Advances in PCB Routing

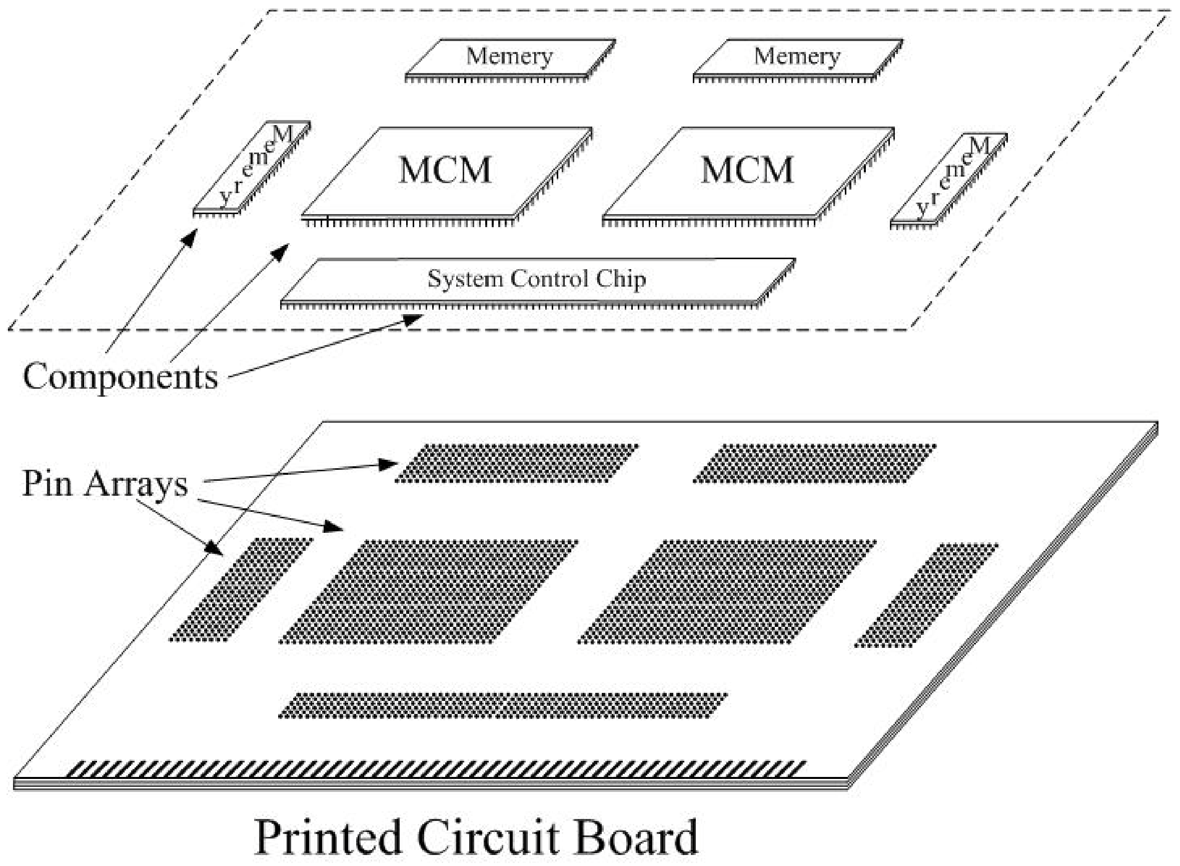
Martin D.F. Wong

Dept of Electrical and Computer Engineering

University of Illinois at Urbana-Champaign

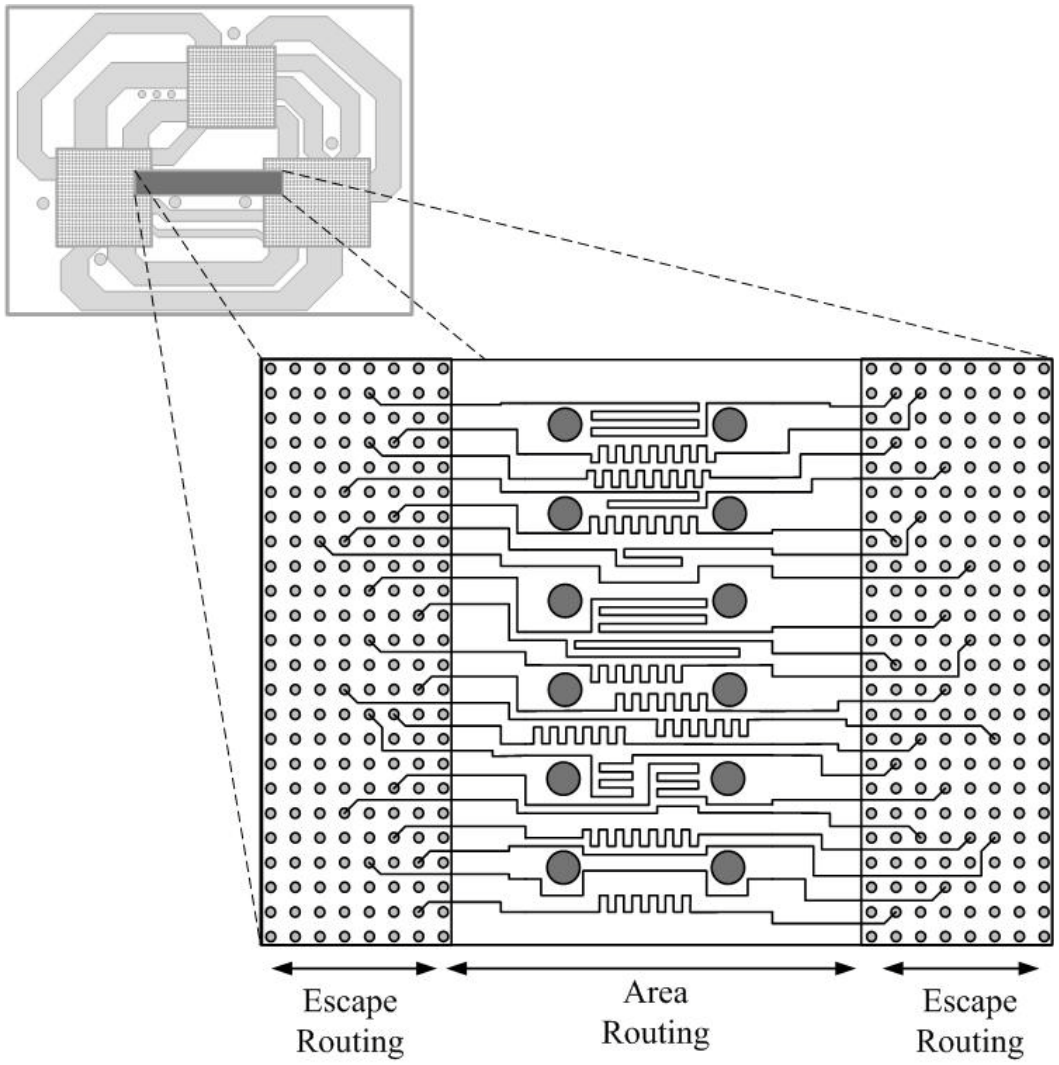
SLIP-2010

Printed Circuit Board (PCB)



- Components plug in or mounted on to PCB
- Each component corresponds to a pin array on the board
- Multiple routing layers

PCB Routing



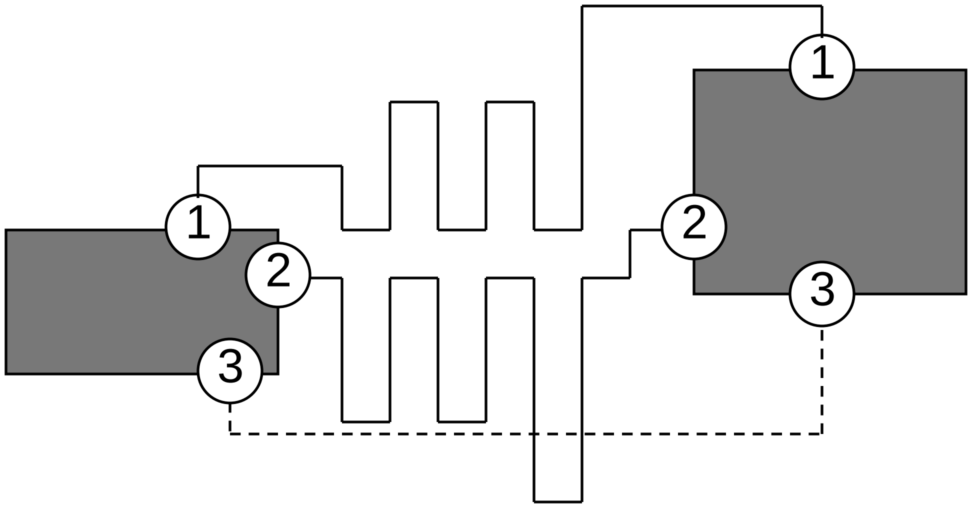
- Planar routing on each layer
- Escape routing
  - Pin to boundary
  - Satisfying constraints
- Area routing
  - Between boundaries
  - Length matching

Length-Matching Routing

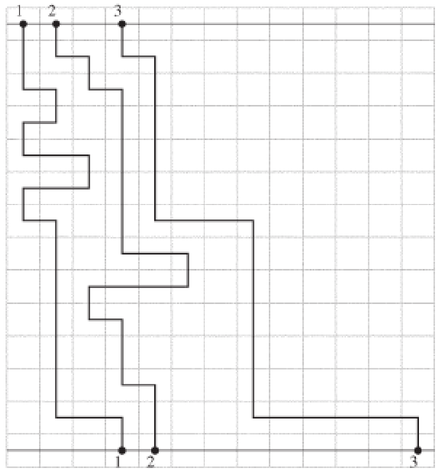
T. Yan & M. Wong, ICCAD-2008

Length-matching routing in PCB

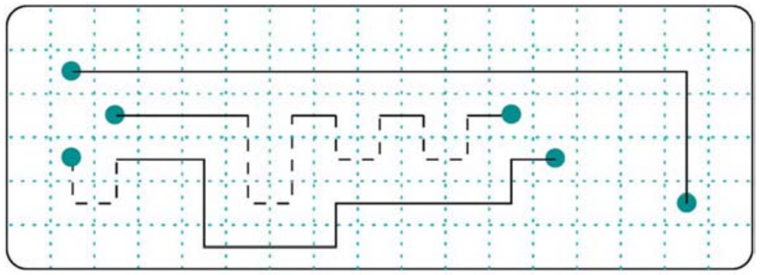
- In high frequency boards, the timing requirements on wires are very tight
- Most wires are assigned min-max length bounds
- Difficult due to the competition for resources



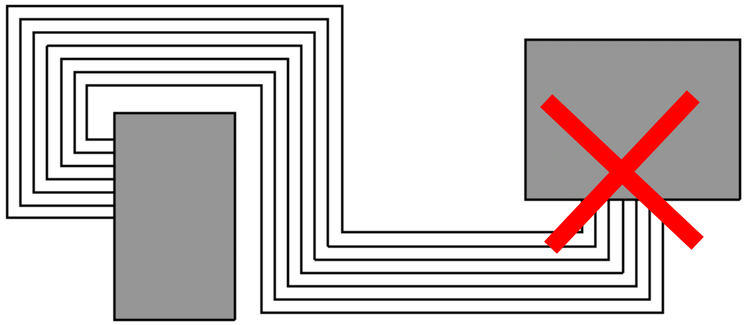
Previous Works



Min-max river routing



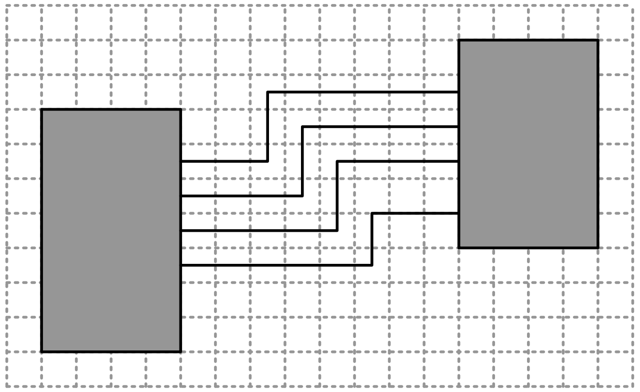
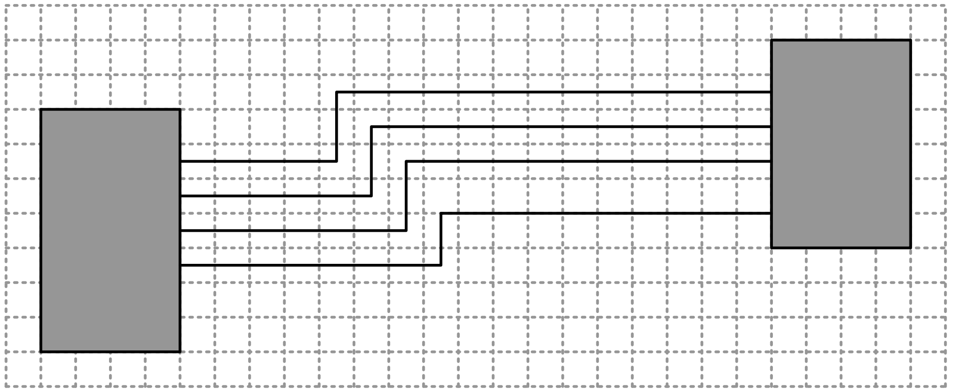
LR based monotonic routing



