BOB-Router: A New Buffering-Aware Global Router with Over-the-Block Routing Resources Optimization

Yilin Zhang[1], Salim Chowdhury[2] and David Z. Pan[1]

1 ECE, Univ. of Texas At Austin

2 Oracle, Austin

Outline

- Background, Motivation and Difficulties
- BOB-Router
  - Formulation
  - Four steps
- Experimental Results
- Conclusion

Blooming Research on Routing

- Two SRC and IEEE-CEDA sponsored ISPD Global Routing Contests (ISPD 07 and 08)
- Exciting results with traditional objectives such as wirelength, via count and routability are reported by well-known routers and their extensions, such as NTUgr, GRIP, FGR, MaizeRouter, Archer, NCTU-GR, FastRoute, BoxRouter and NTHU-Route

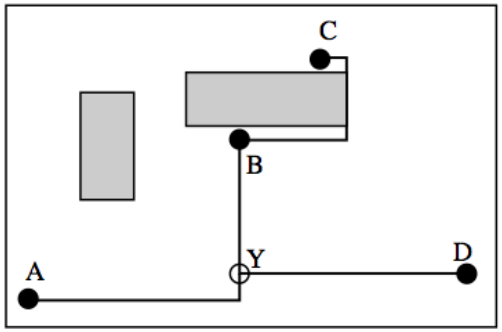
More Practical Objectives

- Most published modern routers aim at the same Objectives
  - Minimizing wirelength (WL) and via count in addition to alleviating congestion
- Pre-placed IP blocks or macros introduce serious signal integrity issues (slew violation) for routing over the blocks
- Routing over the IP blocks not allow periodic placement of repeaters (outside the IP blocks) to alleviate the slew violations.

Why They Are Not Practical Enough

- How to deal with pre-placed IP blocks and macros?
  - Buffer-unaware routing will result in long net over the block
    - Unable to buffer adequately
    - Large slew and bad timing

  - Totally avoid over-the-block routing
    - Losing routing resources over-the-block
    - More congestion outside



[Li+,ICCAD08]

Need Better Routing/Buffering

- Use over-the-block routing resources properly
- Our definition of Properly:
  - No slew violation for any over-the-block net
    - Help signal integrity and timing
  - Use as much over-the-block routing resources as possible
  - Minimize total (both over-the-block and outside) WL+via cost

Slew Model Used

- For any inside tree (tree inside the block), the worst slew part would occur at escaping points (on the edges of IP blocks)
  - If slew on EPs are good, then no problem everywhere
- Use PERI model for slew calculation
  - The error of PERI is within 1% [Kashyap+,ISPD03]

$$S(v_j) = \sqrt{S(v_i)^2 + S_{step}(v_i, v_j)^2} \qquad (1)$$

- Use Bakoglu's metric for step slew calculation [Bakoglu,90]

$$S_{step}(v_i, v_j) = \alpha * Elmore(v_i, v_j), \alpha = ln9 \qquad (2)$$

Outline

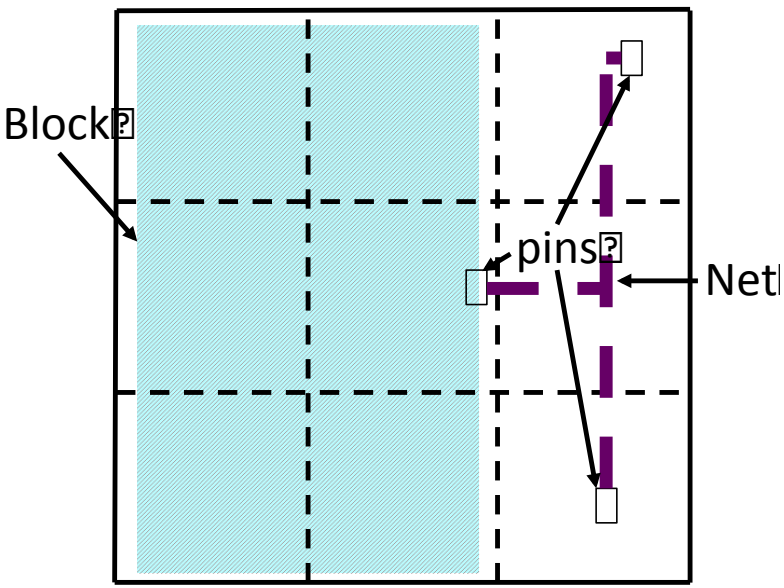
- Background, Motivation and Difficulties
- BOB-Router
  - Formulation
  - Four steps
- Experimental Results
- Conclusion

