EE5333: Introduction to Physical Design Automation

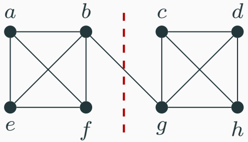
Partitioning

Why partition?

- Exploding design complexity
  - Hard to do layout full-chip as a single entity
- Is logical boundary a good demarcation?
- Emulation
  - Limit on number of pins and available cells
- Divide and conquer

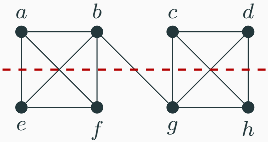
Paritioning problem

- Undirected graph $G = (V, E)$
- Generate k subgraphs of V, $\{V_1, V_2, \ldots, V_k\}$, such that $\bigcup_{i=1}^{k} V_i = V$ and $V_i \bigcap V_j = \emptyset, \forall i, j \in \{1, 2, \ldots, k\}$
- Minimize the number of *cuts* (why?)



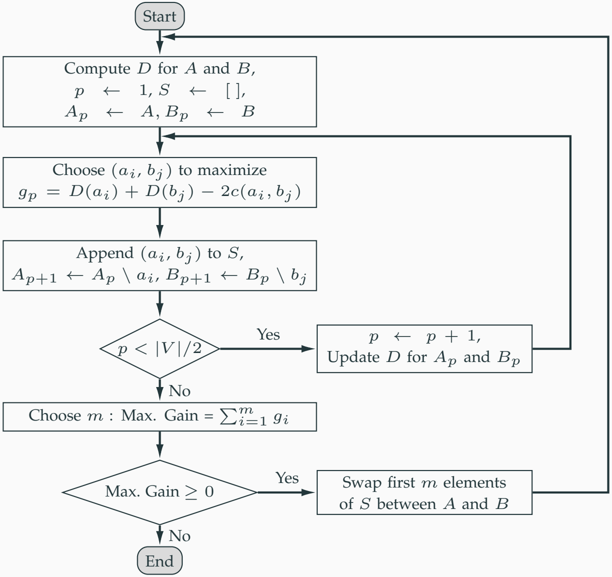
Bipartition

Kernighan-Lin Algorithm - Terminology

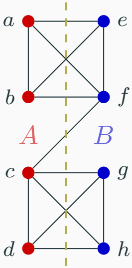


- Undirected graph $G = (V, E)$
- Two partitions A, $B : |A| = |B|$
- Cut size is the sum of costs of edges between A and B
- $E_A(v)$ is the set of edges between v and its neighbours in A
- Cost of moving a vertex v from from A to B is $D(v) = |E_B(v)| - |E_A(v)|$
- Gain on swapping $a \in A$ and $b \in B$ is $g(a, b) = D(a) + D(b) - 2c(a, b)$
- Max. Gain is the maximum sum in a subsequnce of m swaps : $m \left(\leq \frac{|V|}{2}\right)$

Kernighan-Lin Algorithm



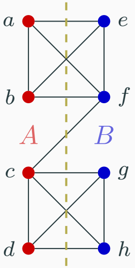
Kernighan-Lin Algorithm



v	$E_A(v)$	$E_B(v)$	$D(v)$
a	1	2	1
b	1	2	1
c	1	3	2
d	1	2	1
e	2	1	1
f	3	1	2
g	2	1	1
h	2	1	1

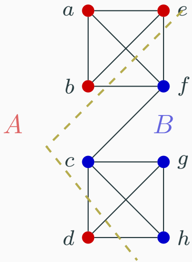
(u, v)	$g(u, v)$
(a, e)	0
(a, f)	1
(a, g)	2
(a, h)	2
(b, e)	0
(b, f)	1
(b, g)	2
(b, h)	2
$(\mathbf{c}, \mathbf{e})$	**3**
(c, f)	2
(c, g)	1
(c, h)	1
(d, e)	2
$(\mathbf{d}, \mathbf{f})$	**3**
(d, g)	0
(d, h)	0

