2020 - 2

2. (10 points) Given a circuit C, a weighted graph G is obtained from C by modeling each k-pin net of C as a clique (i.e., a complete graph) on the k vertices and assigning a weight of $1/(k\text{-}1)$ to each edge. The cut size of a two-way partitioning of G (C, respectively) is defined to be the sum of the weights of all cut edges (the number of all cut nets, respectively). Prove that if each net in C has at most 3 pins, an optimal balanced two-way partitioning of G corresponds to an optimal balanced two-way partitioning of C.

show balance

2. graph G : each k-pin net as a clique

-2

weight = $\frac{1}{k-1}$ to each edge

在 $k \le 3$，轉成 graph G，每個 net 共有不超过 $\binom{3}{2} = 3$ 條 2-pin nets

並且延用 Kernighan-Lin (KL) Algorithm

因 KL 也是 reducing a vertex with weight w(v) into a clique

我們假設此 G 的 vertex 皆是 unit vertex weight.

因此 if each net in C has at most 3 pins,

an optimal balanced two-way partitioning of G corresponds

to an optimal balanced two-way partitioning of C. #

2020 - 5

5. (5 points) Prove that half-perimeter wirelength (HPWL) is a lower bound of rectilinear Steiner minimal tree (RSMT) wirelength.

在一個包含所有pin的Bounding Box中，欲建一個RSMT連接所有pin. 最理想的情況就是：任一x or y座標，最多只會被此RSMT中的一線段經過，在此情況下將RSMT上水平及垂直方向的所有線段加總，可分別得 (Max of x − Min of x) 和 (Max of y − Min of y)，兩者相加即為HPWL

∴ HPWL 為 RSMT WL 的 lower bound #

2019 -5

5. (10 points) Given a slicing floorplan tree where the cut direction of each internal node is undecided yet, describe how to extend Stockmeyer algorithm to simultaneously determine the cut direction for each internal node and the shape for each module such that the total area of the floorplan is minimized. You may assume each module has at most two possible shapes (the ones with and without rotation). Note that the resultant slicing tree could be skewed or non-skewed. The time complexity of your method could be exponential, but it must be as low as possible. You should also analyze the time complexity of your algorithm.

同時 $\frac{1}{2}$ exp

5. 假設未知的 expression 像 12N3N45N6……N，N ∈ {V,H}，但目前未知。

想法：像 original 的 stockmeyer 用 stack 去做運算（讀到 N 就 pop 2個 operand 出來運算，反之則將 operand push in。當 operand 為 a,b，operator 為 N，做以下動作：

A：如果 a,b 皆為 cell（非 module），做①~③。

① 將 a（有無 rotate），b（有無 rotate），N（V or H）做所有可能的排列，會有 2×2×2 種情形

Not Stockmeyer −2

② 去除掉其結果之長寬皆大於某一 other 結果的 module

③ 儲存剩餘可能的排列情形，並將 module push 回

B：如果 a,b 有 module 存在，一般性假設 a 是 module，b 是 cell（當然可以皆為 module）。做④~⑥

④ 將 a（所有可能的排列），b（有無 rotate），N（V or H）做所有可能的排列，共 $n_a \times 2 \times 2$（n_a 為 a module 未被排除的可能排列數）。

（若 a,b 皆為 module，最差情況下有 $n_a \times n_b \times 2$ 種可能）。

⑤ 同②

⑥ 同③

C：直到 stack 為空，則以最後 module 之最小 area 的情況為解，再往前推出所有子 module 的 operator 為 H or V 及 cell 的 rotate 情形。

−2

☆ time complexity：n is # of modules，d is the depth of tree，因 shape 為 constant，考慮 H 及 V 個數為 (n-1)，所以 time complexity 應為 $O(n^2 d)$

2019 - 8

8. (10 points) In class we claimed that the BoundingBox net model can accurately model HPWL in a quadratic placement framework. Prove this claim for the x direction.

