Spring 1989

Efficient Multi-Layer Channel Routing

Ronald I. Greenberg
Rgreen@luc.edu

Efficient Multi-Layer Channel Routing

Ronald I. Greenberg
MIT

Overview

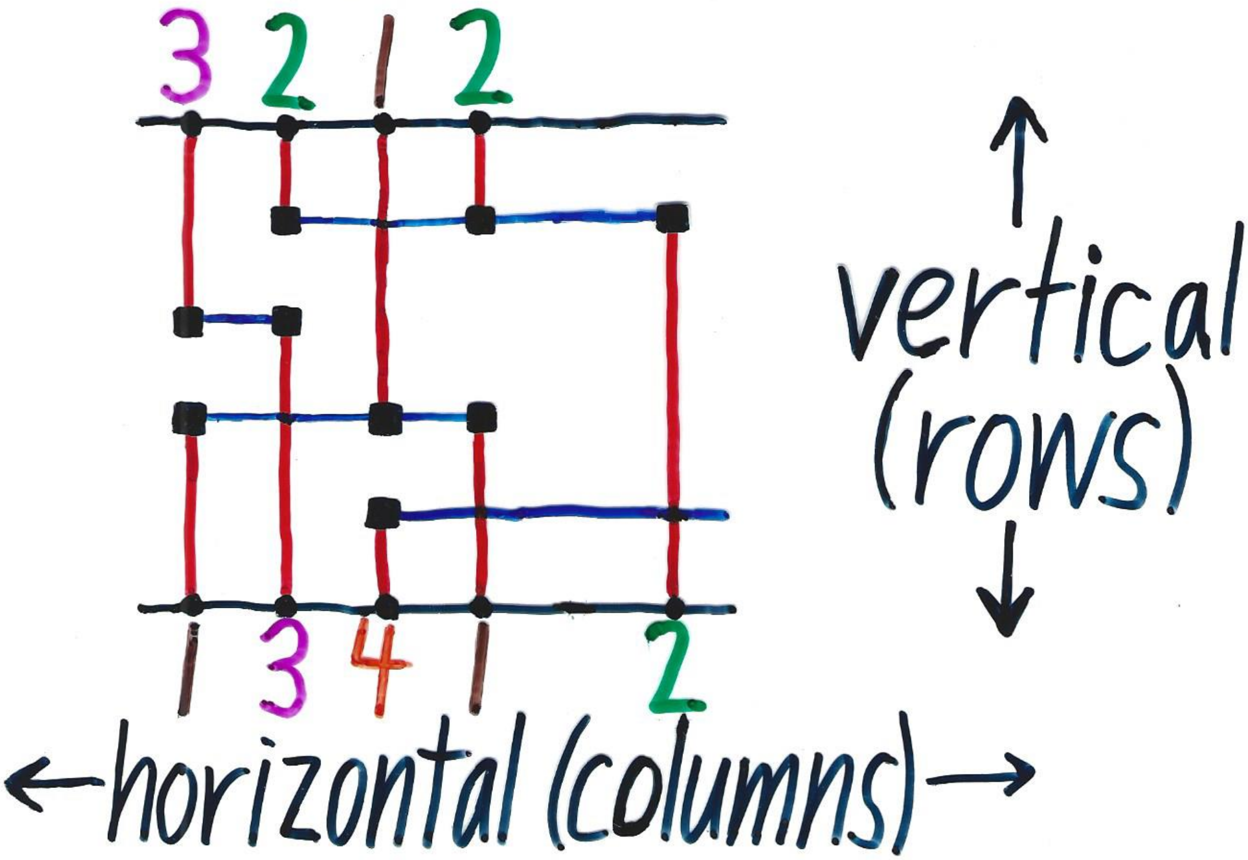
1. Background and a Problem:
   Four layers may be no better than three.