Kernighan-Lin Algorithm



v	$E_A(v)$	$E_B(v)$	$D(v)$
a	1	2	1
b	1	2	1
c	1	3	2
d	1	2	1
e	2	1	1
f	3	1	2
g	2	1	1
h	2	1	1

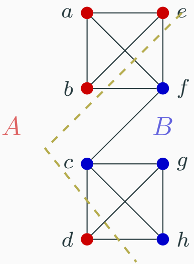
(u,v)	$g(u,v)$
(a,e)	0
(a,f)	1
(a,g)	2
(a,h)	2
(b,e)	0
(b,f)	1
(b,g)	2
(b,h)	2
$(\mathbf{c},\mathbf{e})$	**3**
(c,f)	2
(c,g)	1
(c,h)	1
(d,e)	2
$(\mathbf{d},\mathbf{f})$	**3**
(d,g)	0
(d,h)	0



v	$E_A(v)$	$E_B(v)$	$D(v)$
a	2	1	-1
b	2	1	-1
c	–	–	–
d	0	3	3
e	–	–	–
f	3	1	2
g	1	2	-1
h	1	2	-1

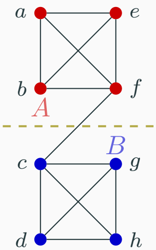
(u,v)	$g(u,v)$
(a,f)	-1
(a,g)	-2
(a,h)	-2
(b,f)	-1
(b,g)	-2
(b,h)	-2
$(\mathbf{d},\mathbf{f})$	**5**
(d,g)	0
(d,h)	0

Kernighan-Lin Algorithm



v	$E_A(v)$	$E_B(v)$	$D(v)$
a	2	1	-1
b	2	1	-1
c	–	–	–
d	0	3	3
e	–	–	–
f	3	1	2
g	1	2	-1
h	1	2	-1

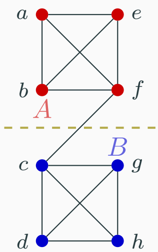
(u, v)	$g(u, v)$
(a, f)	-1
(a, g)	-2
(a, h)	-2
(b, f)	-1
(b, g)	-2
(b, h)	-2
$(\mathbf{d}, \mathbf{f})$	**5**
(d, g)	0
(d, h)	0



v	$E_A(v)$	$E_B(v)$	$D(v)$
a	3	0	-3
b	3	0	-3
c	–	–	–
d	–	–	–
e	–	–	–
f	–	–	–
g	0	3	-3
h	0	3	-3

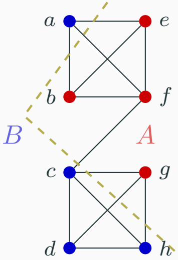
(u, v)	$g(u, v)$
$(\mathbf{a}, \mathbf{g})$	**-6**
(a, h)	-6
(b, g)	-6
(b, h)	-6

Kernighan-Lin Algorithm



v	$E_A(v)$	$E_B(v)$	$D(v)$
a	3	0	-3
b	3	0	-3
c	–	–	–
d	–	–	–
e	–	–	–
f	–	–	–
g	0	3	-3
h	0	3	-3

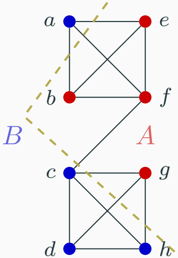
(u, v)	$g(u, v)$
$(\mathbf{a}, \mathbf{g})$	**-6**
(a, h)	-6
(b, g)	-6
(b, h)	-6



v	$E_A(v)$	$E_B(v)$	$D(v)$
a	–	–	–
b	2	1	-1
c	–	–	–
d	–	–	–
e	–	–	–
f	–	–	–
g	–	–	–
h	1	2	-1

