HPWL_AP_2000

Wednesday, July 17, 2024 5:22 PM

Analytical Minimization of Half-Perimeter Wirelength

Andrew Kennings and Igor Markov*
Dept. Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, N2L 3G1
UCLA Computer Science., Los Angeles, CA 90095-1596
akenning@odysseus.uwaterloo.ca, imarkov@cs.ucla.edu

2) ANALYTICAL PLACEMENT

Hypergraphs : $G_H(V_H, E_H)$

Vertices : $V_H = \{v_1, v_2, \cdots, v_n\}$

- Vertices correspond to **modules** (or transistors, gates, nodes, etc.).
- Vertex **weights** correspond to module **areas**.
- Vertices are either fixed or free (graph can be static or dynamic).

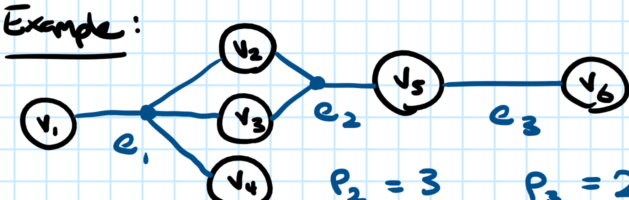
Hyperedges : $E_H = \{e_1, e_2, \cdots, e_n\}$

- Hyperedges correspond to **signal nets** (nets have one source, one or many sinks).
- Hyperedge weights correspond to criticalities and/or multiplicities.

$e_k \in E$ connects to $p_k \geq 2$ vertices $\longrightarrow$ p_k : "vertex degree"

$v_i \in V$ incident to $d_i \geq 0$ hyperedges $\longrightarrow$ d_i : "hyperedge degree"

Example :



$p_1 = 4$
$d_1 = 1$

$p_2 = 3$
$d_2 = 2$

$p_3 = 2$
$d_3 = 1$

e_2 is incident to e_1 and e_3

$G_H(V_H, E_H)$

$V_H = \{v_1, v_2, v_3, v_4, v_5, v_6\}$

$E_H = \{e_1, e_2, e_3\}$

e_1 : $p_1 = 4$, $d_1 = 1$

e_2 : $p_2 = 3$, $d_2 = 2$

e_3 : $p_2 = 2$, $d_3 = 1$

v_i has d_i "pins" and e_k has p_k "pins" for a total of :

$$P = \sum_{k=1}^{m} d_k = \sum_{i=1}^{n} p_i \text{ pins}$$

Module placements (positions) in x and y directions :

Placement vectors x and y :

$x = (x_1, x_2, \cdots, x_n)$

- Circuits are mostly just graphs on a 2-dimensional plane (components on a PCB, transistors on silicon die, CLBs on an FPGA, etc.)

Placement Vectors x and y:

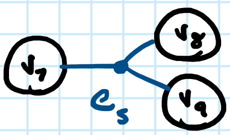
$$x = (x_1, x_2, \cdots, x_n)$$
$$y = (y_1, y_2, \cdots, y_n)$$

- Circuits are mostly just graphs on a 2-dimensional plane (components on a PCB, transistors on silicon die, CLBs on an FPGA, etc.).
- In VLSI, can be 3-dimensional, or "2.5-dimensional"
- x_5 and y_5 are x-y coordinates of module 5.

↳ Graph G_H has placement vectors x and y.
(x_1, y_1) represents placement (position) of module 1

2.1) Hypergraph Placement

C_k : index set of hypergraph vertices incident to net $e_k \in E_H$

Example: 

$C_5 = \{7, 8, 9\}$

x-Direction of HPWL: (Linear Distance)

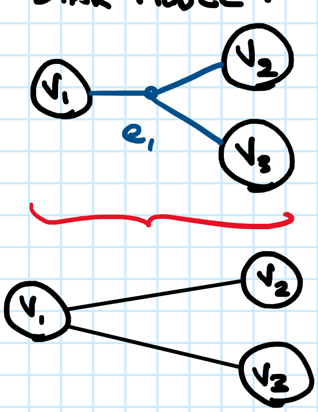
$$HPWL_x(x) = \sum_{e_n \in E_H} \max_{i,j \in C_k} \underbrace{|x_i - x_j|}_{\text{Manhattan / Linear Distance}}$$

- HPWL is a convex function of x since $|x_i - x_j|$ is convex for all i,j
    - However, Linear HPWL is **NOT strictly convex** and most often has uncountably many minimizers.
    - Max function is **non-differentiable**, so cannot use smooth methods like Newton's method.
    - HPWL requires **fixed vertices** with at least two different locations.
        - Trivial solution is to place all modules at precisely the same position for wirelength of zero, so HPWL also requires at least two uniquely positioned vertices.
        - Fixed vertices represent I/O pads or external pins in VLSI or FPGA context.

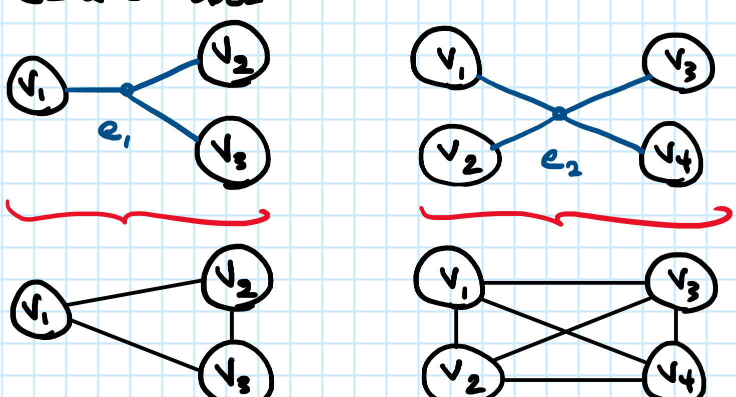
2.2) Reduction To Graphs

In a regular graph, edges are incident to only 2 vertices.

Star Model:



Clique Model:



Source module as central vertex.

All vertices incident to hyperedge become fully connected.

Source module as central vertex.

All vertices incident to hyperedge become fully connected. Quadratic growth of edges.

- There exists a Euclidean / Square Distance HPWL:

$$\min_x \left\{ \sum_{i>j} a_{ij} (x_i - x_j)^2 \; : \; Hx = b \right\}$$

  - The **Square** HPWL is **strictly convex**, so it yields a **unique minimizer**. Can use smooth methods.
  - However, the **Squared** HPWL tends to produce **lower-quality** results.

- Linear HPWL minimization problem:

$$\min_x \left\{ \sum_{i>j} a_{ij} |x_i - x_j| \; : \; Hx = b \right\}$$

a_{ij} : vector of weights that influence the importance of distance between pairs of vertices over others.

$Hx = b$: set of linear constraints

- GORDIAN-L Relaxation: $\min_x \left\{ \sum_{i>j} \dfrac{a_{ij}}{|x_i^{v-1} - x_j^{v-1}|} (x_i^v - x_j^v)^2 \; : \; Hx = b \right\}$

↳ Iterated quadratic minimizations

v : algorithm iteration index, NOT power!

- Regularization Relaxation: $\min_x \left\{ \sum_{i>j} a_{ij} \sqrt{(x_i - x_j)^2 + \beta} \; : \; Hx = b \right\}$

↳ 2 solution methods:

1) Linearly-convergent fixed-point method.
2) Primal-Dual Newton method w/ quadratic convergence.