Lecture 13: Placement – III

Tsung-Wei (TW) Huang
Department of Electrical and Computer Engineering
University of Utah, Salt Lake City, UT

Programming Assignment #2

- **Deadline is extended to 11/17 23:59 PM**
  - https://github.com/tsung-wei-huang/ece5960-physical-design/issues/3
- **Checkpoint due on every Wed until 11/17**
  - https://github.com/tsung-wei-huang/ece5960-physical-design/issues/4
- **Implementation details**
  - Slicing tree: parenthesis checking algorithm
  - Sequence pair: shortest path implementation
  - Simulated annealing-based optimization

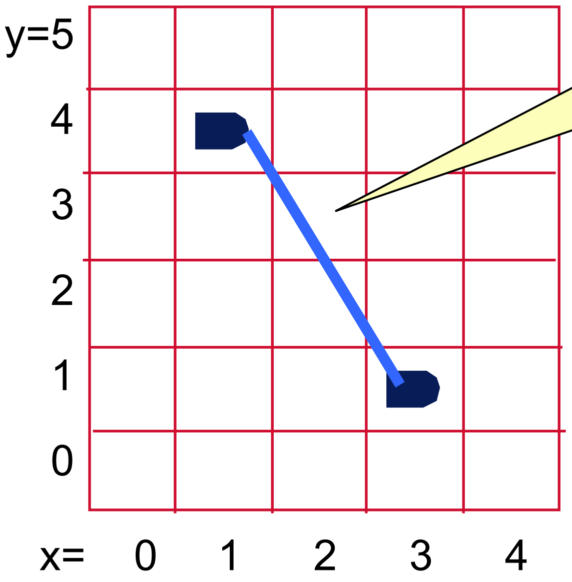
Recap: Analytical Placer

- **Write an *equation* whose *minimum* is the placement**
  - If you have a million gates, need a million **(xi, yi)** values as result
  - Formulate an appropriate **cost function** for all the gate-level **(xi, yi)**:
    $\mathbf{F}(x_1, x_2, \ldots x_{1M}, y_1, y_2, \ldots y_{1M})$
  - Solve analytically for $X^*=(x_1, x_2, \ldots x_{1M}), Y^*=(y_1, y_2, \ldots y_{1M})$ to minimize **F( )**
  - The resulting values of **X*, Y*** give you the placement of all 1M gates
- **This sounds sort of crazy… but it works great**
  - All modern placers for big ASICs and SOCs are "analytical"
  - Big trick is write the wirelength in mathematically "friendly" form we can optimize

Recap: Quadratic Wirelength Model

- We optimize squared length of "distance" line between points:
  $(x1-x2)^2 + (y1-y2)^2$

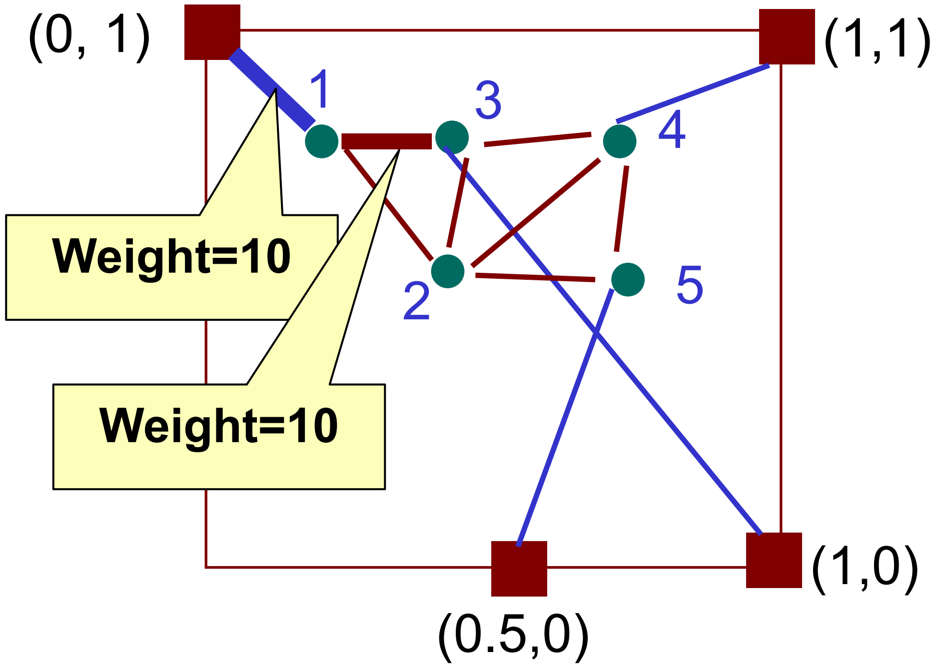
A "2-point net"