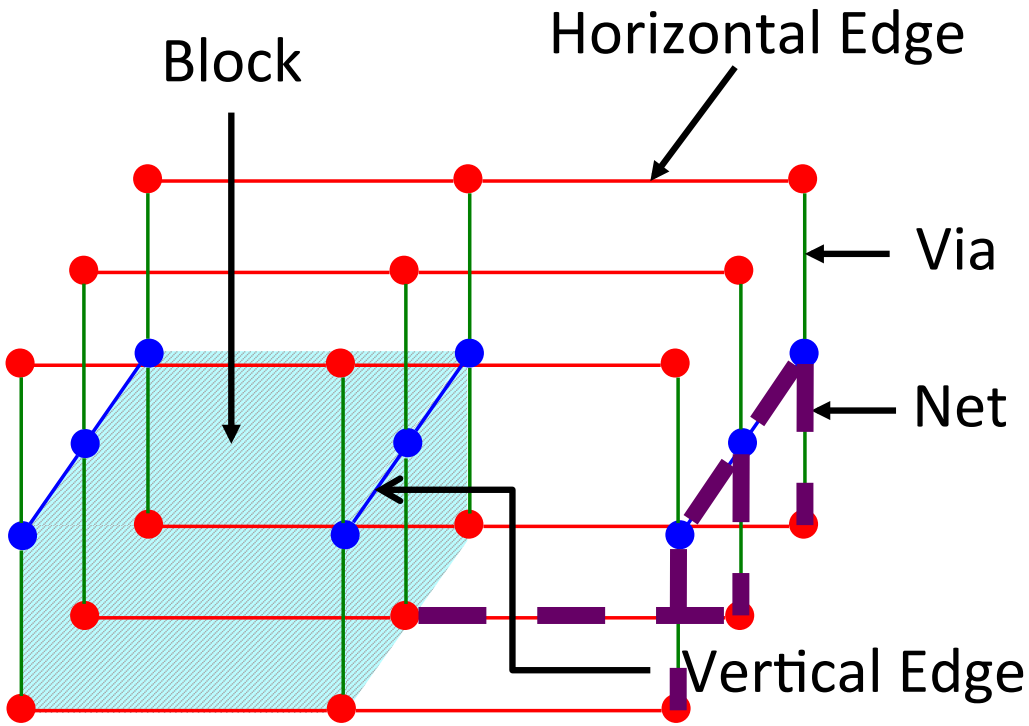
New Problem Formulation

- New objectives:
  - Reducing total overflow (TOF)
  - Minimizing total WL+via cost in additional
  - Over-the-block trees have to satisfy the slew constraints which ensure that every topology has feasible buffering solutions