2. MulCh:
   A solution by relaxing the wiring model.

3. Potential Improvements to MulCh.

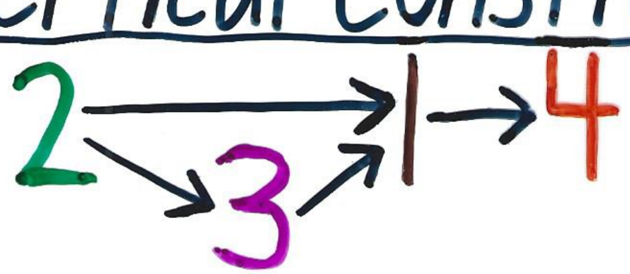
Terminology

channel:



density: $d=3$

vertical constraint graph (VCG):



Traditional Channel Routing

Two layers with preferred directions: horizontal (H) or vertical (V)

- Hashimoto & Stevens (1971)
- Deutsch (1976)
- Kawamoto & Kajitani (1979)
- Rivest & Fiduccia (1982)
- Yoshimura & Kuh (1982)
- Reed, Sangiovanni-Vincentelli, & Santomauro (1985) (YACR2)

Multi-Layer

≥3 layers, preferred directions

- Chen & Liu (1984) (3-layer) (based on Yoshimura/Kuh)
- Bruell & Sun (1984) (3-layer) (based on Rivest/Fiduccia)
- Hambrusch (1985) (n-layer) (overlap model)
- Enbody & Du (1986) (n-layer)
- Braun, Burns, Devadas, Ma, Mayaram, Romeo, & Sangiovanni-Vincentelli (1986) (n-layer) (based on YACR2) Chameleon

B layers

Unlike previous multi-layer channel routers, MulCh uses B layers as well as H & V.

B layers allow wire runs in **both** the horizontal and vertical directions

Chameleon

Divide into subproblems (groups) to be routed using 2 layers (HV) or 3 layers (HVH).

1. Pick group types (HVH or HV) (partition the layers)
2. Assign nets to groups (partition the nets)

---

3. Assign nets to rows (route horizontal segments)
4. Maze route to terminals (route "vertical" segments)

Selecting Group Types

Density lower bound (DLB) on channel width is

$$\frac{\text{density}}{\#\text{layers allowing horiz. routing}}$$

1. Use as many HVH groups as possible.
2. If two layers left over, use an HV group.
3. If one layer left over, combine with HVH group to yield two HV groups.

Assigning Nets

Consider nets sequentially. Add each net to the group with lowest resulting cost.

Cost of a group is based on the density and maximum VCG path length of the group.

Example: Assign net N

	cost	cost with net N
group 1:	5	→ 8 ←
group 2:	10	10
group 3:	7	9

Problem

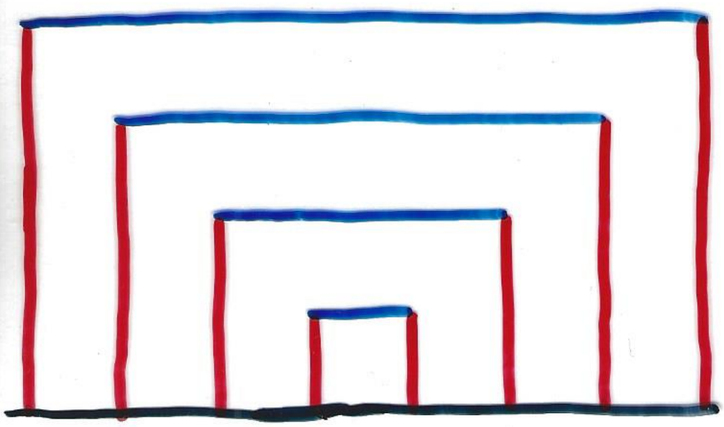
Additional layers do not always add H layers.

3 layers	⇒	4 layers
HVH		HV HV

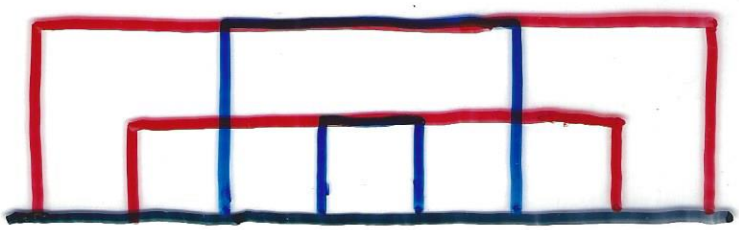
For many problems, a good channel router needs just as wide a channel using 4 layers as it does using 3! (same lower bound)

The Real Problem

Density lower bound is dependent on wiring model.



with preferred directions



without preferred directions

Why Not B Layers?

