Dynamic Planning of Local Congestion from Varying-Size Vias for Global Routing Layer Assignment

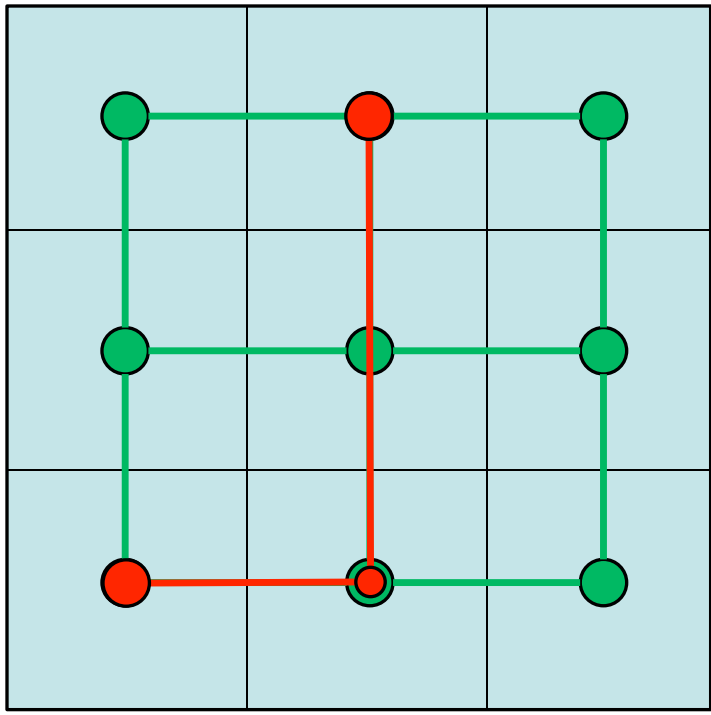
Daohang Shi, Edward Tashjian, Azadeh Davoodi
University of Wisconsin - Madison

Motivation

- In advanced technology nodes there is increasing mismatch between Global Routing (GR) and Detailed Routing (DR)
  - Due to complex and increasing number of design rules which have to be considered during DR but are ignored during GR
  - This mismatch reduces the utility of global routing as an effective starting point for DR which may result in significant increase in runtime of DR to resolve design rule check (DRC) violations

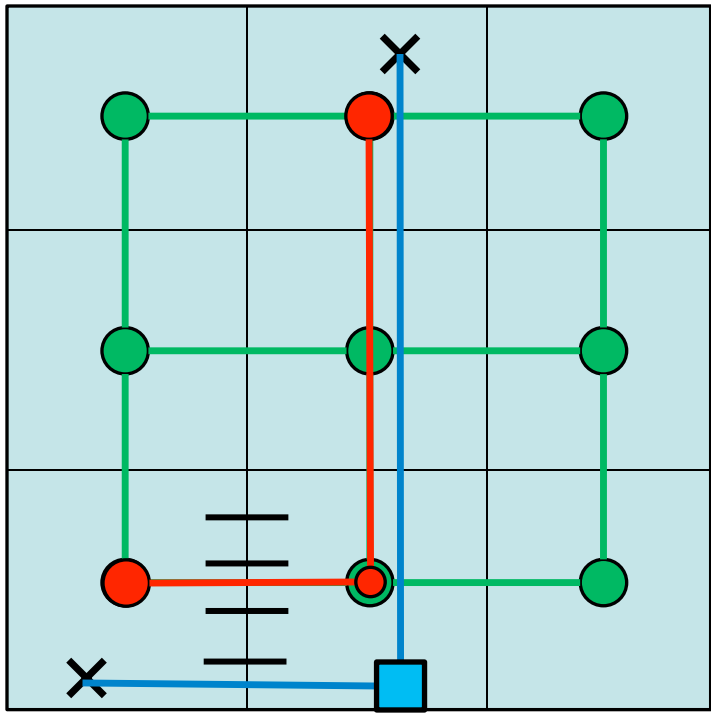
Introduction and Preliminaries

- Relationship between GR and DR
  - Global Routing
    1. Partitions the routing region into g-cells
    2. Constructs a grid graph
       - G-cell → vertex
       - Boundary between two g-cells → edge
    3. Finds routing trees on the grid graph



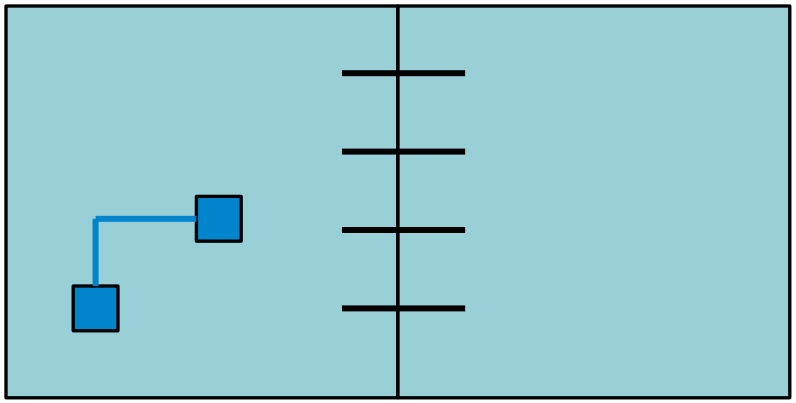
Introduction and Preliminaries

- Relationship between GR and DR
  - Detailed Routing
    - Uses global routing solution as the starting point
    - Determines the exact wirings for each net
      - Assigns every routing wire to a specific routing track
      - Determines the exact locations of vias to connect adjacent layers
      - Pins, rather than g-cells, are connected



Motivation

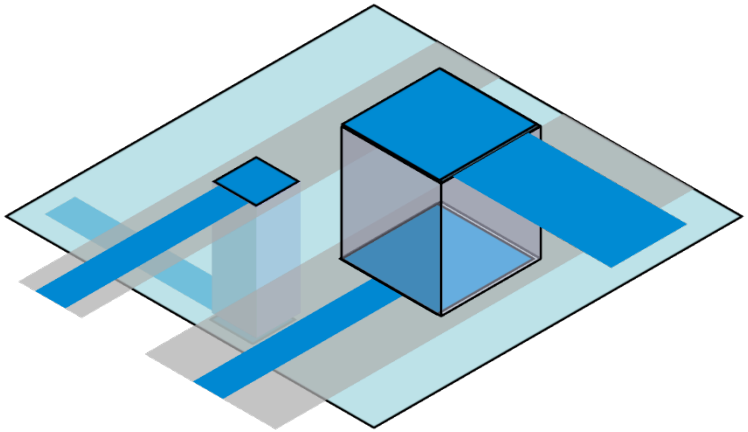
- Sources of mismatch between GR and DR
  - Example: local nets
    - Local nets are traditionally ignored in global routing but they are becoming a considerable percentage of total number of nets [1]
    - Few research works consider local nets during GR [2-3]

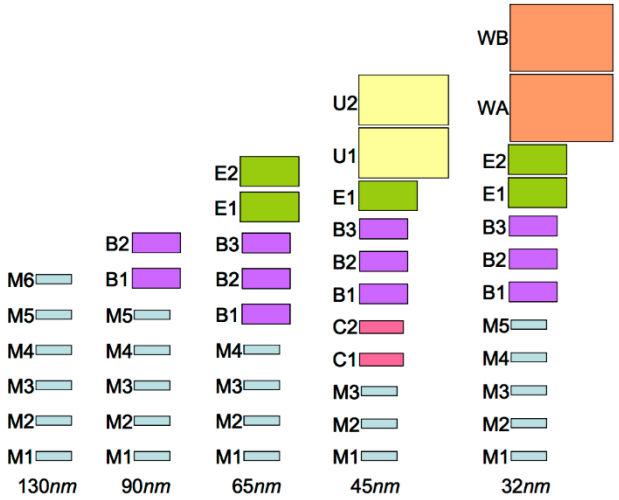


1. N. Viswanathan *et al*. “The ISPD-2011 routability-driven placement contest and benchmark suite”, *ISPD*, 2011.
2. H. Shojaei *et al*. “Planning for local net congestion in global routing”, *ISPD*, 2013.
3. Y. Wei *et al*. “GLARE: global and local wiring aware routability evaluation”, *DAC*, 2012.

