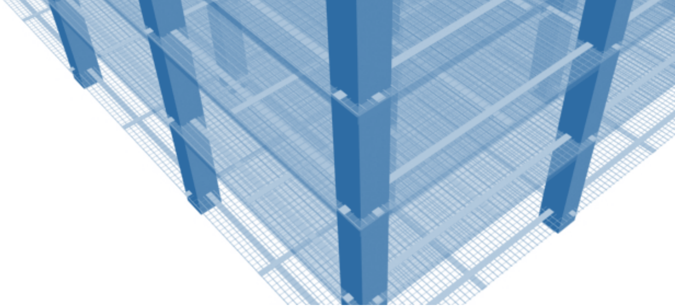
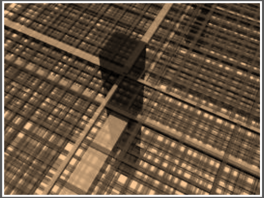
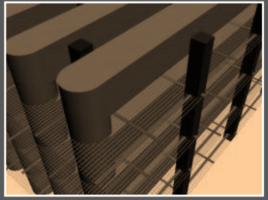
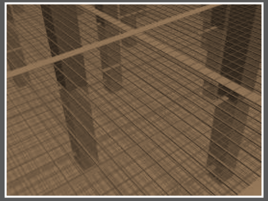
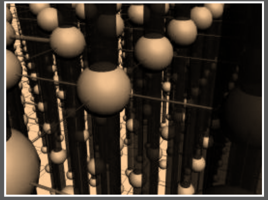


ECE 6133 Project Overview

Spring 2022

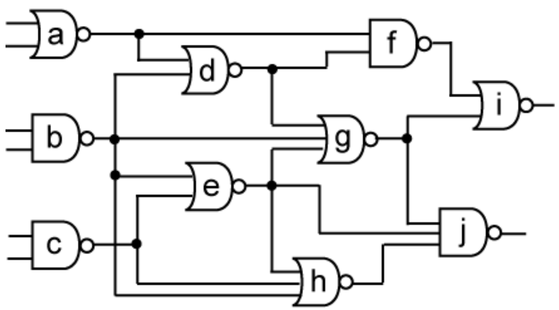
Sung Kyu Lim

limsk@ece.gatech.edu

Georgia Institute of Technology

1/4. EIG Partitioning

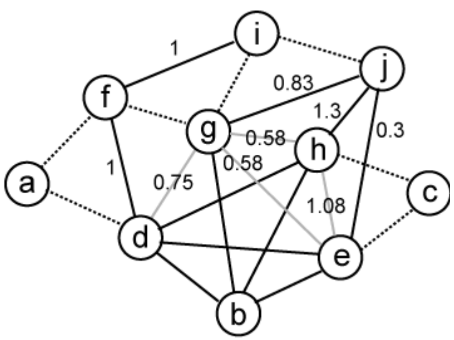
- **Circuit partitioning using matrices and eigenvectors**



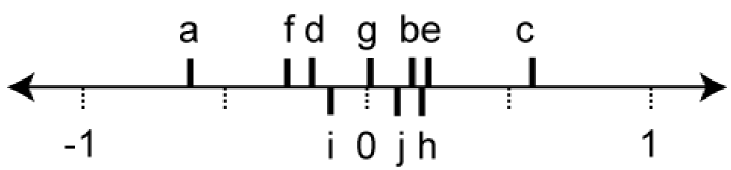
circuit

	a	b	c	d	e	f	g	h	i	j
a	1.0	0	0	-0.5	0	-0.5	0	0	0	0
b	0	1.0	0	-0.25	-0.25	0	-0.25	-0.25	0	0
c	0	0	1.0	0	-0.5	0	0	-0.5	0	0
d	-0.5	-0.25	0	3.0	-0.25	-1.0	-0.75	-0.25	0	0
e	0	-0.25	-0.5	-0.25	2.99	0	-0.58	-1.08	0	-0.33
f	-0.5	0	0	-1.0	0	3.0	-0.5	0	-1.0	0
g	0	-0.25	0	-0.75	-0.58	-0.5	3.99	-0.58	-0.5	-0.83
h	0	-0.25	-0.5	-0.25	-1.08	0	-0.58	3.99	0	-1.33
i	0	0	0	0	0	-1.0	-0.5	0	2.0	-0.5
j	0	0	0	0	-0.33	0	-0.83	-1.33	-0.5	2.99