General topology

They are all gridded

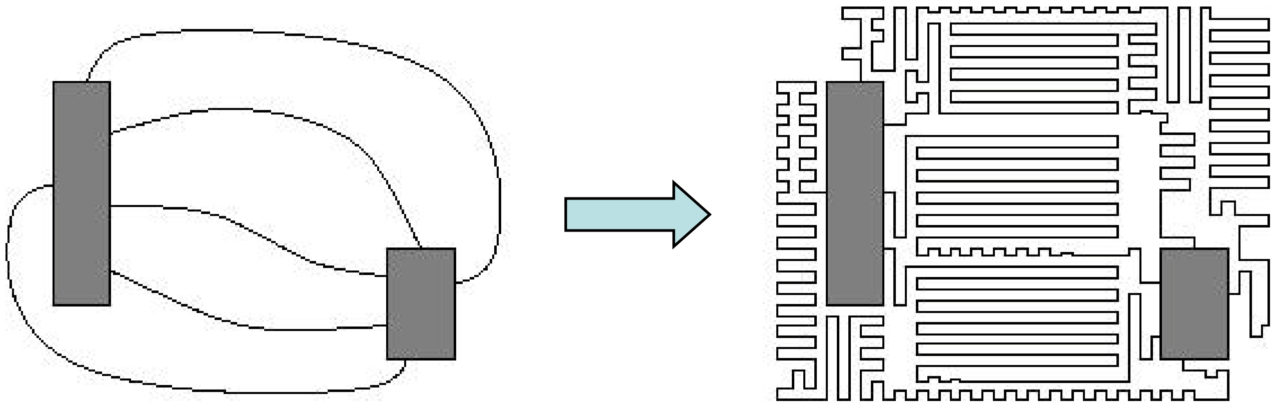
- Major drawback: problem size determined by physical distance, not routing difficulty

Same topology, but different problem size for gridded router

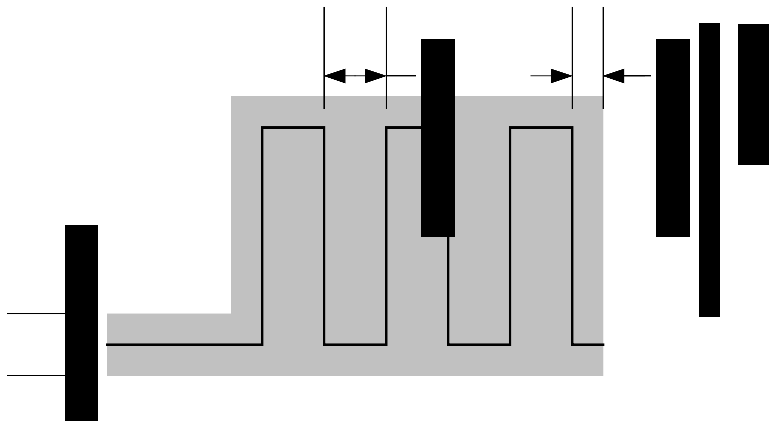
Problem definition

- Input:
  - Two components and a set of nets connecting them (net ordering on the boundary guarantees planarity)
  - Design rules
  - Length bounds for the nets
- Output
  - Rectilinear routing that satisfies design rule and length bounds



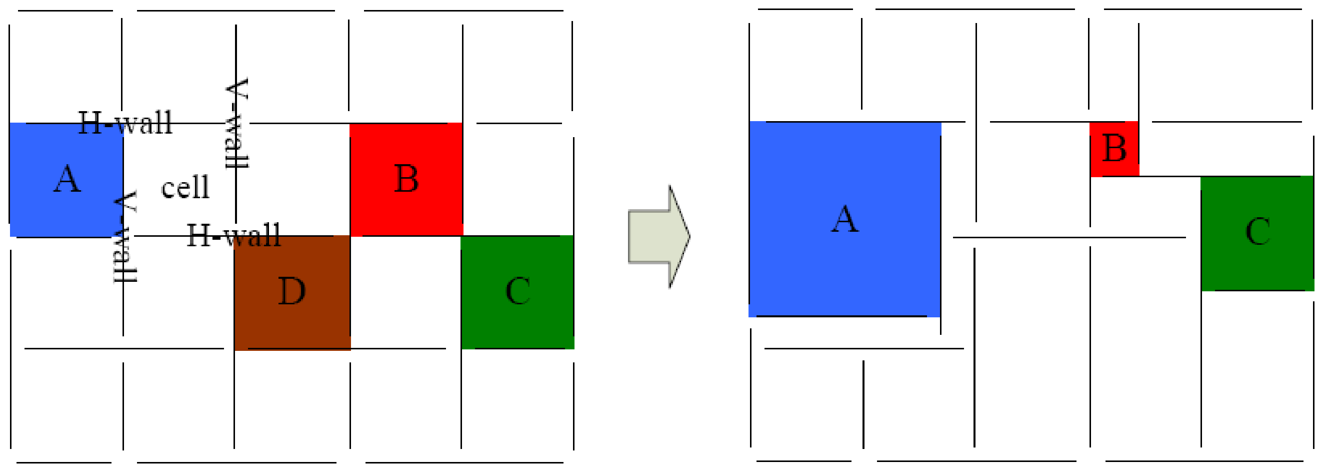
Length-matching = Area-matching

- If we consider the wire as a fat wire
  width (ε) = wire_width + separation
- Then the length of the wire is proportional to the area it occupies
- Instead of control the length, we control the area each wire occupies.

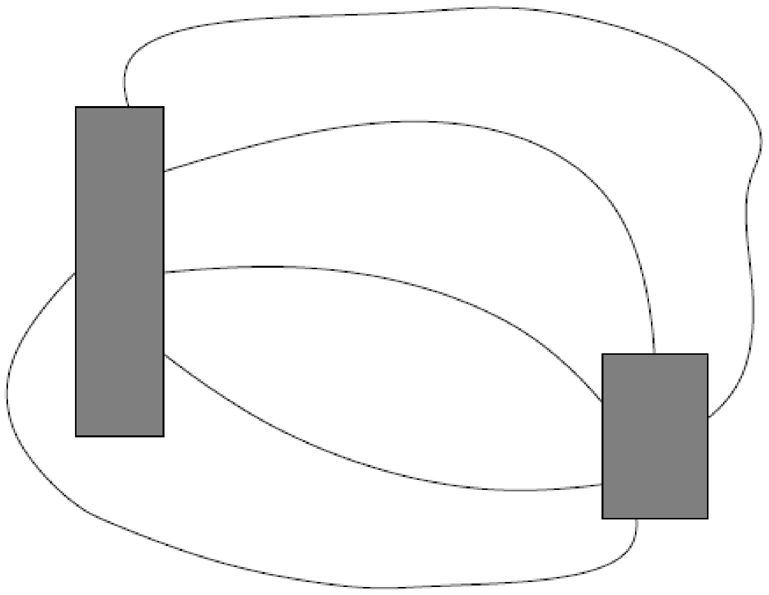


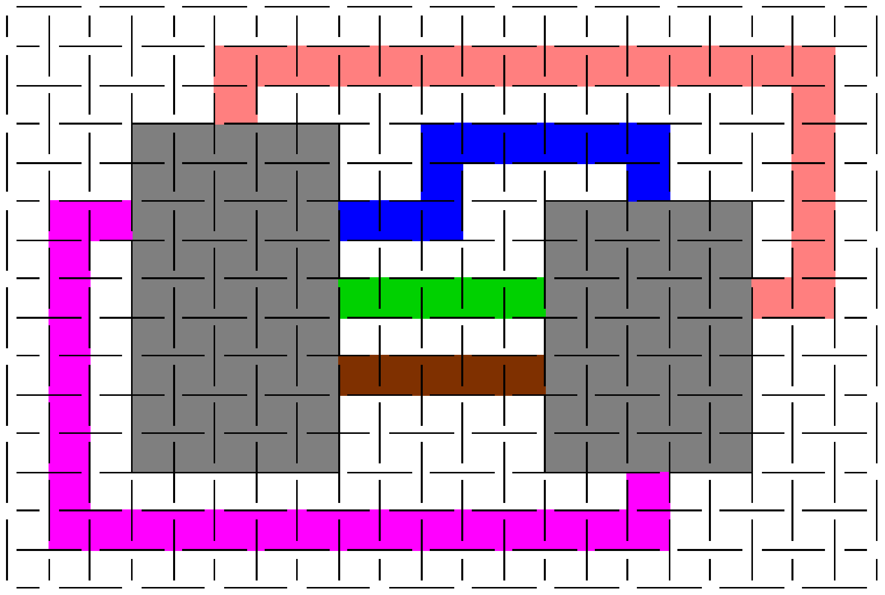
Bounded-Sliceline Grid

- Originally proposed for placement
- Dissect the plane into cells by short segments (walls)
- By moving the segments, we can enlarge, shrink and move cells

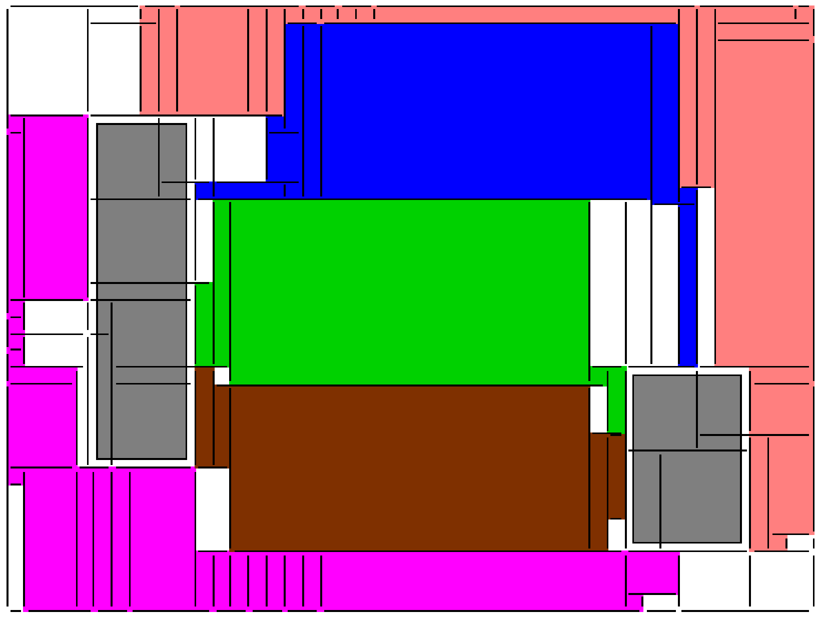


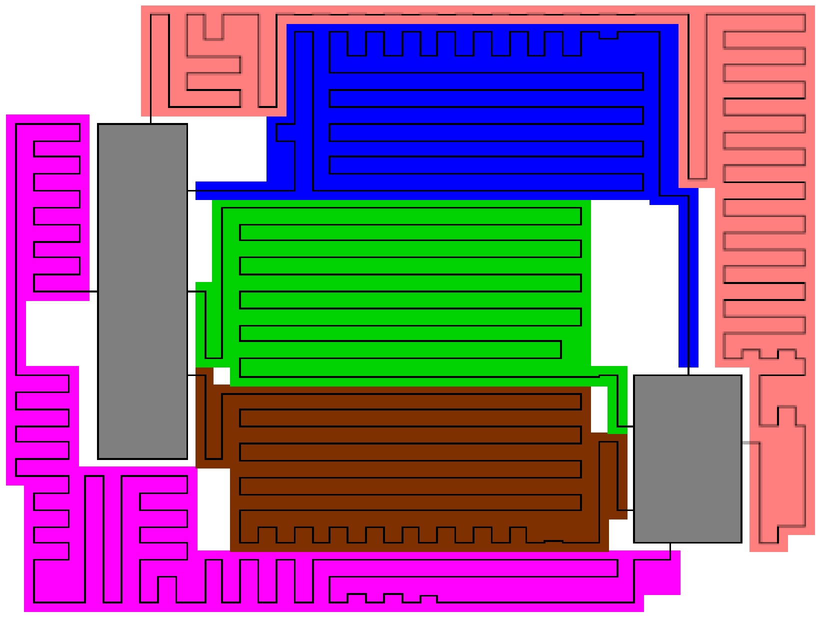
Workflow of our router


(a) Input Topology


(b) BSG Embedding

Workflow of our router (cont')