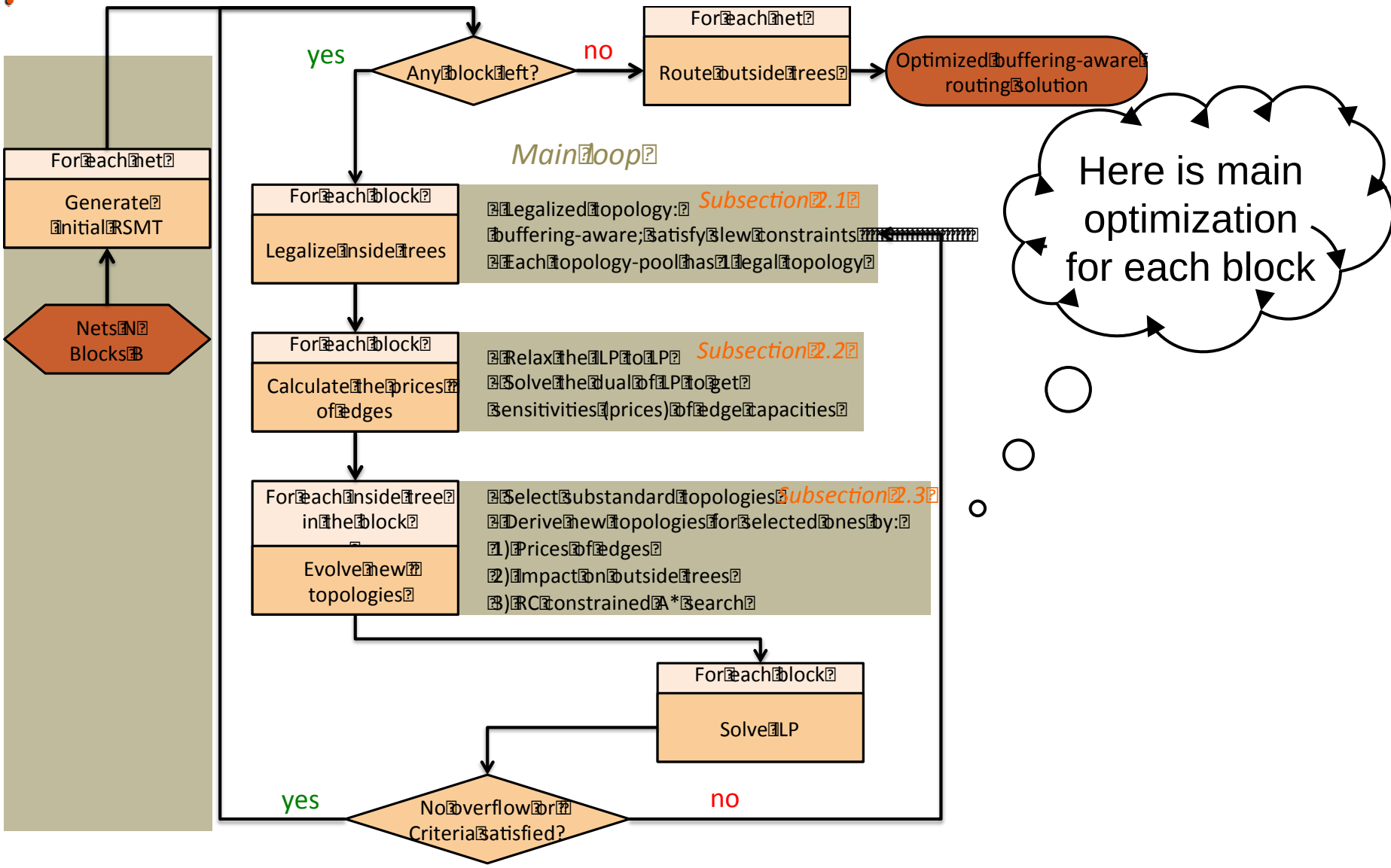
We Use 3D Routing Model

- 3D Grid Graph
- 2D Global Routing Bins





BOB-Router Overall Approach

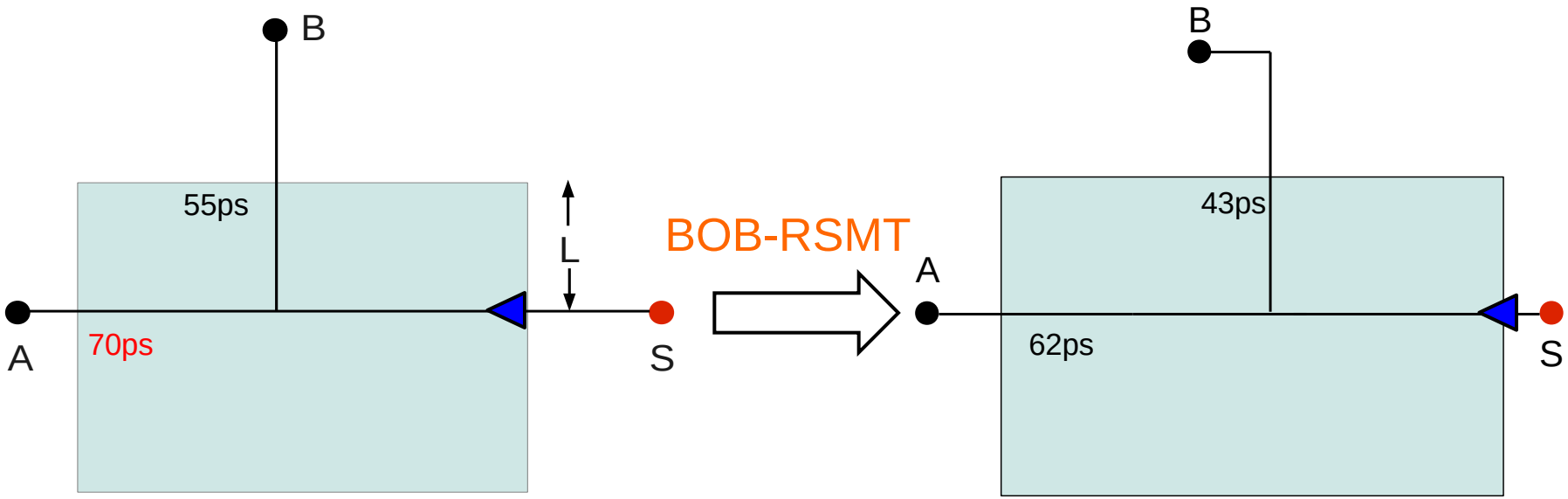


Main Steps within Each Block

- Step 1: Build initial legal inside tree for each net in block b
- Step 2: Calculate the prices for each edge in block b
- Step 3: Using 3-level progressive method to select topologies to reroute
- Step 4: Reroute selected topologies to generate new topologies

