p \text{ connects } T_u \text{ to } T_v} x_p \geq y_e \quad \forall e \in T \] \text{ e biconnected if } \exists p \text{ connecting } T_u \text{ & } T_v

\[ x_p \in \{0,1\} \quad \forall p \in P \] \{e|x_e=1\} gives augmenting paths

\[ y_e \in \{0,1\} \quad \forall e \in T \] \{e|y_e=1\} gives biconnected tree edges

- \( P \) = set of -- at most \( O(n^2) \) -- augmenting paths
- WL budget is fully utilized by (implicit) parallel paths
Integer Linear Program (type D paths)

Maximize \( \sum_{e \in T} y_e l(e) + \left( W - \sum_{e \in H} x_e l(e) \right) \)

Subject to

\[ \sum_{e \in H} x_e l(e) \leq W \]

\[ \sum_{e \in X} x_e \geq y_{(u,v)} \forall (u,v) \in T, X \text{ cut separating } u \text{ and } v \]

\[ x_e \in \{0,1\} \forall e \in H \]

\[ y_e \in \{0,1\} \forall e \in T \]

- \( H = \) Hanan grid defined tree nodes and FRR corners
- Exponentially many cut constraints
  - Fractional relaxation can still be solved using the ellipsoid algorithm
Outline

• Manufacturing defect models
• Previous work on routing for reliable manufacturing
• Motivation for non-tree routing
• Problem formulation and basic properties
• Exact solution by integer programming
• Greedy heuristic
• Experimental results and conclusions
Greedy MRTA Algorithm

Input: Routed tree T, wirelength budget W, feasible routing region, set V of allowed augmenting path endpoints
Output: Augmented routing T ∪ A, with l(A) ≤ W

1. A = {}; mark all edges of T as bridges
2. Compute augmenting path lengths between every u,v ∈ V by |V| Dijkstra calls
3. Compute length of bridges on tree path between every u,v ∈ V by |V| DFS calls
4. Find path p with l(p) ≤ W and max ratio between length of bridges on the tree path between ends of p and l(p)
5. If ratio ≥ 1 then
   - Add p to A
   - Mark all edges on the tree path between ends of p as biconnected
   - Update V and compute lengths for newly allowed paths (C type augmentation)
   - Go to step 3
6. Else exit

• Runtime = O(N*D + K*N²), reduced to O(K*N²) w/o obstacles, where N = #allowed endpoints, K = #added paths, D = Dijkstra runtime
Outline

• Manufacturing defect models
• Previous work on routing for reliable manufacturing
• Motivation for non-tree routing
• Problem formulation and basic properties
• Exact solution by integer programming
• Greedy heuristic
• Experimental results and conclusions
Experimental Setup

• **Compared algorithms**
  – Greedy
  – Integer program solved with CPLEX 7.0
  – Best-drop E2AUG heuristic (Khuller-Raghavachari-Zhu 99)
    • Uses min-weight branching to select best path and multiple restarts
    • Modified to enforce WL budget
  – Recent E2AUG genetic algorithm (Raidl-Ljubic 2002)
    • Features compact edge-set representation + stochastic local improvement for solution space reduction

• **Test Cases**
  – WL increase budget = 1%, 2%, 5%, 10%, 20%, unbounded
  – Net size between 5 and 1000 terminals
    • Random nets routed using BOI heuristic
    • Min-area and timing driven nets extracted from real designs
  – No routing obstacles
### Extra wirelength (%) and runtime (sec.) for Unlimited WL Increase Budget

<table>
<thead>
<tr>
<th>#sinks</th>
<th>CPLEX</th>
<th>Genetic</th>
<th>Best-Drop</th>
<th>Greedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>77.86/0.01</td>
<td>77.86/0.01</td>
<td>77.90/0.26</td>
<td>78.33/0.00</td>
</tr>
<tr>
<td>10</td>
<td>57.97/0.01</td>
<td>57.97/0.52</td>
<td>58.19/7.54</td>
<td>59.70/0.00</td>
</tr>
<tr>
<td>20</td>
<td>45.58/0.07</td>
<td>45.59/4.33</td>
<td>45.77/226.71</td>
<td>46.82/0.03</td>
</tr>
<tr>
<td>50</td>
<td>38.64/0.88</td>
<td>39.44/37.35</td>
<td>--</td>
<td>40.16/0.41</td>
</tr>
<tr>
<td>100</td>
<td>35.11/18.21</td>
<td>39.30/181.08</td>
<td>--</td>
<td>36.28/3.26</td>
</tr>
<tr>
<td>200</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>34.10/26.12</td>
</tr>
<tr>
<td>500</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>32.85/405.45</td>
</tr>
<tr>
<td>1000</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>32.27/3160.24</td>
</tr>
</tbody>
</table>

- CPLEX finds optimum (least) wirelength increase with practical runtime for up to 100 sinks
- Greedy always within 3.5% of optimum; runtime practical for up to 1000 sinks
Biconnectivity (%) and runtime (sec.) for 10% WL Increase