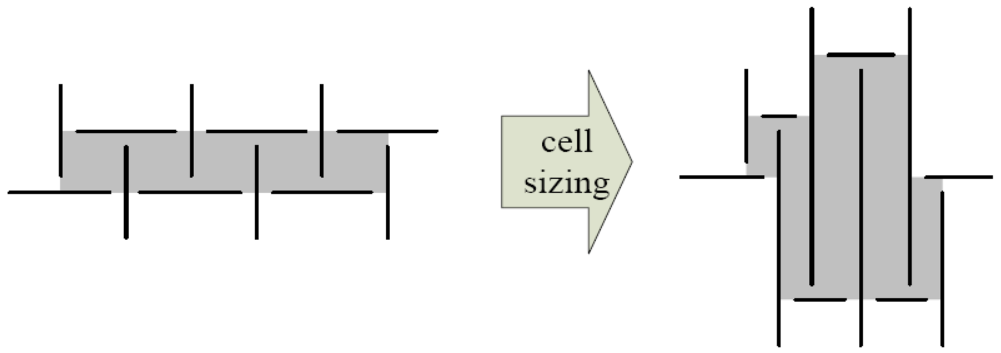

(c) Cell Sizing


(d) Detail Routing

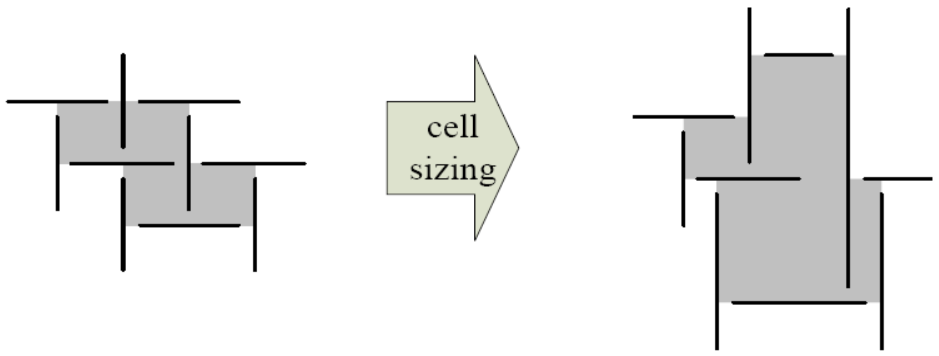
BSG embedding

- Can be done by heuristics like maze router
- Need to follow some guidelines:
  - Allow empty rooms between adjacent nets
  - Use proper number of BSG cells
  - Keep the topological relationships between components and pins

BSG embedding



(a) One BSG embedding and its cell sizing result



(b) Another BSG embedding and its cell sizing result

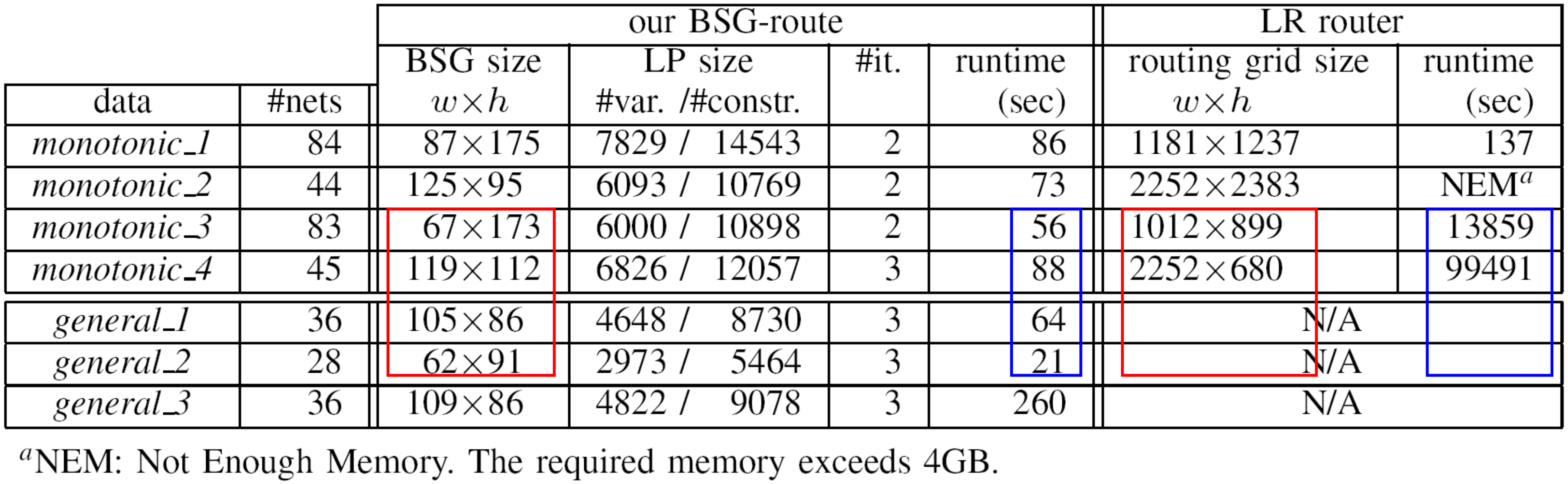
Key step: Cell sizing

- Need to size the cells so that the following constraints are satisfied:
  - Design rule
  - Component and pin location
  - Length(area) bounds for each net
- We formulate this problem as a mathematical programming problem

Experimental results

- Compare with Ozdal & Wong TCAD’06 LR-based router
- Tested on 7 data:
  - monotonic_1*, monotonic_2*, monotonic_3, monotonic_4
  - general_1*, general_2, general_3

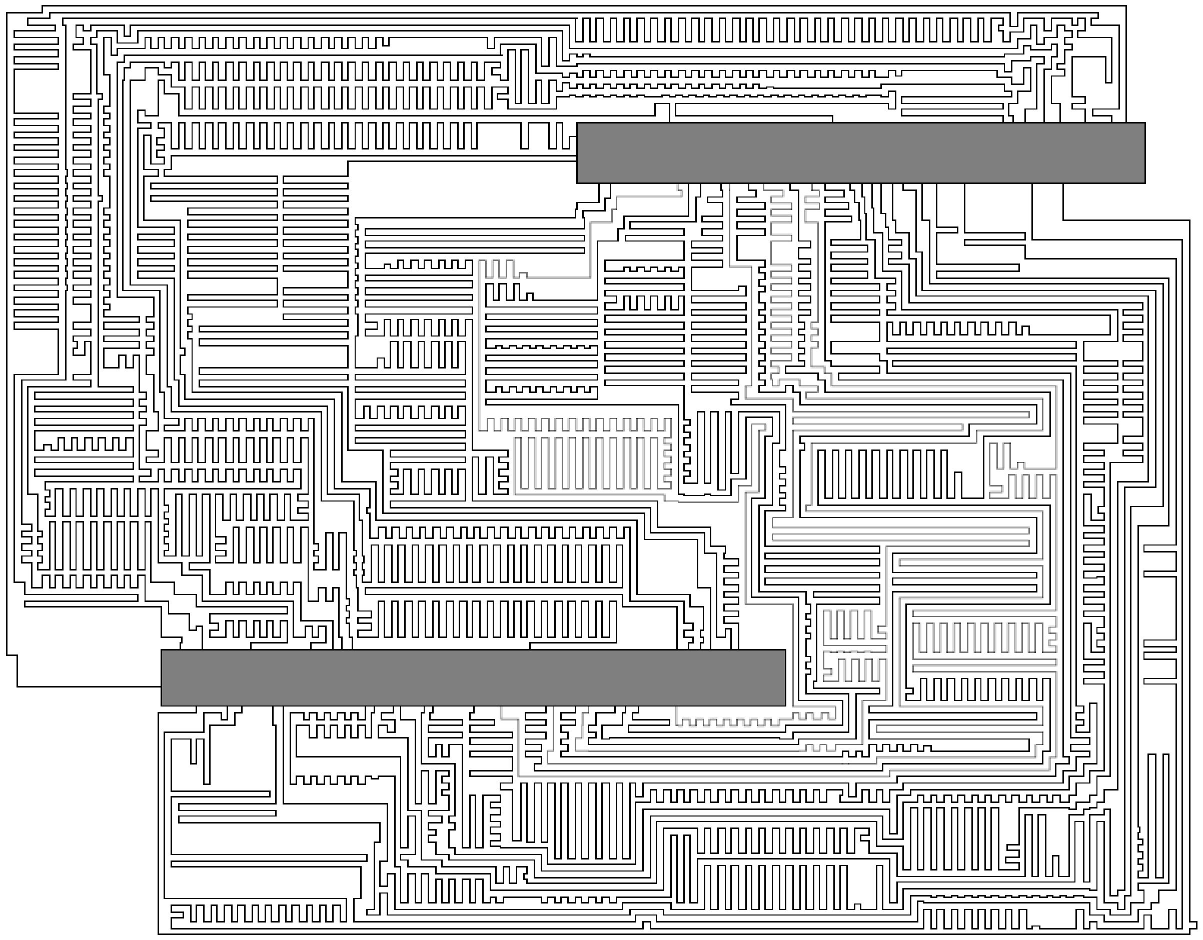
Expeirmental results



		our BSG-route				LR router	
		BSG size	LP size	#it.	runtime	routing grid size	runtime
data	#nets	$w \times h$	#var. /#constr.		(sec)	$w \times h$	(sec)
monotonic_1	84	87×175	7829 / 14543	2	86	1181×1237	137
monotonic_2	44	125×95	6093 / 10769	2	73	2252×2383	NEM[a]
monotonic_3	83	67×173	6000 / 10898	2	56	1012×899	13859
monotonic_4	45	119×112	6826 / 12057	3	88	2252×680	99491
general_1	36	105×86	4648 / 8730	3	64	N/A	
general_2	28	62×91	2973 / 5464	3	21	N/A	
general_3	36	109×86	4822 / 9078	3	260	N/A	

[a]NEM: Not Enough Memory. The required memory exceeds 4GB.

