OpenOrd: An Open-Source Toolbox for Large Graph Layout

S. Martin¹, W. M. Brown², R. Klavans³, and K. Boyack³,*

¹Sandia National Laboratories, Albuquerque, NM
²Oak Ridge National Laboratories, Oak Ridge, TN
³SciTech Strategies, Inc., Berwyn, PA

*Presenter
Motivation for OpenOrd Toolbox

• Force-directed layout doesn’t scale well to large graphs
  – Computational complexity $O(n^2)$
  – Poor global structure
  – Visually unappealing “hairball”

• Most existing algorithms don’t work well on “real-world” data, e.g. not mesh, not scale-free, etc.
Basic Force-Directed Graph Layout

• Force-directed layout optimizes

$$
\min_{x_1, \ldots, x_n} \sum_i \left( \sum_j (w_{ij} d(x_i, x_j)^2) + D_{xi} \right),
$$

where $x_i$ are positions of vertices, $w_{ij}$ are edge weights and $D_{xi}$ is the density of edges near $x_i$.

  – Large edge weights encourage vertices to group together.
  – High density is discouraged.

• OpenOrd is based on predecessor VxOrd and uses simulated annealing to solve this problem, with a five stage cooling schedule (liquid, expansion, cool-down, crunch, simmer).

• Density term $D_{xi}$ is costly, so we use a grid-based method to reduce computation from $O(n^2)$ to $O(n)$. 
Basic Force-Directed Graph Layout

- Movie showing simulated annealing schedule.
Edge-Cutting

• OpenOrd uses an edge-cutting heuristic in order to provide user control of amount of white space in layout.
  – Edges are cut if they are both long (in layout) and large weight.
• Edge-cutting allows trade-off between attractive $w_{ij}d(x_i,x_j)^2$ term and repulsive $Dx_i$ term in optimization.

- less ___________ edge-cutting ___________ more ➔

[Graph showing normalized energy vs. edge-cutting, with attraction and repulsion curves]
Parallel Layout

- OpenOrd can be run in parallel: each processor keeps track of all vertex positions, but is only responsible for moving a subset of vertices.
  - Maintains similar layout.
  - Increases effective memory of computer for truly enormous graphs.
  - Increases speed.
Multi-Level Layout

- Multi-level layout: cluster vertices, coarsen, repeat, layout, refine, repeat.

- OpenOrd uses multi-level layout to untangle global structure.
Average-Link Clustering

• When we coarsen the graph for multi-level layouts, we use average-link heirarchical clustering.
  – Clustering is based on distance in a layout with maximum edge-cutting.
  – A distance threshold is chosen based for forming clusters.
  – Distance threshold can be chosen manually or automatically using a normalized curve.
Parameter Testing

- OpenOrd has many parameters that must be chosen in order to function, such as edge-cutting and layout level.
- We identified good defaults using a variety of datasets as test cases.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Nodes</th>
<th>Edges</th>
<th>Level</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yeast</td>
<td>6,147</td>
<td>61,646</td>
<td>3</td>
<td>1:29</td>
</tr>
<tr>
<td>Journals</td>
<td>8,712</td>
<td>98,705</td>
<td>3</td>
<td>2:13</td>
</tr>
<tr>
<td>Swiss Roll</td>
<td>20,000</td>
<td>400,000</td>
<td>9</td>
<td>4:01</td>
</tr>
<tr>
<td>Solid State Lighting</td>
<td>32,776</td>
<td>222,626</td>
<td>4</td>
<td>7:16</td>
</tr>
<tr>
<td>Quarter Year ISI 2003</td>
<td>218,716</td>
<td>1,821,976</td>
<td>5</td>
<td>1:09:36</td>
</tr>
<tr>
<td>Wikipedia</td>
<td>659,388</td>
<td>16,582,426</td>
<td>6</td>
<td>3:39:23</td>
</tr>
<tr>
<td>Full Year ISI 2004</td>
<td>849,888</td>
<td>5,843,729</td>
<td>7</td>
<td>3:40:23</td>
</tr>
</tbody>
</table>
Some Real-World Examples

• Map of Last.fm music database by Tamas Nepusz (Royal Holloway University)

• Netflix Movie Database by Todd Holloway (Indiana University)

• Maps of Science (lots of variations) by R. Klavans and K. Boyack
Conclusions

• OpenOrd combines many of the state-of-the-art ideas in large-scale graph layout in an easy-to-use open source package.
  – Edge-cutting (novel to OpenOrd and predecessor VxOrd)
  – Parallel operation.
  – “Smart” (using average-link clustering) multi-level graph partitioning.

• OpenOrd (and predecessor VxOrd) has been used successfully in a number of applications.
  – Scientific literature analysis.
  – Bioinformatics applications.
  – Music databases.
  – Movie databases.

• OpenOrd is available at www.cs.sandia.gov/~smartin.