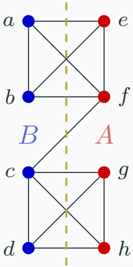
(u, v)	$g(u, v)$
$(\mathbf{b}, \mathbf{h})$	**-2**

Kernighan-Lin Algorithm



v	$E_A(v)$	$E_B(v)$	$D(v)$
a	–	–	–
b	2	1	-1
c	–	–	–
d	–	–	–
e	–	–	–
f	–	–	–
g	–	–	–
h	1	2	-1

(u, v)	$g(u, v)$
(b, h)	-2



v	$E_A(v)$	$E_B(v)$	$D(v)$
a	–	–	–
b	–	–	–
c	–	–	–
d	–	–	–
e	–	–	–
f	–	–	–
g	–	–	–
h	–	–	–

Kernighan-Lin extensions

- Handling unequal partition sizes : Initialize with the required number of vertices and restrict number of moves in an iteration to $\min(|A|, |B|)$
- k-way partitioning : Iteratively apply KL to every pair of bipartitions until no improvement in an iteration

Logical operations as linear constraints

- F is a boolean function in product-of-sums form. e.g. $F = (x_1 + \overline{x_2} + x_3) \cdot (\overline{x_1} + x_2 + \overline{x_3})$
- An equivalent ILP constraint:

$$\begin{aligned} x_1 + (1 - x_2) + x_3 &\geq 1 \\ (1 - x_1) + x_2 + (1 - x_3) &\geq 1 \end{aligned}$$

- Logical implication : $a \rightarrow b = \overline{a} + b$
- Logical equivalence or bi-implication $a \leftrightarrow b = a \rightarrow b \wedge b \rightarrow a = (\overline{a} + b) \cdot (a + \overline{b})$
- To translate any logical function to a POS form we use the logical equivalence operation

Logical operations as linear constraints

- AND operation

$$
\begin{aligned}
F = (y \leftrightarrow a \wedge b) &= (\overline{y} + ab) \cdot (y + \overline{ab}) \\
&= (\overline{y} + a) \cdot (\overline{y} + b) \cdot (y + \overline{a} + \overline{b})
\end{aligned}
$$

$$
\begin{aligned}
(1 - y) + a &\geq 1 \\
(1 - y) + b &\geq 1 \\
y + (1 - a) + (1 - b) &\geq 1
\end{aligned}
$$

- OR operation

$$
\begin{aligned}
F = (y \leftrightarrow a \vee b) &= (\overline{y} + a + b) \cdot (y + \overline{a + b}) \\
&= (\overline{y} + a + b) \cdot (y + \overline{a}) \cdot (y + \overline{b})
\end{aligned}
$$

$$
\begin{aligned}
(1 - y) + a + b &\geq 1 \\
y + (1 - a) &\geq 1 \\
y + (1 - b) &\geq 1
\end{aligned}
$$

- XOR operation

$$
\begin{aligned}
F = (y \leftrightarrow a \oplus b) &= (\overline{y} + a \oplus b) \cdot (y + \overline{a \oplus b}) \\
&= (\overline{y} + (a + b) \cdot (\overline{a} + \overline{b})) \cdot (y + (a + \overline{b}) \cdot (\overline{a} + b)) \\
&= (\overline{y} + a + b) \cdot (\overline{y} + \overline{a} + \overline{b}) \cdot (y + a + \overline{b}) \cdot (y + \overline{a} + b)
\end{aligned}
$$

$$
\begin{aligned}
(1 - y) + a + b &\geq 1 \\
(1 - y) + (1 - a) + (1 - b) &\geq 1 \\
y + a + (1 - b) &\geq 1 \\
y + 1 - a + b &\geq 1
\end{aligned}
$$