Laplacian matrix L



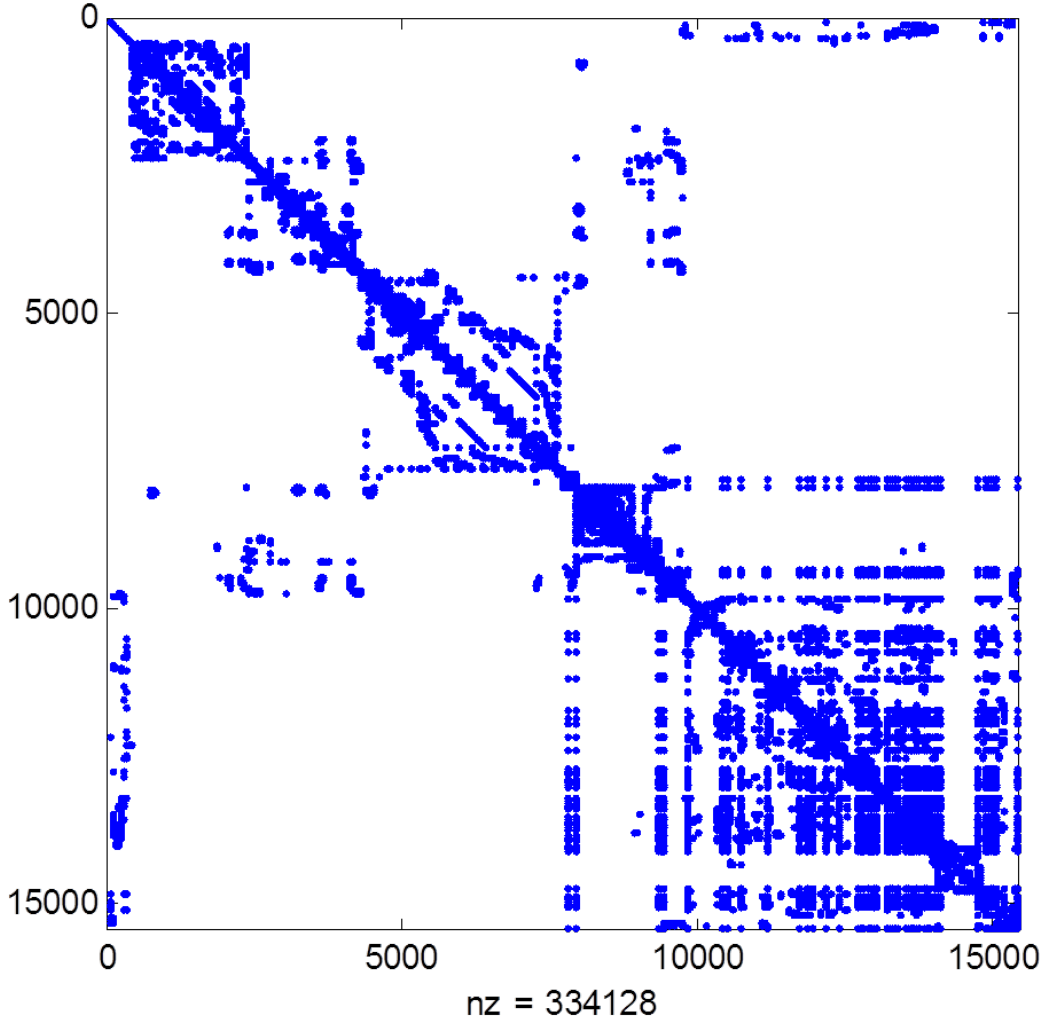
clique-based model



2nd smallest eigenvector of L gives linear placement

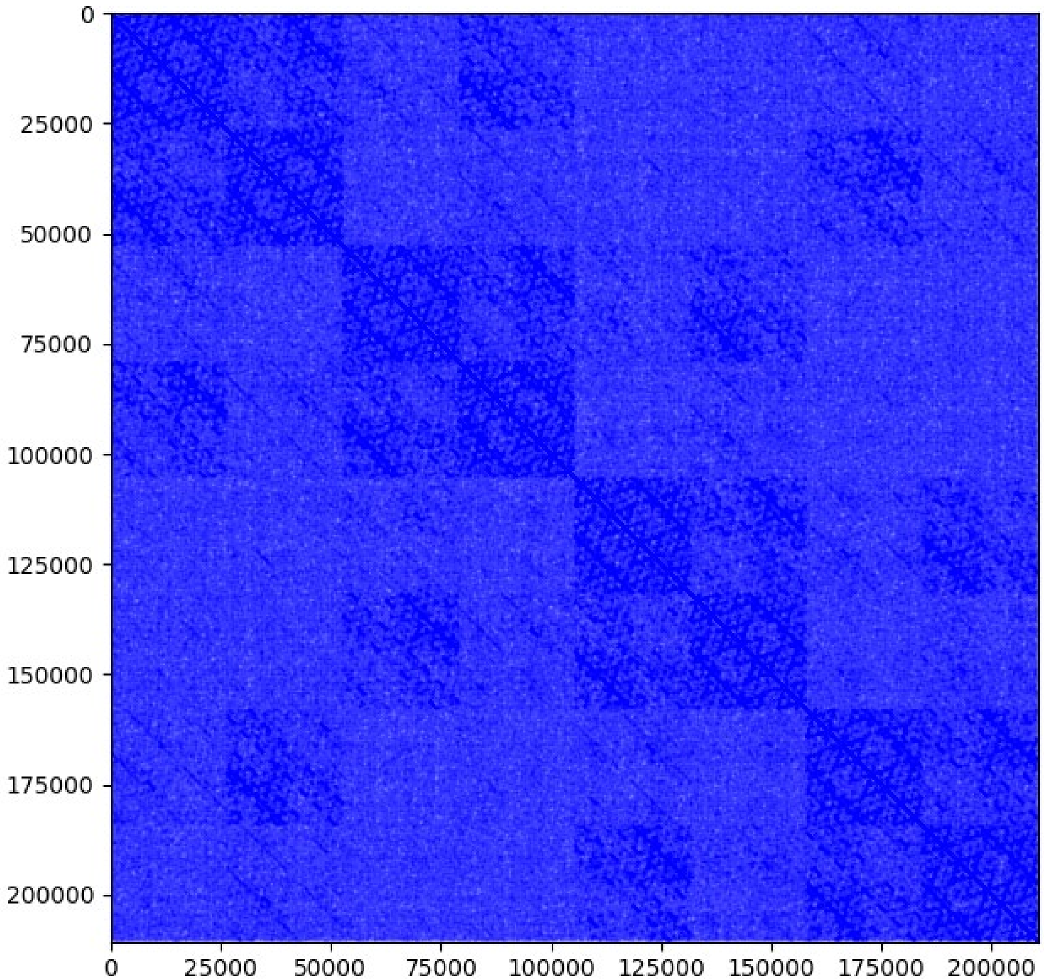
Laplacian Sparsity: Small Circuit

- **industry3.hgr**
  - 15,406 nodes
  - 21,923 nets
  - 237M entries in matrix
  - 334k nonzero entries
  - 0.14% fill factor
- **Sparse matrix methods**
  - enables us to consider systems 700X larger than otherwise possible

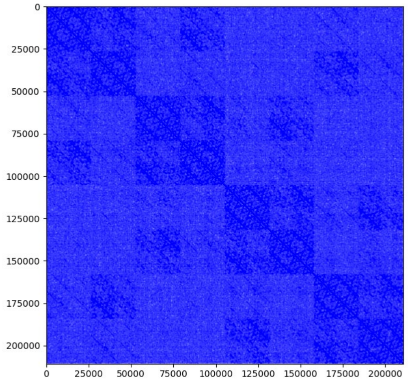


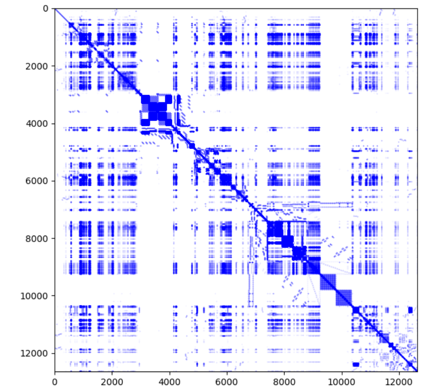
Laplacian Sparsity: Large Circuit

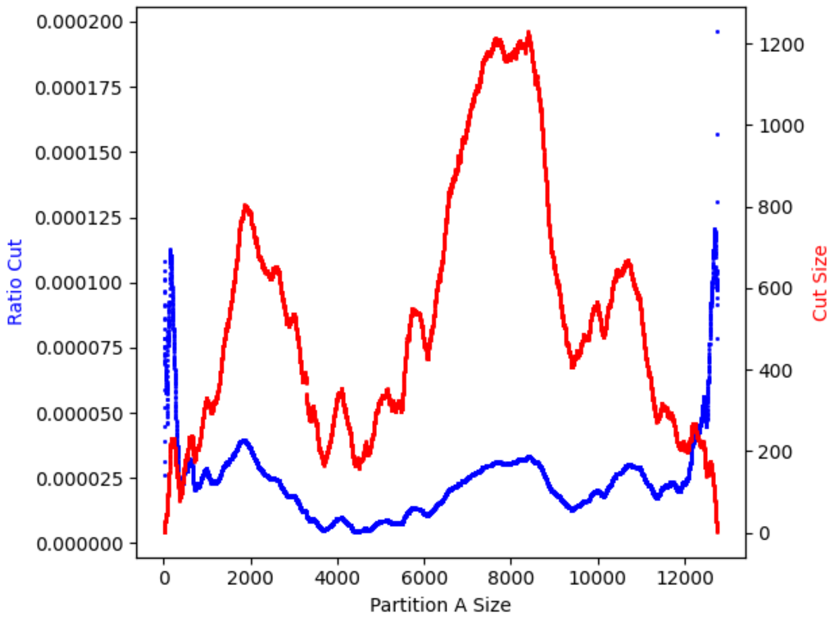
- **IBM18 benchmark**
  - 210,613 cells
  - Laplacian matrix fill rate: 0.01%!
  - 178 GB of memory required to store in a normal matrix
  - Sparse matrix can bring that down to 2.5 MB