Quadratic wirelength
$=(3-1)^2 + (4-1)^2$
$=13$

For k-point net, we use the clique model (complete graph for each net)

Recap: Quadratic Placement Formulation



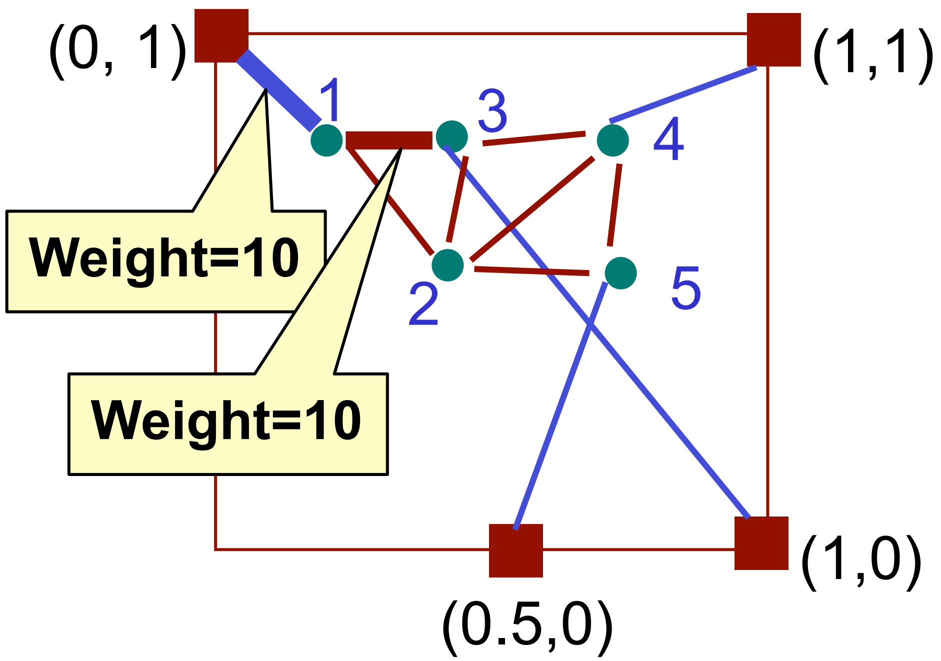
All wire weights = 1 *except* two highlighted:
gate1 to **pad** and **gate1** to **gate2**