Motivation

- In advanced technology nodes, vias are becoming an important source of mismatch between GR and DR
  - Due to significant variation of wire sizes in some *adjacent* metal layers

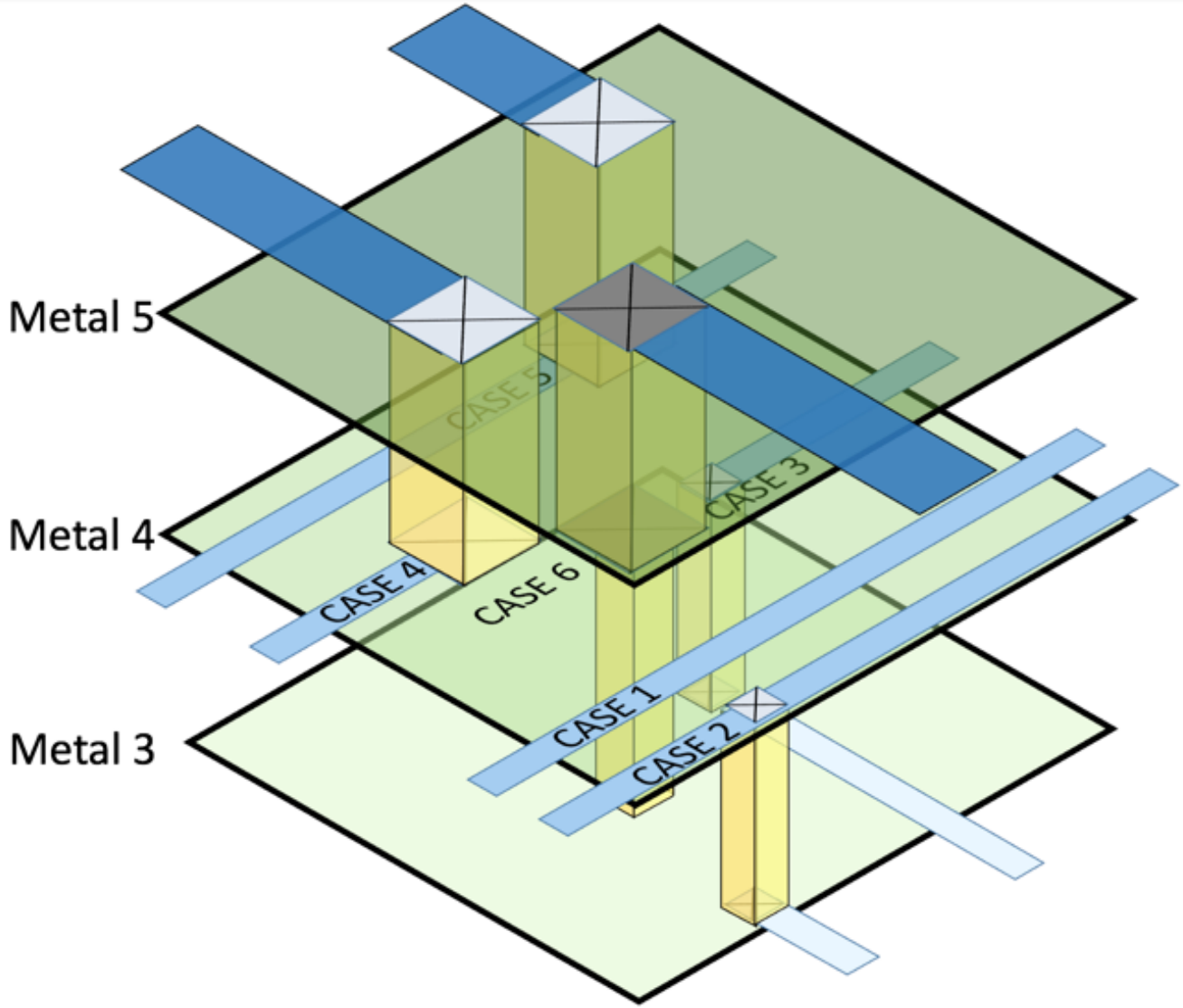




Summary of This Work

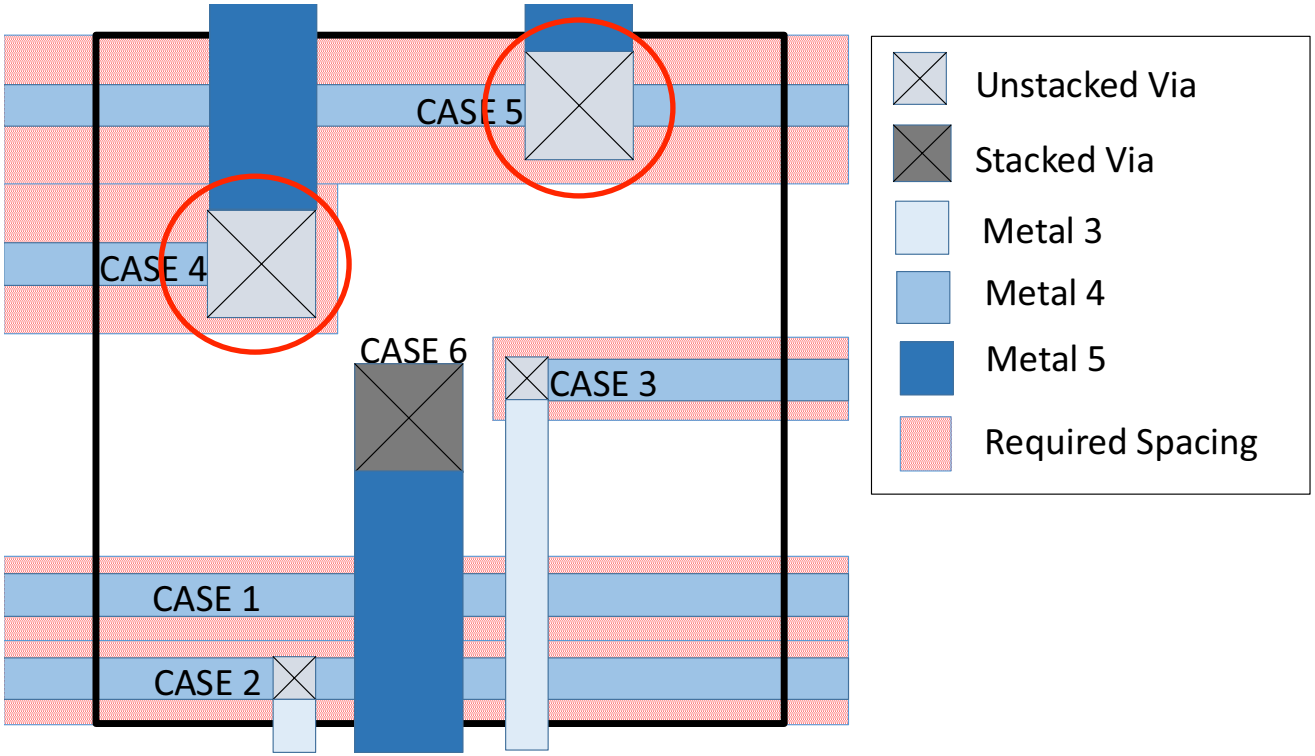
- Propose via-aware overflow models for GR
  - Via-aware edge overflow (VA-EOF)
  - Edge-aware via overflow (EA-VOF)
- Propose via-aware layer assignment algorithm (VALA)
  - Includes an efficient linear programming phase which provably generates integral solutions
- Simulation Results
  - Includes comparison of DRC violations at the DR stage using a commercial tool

