ECE 6133
Implementation of “Efficient Algorithms for Channel Routing” - Yoshimura & Kuh

SHREEPAD A PANTH
Channel Routing Problem

- Route between a top row and bottom row, pins with the same number have to be connected.

- Using only two metal layers.

- Overall area of the channel (ie height of the channel) has to be minimized.
The Algorithm

- \( L = \{ \} \)
- For \( z_s \) to \( z_t \) do:
  - \( L = L + \{ \text{nets that terminate at zone } z \} \)
  - \( R = \{ \text{nets that begin at zone } z+1 \} \)
  - Merge \( L \) and \( R \) so as to minimize the increase in the longest path in VCG
  - \( L = L - \{ \text{merged nets in previous step} \} \)
The Algorithm (contd)

Merging is done heuristically

- Pair to be merged can be computed by knowing the longest source-node \((u)\) and sink-node\((d)\) paths for every node.

- By maximizing \(f(m)\) and minimizing \(g(n,m)\) the pair to be merged is obtained heuristically.
Implementation

- If no cycles are present in the VCG, route the given problem without inserting any doglegs.

- If VCG contains cycles, then break all nets into two terminal nets and perform channel routing.
Results – small circuit

No of nets = 10
Average netsize = 2
Max density = 5
No of tracks = 5
No of vias used = 22
Results – Larger Circuit

No of nets = 60
Average netsize = 4
Max density = 41
No of tracks = 42
No of vias used = 331
## Results

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Nets</th>
<th>Max Den</th>
<th>No doglegs</th>
<th>Doglegs</th>
<th>No Doglegs</th>
<th>Doglegs</th>
</tr>
</thead>
<tbody>
<tr>
<td>dr1</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>dr6</td>
<td>15</td>
<td>9</td>
<td></td>
<td>12</td>
<td></td>
<td>13</td>
</tr>
<tr>
<td>dr2</td>
<td>20</td>
<td>14</td>
<td></td>
<td>14</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>dr7</td>
<td>20</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>dr3</td>
<td>30</td>
<td>20</td>
<td></td>
<td>23</td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>dr8</td>
<td>40</td>
<td>22</td>
<td>23</td>
<td>23</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>dr4</td>
<td>60</td>
<td>41</td>
<td></td>
<td>46</td>
<td></td>
<td>42</td>
</tr>
</tbody>
</table>

### Without Doglegs

![Graph showing the number of tracks vs. number of nets without doglegs.](image)

### With Doglegs

![Graph showing the number of tracks vs. number of nets with doglegs.](image)
Reasons for discrepancies

- Too many doglegs worsen the solution (shown by dr7)
- With a lot of nets, merging of nets sometimes blocks the merging of other nets
- Suppose we merge nets a&d; b&e
- Net f cannot be merged with either net c or net g as cycle will be formed in VCG
- On the other hand if we merge a&d; c&e, net f can be merged with net b
Solutions

- Inserting Doglegs only where necessary to break cycles in VCG, however potential improvement may be lost.
- Algorithm #2 proposed by Yoshimura & Kuh based on bipartite graph and matching of nets.