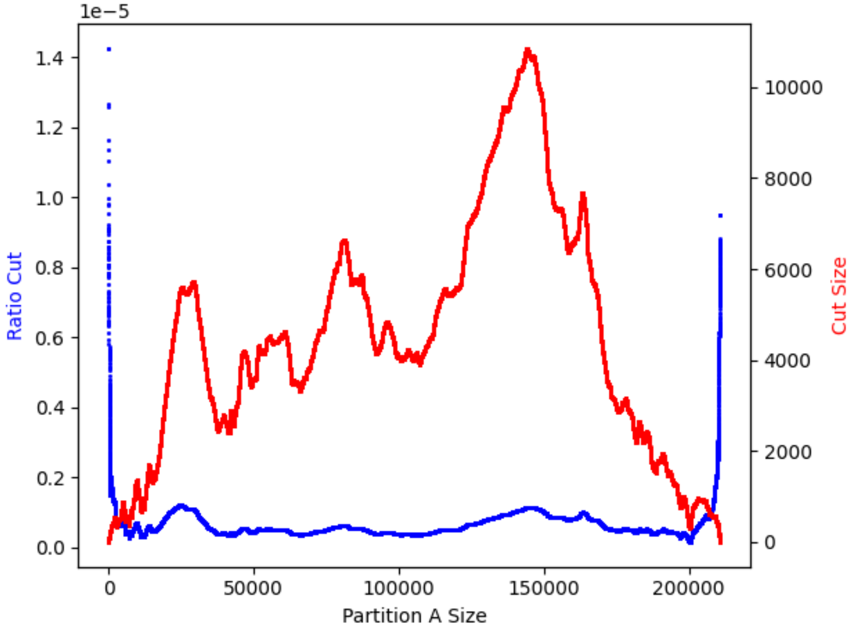


Cutsize vs. Ratio Cut Landscape



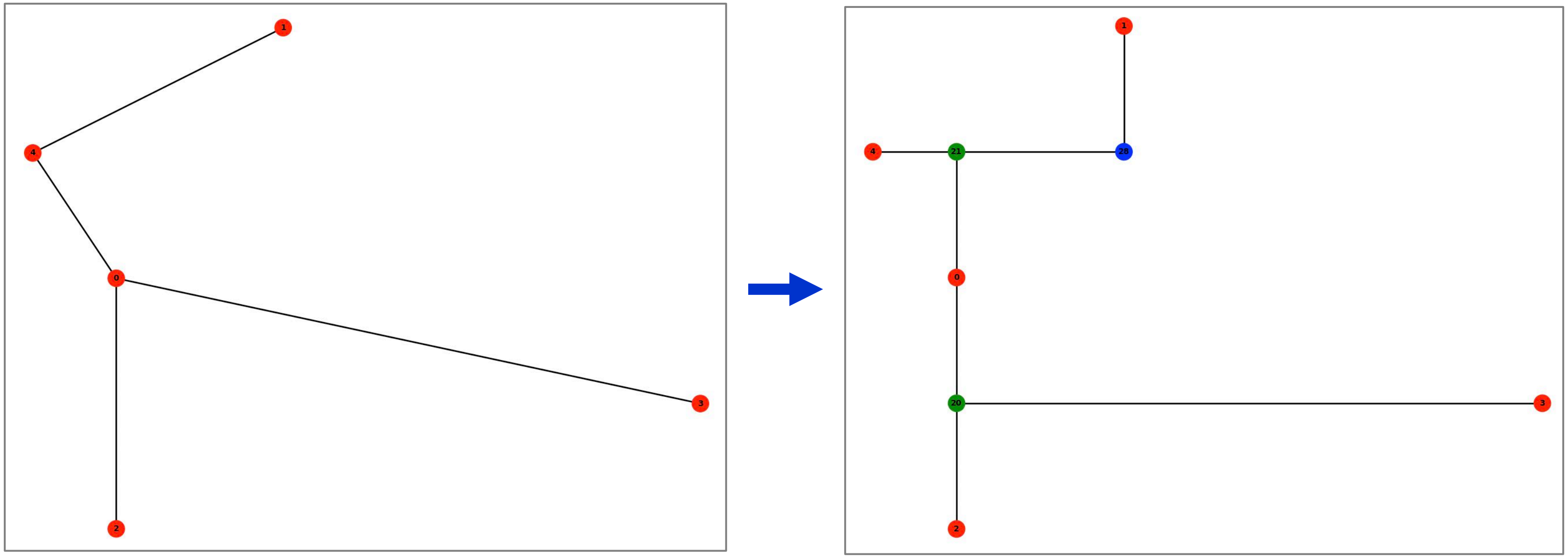






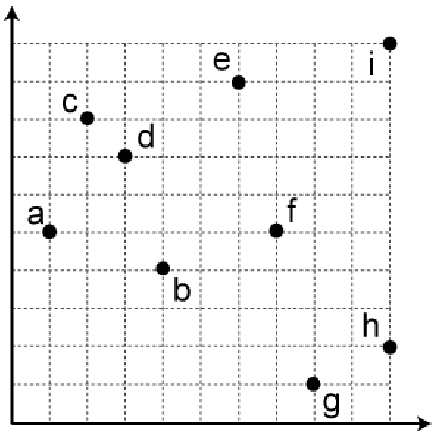
2/4. L-Shaped vs. Borah Routing

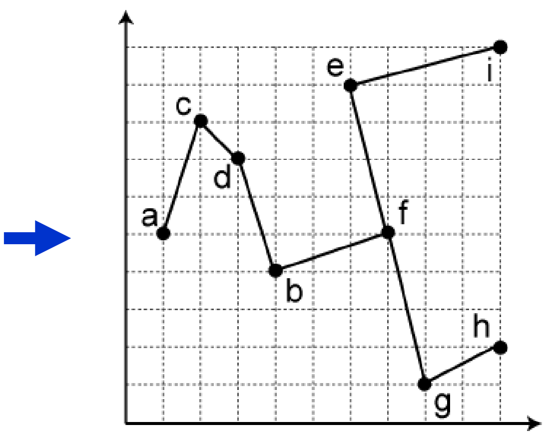
- **Steiner points**
  - Reduce the overall wirelength
  - Finding Steiner points is NP-hard….
  - So, we rely on heuristics