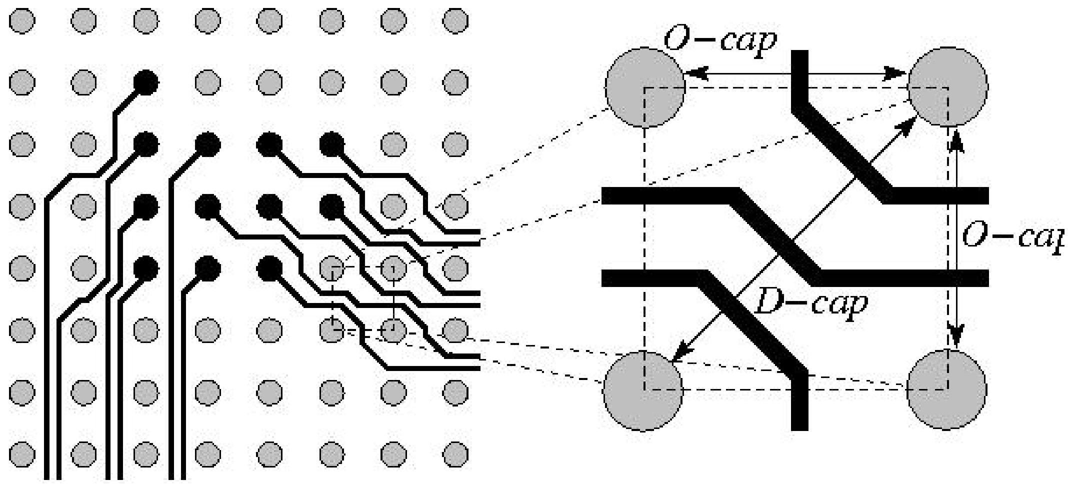
An example (general_3)



Escape Routing

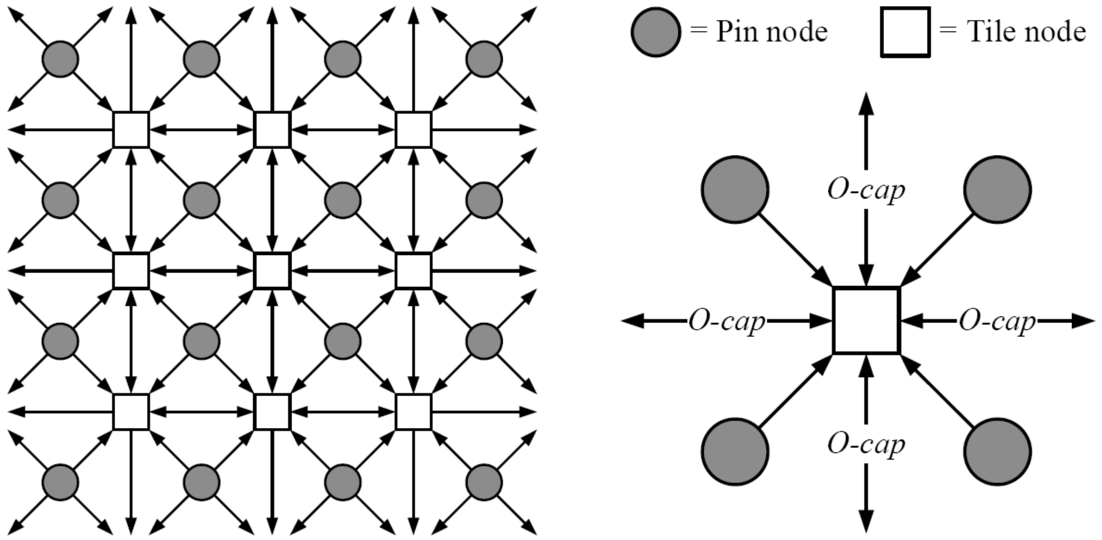
T. Yan and M. Wong, DAC 2009

Escape Routing



- Route (black) pins to the boundary of the pin grid array
- The grid has Orthogonal and Diagonal wiring capacity (*O-cap* and *D-cap*)

Traditional network-flow model

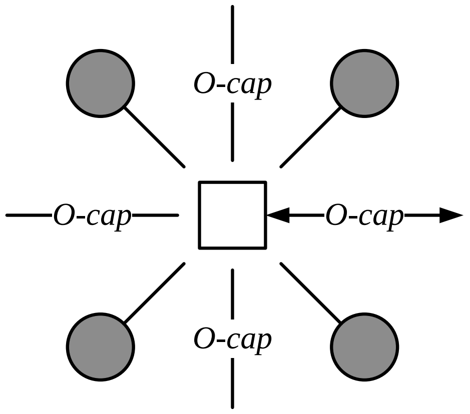


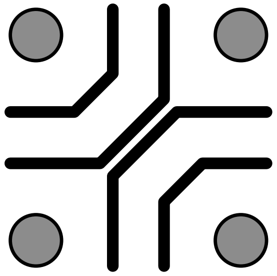
a) Flow model for a pin grid (b) Inside a tile

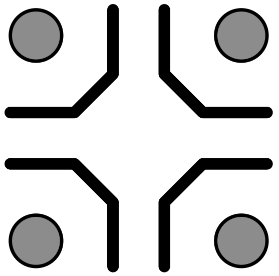
- *O-cap* can be guaranteed, but *D-cap* is not reflected
- Some works assign node capacity to tile node, but it still does not reflect *D-cap* correctly

When traditional model fails

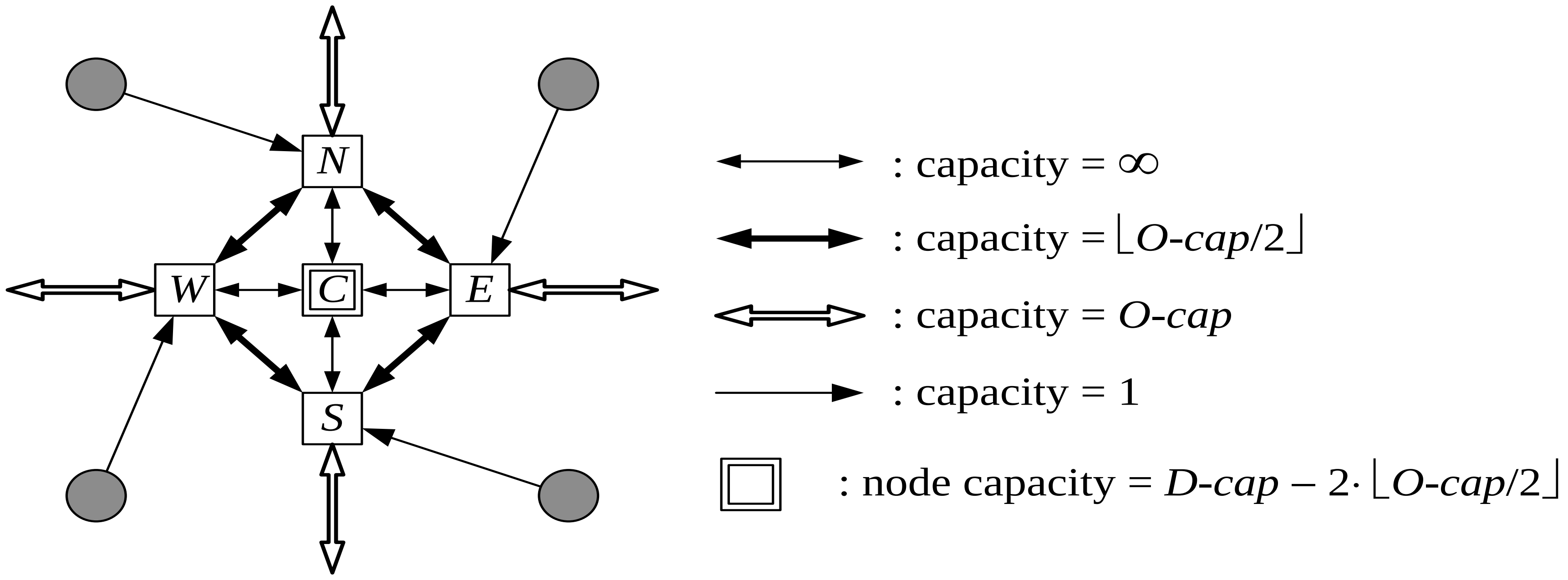
- Assume *O-cap* = 2 and *D-cap* = 3
- No constraints on tile node leads to illegal routing (a)
- Let tile node capacity = 3 or less misses the legal routing case (b)
- Traditional network-flow model is not capable of capturing diagonal capacity



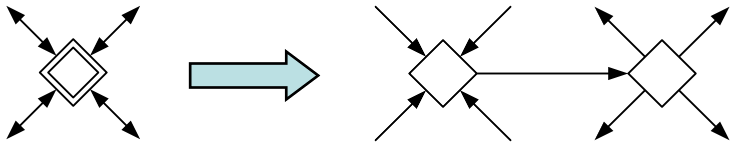

(a)


(b)

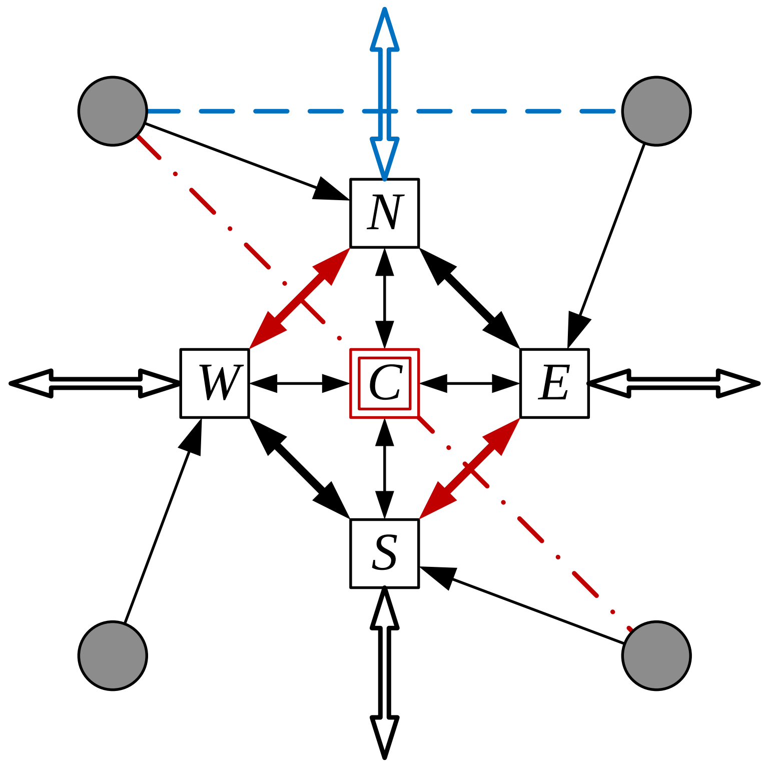
Our Network Flow Model



Node capacity implementation



Capture *O-cap* and *D-cap*



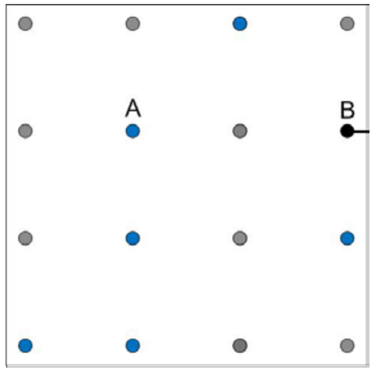
An orthogonal cut cuts only one hallow edge, which has capacity exactly *O-cap*

An diagonal cut cuts two solid edges and center node. The sum of their capacities is

$2\times\lfloor O\text{-}cap/2\rfloor+(D\text{-}cap - 2\times\lfloor O\text{-}cap/2\rfloor)$
$= D\text{-}cap$

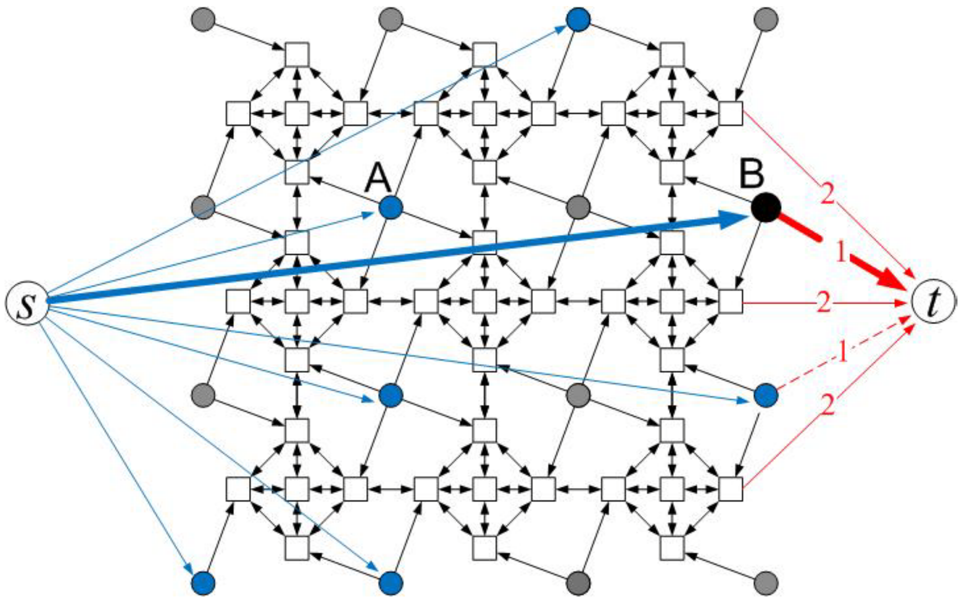
The Entire Flow Network

Optimal single-layer escape routing!