VALA: Motivation



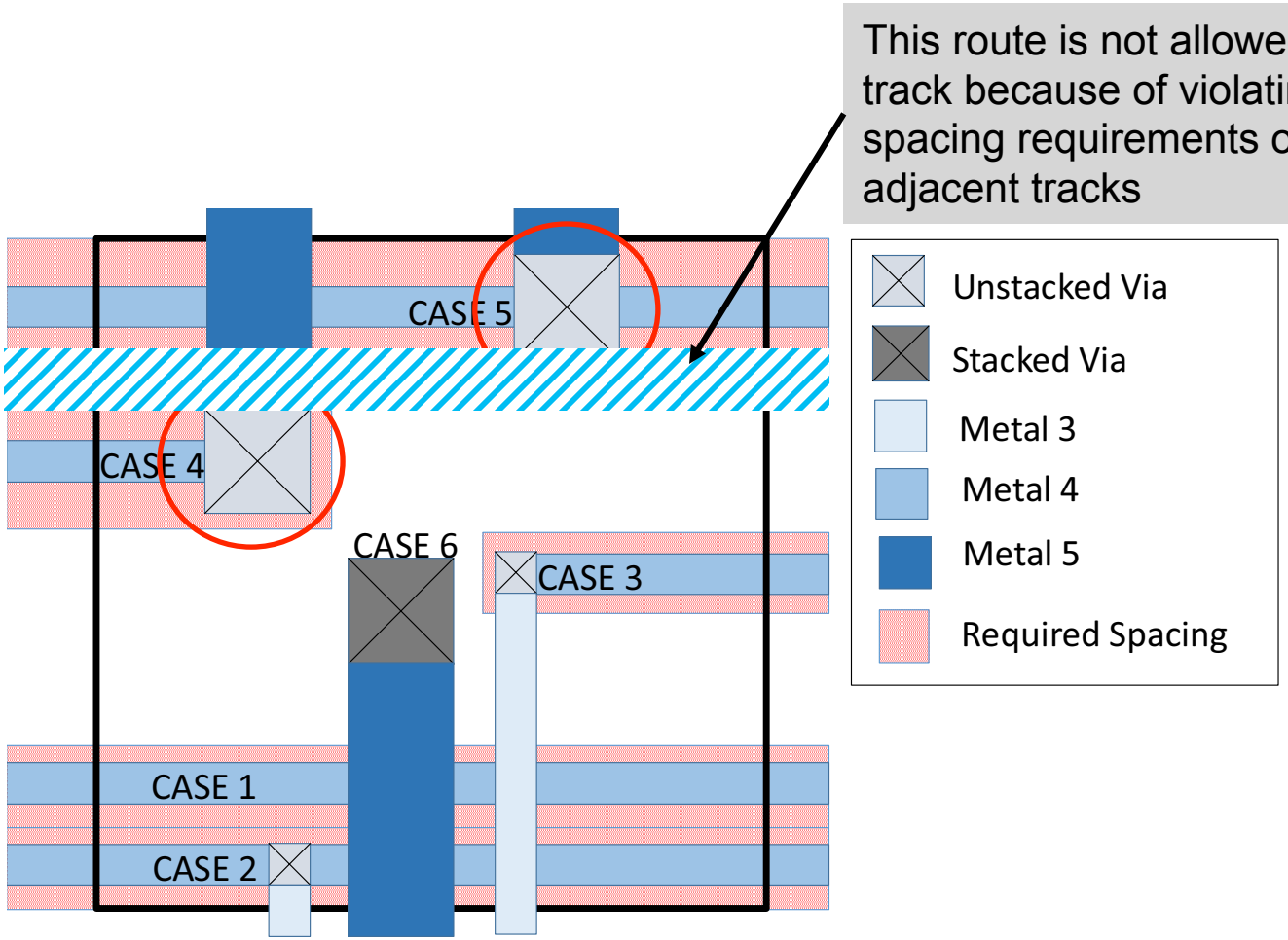
VALA: Motivation

- Issue 1: unstacked vias
  - May block adjacent routing tracks when connecting metal layers
    - Blockage happens on the g-cell with the smaller wire size