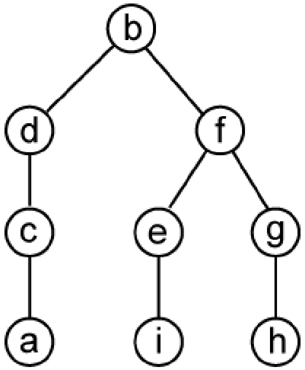


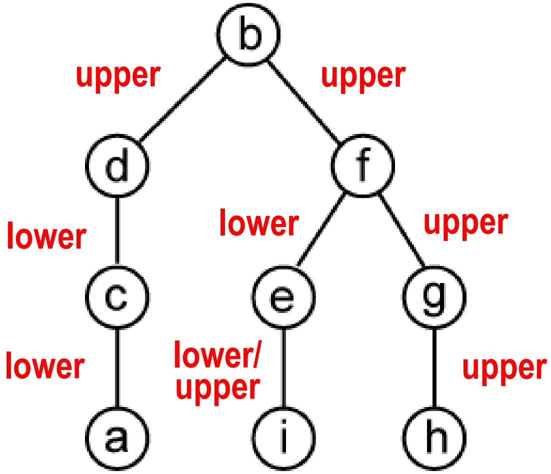
L-Shaped Steiner Routing

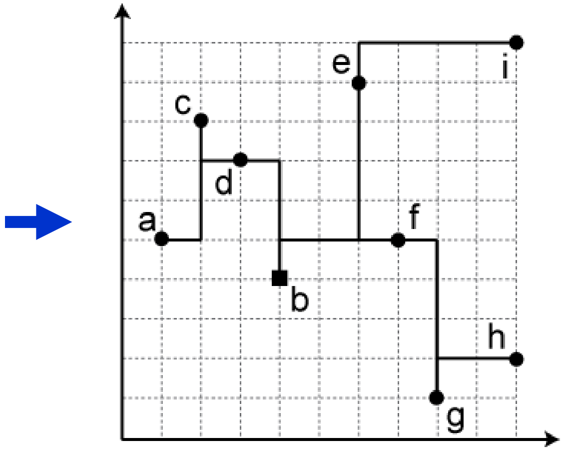
- Strategy: build **separable MST** and improve with **L-edges**