- Partitioner complexity
  Nets assigned to a B layer group must be coplanar. Choosing group types is harder.
- Router complexity
  Must find planar routings for groups using B layers.
- Questionable benefit
  Do real channels have enough planarity to justify difficulties?

What MulCh Is:

- A "sales pitch" for B layers
- Experimental
- Implemented as an extended version of Chameleon

What MulCh Is Not:

- Industrial strength
- The final word on the techniques that it uses

Assumptions

- Grid model
- Terminals can be contacted from any layer.
- Interconnect technology is uniform across all layers.

MulCh Outline

- Start with Chameleon group types & assign nets to groups.
- Repeat the following:
  1. Add more B layers.
  2. Choose other group types.
  3. Assign nets to groups.
  4. Estimate channel width.

  until width estimate increases.

Note: Expect to do no worse than Chameleon.

Assigning Nets

As in Chameleon, but with different cost function.

For each group, let

DLB = density lower bound

p = max. VCG path length for the group.

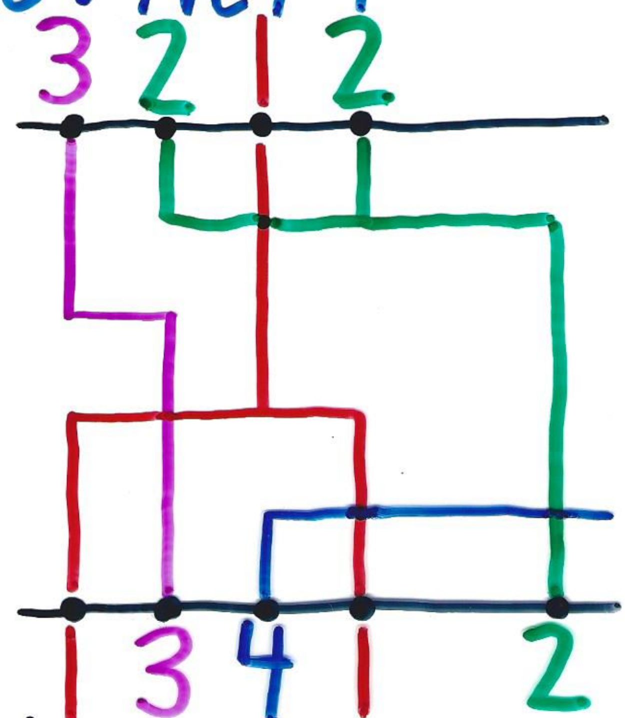
Cost for HV or HVH group is

max(DLB, p)

B Layer Cost

For B layers, penalize nets which make adding other nets difficult.

