Spectral Graph Drawing

Rodrigo Arias

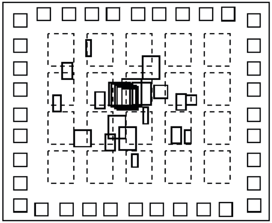
January 21, 2018

Outline

- Background
- Motivation
- Ideas from the paper
- Methods
- Examples
- Conclusions

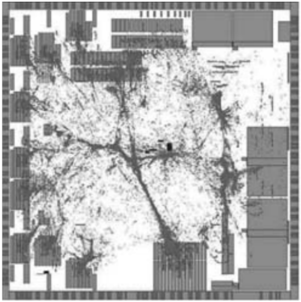
Background

The placement problem.



- Analytic solution: Minimize an **objective function** (wire, area...) via mathematical analysis.
- Global placement must be legalized afterwards (detailed placement).

Motivation



- Cells and nets can be modeled as an undirected graph
- Spectral methods can be used to draw complex graphs

Ideas from the paper

In the Koren paper [1] two methods to find the placement of the nodes are described, using the following properties.

- Method 1: The Laplacian matrix L
- Method 2: The normalized Laplacian matrix $\mathcal{L}$

The methods minimize the total **wire-length**

Method 1: Laplacian matrix L

- A graph can be represented by the adjacency matrix A, were

$$a_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are connected} \\ 0 & \text{otherwise} \end{cases}$$

- Also, the degree matrix D has the degree of each node in the diagonal, $d_{ii} = deg(i)$.
- Finally the Laplacian matrix is defined as $L = D - A$

Method 1: Laplacian matrix L

- The problem is formulated as the minimization of the square wire-length

$$\begin{aligned} &\min_x E(x) = \sum (x(i) - x(j))^2 \\ &\text{s.t. } \mathrm{Var}(x) = 1 \end{aligned} \tag{1}$$

- The eigenvalues $\lambda_1 \leq \lambda_2 \cdots \lambda_m$ and eigenvectors $v_1, v_2, \cdots v_m$ of L are computed.
- The positions of the nodes are given by the coordinates of $x = v_2$ and $y = v_3$.

Method 2: Normalized Laplacian matrix

- Let d_i be the degree of the node i, we define the normalized Laplacian matrix $\mathcal{L}$ as:

$$\mathcal{L}(i,j) = \begin{cases} 1 & \text{if } i = j \text{, and } d_i \neq 0 \\ -\frac{1}{\sqrt{d_i d_j}} & \text{if } i \text{ and } j \text{ are connected} \\ 0 & \text{otherwise} \end{cases}$$

- We give more weight to the nodes with higher degree, in order to cluster neighbours around centric nodes.

Method 2: Normalized Laplacian matrix

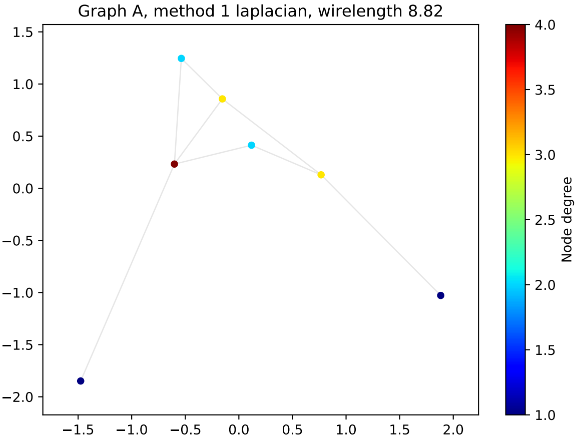
- The solution is computed as before, but now using the matrix $\mathcal{L}$.
- The eigenvalues $\lambda_1 \leq \lambda_2 \cdots \lambda_m$ and eigenvectors $v_1, v_2, \cdots v_m$ of $\mathcal{L}$ are computed.
- The positions of the nodes are given by the coordinates of $x = v_2$ and $y = v_3$.