(25/15)

If $l_{\{i,j\}}$ is set to $|x_i - x_j|$ for all $\{i,j\} \in N$, then:

$$
\begin{aligned}
\tilde{L}^{BB} &= \frac{1}{2}\left(\sum_{\{i,j\}\in N} \frac{2}{k-1} \times \frac{1}{|x_i - x_j|} \times (x_i - x_j)^2\right) \\
&= \frac{1}{k-1}\left(\sum_{\{i,j\}\in N} |x_i - x_j|\right) \\
&= \frac{1}{k-1}\left(|x_1 - x_k| + \sum_{2\le i\le k-1} (|x_1 - x_i| + |x_i - x_k|)\right) \\
&= \frac{1}{k-1}(|x_1 - x_k| + (k-2) \times |x_1 - x_k|) \\
&= |x_1 - x_k|
\end{aligned}
$$

2019 - 10

10. (10 points) For a net with all its pins located on a two-dimensional plane, a single vertical-trunk Steiner tree for the net consists of a signal vertical trunk (i.e., a vertical line segment) and all the pins of the net are connected to the trunk by horizontal line segments. See the following figure for an example. Note that it is possible for a single vertical-trunk Steiner tree to have the trunk or a horizontal line segment degenerate as a point. Also note that there are many possible single vertical-trunk Steiner trees for a net. Given a k-pin net with the pin coordinates, (x_i, y_i), $1 \leq i \leq k$, in a plane, develop an algorithm for constructing a single vertical-trunk Steiner tree that has the minimum wirelength among all possible single vertical-trunk Steiner trees. Your algorithm should be as efficient as possible. You should also analyze the time complexity of your algorithm.

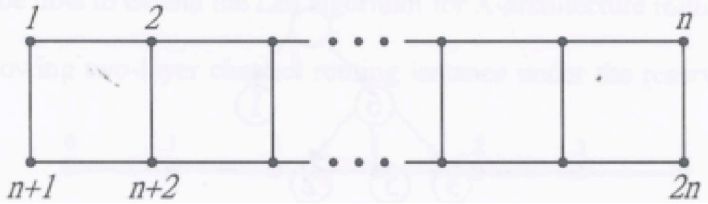
Single-trunk Steiner trees[edit]

The single-trunk Steiner tree is a tree that consists of a single horizontal segment and some vertical segments. A minimum single-trunk Steiner tree problem (MSTST) may be found in linear time.

The idea is that STSTs for a given point set essentially have only one "degree of freedom", which is the position of the horizontal trunk. Further, it is easy to see that if the Y-axis is split into segments by Y-coordinates of input points, then the length of a STST is constant within any such segment. Finally, it will be minimal if the trunk has the closest possible numbers of points below and above it. Therefore, an optimal position of the trunk is defined by a median of the set of Y-coordinates of the points, which may be found in linear time. Once the trunk is found, the vertical segments may be easily computed. Notice however that while the construction of the connecting net takes linear time, the construction of the tree which involves both input points and Steiner points as its vertices will require O(n log n) time, since it essentially accomplishes sorting of the X-coordinates of the input points (along the split of the trunk into the edges of the tree).

2017 - 2

(5 points) Consider the following ladder graph with $2n$ vertices, and an initial bipartition $A=\{1, 2, \ldots, n\}$ and $B=\{n+1, n+2, \ldots, 2n\}$. What is the resulting bipartition if one pass of the KL algorithm is applied?



We shall prove that there does not exist $u \in A$ and $v \in B$ as determined by the KL algorithm such that the gain of exchanging u and v is greater than 0. Consider two sets of vertices: the set of vertices in the four corners, $C = \{1, n, n+1, 2n\}$, and the set of the remaining vertices, $R = \{2, \ldots, n-1, n+2, \ldots, 2n-1\}$. For any $a \in C$ and $b \in R$, we have $\mathrm{D_a} = \mathrm{E_a} - \mathrm{I_a} = 1 - 1 = 0$ and $D_b = E_b - I_b = 1 - 2 = -1$. Further, for any pair $u \in A$ and $v \in B$, $c_{uv} = 0\ or\ 1$. Hence, $g_{uv} = D_u + D_v - 2c_{uv} \leq D_u + D_v \leq 0$. The claim thus follows.

9. (10 points) X-architecture routing allows vertical, horizontal, 45-degree, and 135-degree wire segments. Describe how to extend the Lee algorithm for X-architecture routing.

9、將lee-alogrithm之水波擴散方向改成8個，往斜方行進之格子標記為現有步數加上$\sqrt{2}$，若欲標記之格子已被標記則不標記之 (同lee alg)

* 每次從標記最小的點向外擴

，backtrace之方向改為"從"終點開始找，每次皆從八個方向找到小於自身標記步數之格子中的最小值做為backtrace之方向。