Example: Net 1



d = density

cp = crossing penalty

Cost for a B layer group is

$$d + cp$$

Width Estimates

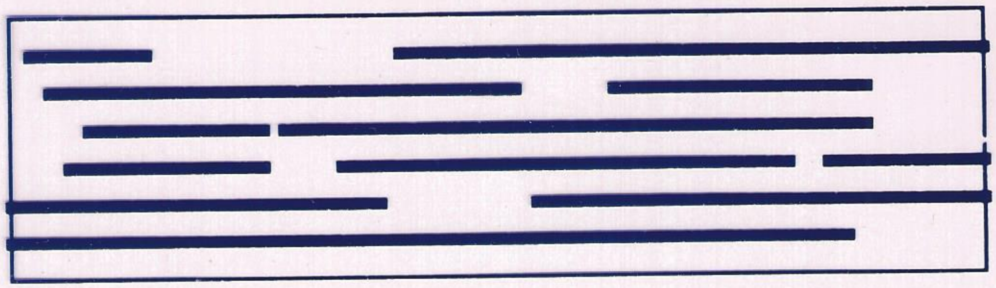
B groups:
actual routing

other groups:
DLB

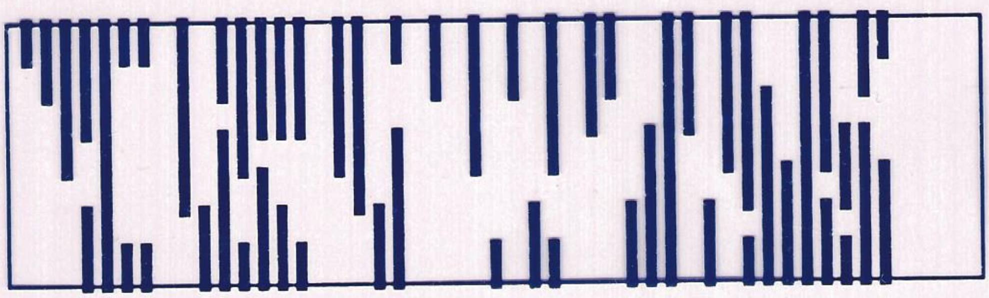
channel:
max. of estimates for group widths

Results

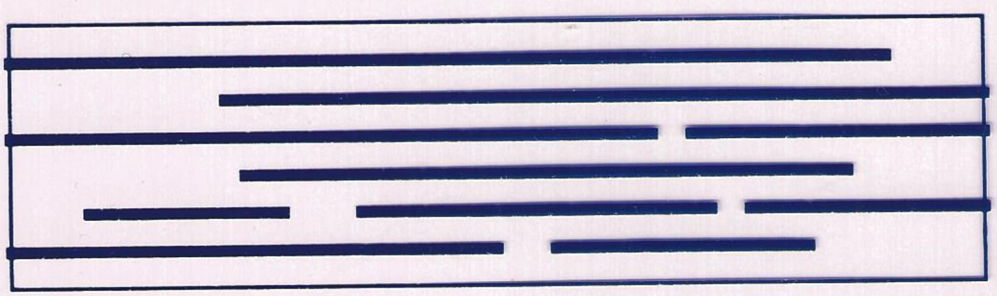
Example Channel:



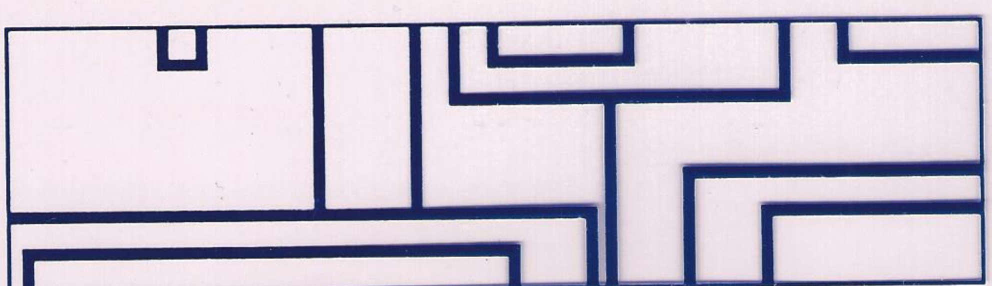
Layer1
H



Layer1
V



Layer1
H



Layer1
B

Test Cases

real channels	(5)
Yoshimura & Kuh	(3)
randomly generated	(4)
Deutsch's Difficult Example	(1)

Total of 13 example channels.

Improvements over Chameleon

Best case,
average case,
worst case

% change in

- channel width (width)
- total net length (net l.)
- total # vias (vias)

4 layers

	width	net l.	vias
best	-25%	-9.9%	-31.7%
avg.	-9.9%	-4.8%	-20.8%
worst	0%	-1.2%	-0.9%

Note:

7 cases were narrower than DLB for H-V wiring model.

9 cases were narrower than Chameleon.

3, 5, and 6 layers

no width reductions, but savings in net length and vias can be substantial.

5 layers

	width	net l.	vias
ex3	0%	-6.4%	-56.3%

7 layers

	width	net l.	vias
best	-25%	-6.1%	-32.2%
avg.	-7.4%	-3.4%	-19.2%
worst	0%	+0.9%	+0.5%

Note:

2 cases were narrower than DLB for H-V wiring model.