Examples

In order to test the methods, some graphs were generated

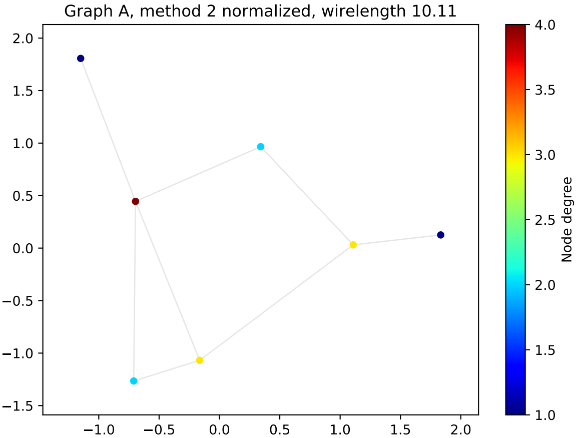
- Small graph crafted by hand with 7 nodes.
- Erdös-Rényi graph with 100 nodes
- A FPGA network from a real Verilog TX module with 345 nodes

Small graph



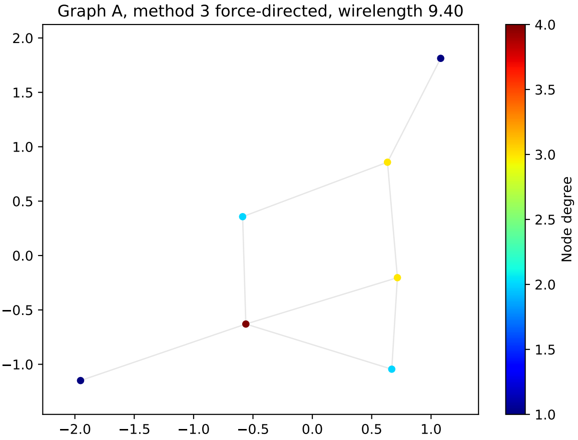
Using the method 1, the Laplacian matrix L.
Wire-length: 8.82

Small graph



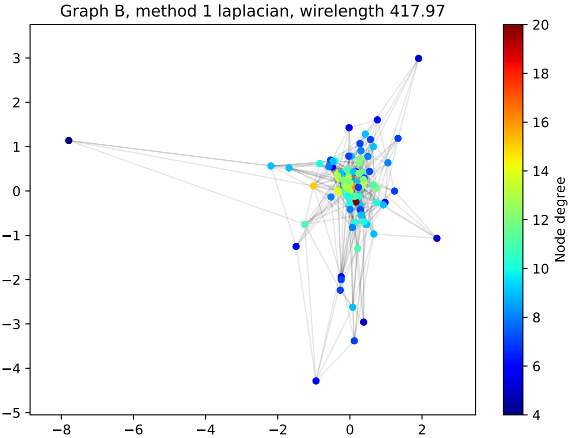
Using the method 2, the normalized Laplacian matrix $\mathcal{L}$.
Wire-length: 12.87

Small graph



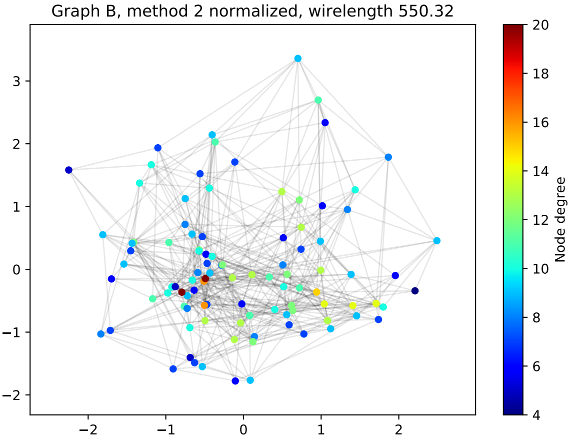
Using the directed force method (for comparison).
Wire-length: 9.40

ER graph



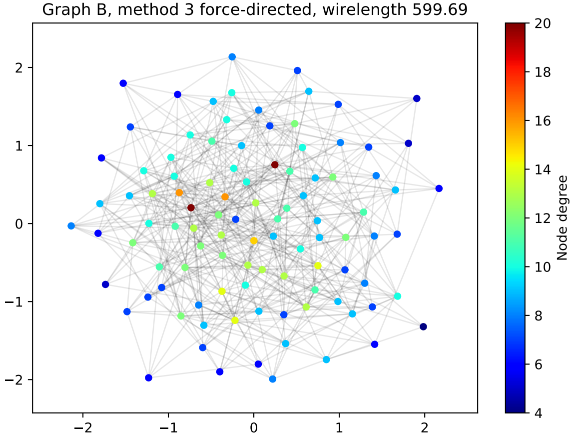
Using the method 1, the Laplacian matrix L.
Wire-length: 417.97

ER graph



Using the method 2, the normalized Laplacian matrix $\mathcal{L}$.
Wire-length: 722.86