Pin Grid

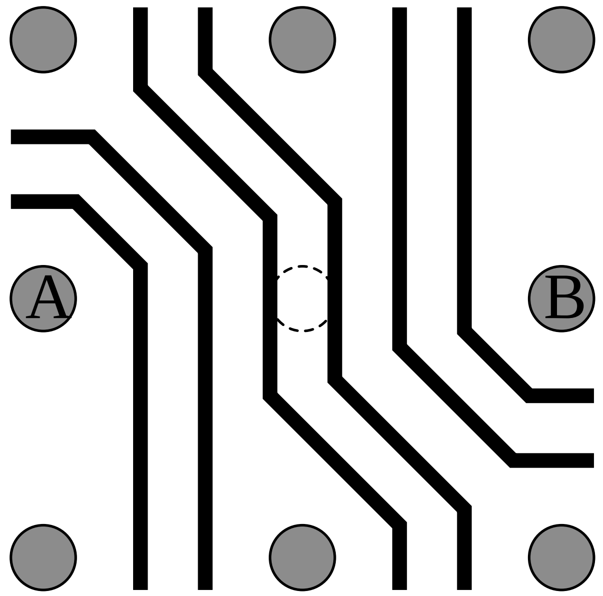
(O=2, D=3)



Network

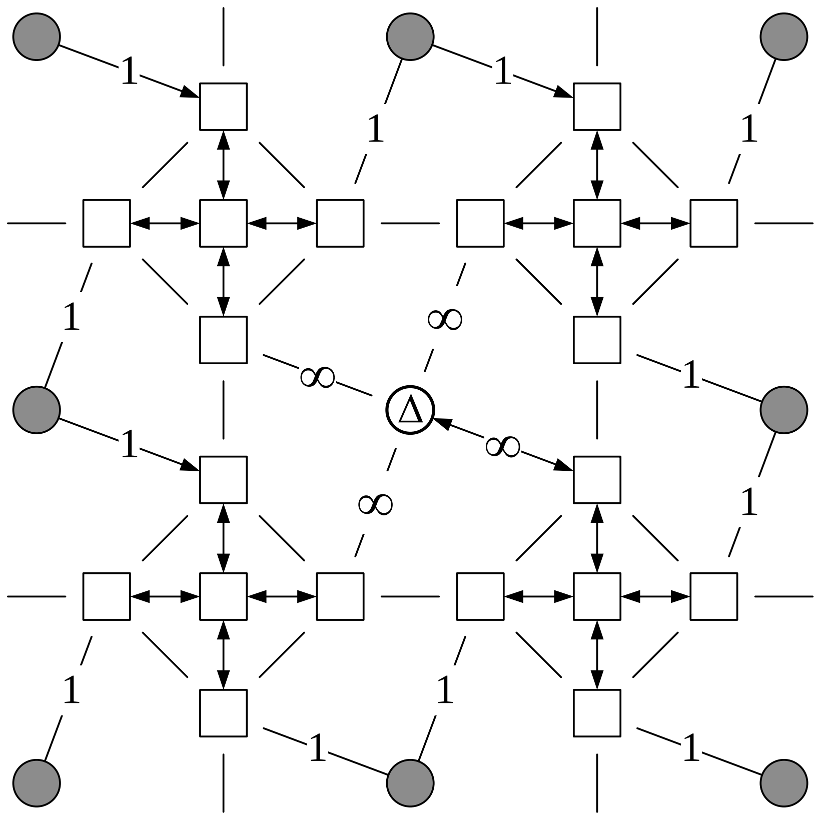
Missing pin

- In real designs, some pins in the array maybe removed. This leads to extra routing resource



The wiring capacity between A and B increases from 4 to 6 due to the missing pin. We call the difference, 2, *extra capacity* and denote it as Δ

Consider missing pins in our model



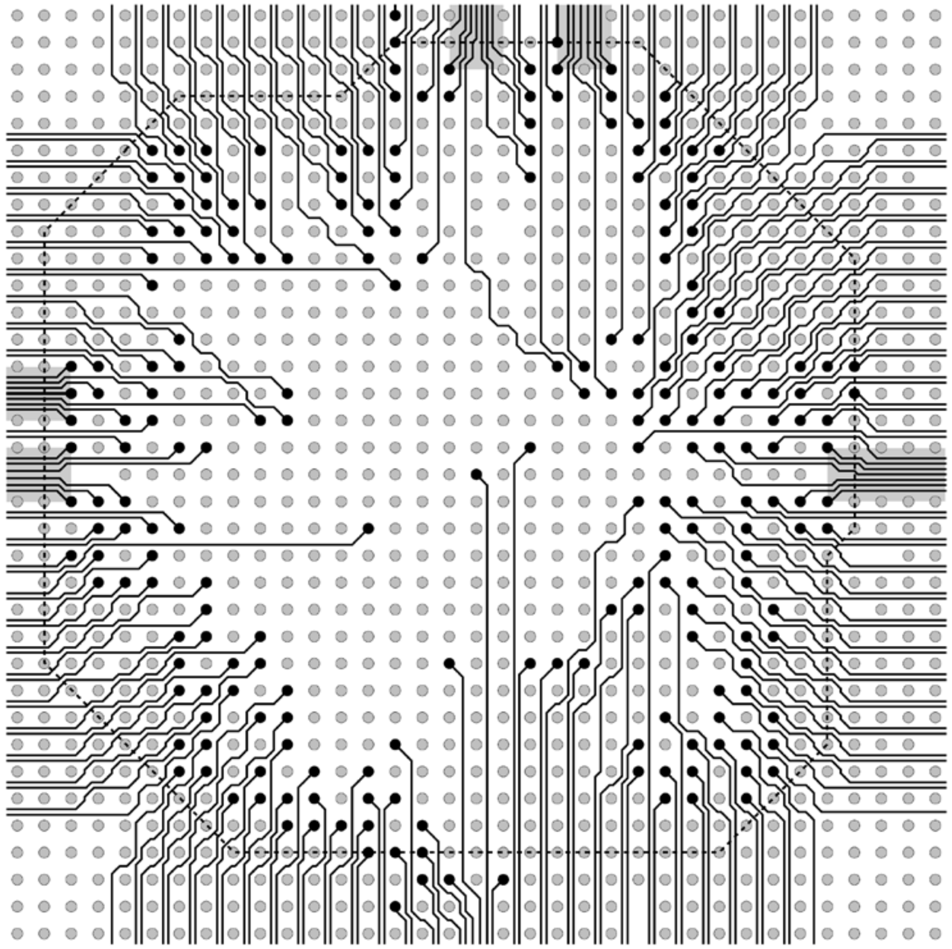
- Consider the missing pin as a resource node
- The capacity of the resource node is exactly the extra capacity

Experimental results

- Tested on industrial data
- Results indicate that our model has zero *D-cap* violation for all data while traditional model has violations
- Though our model is more complicated, the runtime are comparable

	array	escape	missing	capacities			our model		traditional model	
	$w \times h$	pin no.	pin no.	O	D	Δ	*D-cap* vio.	runtime	*D-cap* vio.	runtime
industrial_1	14×16	78	13	2	3	3	0	0.16 s	0	0.14 s
industrial_2	29×11	66	20	2	3	3	0	0.22 s	10	0.17 s
industrial_3	33×14	120	46	2	3	3	0	0.33 s	6	0.28 s
industrial_4	35×17	160	30	2	3	3	0	0.61 s	9	0.28 s
industrial_5	35×35	108	105	2	3	3	0	0.86 s	1	0.68 s
industrial_6	35×17	143	38	2	3	3	0	0.39 s	0	0.30 s
industrial_7	35×35	220	106	2	3	3	0	1.01 s	53	0.79 s
modified_8	35×35	225	33	2	3	3	0	0.99 s	53	0.79 s

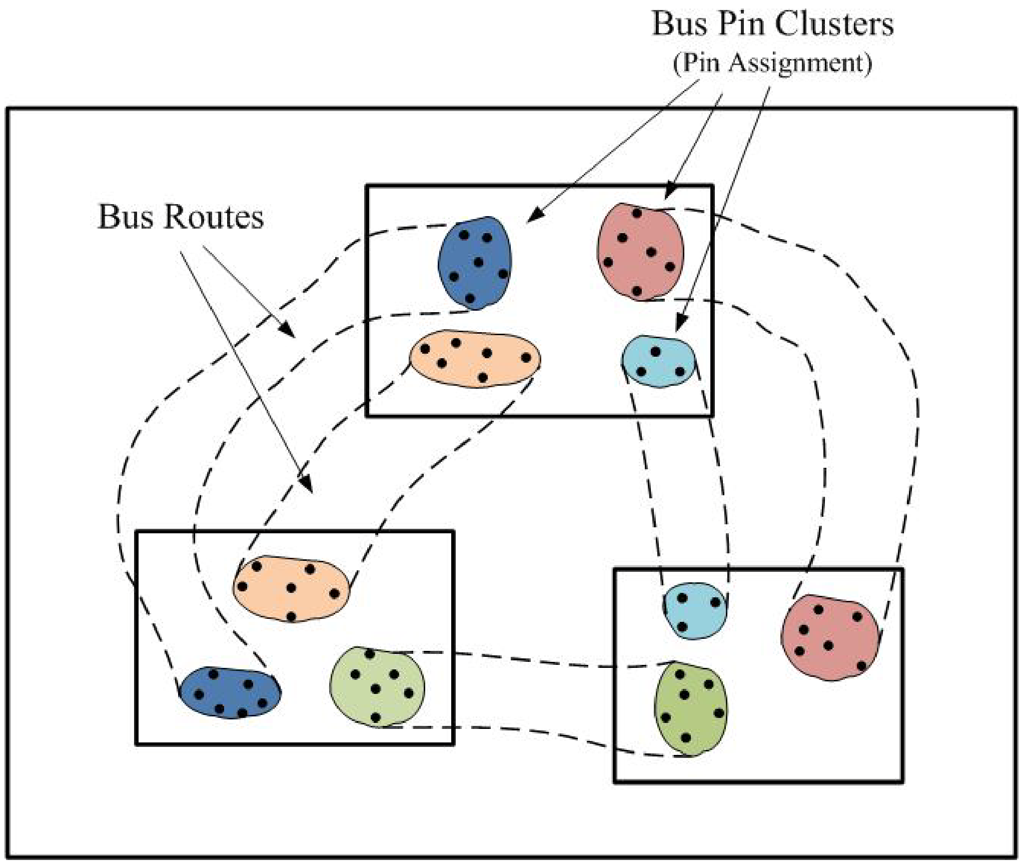
A sample result



Pin Assignment

H. Kong, T. Yan and M. Wong, ASP-DAC 2010

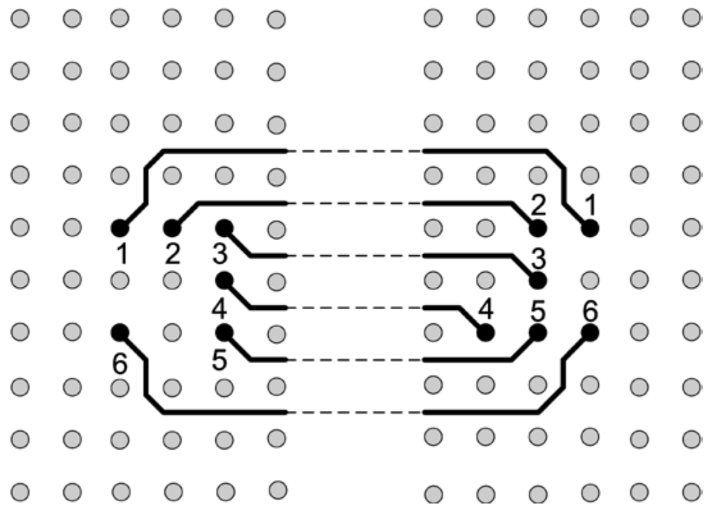
Pin Assignment for Buses



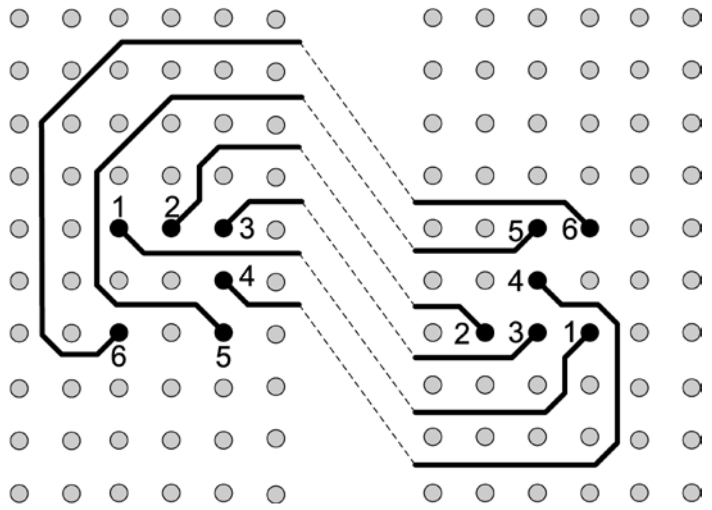
One routing layer

Pin Assignment Problem

- Assign signals (nets) to Pins
- Significant influence on later routing
- Existing algorithms
  - Based on heuristic metrics to estimate routability
  - No routability guarantee