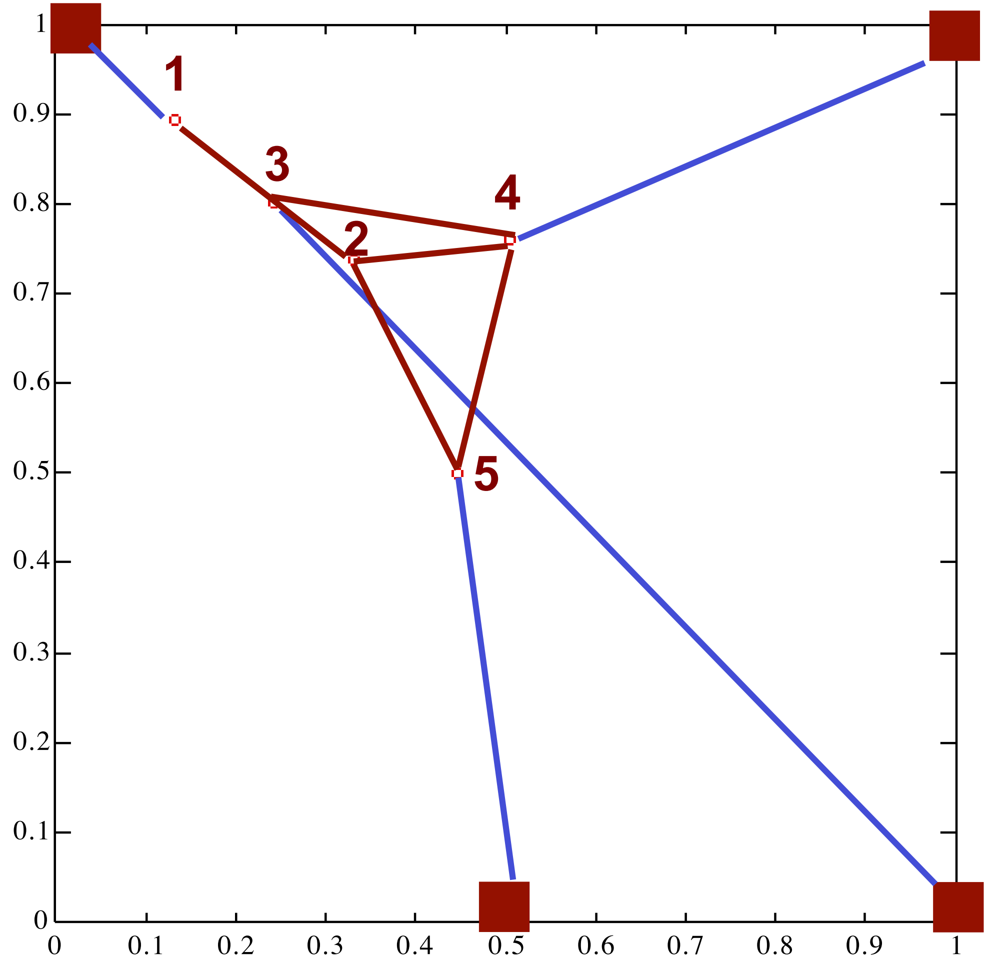
$$\mathbf{C} = \begin{pmatrix} 0 & 1 & 10 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 10 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \qquad \mathbf{A} = \begin{pmatrix} 21 & -1 & -10 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ -10 & -1 & 13 & -1 & 0 \\ 0 & -1 & -1 & 4 & -1 \\ 0 & -1 & 0 & -1 & 3 \end{pmatrix}$$

$$\mathbf{b_x} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0.5 \end{pmatrix} \qquad \mathbf{b_y} = \begin{pmatrix} 10 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Recap: Quadratic Placement Result

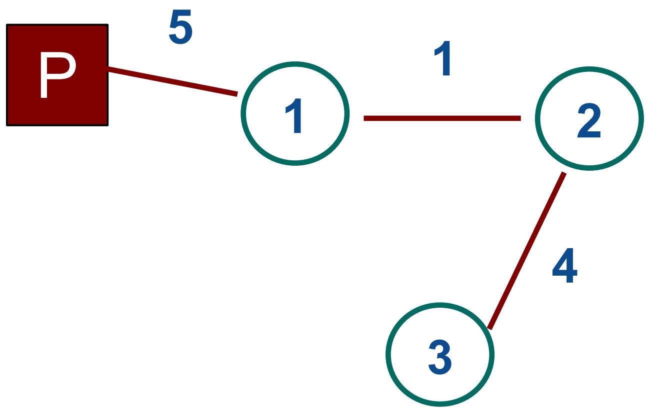
Placement → Solving a linear system





Recap: What is Matrix A?