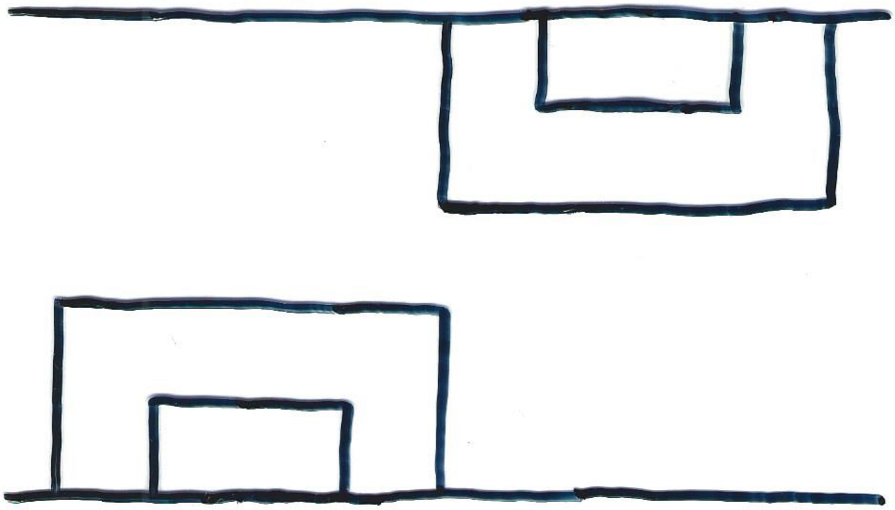
5 cases were narrower than Chameleon.

Improving MulCh

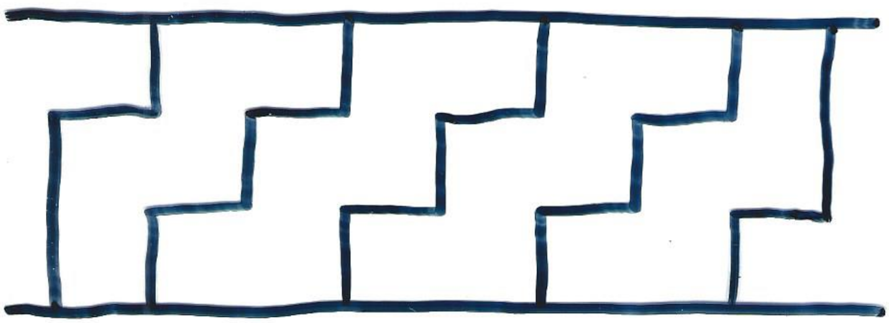
- Determine B group widths quickly

- Incremental algorithms
  - B group widths
  - density
  - VCG path length

B groups are different



density=2
width=4



p=4
width=2

Can find width in linear time!
Single-sided connections!
Assume two-point nets.