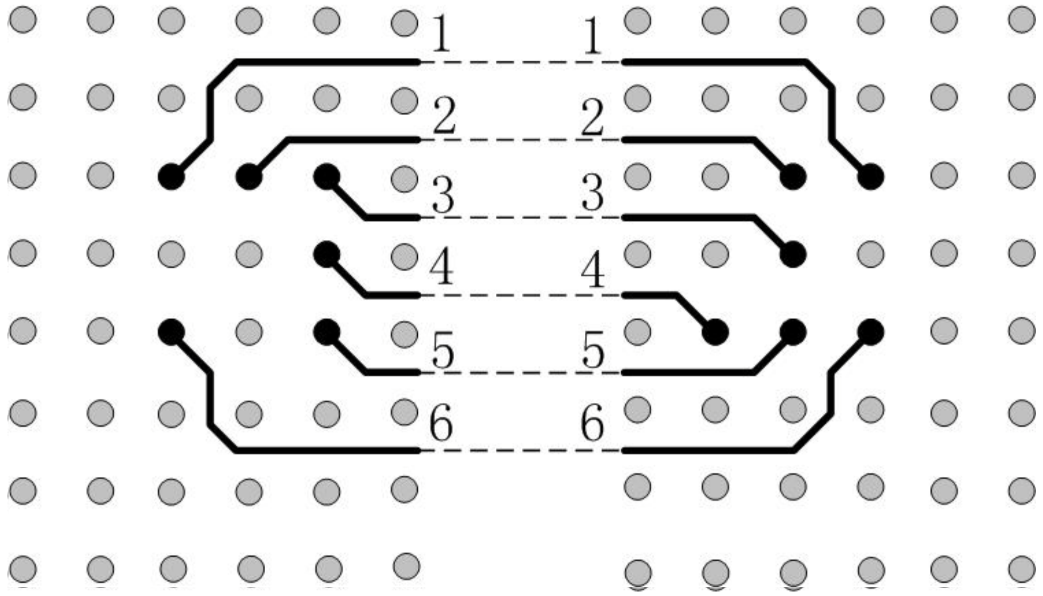
Simple routing solution



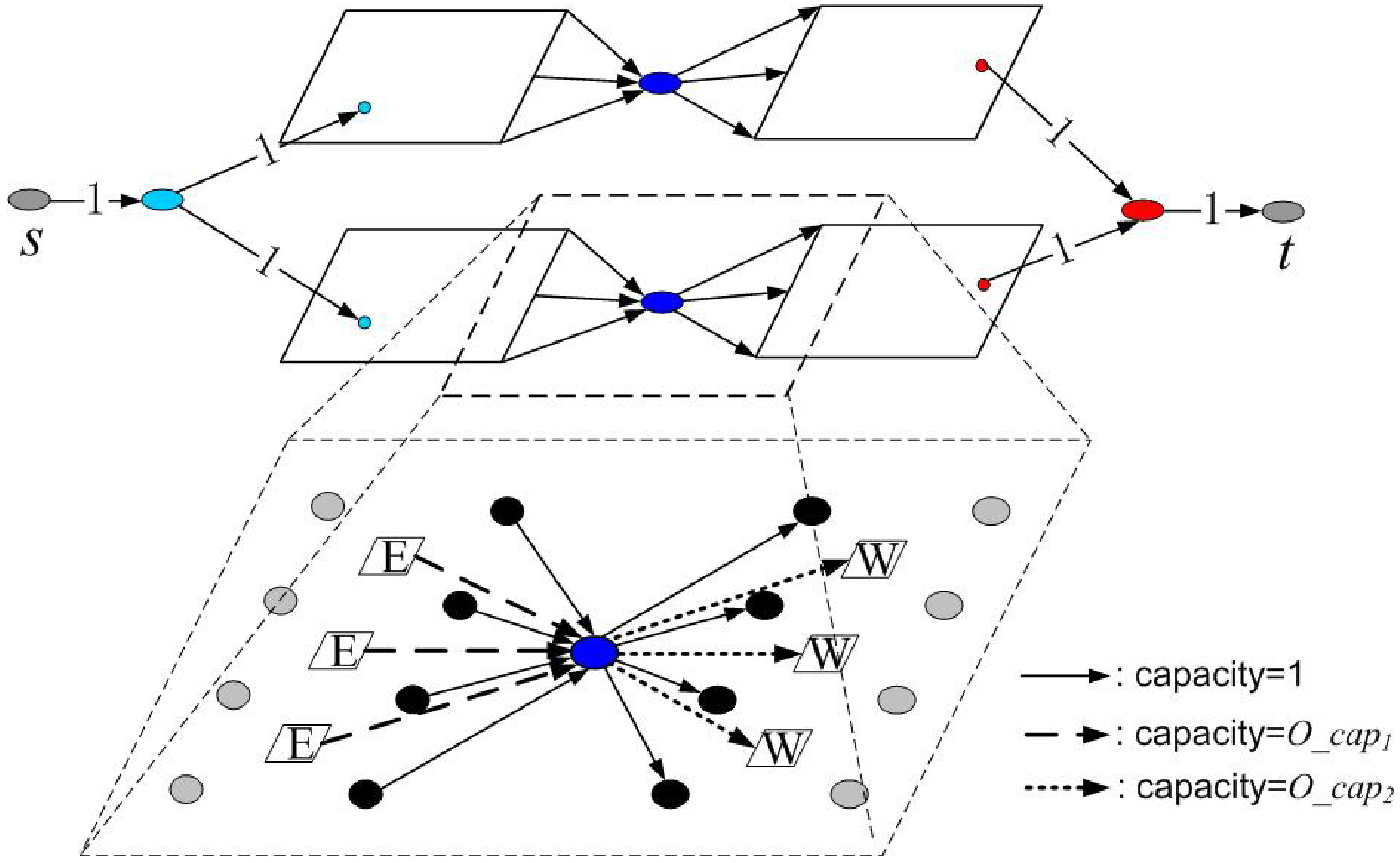
Complex routing solution

Single-Layer Pin Assignment and Routing

- Solve two *independent* escape routing problems
- Assign pins to nets by sweeping pin escape positions
- Optimal pin assignment and routing

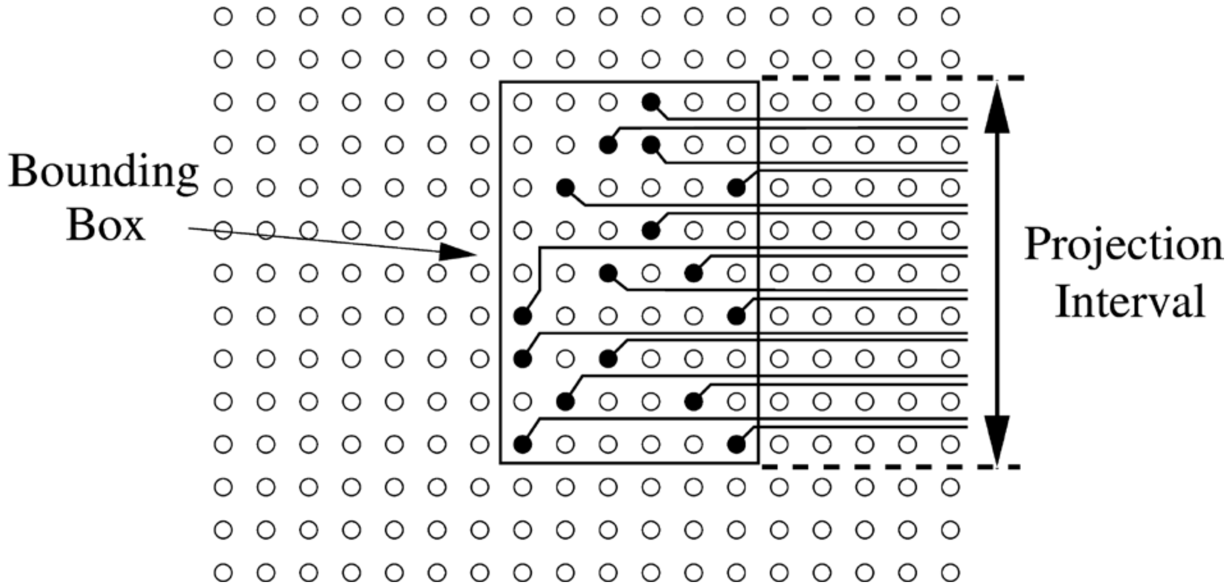


Multi-Layer Pin Assignment and Routing



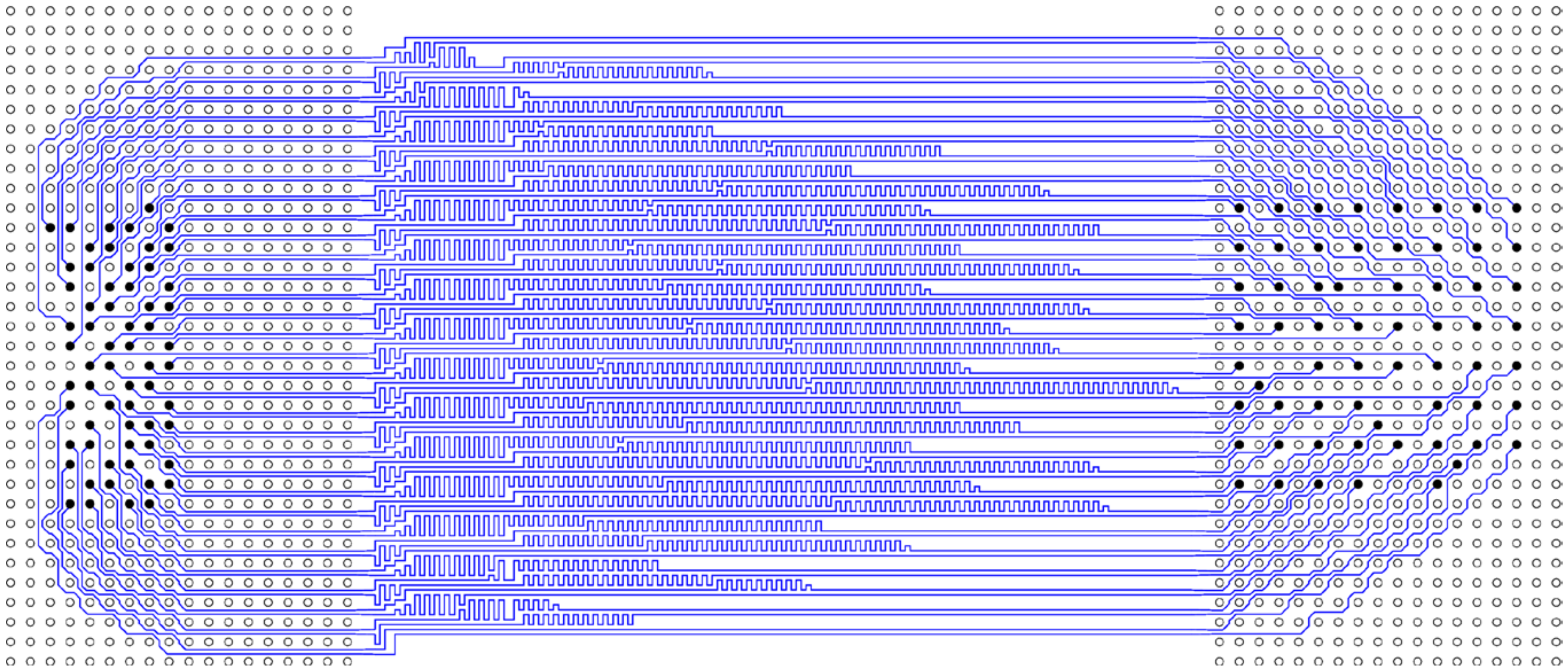
Projection Style Escape

- **Projection Interval**: Project the bounding box to the component boundary.
- Nets escape the component from its bus projection interval
- Need multiple layers
- Different pin/layer assignments affect routing resource utilization