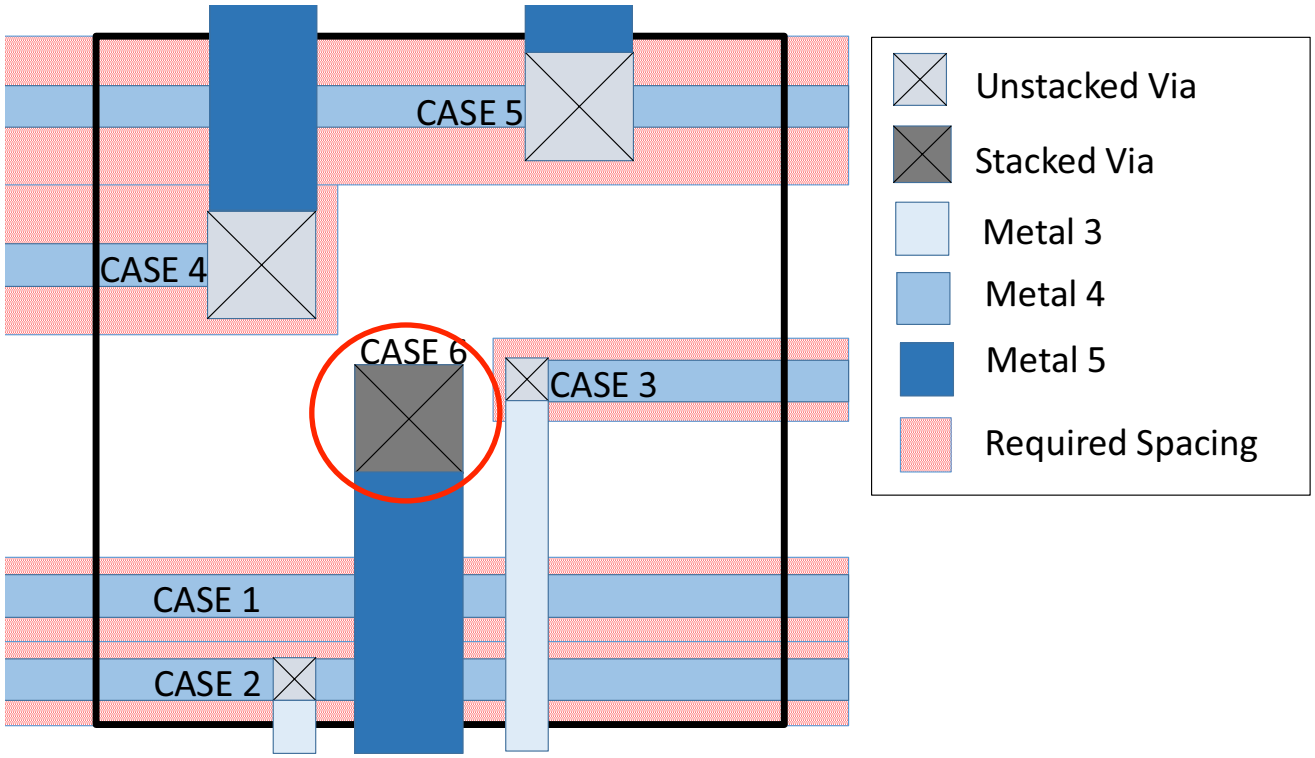
VALA: Motivation

- Issue 1: unstacked vias
  - May block adjacent routing tracks when connecting metal layers
    - Blockage happens on the g-cell with the smaller wire size



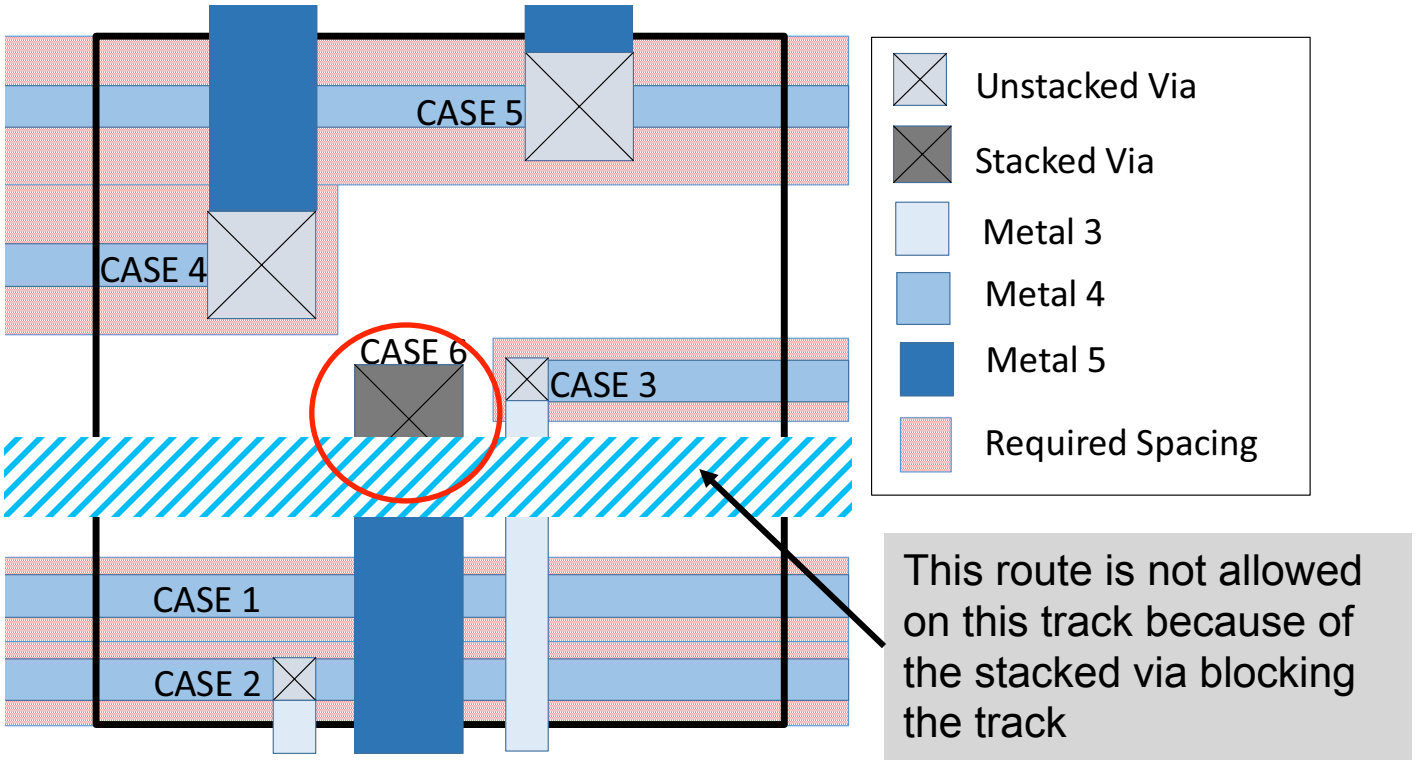
VALA: Motivation

- Issue 2: stacked vias
  - Stacked vias (i.e. vias only “passing” within a g-cell) can also block routing tracks such as case 6 in the figure



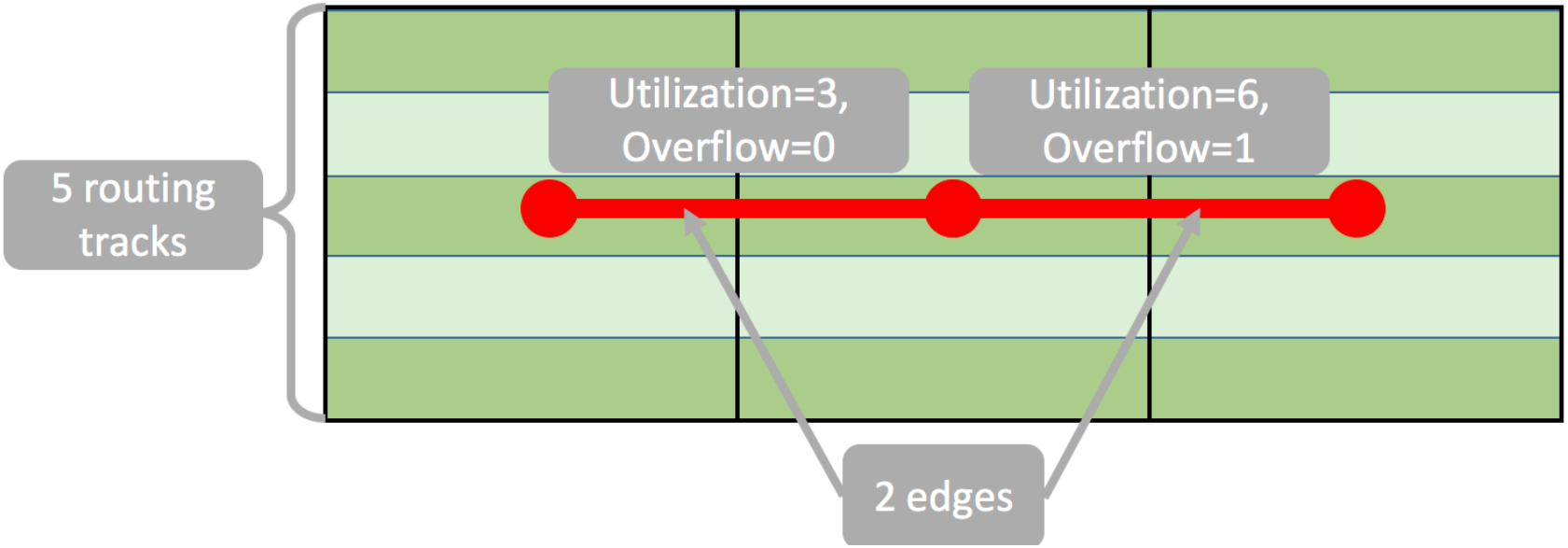
VALA: Motivation

- Issue 2: stacked vias
  - Stacked vias (i.e. vias only “passing” within a g-cell) can also block routing tracks such as case 6 in the figure



Existing Overflow Models

- Edge overflow (EOF)
  - Edge capacity is the number of available routing tracks that can cross the boundary of two adjacent g-cells
    - Edge utilization is the number of routes used by a GR solution
    - EOF = $\max(\text{edge\_utilization} - \text{edge\_capacity}, 0)$



This edge overflow model ignores the fact that larger vias may block neighboring routing tracks and result in a higher edge utilization from the GR grid graph.