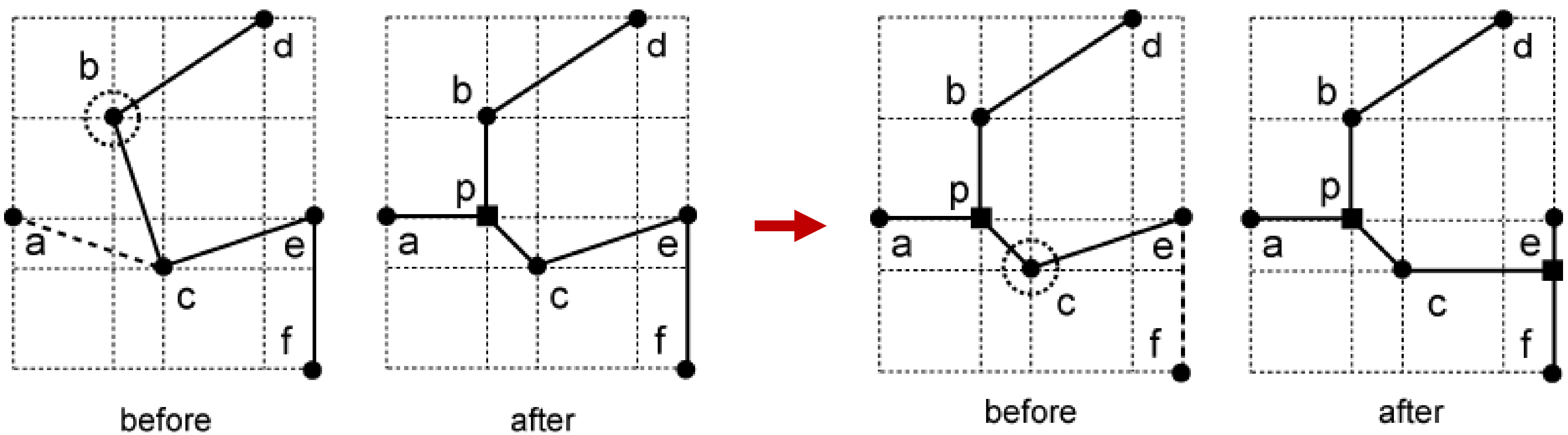






Borah Routing

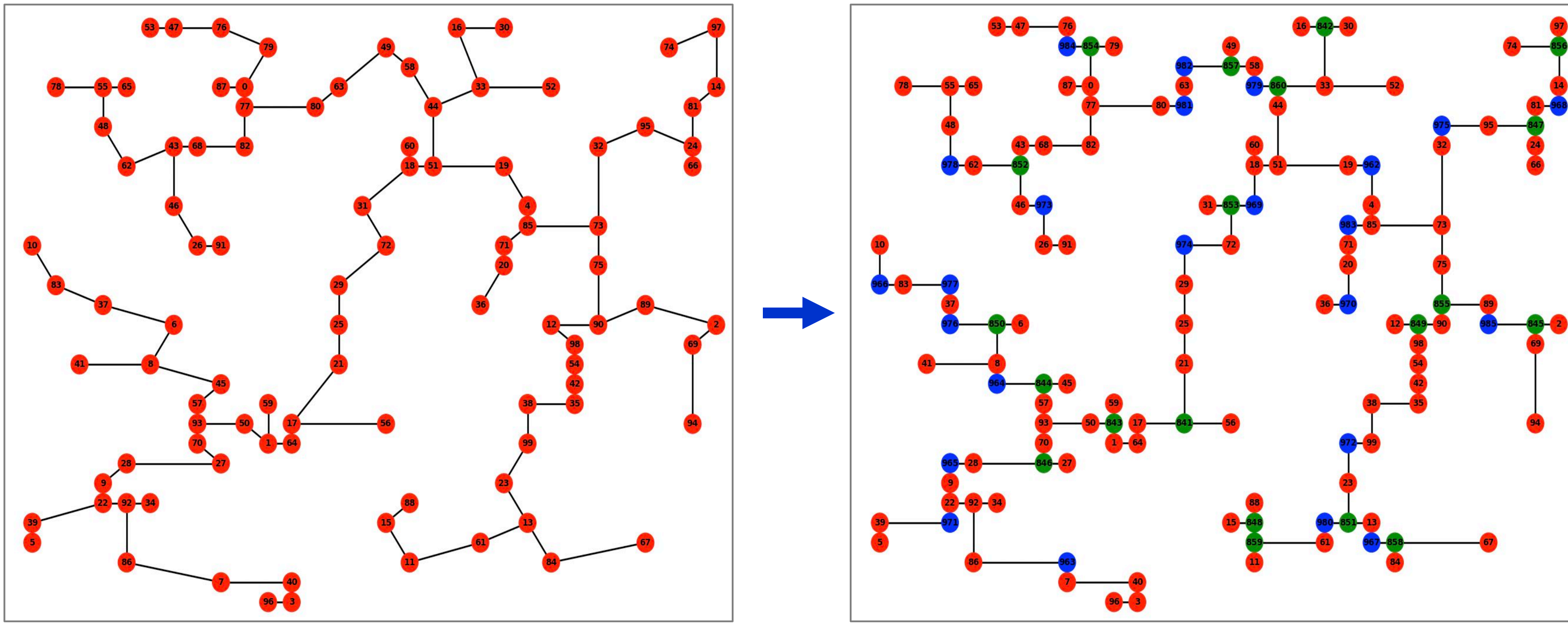
- Based on node/edge pairing



WL reduces from 20 to 18

WL reduces from 18 to 17

Borah Routing Sample

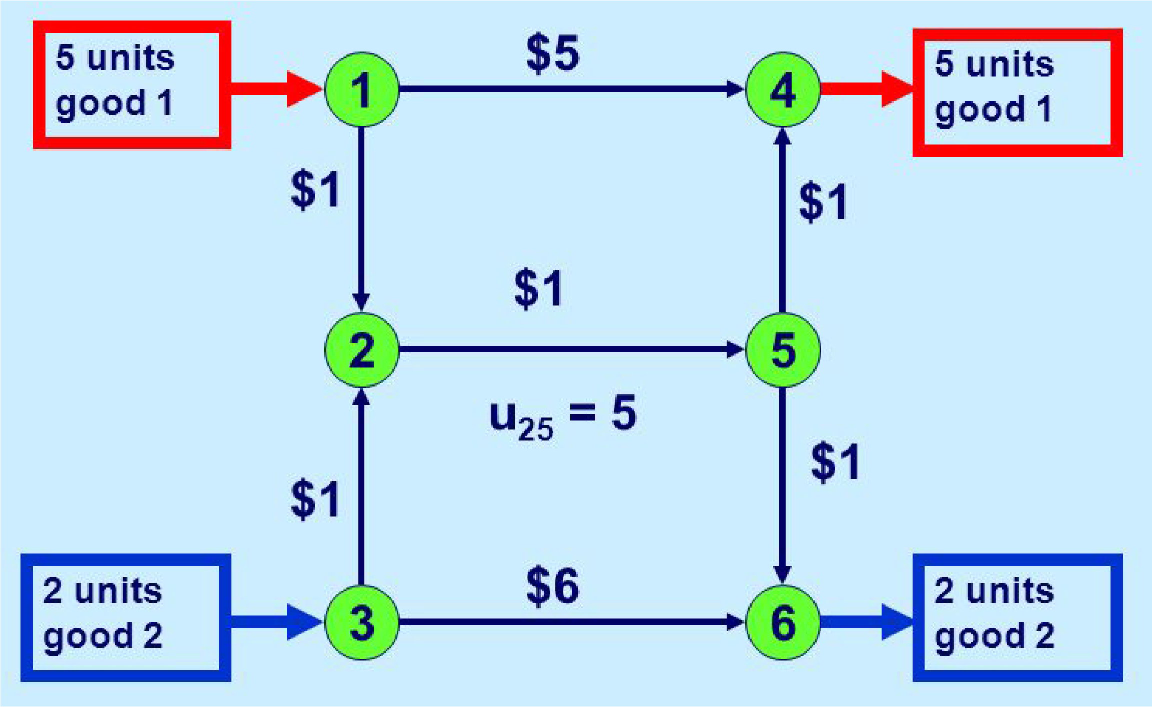


initial (WL = 242)

final (WL = 221), 20 Steiner pt added
Took 1.2 sec

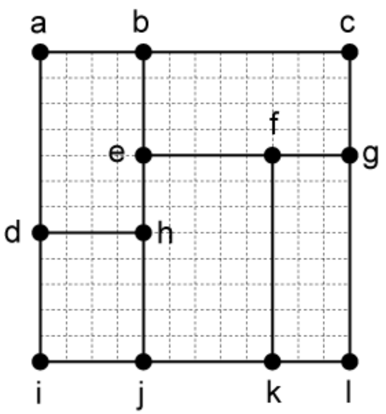
3/4. Multi-Commodity Flow Routing

- Cost minimization problem
  - How do we ship the units so that the overall cost is minimized?
  - Assume the capacity of each edge is 5 units

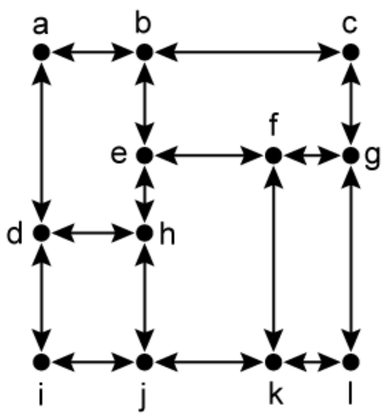


MCF-based Multi-net Routing

- **Set up ILP for MCF routing**
  - **Capacity of each edge in *G* is 2**
  - **Each edge in *G* becomes a pair of bi-directional arcs in *F***
  - $n_1 = \{a,l\}$, $n_2 = \{i,c\}$, $n_3 = \{d,f\}$, $n_4 = \{k,d\}$, $n_5 = \{g,h\}$, $n_6 = \{b,k\}$



routing graph G



flow network F

arc	cost	arc	cost	arc	cost	arc	cost
(a,b)	4	(b,a)	4	(b,c)	8	(c,b)	8
(d,h)	4	(h,d)	4	(e,f)	5	(f,e)	5
(f,g)	3	(g,f)	3	(i,j)	4	(j,i)	4
(j,k)	5	(k,j)	5	(k,l)	3	(l,k)	3
(a,d)	7	(d,a)	7	(d,i)	5	(i,d)	5
(b,e)	4	(e,b)	4	(e,h)	3	(h,e)	3
(h,j)	5	(j,h)	5	(f,k)	8	(k,f)	8
(c,g)	4	(g,c)	4	(g,l)	8	(l,g)	8

ILP Formulation

$$4(x^1_{a,b} + \cdots + x^6_{a,b}) + 4(x^1_{b,a} + \cdots + x^6_{b,a}) + 8(x^1_{b,c} + \cdots + x^6_{b,c}) +$$
$$8(x^1_{c,b} + \cdots + x^6_{c,b}) + 4(x^1_{d,h} + \cdots + x^6_{d,h}) + 4(x^1_{h,d} + \cdots + x^6_{h,d}) +$$
$$5(x^1_{e,f} + \cdots + x^6_{e,f}) + 5(x^1_{f,e} + \cdots + x^6_{f,e}) + 3(x^1_{f,g} + \cdots + x^6_{f,g}) +$$
$$3(x^1_{g,f} + \cdots + x^6_{g,f}) + 4(x^1_{i,j} + \cdots + x^6_{i,j}) + 4(x^1_{j,i} + \cdots + x^6_{j,i}) +$$
$$5(x^1_{j,k} + \cdots + x^6_{j,k}) + 5(x^1_{k,j} + \cdots + x^6_{k,j}) + 3(x^1_{k,l} + \cdots + x^6_{k,l}) +$$
$$3(x^1_{l,k} + \cdots + x^6_{l,k}) + 7(x^1_{a,d} + \cdots + x^6_{a,d}) + 7(x^1_{d,a} + \cdots + x^6_{d,a}) +$$
$$5(x^1_{d,i} + \cdots + x^6_{d,i}) + 5(x^1_{i,d} + \cdots + x^6_{i,d}) + 4(x^1_{b,e} + \cdots + x^6_{b,e}) +$$
$$4(x^1_{e,b} + \cdots + x^6_{e,b}) + 3(x^1_{e,h} + \cdots + x^6_{e,h}) + 3(x^1_{h,e} + \cdots + x^6_{h,e}) +$$
$$5(x^1_{h,j} + \cdots + x^6_{h,j}) + 5(x^1_{j,h} + \cdots + x^6_{j,h}) + 8(x^1_{f,k} + \cdots + x^6_{f,k}) +$$
$$8(x^1_{k,f} + \cdots + x^6_{k,f}) + 4(x^1_{c,g} + \cdots + x^6_{c,g}) + 4(x^1_{g,c} + \cdots + x^6_{g,c}) +$$
$$8(x^1_{g,l} + \cdots + x^6_{g,l}) + 8(x^1_{l,g} + \cdots + x^6_{l,g})$$

objective function

$$x^1_{a,b} + x^1_{a,d} - x^1_{b,a} - x^1_{d,a} = 1$$
$$x^2_{a,b} + x^2_{a,d} - x^2_{b,a} - x^2_{d,a} = 0$$
$$x^3_{a,b} + x^3_{a,d} - x^3_{b,a} - x^3_{d,a} = 0$$
$$x^4_{a,b} + x^4_{a,d} - x^4_{b,a} - x^4_{d,a} = 0$$
$$x^5_{a,b} + x^5_{a,d} - x^5_{b,a} - x^5_{d,a} = 0$$
$$x^6_{a,b} + x^6_{a,d} - x^6_{b,a} - x^6_{d,a} = 0$$