- **Surprisingly simple recipe to build the required A matrix**
  - First, build the **NxN** connectivity matrix, called **C**
  - If gate **i** has a 2-point wire to gate **j** with weight **w**, **c[i,j] = w**, else = **0**
- **Another example of 3 gates, 3 wires, and 1 I/O pad (P)**

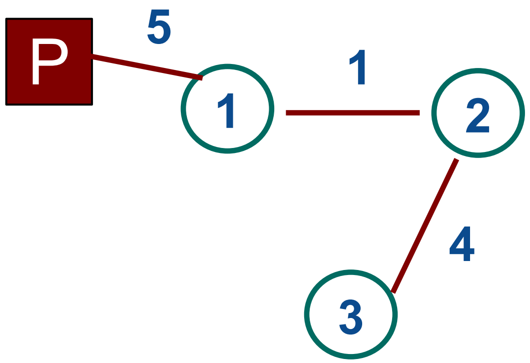


$$C = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 4 \\ 0 & 4 & 0 \end{bmatrix}$$

Note: **C** matrix ignores the pads

Recap: What is Matrix A? (cont'd)

- **Use the connectivity C matrix to build A matrix**
  - Elements **a[i,j]** not on the matrix diagonal are just **a[i,j] = -c[i,j]**
  - Elements on the diagonal are **a[i,j]=** $\sum_{j=1,n}$ **c[i,j] + (weight of any pad wire)**
    - …ie, add up the **i**th row of **C** and then add in weight on a (possible) wire to pad

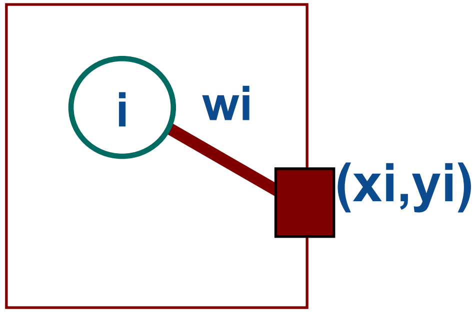


$$C = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 4 \\ 0 & 4 & 0 \end{bmatrix} \qquad A = \begin{bmatrix} 6 & -1 & 0 \\ -1 & 5 & -4 \\ 0 & -4 & 4 \end{bmatrix}$$

diagonal

Note: **A** matrix accounts for pads

Recap: What is Vector b?



- For **Ax = b_x** vector:
  - If gate **i** connects to a pad at **(xi, yi)** with a wire with weight **wi**
  - Then set **b_x[i] = wi • xi**

$$\begin{bmatrix} A \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} b_x \end{bmatrix}$$

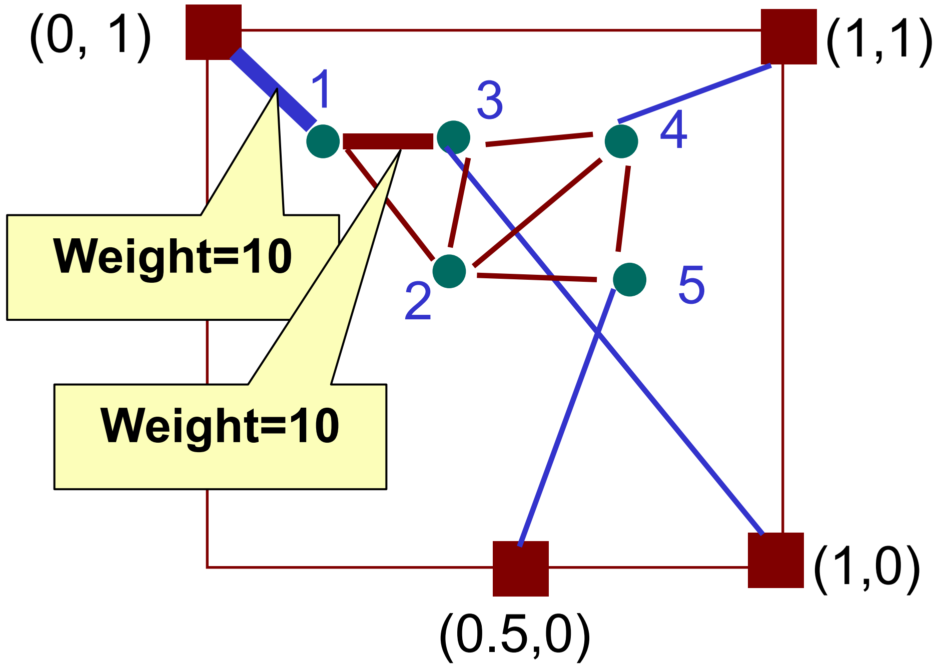
i^{th} element of b_x vector

- For **Ay = b_y** vector:
  - If gate **i** connects to a pad at **(xi, yi)** with a wire with weight **wi**
  - Then set **b_y[i] = wi • yi**

$$\begin{bmatrix} A \end{bmatrix} \begin{bmatrix} y \end{bmatrix} = \begin{bmatrix} b_y \end{bmatrix}$$

i^{th} element of b_y vector

Recap: Revisited



All wire weights = 1 *except* two highlighted:
gate1 to **pad** and **gate1** to **gate2**