Existing Overflow Models

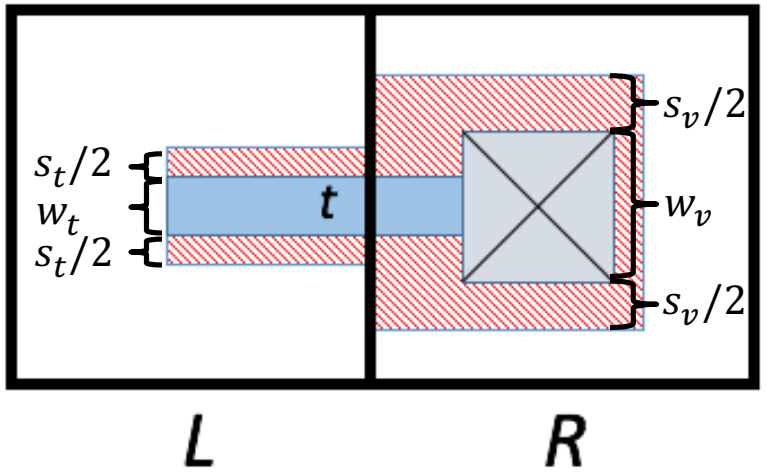
- Via overflow
  - Via capacity [1]
    - Defined for every g-cell
    - Captures how many stacked vias can pass a g-cell
      - After subtracting already consumed routing tracks
  - Via overflow
    - Max(#stacked vias – via capacity, 0)

This model ignores unstacked vias and assumes the same via size is used inside a g-cell.

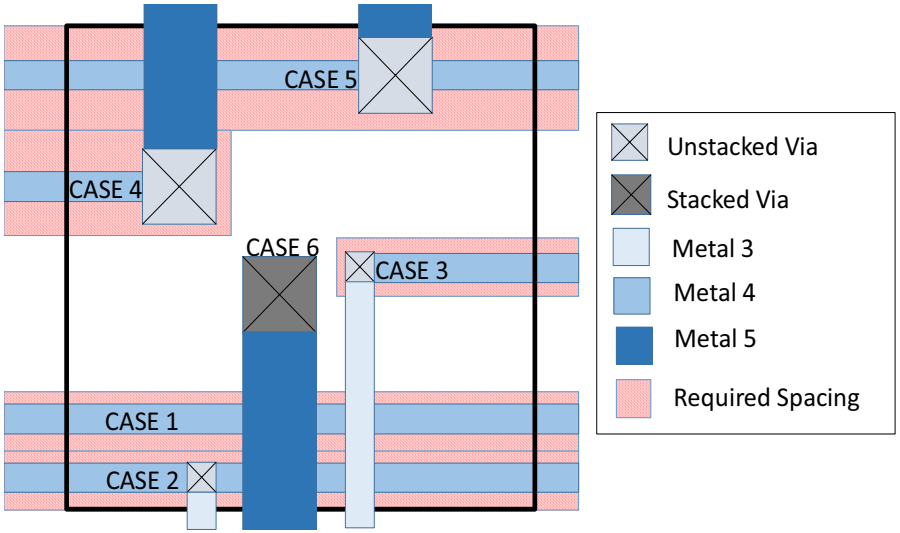
1. C. Hsu *et al*. "Multi-layer global routing considering via and wire capacities", *ICCAD,* 2008.

VALA: Proposed Overflow Models

- Via-aware edge models
  - For all routes t that pass edge e
  - Via-aware edge utilization
    - $u_e^{VA} = \max(\sum_{\forall t} r_{t_e}^L, \sum_{\forall t} r_{t_e}^R)$
    - The value of each r_{t_e} depends on which case t belongs to
  - Total via-aware edge overflow
    - $VA\text{-}EOF = \sum_{e \in E} \max(u_e^{VA} - c_e, 0)$

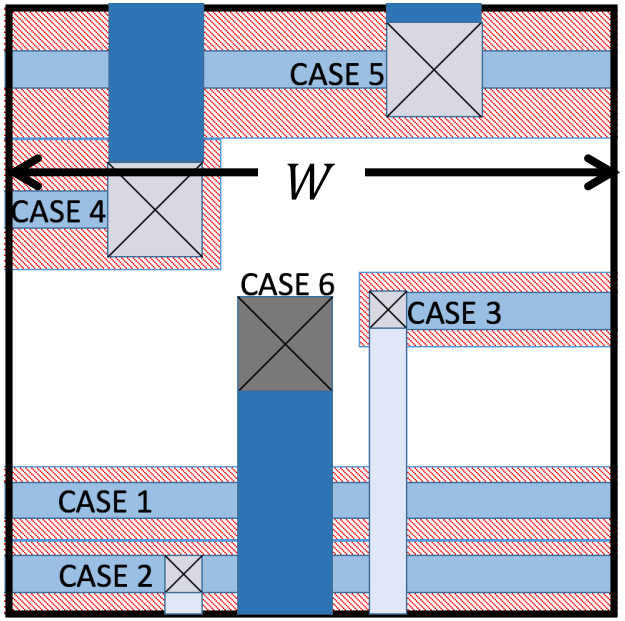


Case	$r_{t_e}^{L\|R}$
#1	$(w_t + s_t)$
#2	$(w_t + s_t)$
#3	$(w_t + s_t)$
#4	$(w_v + s_v)$
#5	$(w_v + s_v)$
#6	N/A