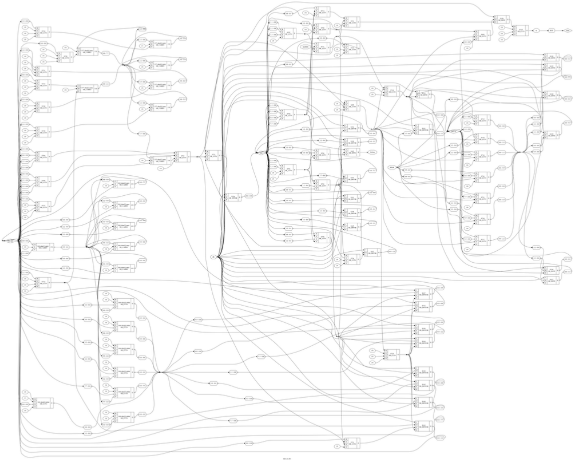
ER graph



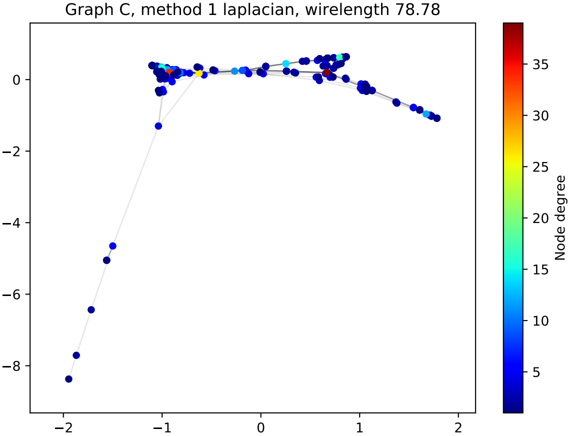
Using the spring method (for comparison).
Wire-length: 599.69

TX graph

Generated using Yosys [2] from the Verilog file `uart_trx.v`.

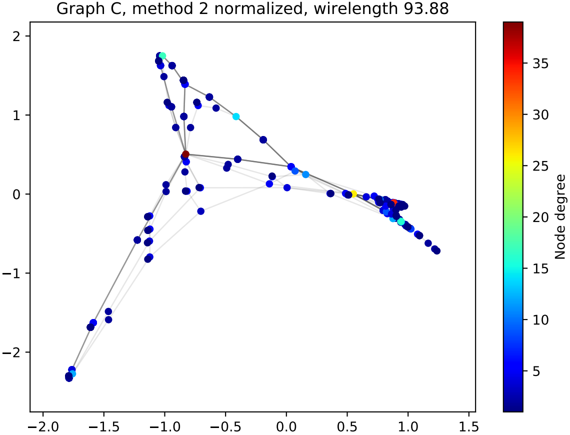


TX graph



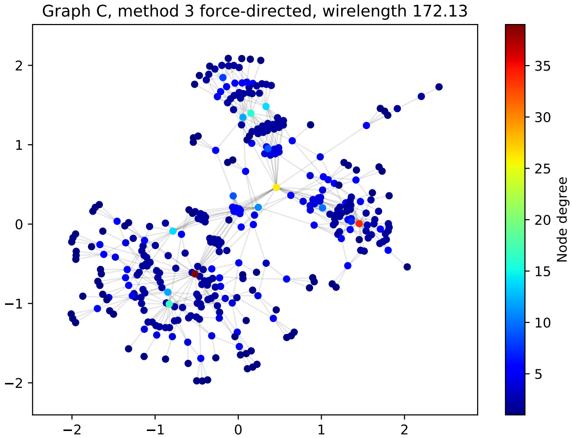
Using the method 1, the Laplacian matrix L.
Wire-length: 78.78

TX graph



Using the method 2, the normalized Laplacian matrix $\mathcal{L}$.
Wire-length: 733.78

TX graph



Using the spring method (for comparison).
Wire-length: 172.13

Observations

- The proposed model doesn't take into account the size of the cells
- Multiple cells can overlap in the same point
- The behavior is not close to reality

Conclusions

- The spectral methods can be used in the global placement.
- Don't produce great results in comparison with spring layout.
- The underlying working principle is complex to understand.
- May be suitable for larger graphs, as the matrices are usually very sparse.

References

- Yehuda Koren *On Spectral Graph Drawing* Proceedings of the 9th Annual International Conference on Computing and Combinatorics, p496–508, 2003.
- Clifford Wolf, Johann Glaser. *Yosys - A Free Verilog Synthesis Suite.* In Proceedings of Austrochip 2013