Bipartitioning using ILP

- Undirected graph $G = (V, E)$
- Partition V into V_1 and V_2 such that $V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$, and $|V_1| = |V_2| = \frac{|V|}{2}$
- x_v is the indicator variable for v being in V_1
- $x_{u,v}$ is the indicator variable for $(u, v) \in E$ being cut

$$x_{u,v} = x_u \oplus x_v$$

$$\begin{aligned}
\underset{x_{u,v}}{\text{minimize}} \quad & \sum_{(u,v)\in E} x_{u,v} \\
\text{subject to} \quad & \sum_{v\in V} x_v = \frac{|V|}{2} \\
& x_u - x_v \leq x_{u,v}, \quad \forall (u,v) \in E \\
& x_v - x_u \leq x_{u,v}, \quad \forall (u,v) \in E \\
& x_u + x_v \geq x_{u,v}, \quad \forall (u,v) \in E \\
& x_u + x_v + x_{u,v} \leq 2, \quad \forall (u,v) \in E \\
& x_v \in \{0,1\}, \quad \forall v \in V \\
& x_{u,v} \in \{0,1\}, \quad \forall (u,v) \in E
\end{aligned}$$

k-way partitioning using ILP

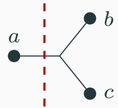
- Undirected graph $G = (V, E)$
- Partition V into $V_1, V_2 \ldots V_k$ such that $\bigcup_{i=1}^{k} V_i = V$, $V_i \cap V_j = \emptyset \; \forall i \neq j$, and $|V_i| = \dfrac{|V|}{k}$
- $x_{v,i}$ is the indicator variable for v being in V_i
- $x_{u,i,v,j}$ is the indicator variable for $(u, v) \in E$ being cut if u belongs to V_i and v belongs to V_j

$$x_{u,i,v,j} = x_{u,i} \oplus x_{v,j}$$

$$
\begin{aligned}
\underset{x_{u,i}}{\text{minimize}} \quad & \sum_{(u,v)\in E} \sum_{i=1}^{k} \sum_{\substack{j=1 \\ j\neq i}}^{k} x_{u,i} \oplus x_{v,j} \\
\text{subject to} \quad & \sum_{i=1}^{k} x_{v,i} = 1, & \forall v \in V \\
& \sum_{v \in V} x_{v,i} = \frac{|V|}{k}, & \forall i \in \{1, 2, \ldots k\} \\
& x_{u,i} - x_{v,j} \leq x_{u,i,v,j}, & \forall (u, v) \in E, \forall i \neq j \\
& x_{v,j} - x_{u,i} \leq x_{u,i,v,j}, & \forall (u, v) \in E, \forall i \neq j \\
& x_{u,i} + x_{v,j} \geq x_{u,i,v,j}, & \forall (u, v) \in E, \forall i \neq j \\
& x_{u,i} + x_{v,j} + x_{u,i,v,j} \leq 2, & \forall (u, v) \in E, \forall i \neq j \\
& x_{u,i}, x_{u,i,v,j} \in \{0, 1\}
\end{aligned}
$$

Fiduccia-Mattheyses Algorithm

- Hypergraph $H = (V, E)$; $c(v)$ is the cost of vertex $v \in V$
- A hyperedge (net) is *cut* if connected vertices occupy more than one partition; *uncut* otherwise
- Gain of vertex v ($g(v)$)is the change in cutset size if a vertex moves from one partition to another
  - $FS(v)$ is the number of cut edges connected only to v
  - $TE(v)$ is the number of uncut edges connected to v



- Max. positive gain
- Balance criterion : ratio $r = \frac{\sum_{v \in A} c(v)}{\sum_{v \in V} c(v)} = \frac{c(A)}{c(V)}$
  - Tolerance : $c_{max}(V) = \max(\{c(v) : v \in V\})$
  - $r \cdot c(V) - c_{max}(V) \leq c(A) \leq r \cdot c(V) + c_{max}(V)$
- A hyperedge e is *critical* if it is connected to a cell v whose move will affect its cut status
  - All vertices connected to e are in same partition
  - Exactly one vertex connected to e is in a different partition