Step1: Initial Legal Inside Trees

- First Use BOB-RSMT [Zhang+,ICCAD12] to legalize slew for each inside tree

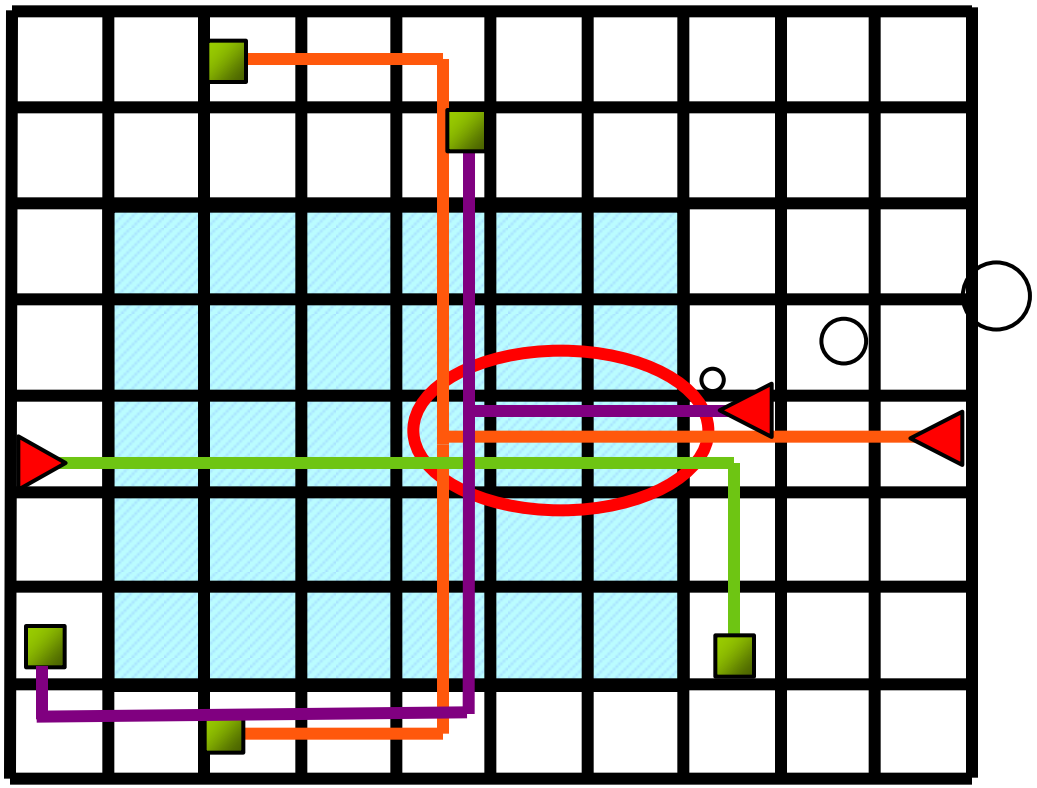


Use over-the-IP-block routing resources

Use over-the-IP-block routing resources with slew and buffering consideration

Step1: Initial Legal Inside Trees

- Congestion is allowed in current stage



All initial topologies are slew-satisfied; congestion may exist

Driver

Receiver

Step2: Calculate Price for Each Edge

- We have topology-pool for each inside tree
  - Initially the topology-pool only has one topology
- We use the formulation to calculate prices for each edge (also is the sensitivity of edge-capacitance)

ILP Problem formulation
[Wu+,2009 DAC]

$$\min. \sum_{i=1}^{n} \sum_{t \in \zeta_i} X_{it} W_{it} + M \sum_{i=1}^{n} S_i$$

$$\text{s.t. } \forall i, S_i + \sum_{t \in \zeta_i} X_{it} = 1$$

$$\sum_{i=1}^{n} \sum_{t \in \zeta_i} X_{ite} <= C_e \quad \forall e \prec b$$

$$X_{it} \in \{0,1\} \quad \forall i \in \{1,2,\ldots,n\} \quad \forall t \in \zeta_i$$

$$S_i \in \{0,1\} \quad \forall i \in \{1,2,\ldots,n\}$$

Relax ILP to LP
(Dual-LP will provide price for each edge)

$$\min. \sum_{i=1}^{n} \sum_{t \in \zeta_i} X_{it} W_{it} + M \sum_{i=1}^{n} S_i$$

$$\text{s.t. } \forall i, S_i + \sum_{t \in \zeta_i} X_{it} = 1$$

$$\sum_{i=1}^{n} \sum_{t \in \zeta_i} X_{ite} <= C_e \quad \forall e \prec b$$

$$X_{it} >= 0 \quad \forall i \in \{1,2,\ldots,n\} \quad \forall t \in \zeta_i$$

$$S_i >= 0 \quad \forall i \in \{1,2,\ldots,n\}$$

Step2: Calculate Price for Each Edge

- Derive dual problem
  - λ_i and ρ_e are Lagrange Multipliers
  - Lagrangian function:

$$\Lambda(X,\lambda,\rho)=\sum_{i=1}^{n}\sum_{t\in\zeta_i}X_{it}W_{it}+M\sum_{i=1}^{n}S_i+\sum_{i=1}^{n}\lambda_i(S_i+\sum_{t\in\zeta_i}X_{it}-1)+\sum_{e\prec b}\rho_e(\sum_{i=1}^{n}\sum_{t\in\zeta_i}X_{ite}-C_e)$$

$$\rho_e\geq 0$$

  - Dual-problem

$$g(\lambda,\rho)=\inf_{X,S}\Lambda(X,\lambda,\rho)=\sum_{i=1}^{n}\lambda_i(-1)+\sum_{e\prec b}\rho_e(-C_e)$$

$$\rho_e\geq 0$$

$$\forall t, W_{it}+\lambda_i+\sum_{e\prec t}\rho_e\geq 0$$

$$\forall i, M+\lambda_i\geq 0$$

Step2: Calculate Price for Each Edge

- Solve dual problem will provide ρe which is the price of edge e (also is the sensitivity on edge-capacitance)