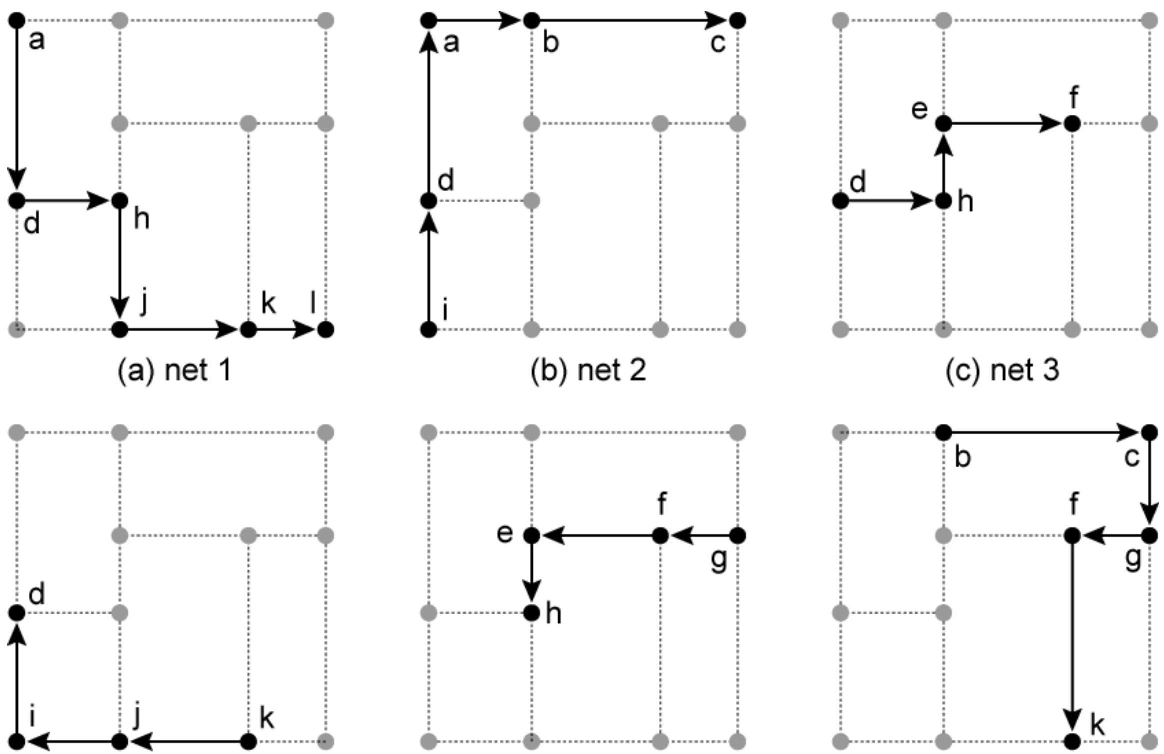
demand constraints

$$x^1_{a,b} + \cdots x^6_{a,b} + x^1_{b,a} + \cdots x^6_{b,a} \leq 2$$
$$x^1_{b,c} + \cdots + x^6_{b,c} + x^1_{c,b} + \cdots + x^6_{c,b} \leq 2$$
$$x^1_{d,h} + \cdots + x^6_{d,h} + x^1_{h,d} + \cdots + x^6_{h,d} \leq 2$$
$$x^1_{e,f} + \cdots + x^6_{e,f} + x^1_{f,e} + \cdots + x^6_{f,e} \leq 2$$
$$\cdots$$
$$x^1_{h,j} + \cdots + x^6_{h,j} + x^1_{j,h} + \cdots + x^6_{j,h} \leq 2$$
$$x^1_{f,k} + \cdots + x^6_{f,k} + x^1_{k,f} + \cdots + x^6_{k,f} \leq 2$$
$$x^1_{c,g} + \cdots + x^6_{c,g} + x^1_{g,c} + \cdots + x^6_{g,c} \leq 2$$
$$x^1_{g,l} + \cdots + x^6_{g,l} + x^1_{l,g} + \cdots + x^6_{l,g} \leq 2$$