VALA: Proposed Overflow Models

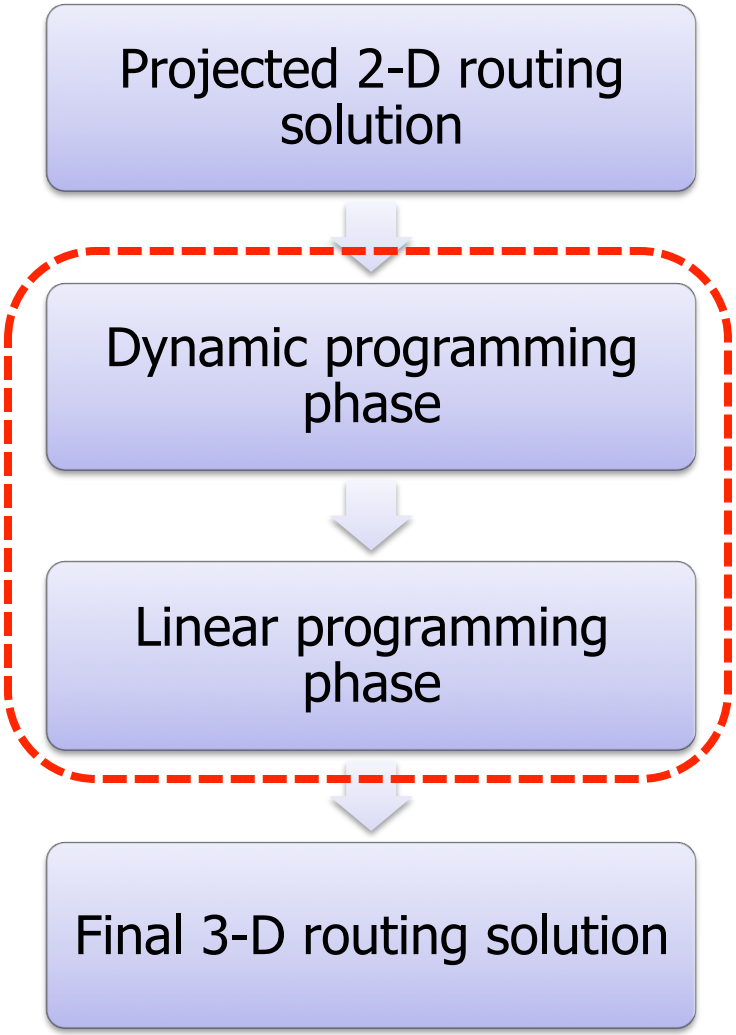
- Edge-aware vertex models
  - Defined for each g-cell g with respect to route t passing from g
  - Edge-aware g-cell utilization
    - $u_g^{EA} = \sum_t a_{t_g}$
    - The value of a_{t_g} estimates the area taken by the route from the track (includes the via area) and is computed from the table based on the case that it falls into
  - Total edge-aware via overflow
    - $EA\text{-}VOF = \sum_g \max(u_g^{EA} - c_g, 0)$



Case	a_{t_g}
#1	$(w_t + s_t) \times W$
#2	$(w_t + s_t) \times W$
#3	$(w_t + s_t) \times W/2$
#4	$(w_v + s_v) \times W/2$
#5	$(w_v + s_v) \times W$
#6	$(w_v + s_v)^2$

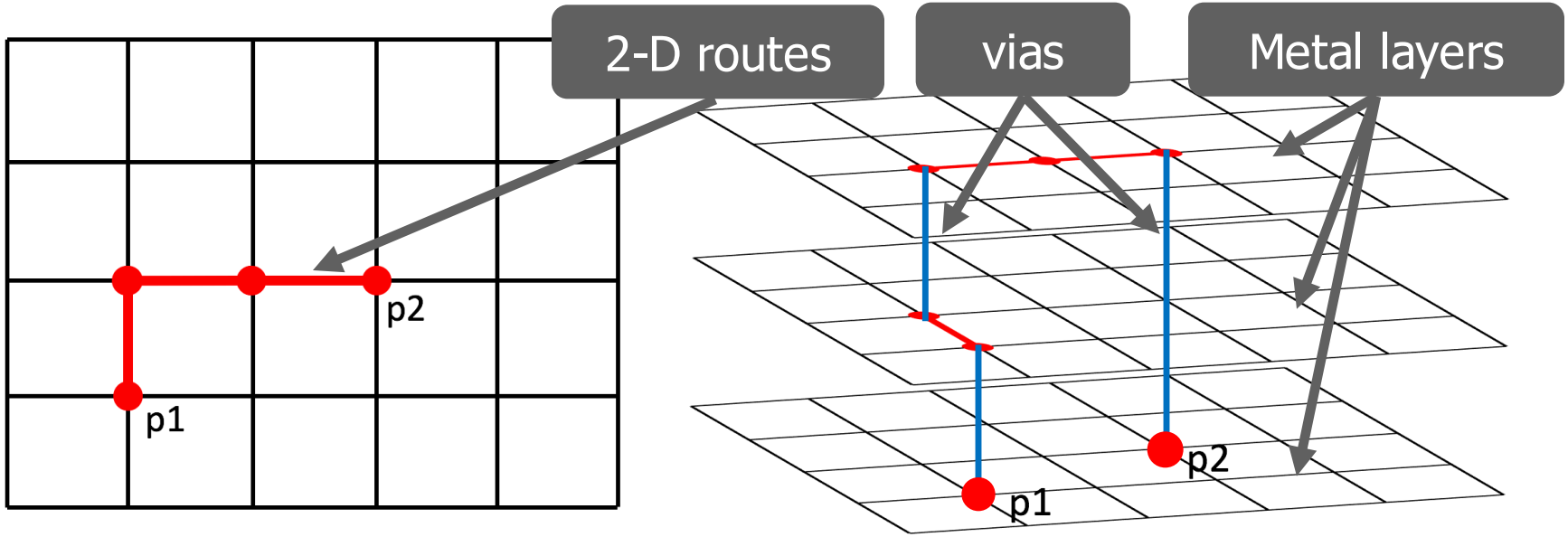
VALA: Overview of Layer Assignment

- Via-aware layer assignment
  - Takes 2D routing results as input
  - Performs a layer assignment procedure where VA-EOF and EA-VOF are additionally optimized



Overview of Layer Assignment

- Vias are determined during the layer assignment phase
- Layer Assignment
  - Receives as input the GR routing results from a 2D grid graph
  - Assigns every global wire (flat segment) of a global route to a metal layer while ensuring the 2D projection remains unchanged
  - Inserts vias to connect segments of a net on different layers