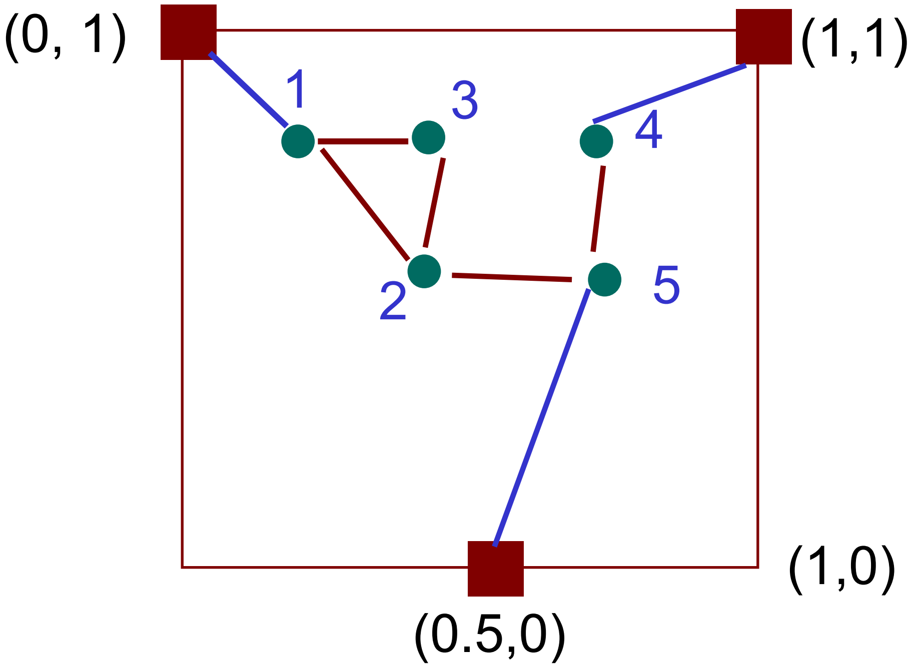
$$\mathbf{C} = \begin{pmatrix} 0 & 1 & 10 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 10 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \qquad \mathbf{A} = \begin{pmatrix} 21 & -1 & -10 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ -10 & -1 & 13 & -1 & 0 \\ 0 & -1 & -1 & 4 & -1 \\ 0 & -1 & 0 & -1 & 3 \end{pmatrix}$$

$$\mathbf{b_x} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0.5 \end{pmatrix} \qquad \mathbf{b_y} = \begin{pmatrix} 10 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Recap: How to Solve Ax=b?

- **Are these *difficult* to do, in practice?**
  - If we have 1M gates, this is a 1M x 1M **A** matrix, with 1M element **x** and **b** vectors!
- **No – these are VERY EASY to solve, even when very large**
  - The **A** matrix has a special form—*it is sparse, symmetric, diagonally dominant*
  - Mathematically: **A** is positive semi-definite—*very simple to solve!*
  - We use iterative, approximate solvers, in practice (i.e., not Gaussian elimination but techniques like conjugate gradient)
    - This means the solver converges gradually to the right answer
    - But, also means that the answers can be a little bit “off”, not quite perfect

Practice #1: What are C, A, bx, and by?