capacity constraints

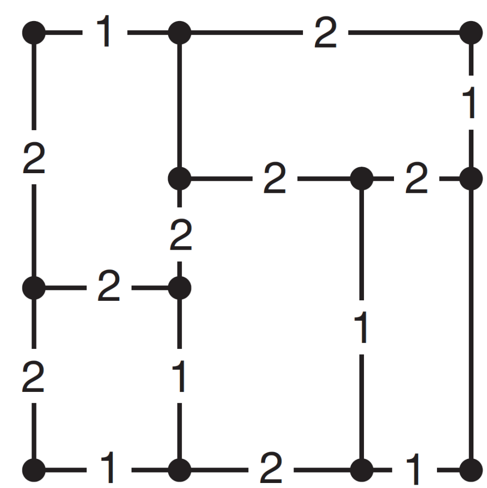
ILP-based Routing Solution

- **Using off-the-shelf ILP solver**



(a) net 1 (b) net 2 (c) net 3

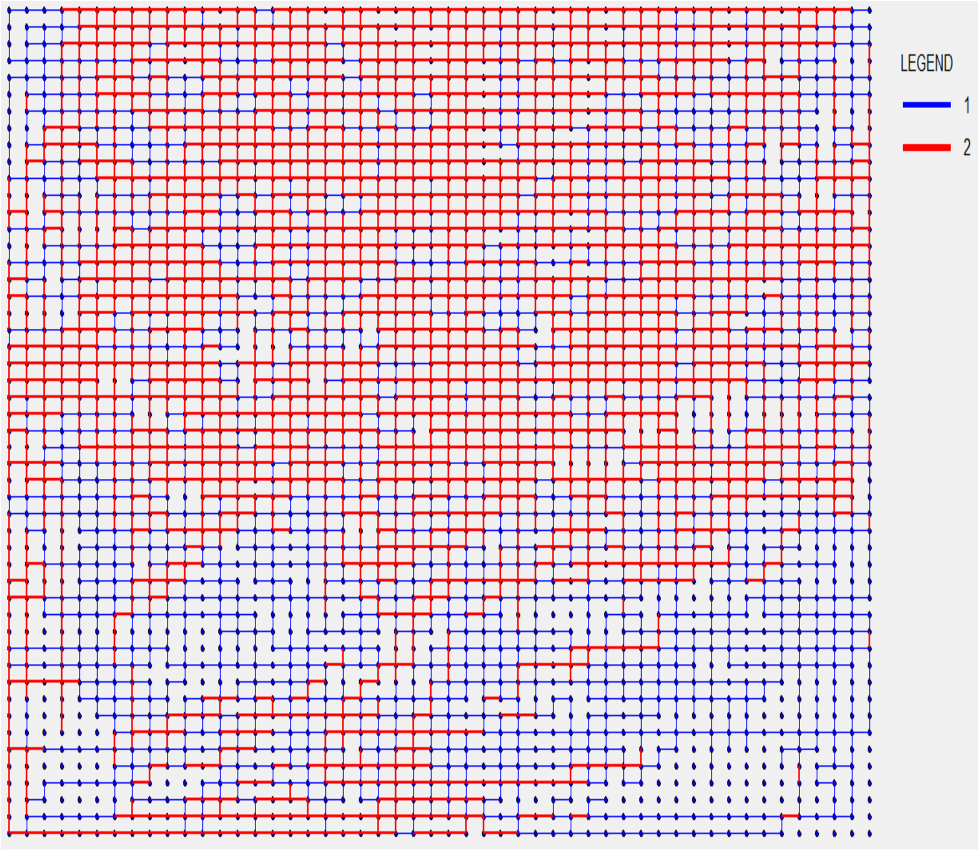
(d) net 4 (e) net 5 (f) net 6



channel usage

Sample Routing Results

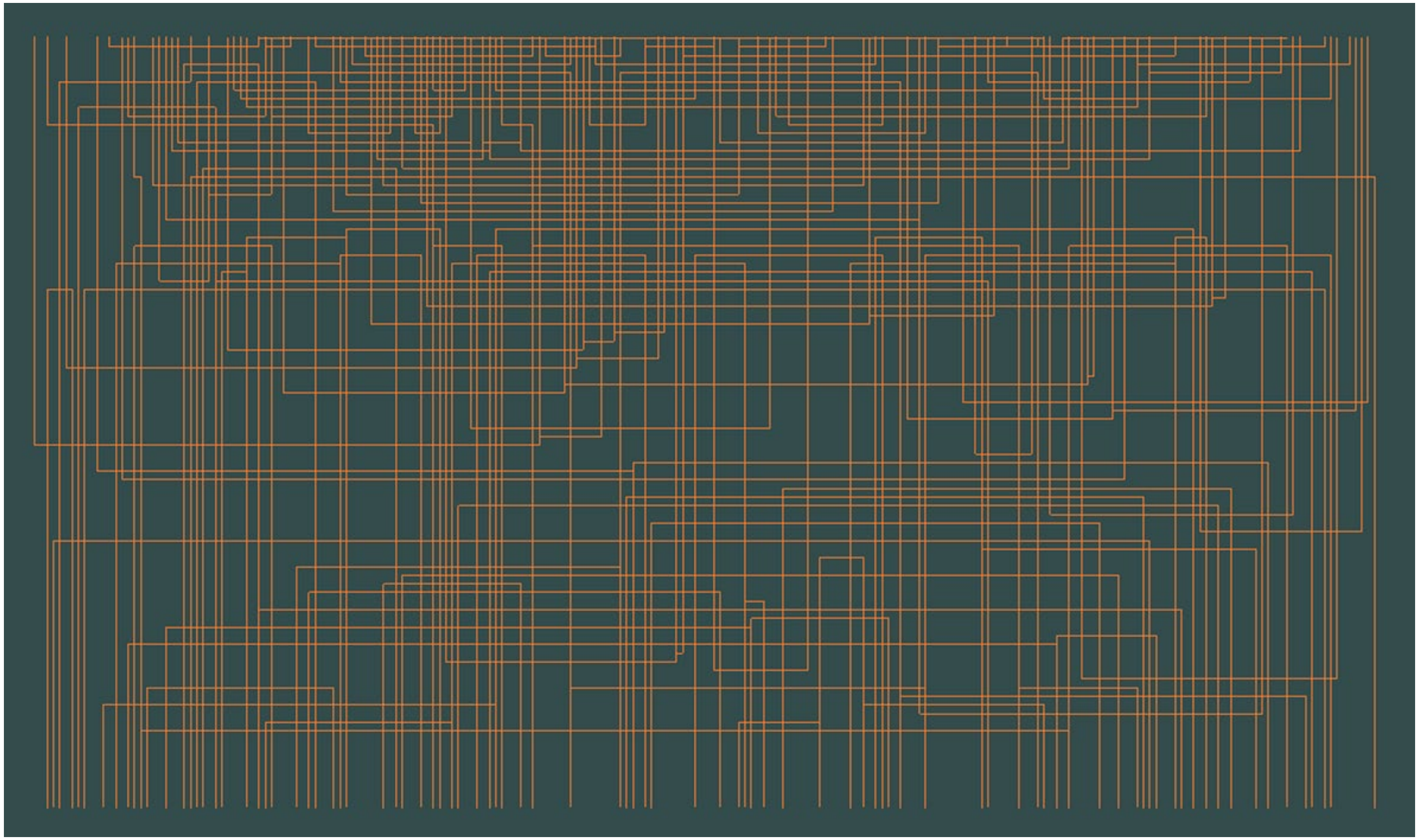
200 net benchmark
usage 1:1298 edges
usage 2: 1412 edges