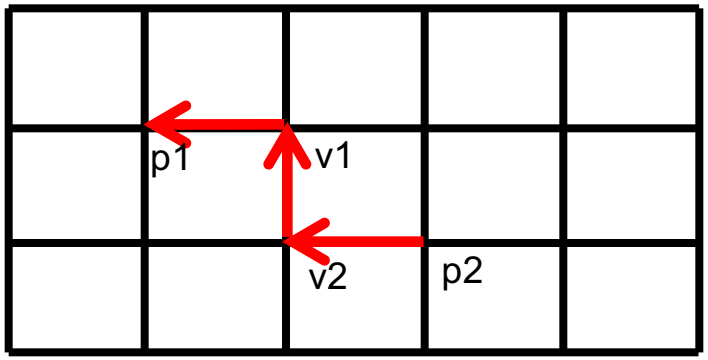
VALA: 1) Dynamic Programming

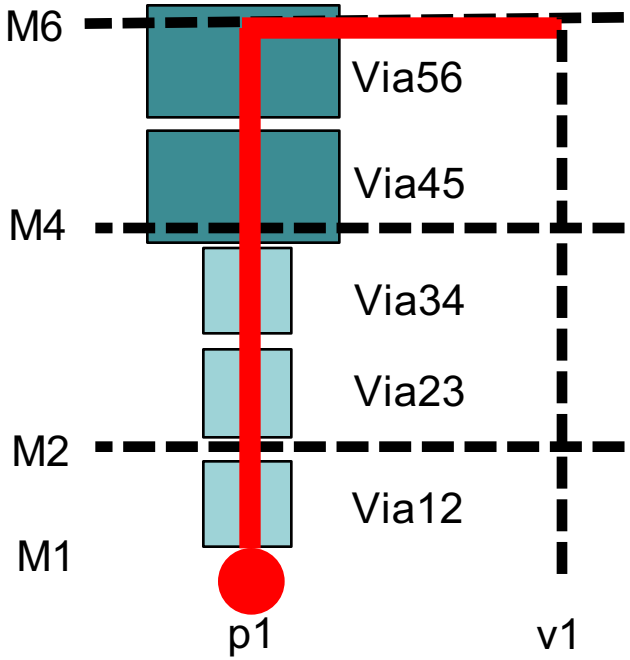
- Via-aware layer assignment
  - Stage 1 : dynamic programming (DP) [1]
    - We modified the cost function of the framework given in [1] to incorporate our proposed metrics
    - Overview:
      - The DP framework is applied to find the layer assignment on a net by net basis
      - Cost function of a net is expressed in terms of traditional metrics via count (VC) and EOF (as in [1]), as well as our proposed metrics VA-EOF and EA-VOF
      - Subproblem f(v,z): Find the minimal cost to perform layer assignment for the sub-tree rooted at node v when the edge between v and its parent node is assigned to layer z

1. W. Liu *et al*. "NCTU-GR 2.0: multithreaded collision-aware global routing with bounded-length maze routing", *TCAD*, 2013.

VALA: 1) Dynamic Programming

- Stage 1 : dynamic programming (DP) for one net
  - Example :
    - For edge (v1, p1), three sub-problems are considered :
      - f(v1, 2), f(v1, 4) and f(v1, 6)
    - Assume the cost function only depends on via count
      - For f(v1,6) we need 5 vias to connect (p1,v1) to p1 on M1
    - Similarly we update the cost function to calculate VAEOF and EAVOF



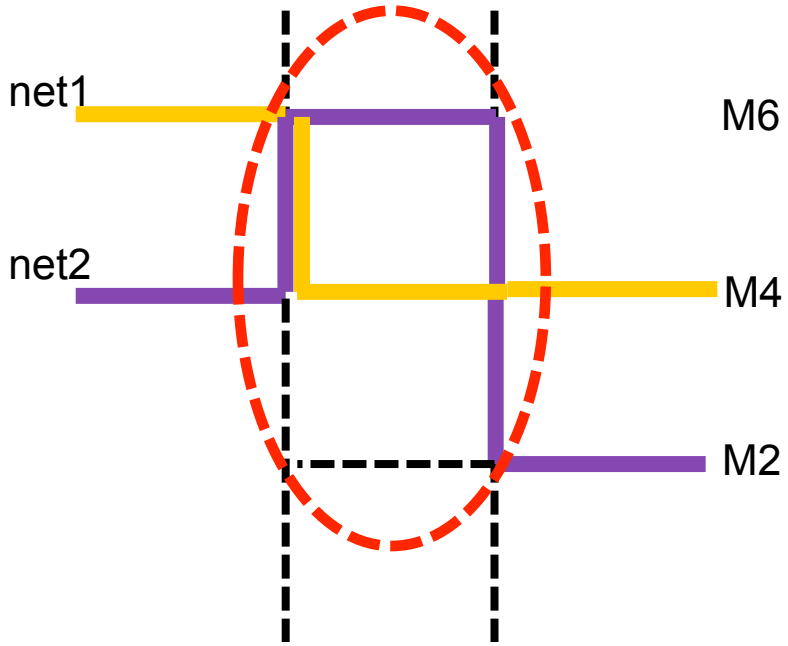


VALA: 2) Linear Programming

- Stage 2 : linear programming (LP)
  - We propose a refinement procedure to further optimize the GR solution generated by the DP stage
  - LP works on an “edge-set”
    - An edge-set is the set containing all the 3D edges mapping to the same 2D edge
    - Each LP is run-time efficient and has the property that the layer assignment generated solution is guaranteed to be integral
  - Procedure of LP stage
    1. Order edge-sets
       - Propose “Nautilus” ordering
    2. Solve LP for each edge-set according to the ordering

VALA: 2) Linear Programming

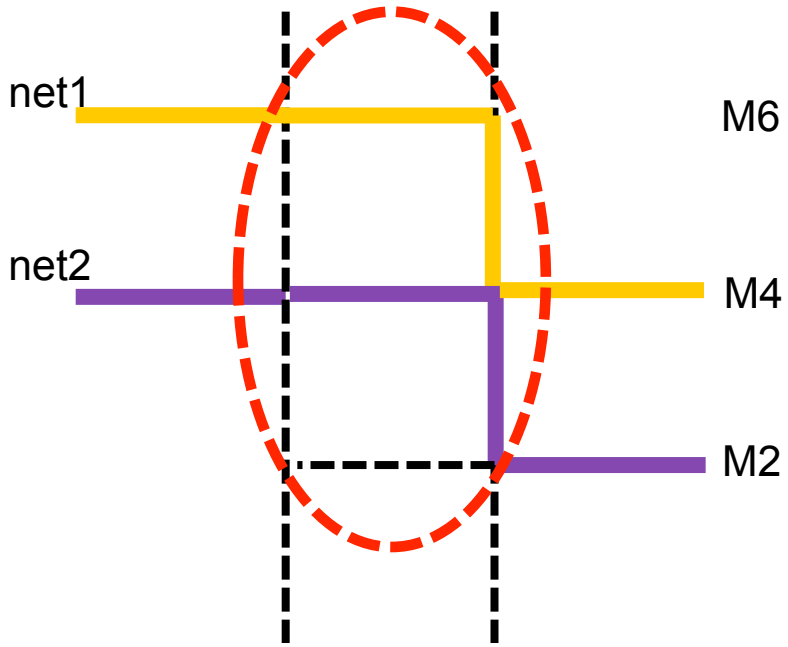
- Example of LP for one edge-set



After the DP stage, 4 vias are needed to connect the route fragments of the two nets for this edge-set

VALA: 2) Linear Programming

- Example of LP for one edge-set



LP changes the routes within the edge-set while keeping the edge utilizations of the 3D edges in the edge-set unchanged compared to DP.

After LP, it requires only 2 vias to make the part of nets within the edge set connected to the other parts outside the edge set.

VALA: 2) Linear Programming

- LP formulation for one edge-set
  - Variables : x_{nl} , for every net n and every candidate layer l
    - A *continuous* variable in the range [0 1] to show whether or not net n will be routed on the layer l within the edge-set
  - Constraints :
    - Selection constraint : Every net is routed on one and only one layer within the edge-set
    - Edge *utilization* constraint : The utilization of each 3D edge doesn't increase compared to the edge utilization of the DP solution (before LP)
      - Note this is different than the typical edge capacity constraint seen in GR papers
  - Objective :
    - Minimizing the via count

- While LP does not explicitly minimizes our proposed overflow metrics, we found out that the combination of minimizing via count as objective *and* our proposed, simple, edge-utilization constraints provides an effective way to optimize our proposed metrics as shown in the results

VALA: 2) Linear Programming

Objective function : $\min_{x} \sum_{\substack{n \in N_e \\ l \in S_e}} w_{nl} x_{nl}$