Experimental Results

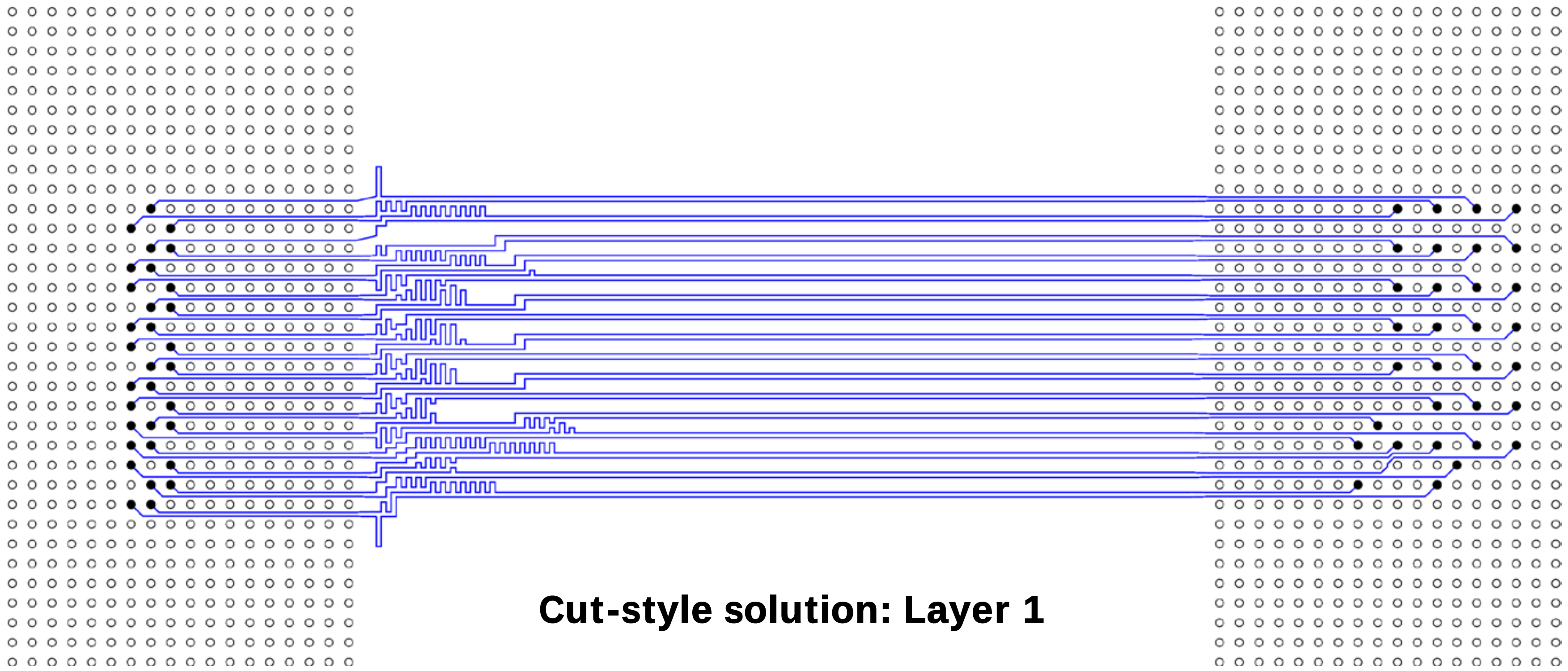
- Equal-length routing
- Back detours in the left component
- Complex length extension between component boundaries



Single-layer solution

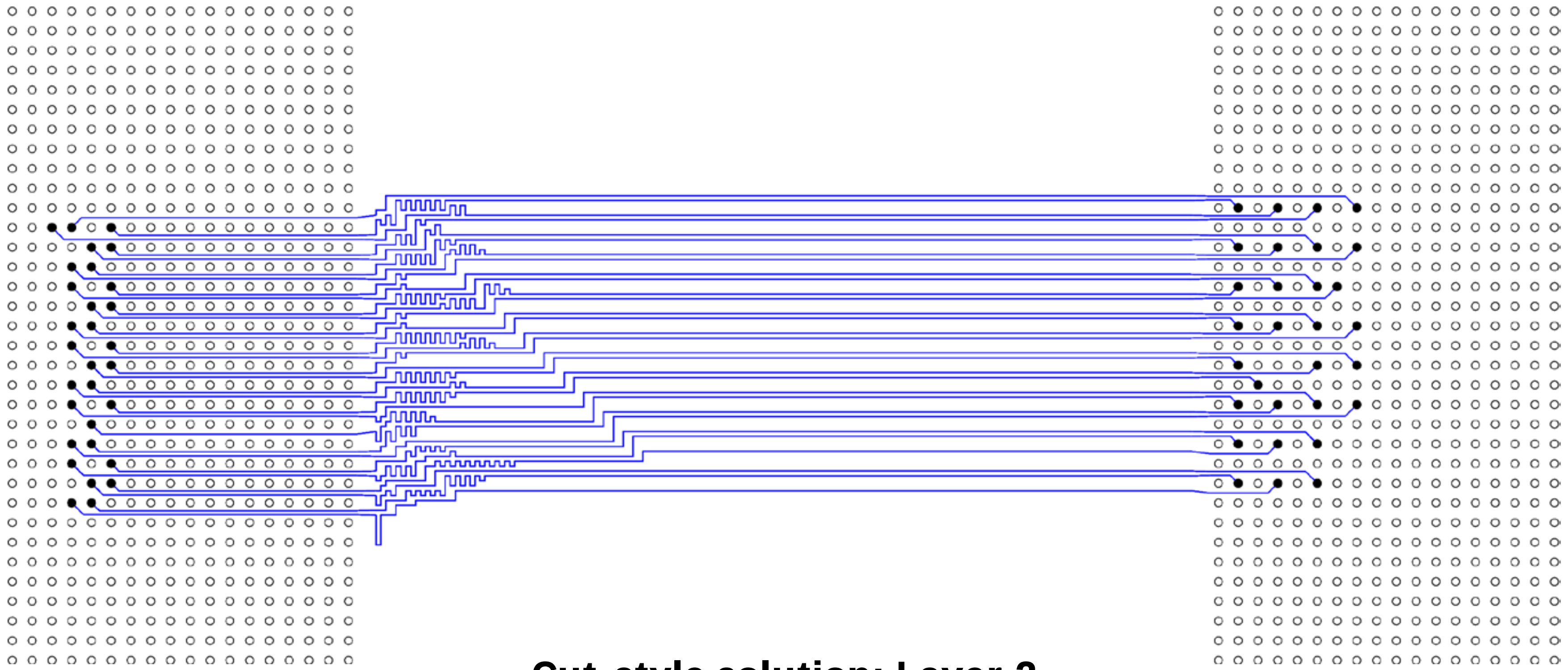
Experimental Results

- Equal-length solution
- No back detours
- Simple length extension between component boundaries
  - 37.15% shorter total length
  - 21.36% shorter escape wire length
  - 44.96% shorter detailed wire length



Cut-style solution: Layer 1

Experimental Results



Cut-style solution: Layer 2

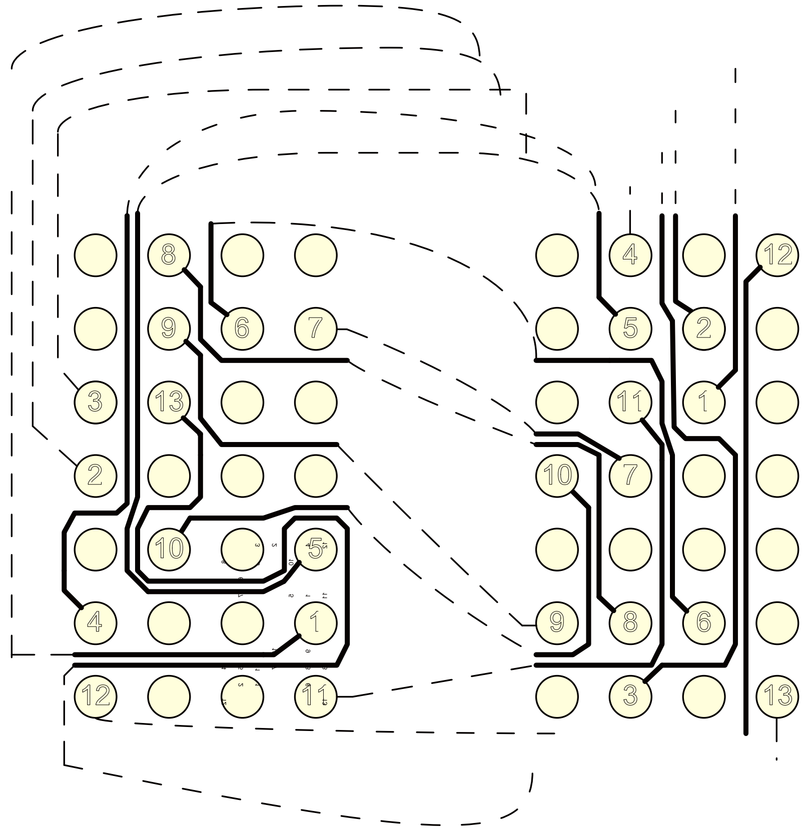
Experimental Results

- State-of-the-art industrial PCB
  - 7000+ nets
  - 80 buses
  - 12 routing layers
  - Previously routed manually
    - manual routing typically takes 2 months per board
- Pin assignment and escape routing results
  - All 80 buses takes less than 5 minutes
  - The largest bus
    - 338 nets
    - Takes 6 routing layers