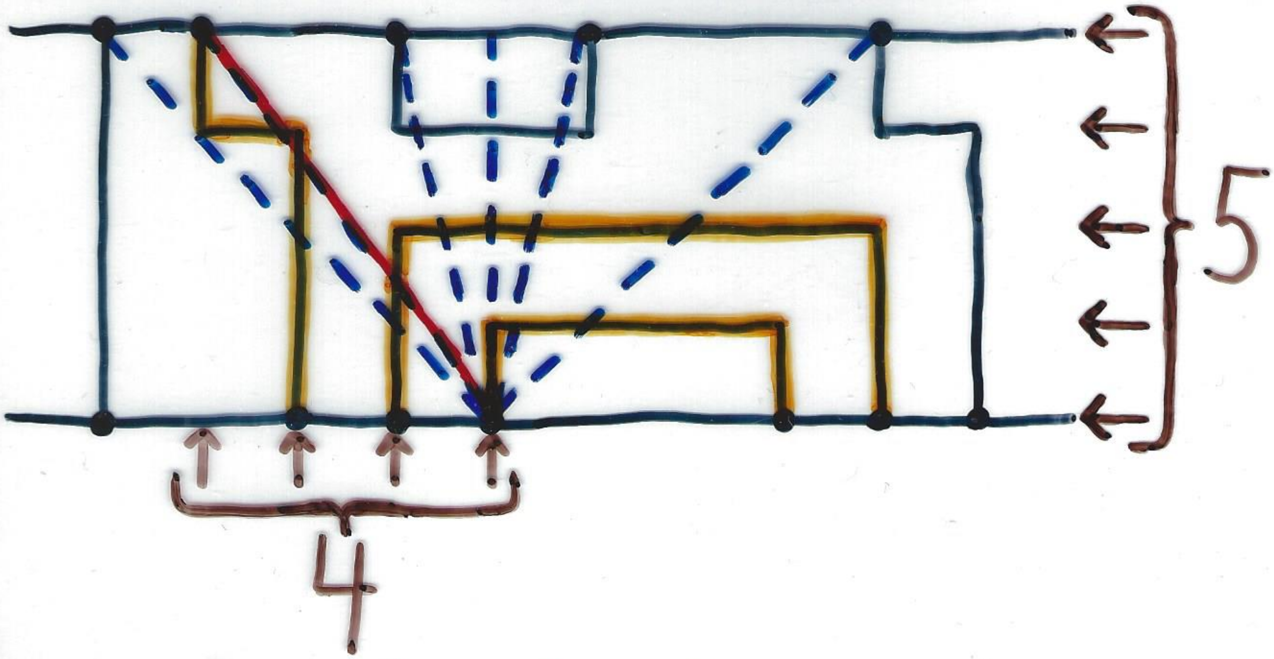
B group width

1. General theory of single-layer routing
   Cole & Siegel (1984)
   Maley (1987)

2. Remove redundant conditions

3. Find width satisfying conditions in linear time

Single-Layer Channel Routing



critical cut: from terminal to: terminal or straight across

capacity: cap(c) = max of grid lines crossing horizontally & vertically

flow: flow(c) = # nets which must cross

safe cut: $flow(c) \leq cap(c)$

Channel is routable iff all critical cuts are safe.

Sparse vs. Dense cuts

$cap_x(c)$ = # grid lines crossing vertically.