350 net benchmark
usage 1:1849 edges
usage 2:2468 edges

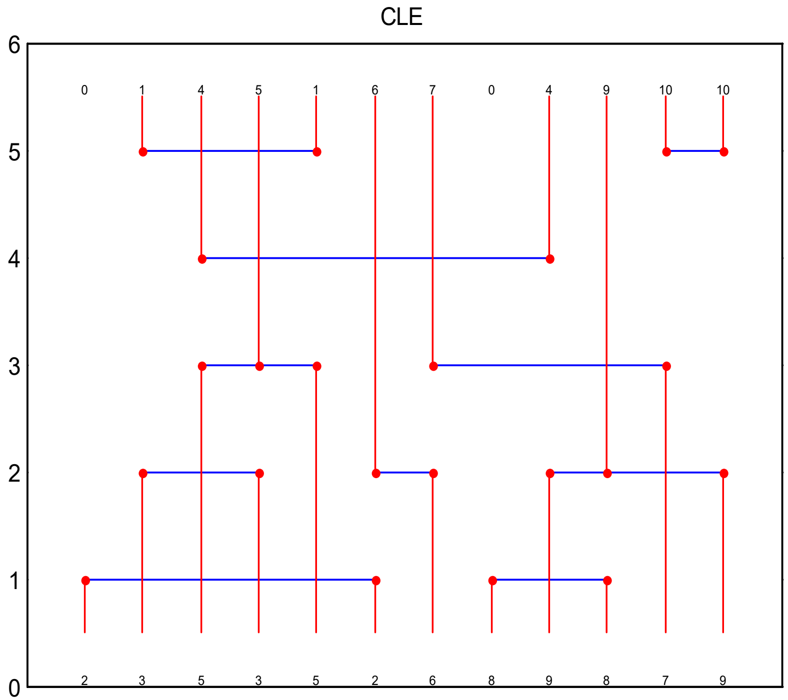
4/4. YK Channel Routing

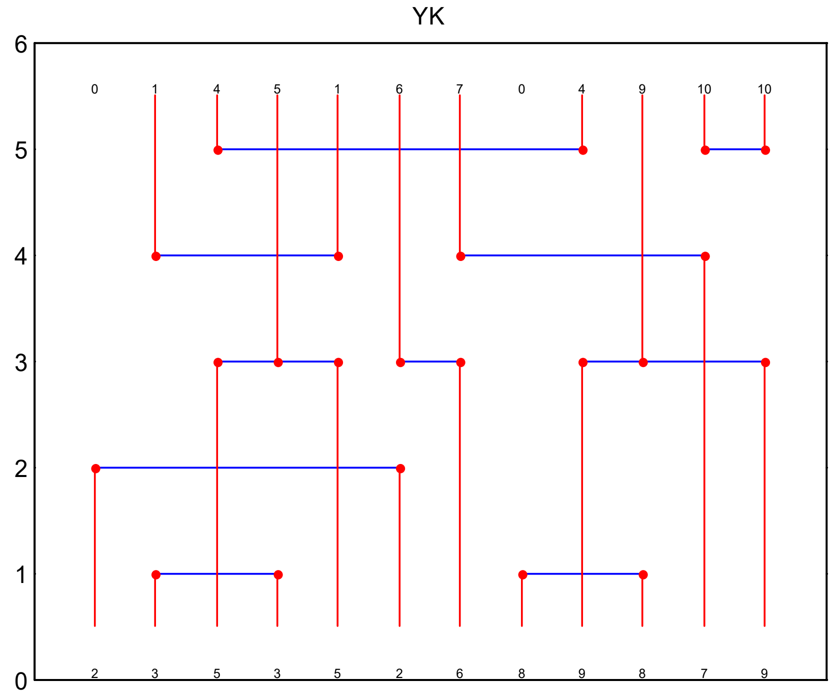
- **Channel routing**
  - **Pins on top/bottom only, 2 metal layers used**



Two Algorithms: Small Circuit

- **CLE vs. YK algorithms**



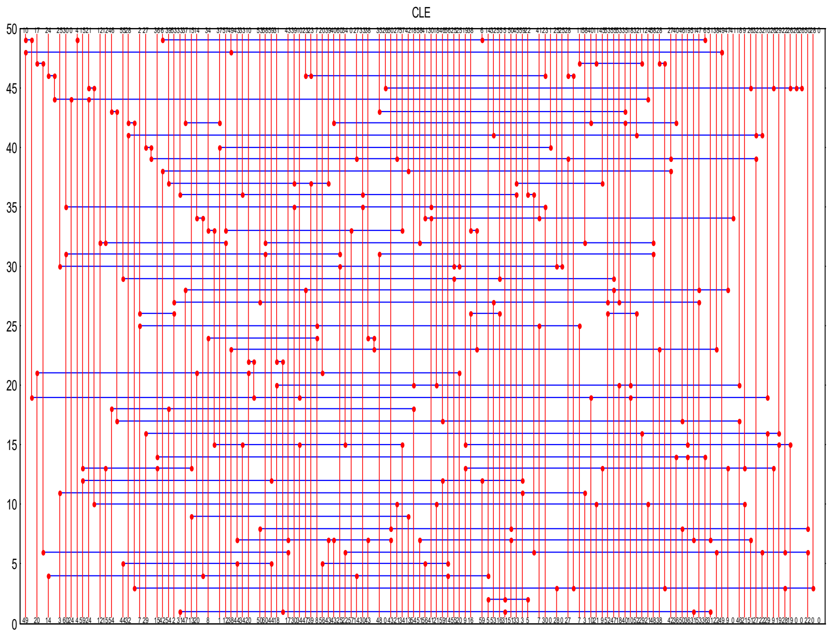


channel density: 5
track used: 5

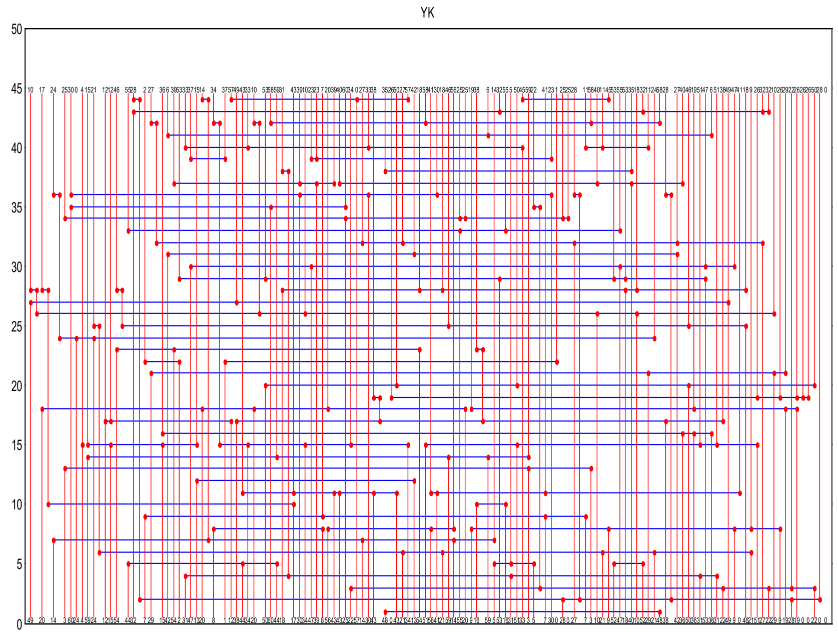
channel density: 5
track used: 5

Two Algorithms: Large Circuit

- **YK offers better solution**
  - **Based on zoning and net merging schemes**



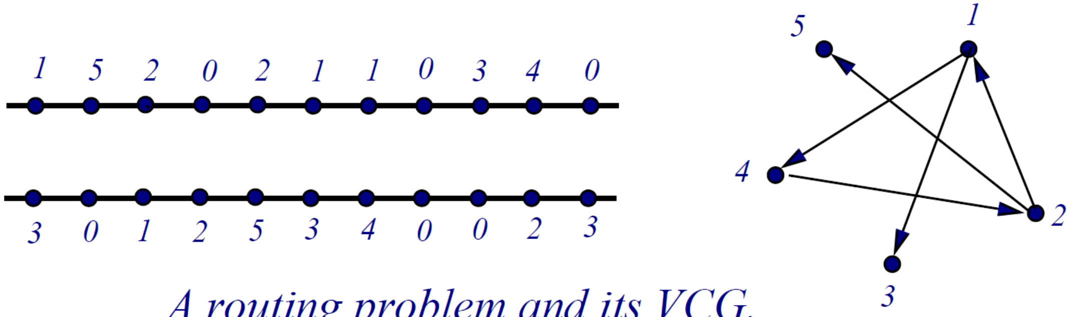
channel density: 41
track used: 49



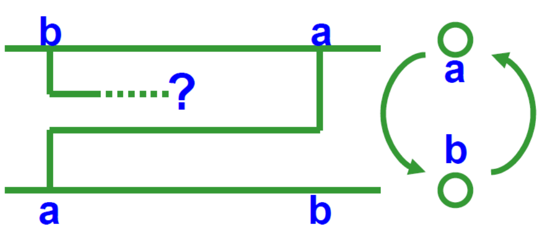
channel density: 41
track used: 44

Cycles in VCG

- **Vertical constraint graph**
  - VCG $G = (V, E)$ is **directed** graph where
    - $V = \{v_i | v_i$ represents a net $n_i\}$
    - $E = \{(v_i, v_j)|$ a vertical constraint exists between n_i and $n_j\}$.
  - For graph G: vertices $\Leftrightarrow$ nets; edge $i \rightarrow j \Leftrightarrow$ net i must be above net j.



A routing problem and its VCG.



how do we handle this?

Doglegs Are Very Useful

- Reduce track usage