$s.t.$

$$\sum_{l \in S_e} x_{nl} = 1, \ \forall n \in N_e \ \ (1)$$

$$\sum_{n \in N_e} x_{nl} \leq u_l, \forall l \in S_e \ \ (2)$$

$$0 \leq x_{nl} \leq 1, \forall n \in N_e, \forall l \in S_e$$

(1) Each routing wire can only be assigned to one layer

(2) The edge utilization cannot become greater after LP

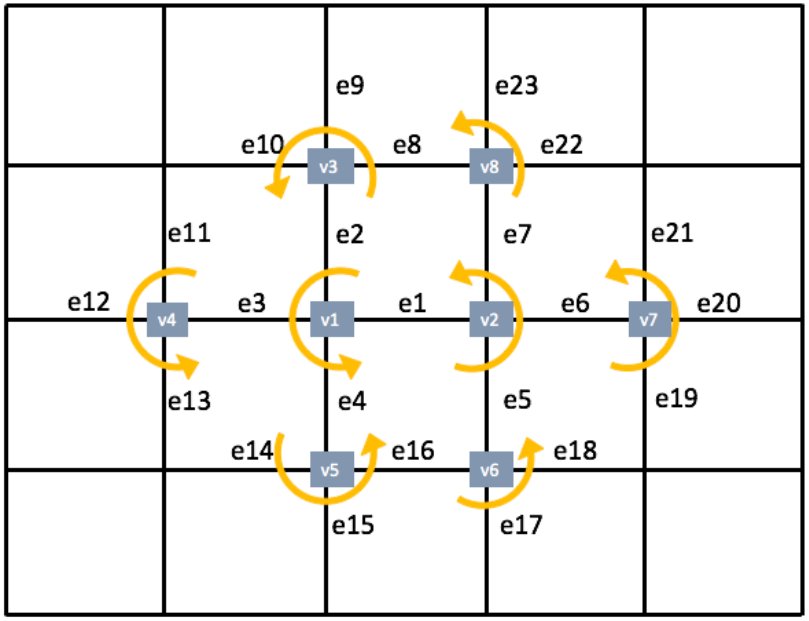
* parameter w_{nl} in the objective is the corresponding via count when connecting the edge of net *n* in the considered edge-set from layer *l* to the rest of the edges of the net outside the edeg-set

VALA: 2) Linear Programming

- Properties of the linear programming formulation
  - The coefficient matrix of LP is totally unimodular (TU) which means:
    - There exists a bi-coloring row partitioning in the matrix
    - The polyhedron determined by a TU matrix is *integral*
  - Time efficiency of solving TU formulations
    - The total run-time of solving tens of thousands small LP problems is typically shorter than a minute

VALA: 2) Linear Programming

- When solving LP for an edge-set, LP uses the solution of the neighboring edge-sets
  - Required to compute the term (*wnl*) in the objective expression
- So there will be error if any of the neighbor edge-sets changes later
  - Therefore important to have adjacent edge-sets solved sequentially as much as possible
- Propose a Nautilus-based ordering of edge-sets



Experimental Results

- Compared the following cases
  - **Base**
    - Implemented the DP framework given in [1]
      - Cost function only depends on via count and EOF
  - **VALA (DP):**
    - Ran the same DP but added our proposed metric VA-EOF and EA-VOF
  - **VALA (DP+LP):** Ran LP after VALA(DP)
  - **VALA (DP(No-EOF)+LP)**
    - Ran DP with a cost function in which traditional EOF got replaced with our proposed metrics VA-EOF and EA-VOF
    - Next, applied LP to the generated DP solution
    - This case allows measuring the impact of our proposed overflow metric as a replacement to the traditional overflow (EOF) metric

[1] W. Liu *et al*. "NCTU-GR 2.0: multithreaded collision-aware global routing with bounded-length maze routing", *TCAD*, 2013.

GR Comparisons

	Base Case					VALA (DP) / Base			
Design	EOF	#Vias	VA-EOF	EA-VOF	Run (s)	#Vias	VA-EOF	EA-VOF	Run (s)
sb1	0	4613288	40160	480516	76	1.03%	-80.81%	-32.97%	87.8
sb2	1686	6441030	311558	1702887	1022	3.91%	-64.34%	-26.64%	1007.4
sb4	272	3332687	141114	1424344	62	2.05%	-61.26%	-14.86%	61.7
sb5	0	4674230	136574	1168377	95	2.12%	-67.40%	-20.23%	107.1
sb12	0	9047463	425732	4827756	105	3.13%	-59.67%	-5.34%	121.8
sb15	0	6204090	220290	1900414	89	2.33%	-66.50%	-23.89%	100.4
sb18	0	3684947	128384	2129915	46	1.87%	-61.17%	-10.99%	52.7
average					**213.6**	**2.35%**	**-65.88%**	**-19.27%**	**219.8**

	VALA (DP+LP) / Base					VALA (DP(No-EOF)+LP) / Base			
Design	EOF	#Vias	VA-EOF	EA-VOF	LP-Run(s)	#Vias	VA-EOF	EA-VOF	
sb1	0	0.61%	-77.55%	-37.31%	63.6	0.08%	-74.69%	-13.99%	
sb2	1658	2.39%	-59.50%	-38.99%	219.9	0.90%	-57.21%	-32.05%	
sb4	262	0.72%	-58.04%	-27.49%	44.9	0.34%	-56.13%	-20.83%	
sb5	0	1.12%	-63.04%	-31.40%	89.2	0.22%	-61.64%	-24.78%	
sb12	0	0.87%	-53.19%	-23.53%	73.7	0.25%	-52.08%	-20.58%	
sb15	0	0.98%	-61.48%	-37.34%	63.4	0.40%	-60.38%	-20.41%	
sb18	0	-0.23%	-55.51%	-30.39%	35.5	-0.61%	-53.53%	-28.96%	
average		**0.92%**	**-61.19%**	**-32.35%**	**84.3**	**0.23%**	**-59.38%**	**-23.09%**	

Significant improvement in the proposed metrics with no or minimal degradation in EOF and via count

DR Comparisons

- Compared the number of DRC violations of global routing solutions of different approaches by doing detailed routing on each
- Our DR evaluation flow
  1. Converted benchmarks into industry-standard LEF/DEF formats
  2. Imported the benchmarks into Olympus SoC tool of Mentor
  3. Imported the global routing results with Olympus built-in commands create_wire and create_via
  4. Ran detailed routing of Olympus
  5. Extracted the number of DRC violations from the DR reports

DR Comparisons

	DRC violations		
Design	Base Case	VALA (DP+LP)	VALA (DP(No-EOF)+LP)
sb1	69471	4.76%	-2.66%
sb2	351969	-4.50%	-11.33%
sb4	158892	-5.23%	-6.23%
sb5	141145	-5.58%	-12.93%
sb12	624414	-0.85%	-4.17%
sb15	295567	-2.83%	-10.62%
sb18	186862	-10.62%	-16.09%
average		**-3.55%**	**-9.15%**

- Able to reduce the DRC violations using both VALA variations: DP+LP and DP(No-EOF)+LP
- More improvements in DP(No-EOF)+LP
  - Therefore our proposed overflow metrics may act as a good replacement for traditional EOF during GR in order to reduce DRC violations in DR

Summary of VALA

- We proposed two overflow metrics to capture the impact of varying-sized vias during GR
- We introduced an effective layer assignment procedure including a linear programming phase that generated integral solutions
- We showed that optimizing the new metrics at the GR stage helps reduce the number of DRC violations at the DR stage using a commercial detail router
- This is the first work to verify the effectiveness of GR with a commercial detailed router


Thank you!