$(cap(c) = \max\{cap_x(c), cap_y(c)\})$

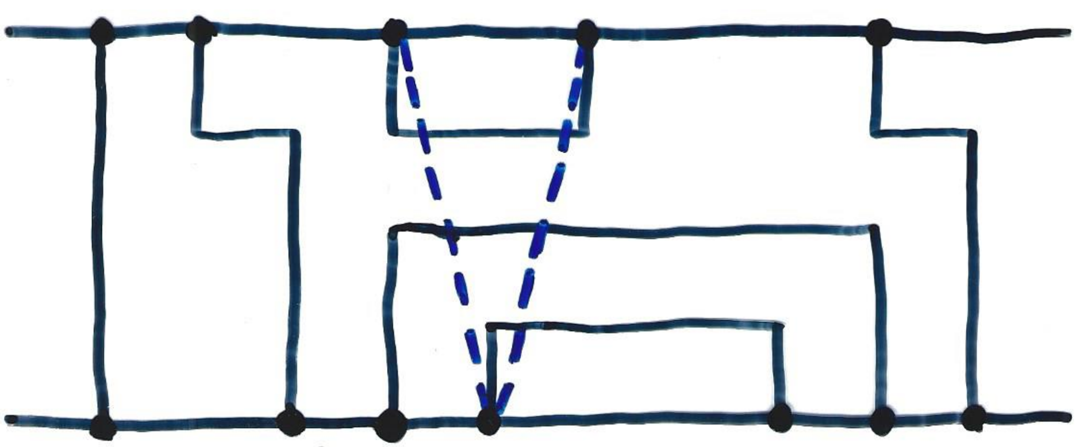
sparse cut: $flow(c) \le cap_x(c)$

dense cut: $flow(c) > cap_x(c)$

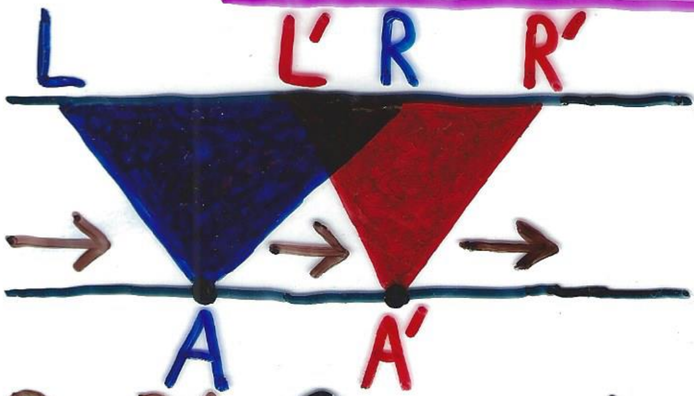
Minimum channel separation is

$$\max_{c \text{ dense}} flow(c)$$

Dense cuts form "cones"



Linear-Time sweep



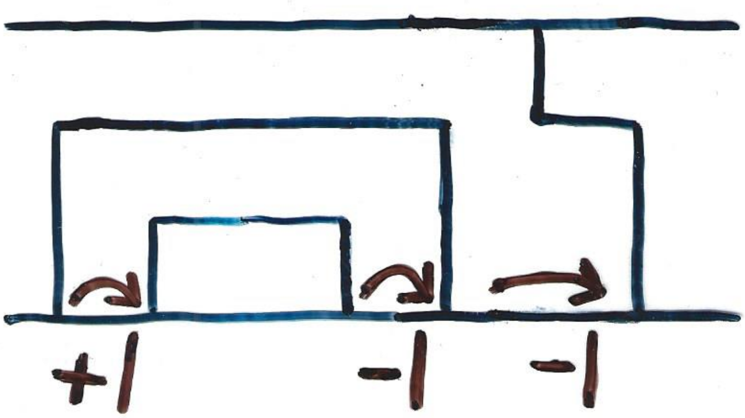
R→R′: Compute new flows differing by at most 1 from preceding flow

L→L′: Remove flows from collection

A→A′: Change flow offset by at most 1

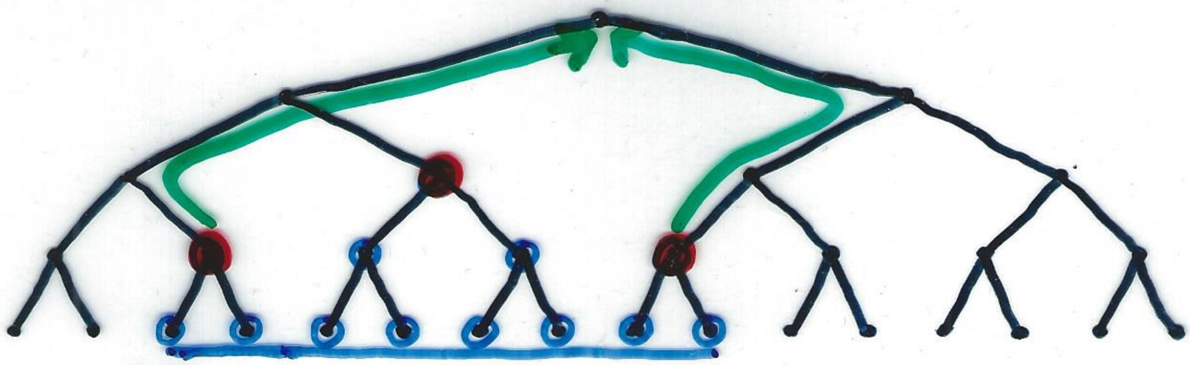
	flow	cuts
max→	3	p, q
	2	r
	1	s

offset: 2



Incremental Density

static case: tree on known columns



$d_{below}(v)$ = max. density of nets covering (at or) below v but not parent

$d_{at}(v)$ = # nets covering v but not parent

$d = d_{below}(root)$

$d_{at}(v) \leftarrow d_{at}(v) + 1$ at $\leq 2 \lg N$ nodes

$d_{below}(v) \leftarrow \max\{d_{below}(left(v)), d_{below}(right(v))\} + d_{at}(v)$ at $\leq 4 \lg N$ nodes