Simultaneous Escape

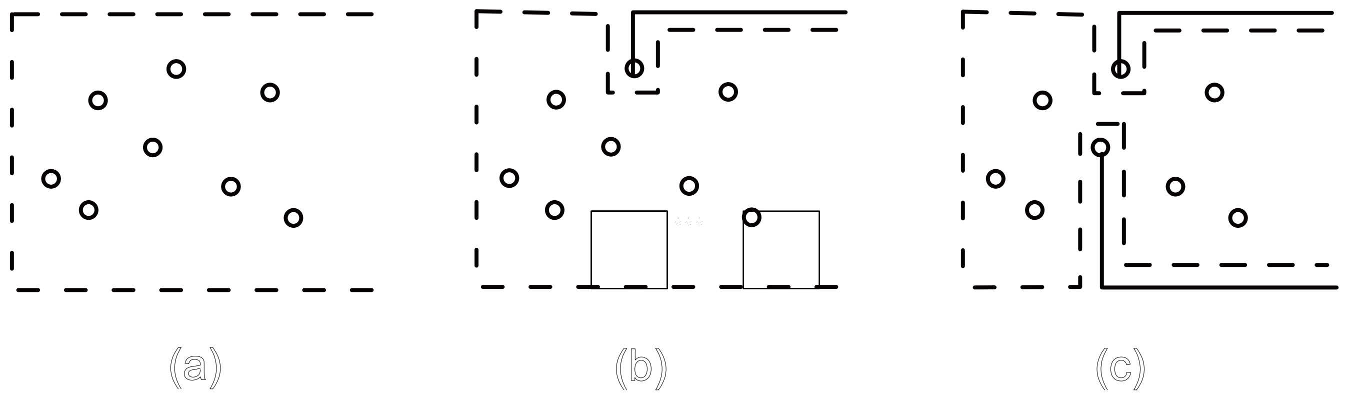
Lijuan Luo et al, ISPD-2010

Simultaneous Escape



Approach

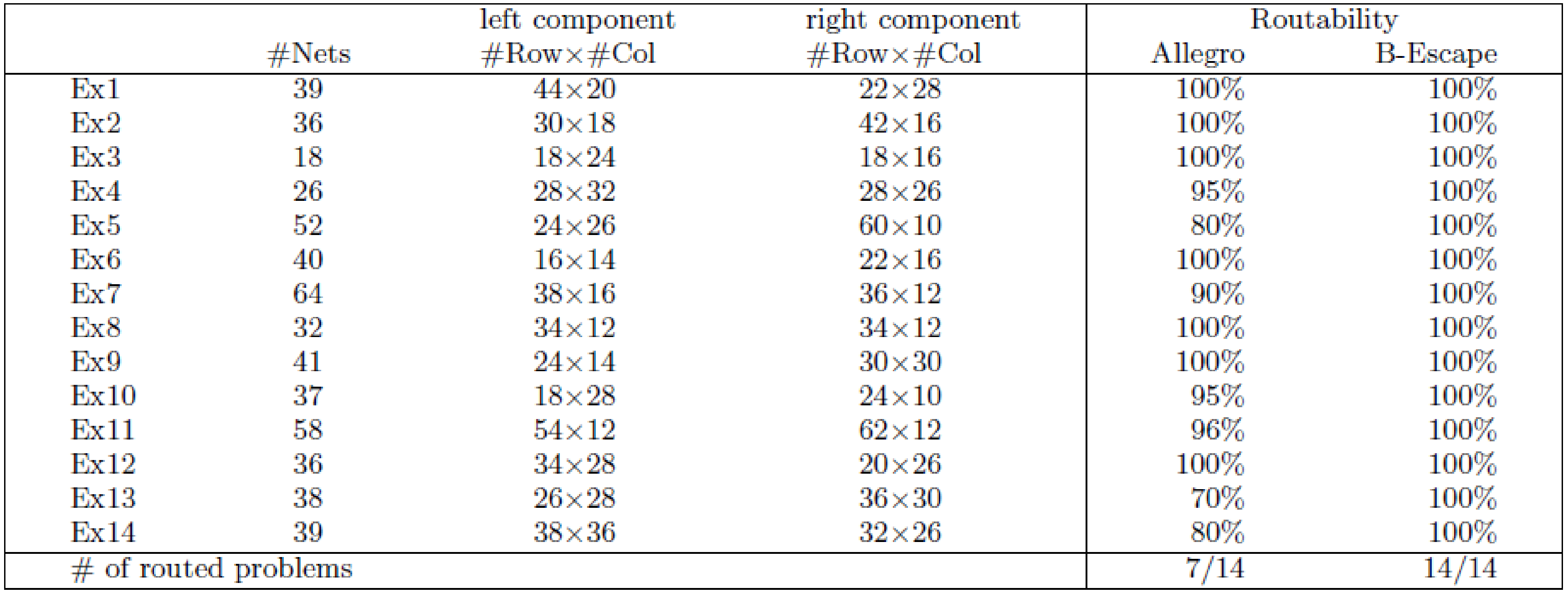
- Net-by-net routing with various routing styles
- Determine next net to route
- Route net along routing boundary



Routing boundary

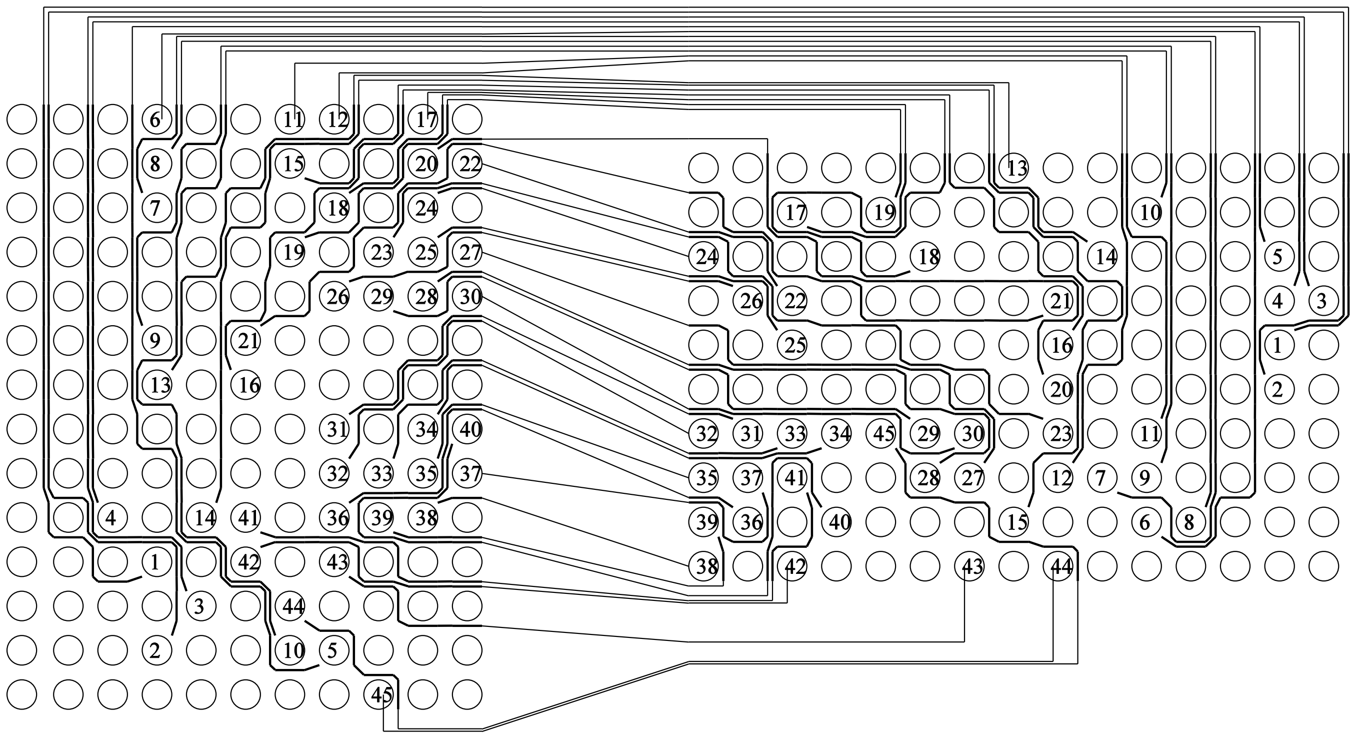
Experimental Results

- Performs significantly better than Cadence Allegro
- Runtimes range from 0.2s to 289s



		left component	right component	Routability	
	#Nets	#Row×#Col	#Row×#Col	Allegro	B-Escape
Ex1	39	44×20	22×28	100%	100%
Ex2	36	30×18	42×16	100%	100%
Ex3	18	18×24	18×16	100%	100%
Ex4	26	28×32	28×26	95%	100%
Ex5	52	24×26	60×10	80%	100%
Ex6	40	16×14	22×16	100%	100%
Ex7	64	38×16	36×12	90%	100%
Ex8	32	34×12	34×12	100%	100%
Ex9	41	24×14	30×30	100%	100%
Ex10	37	18×28	24×10	95%	100%
Ex11	58	54×12	62×12	96%	100%
Ex12	36	34×28	20×26	100%	100%
Ex13	38	26×28	36×30	70%	100%
Ex14	39	38×36	32×26	80%	100%
# of routed problems				7/14	14/14

Experimental Results

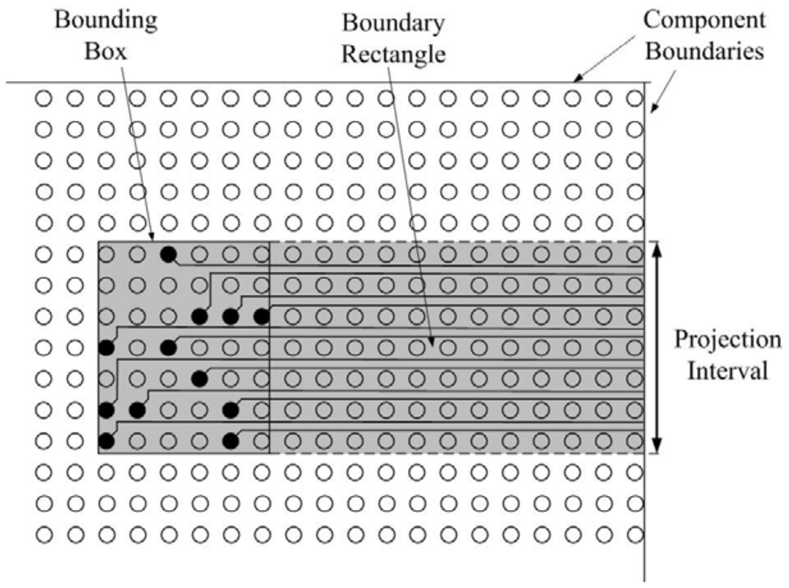


Bus Escape Problem

Hui Kong et al DAC 2010

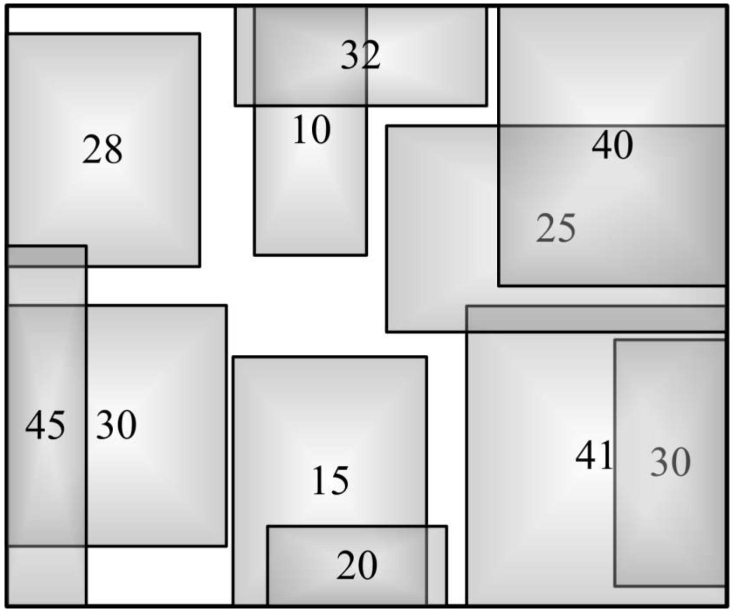
Bus Escape Routing

- Route nets from pins to component boundaries
- Keep bus structures
- Routing region for the bus is a *boundary rectangle*
- Formulate as a Maximum Disjoint Subset problem.

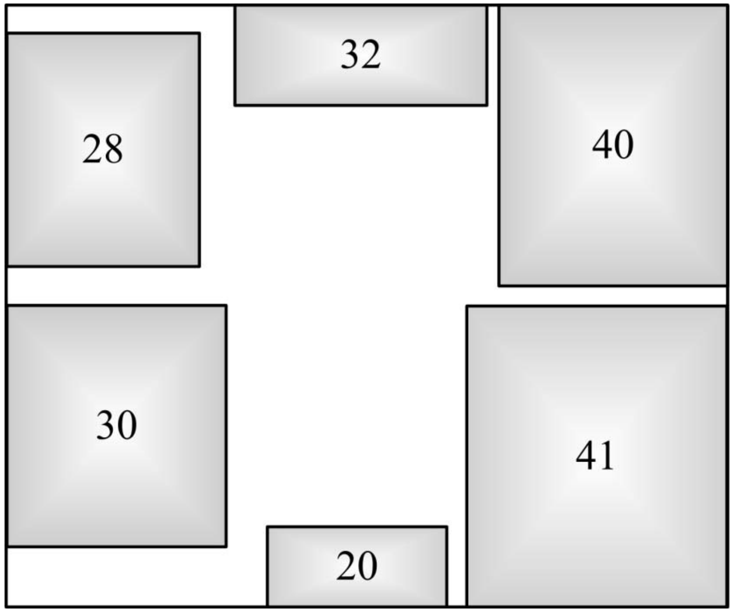


Maximum Disjoint Subset (MDS) Problem

- General rectangles → NP-complete
- Boundary rectangle → Open Problem
  - Rectangle attached to one or more boundaries
- We designed a polynomial time optimal algorithm!



Problem



Solution