All wire weights = 1

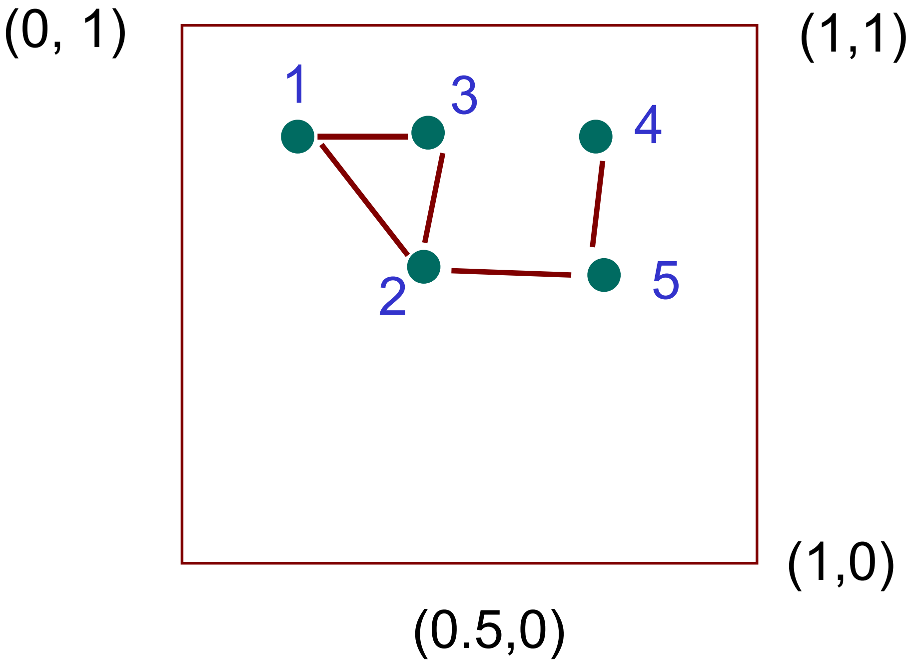
C =

A =

$\mathbf{b_x}$ =

$\mathbf{b_y}$ =

Practice #2: What are C, A, bx, and by?



All wire weights = 1

$\mathbf{C}$ =

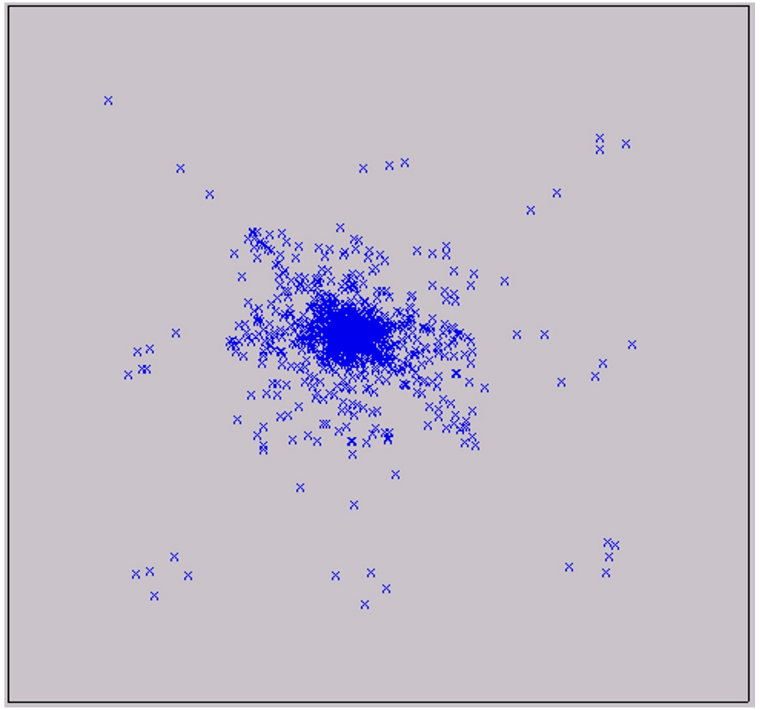
$\mathbf{A}$ =

$\mathbf{b_x}$ =

$\mathbf{b_y}$ =

Takeaway

- **When no pads exists**
  - **b** vectors become zero
  - Solutions of **x** and **y** degenerate to all zero
    - trivial solutions
  - All points lump together – many overlaps
- **New problem**
  - Quadratic model minimizes wirelength for big netlists, in a numerical way
  - But ignores that gates have physical size, cannot be on top of each other
  - Now, we have to fix this



Artificial Pads for b Vectors

- https://github.com/limbo018/DREAMPlace