<table>
<thead>
<tr>
<th>#sinks</th>
<th>Greedy B</th>
<th>Greedy C</th>
<th>Best-Drop B</th>
<th>Best-Drop C</th>
<th>CPLEX B</th>
<th>CPLEX C</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>11.30/0.00</td>
<td>12.24/0.00</td>
<td>9.89/0.20</td>
<td>10.44/0.41</td>
<td>11.30/0.00</td>
<td>12.30/0.00</td>
</tr>
<tr>
<td>10</td>
<td>14.41/0.00</td>
<td>17.95/0.00</td>
<td>10.84/7.11</td>
<td>12.04/61.73</td>
<td>14.46/0.01</td>
<td>18.11/0.02</td>
</tr>
<tr>
<td>20</td>
<td>30.53/0.01</td>
<td>35.56/0.06</td>
<td>23.68/173.12</td>
<td>28.25/5516.75</td>
<td>31.15/0.04</td>
<td>36.32/0.33</td>
</tr>
<tr>
<td>50</td>
<td>56.95/0.11</td>
<td>58.58/1.51</td>
<td>--</td>
<td>--</td>
<td>58.04/0.58</td>
<td>--</td>
</tr>
<tr>
<td>100</td>
<td>65.87/0.66</td>
<td>66.76/16.69</td>
<td>--</td>
<td>--</td>
<td>67.00/5.81</td>
<td>--</td>
</tr>
<tr>
<td>200</td>
<td>71.33/4.28</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>500</td>
<td>73.56/56.77</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>1000</td>
<td>74.49/420.91</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

- Augmenting paths of type C (node projections as endpoints) give extra 1-5% biconnectivity
- Biconnectivity grows with net size
- Greedy within 1-2% of optimum (max) biconnectivity computed by CPLEX
Biconnectivity-Wirelength Tradeoff

- 20-terminal nets
- 68% biconnectivity with 20% WL increase
Max SPICE Delay (ns) Improvement

- 52-56 terminal nets, routed for min-area
- 28.26% average and 62.15% maximum improvement in max-delay for 20% WL increase
- Smaller improvements for timing driven initial routings

<table>
<thead>
<tr>
<th>Testcase</th>
<th>Initial Routing (WarpRoute)</th>
<th>WL Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1%</td>
</tr>
<tr>
<td>1</td>
<td>1.551</td>
<td>1.564</td>
</tr>
<tr>
<td>2</td>
<td>0.366</td>
<td>0.374</td>
</tr>
<tr>
<td>3</td>
<td>0.859</td>
<td>0.869</td>
</tr>
<tr>
<td>4</td>
<td>0.282</td>
<td>0.282</td>
</tr>
<tr>
<td>5</td>
<td>1.002</td>
<td>1.002</td>
</tr>
<tr>
<td>6</td>
<td>0.787</td>
<td>0.794</td>
</tr>
<tr>
<td>7</td>
<td>0.514</td>
<td>0.514</td>
</tr>
</tbody>
</table>
Process Variability Robustness

- Width \( w = w_0 \), \( w_0 \pm 6.67\% \)
- Delay variation computed as \( \frac{\max_w d(w) - \min_w d(w)}{d(w_0)} \)
- 13.79\% average and 28.86\% maximum reduction in delay variation for 20\% WL increase

<table>
<thead>
<tr>
<th>Testcase</th>
<th>Initial Routing (WarpRoute)</th>
<th>20% WL Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.551( \pm 4.126% )</td>
<td>0.873( \pm 3.780% )</td>
</tr>
<tr>
<td>2</td>
<td>0.366( \pm 3.552% )</td>
<td>0.345( \pm 2.898% )</td>
</tr>
<tr>
<td>3</td>
<td>0.859( \pm 3.958% )</td>
<td>0.627( \pm 3.668% )</td>
</tr>
<tr>
<td>4</td>
<td>0.282( \pm 2.837% )</td>
<td>0.262( \pm 3.053% )</td>
</tr>
<tr>
<td>5</td>
<td>1.002( \pm 3.792% )</td>
<td>0.778( \pm 3.470% )</td>
</tr>
<tr>
<td>6</td>
<td>0.787( \pm 3.812% )</td>
<td>0.442( \pm 3.167% )</td>
</tr>
<tr>
<td>7</td>
<td>0.514( \pm 3.502% )</td>
<td>0.273( \pm 2.930% )</td>
</tr>
</tbody>
</table>
Conclusions

• **Summary**
  - Post-processing tree augmentation approach to manufacturing yield improvement
    - Easy to integrate in current flows
    - Appropriate for large non-critical nets
  - Compact integer program, practical up to 100 terminals
  - Faster, near-optimal greedy heuristic
  - Results show significant biconnectivity increase with small increase in wirelength, especially for large nets

• **Ongoing work**
  - Multiple net augmentation
  - Simultaneous tree augmentation and decompaction
  - Reliability with timing constraints
Thank You for Your Attention!

Further details on our work are available on the group’s website http://vlsicad.ucsd.edu