$$\max. \sum_{i=1}^{n}(-\lambda_i) + \sum_{e \prec b}(-\rho_e)c_e$$

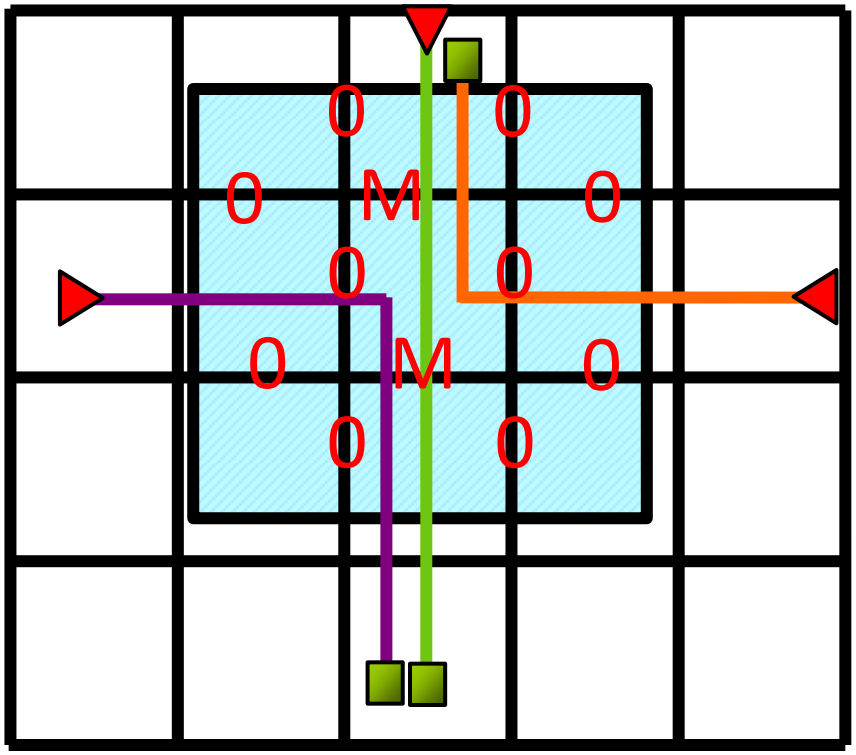
$$\text{s.t. } \lambda_i + \sum_{e \prec t} \rho_e + W_{it} >= 0$$

$$\lambda_i >= -M \qquad \forall i \in \{1, 2, \ldots, n\}$$

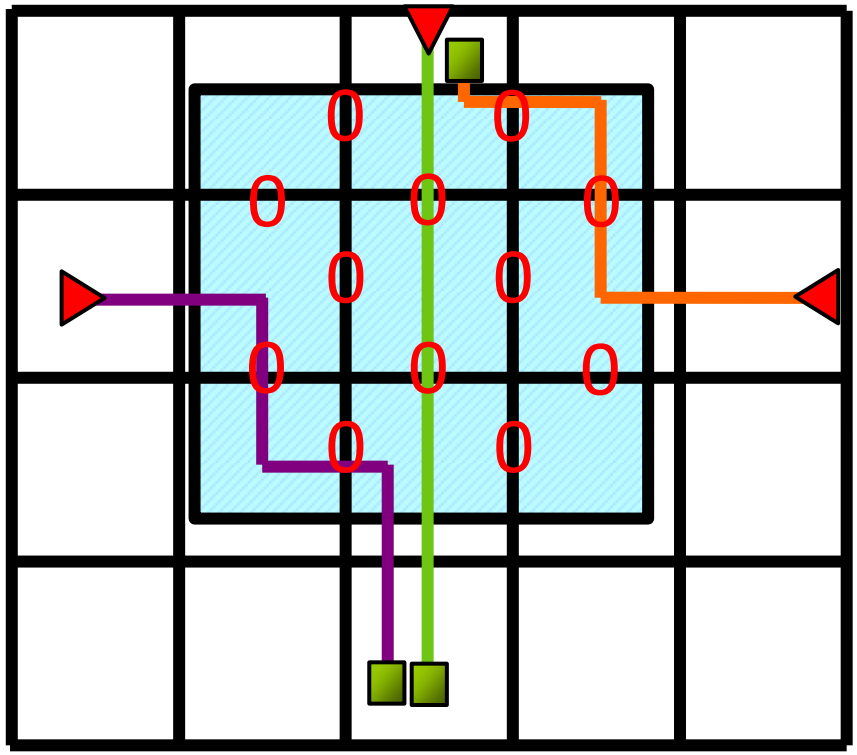
$$\rho_e >= 0 \qquad \forall e \prec b$$

Step2: Calculate Price for Each Edge

Expensive ρ_e due to congestion



Cheaper ρ_e for less congestion

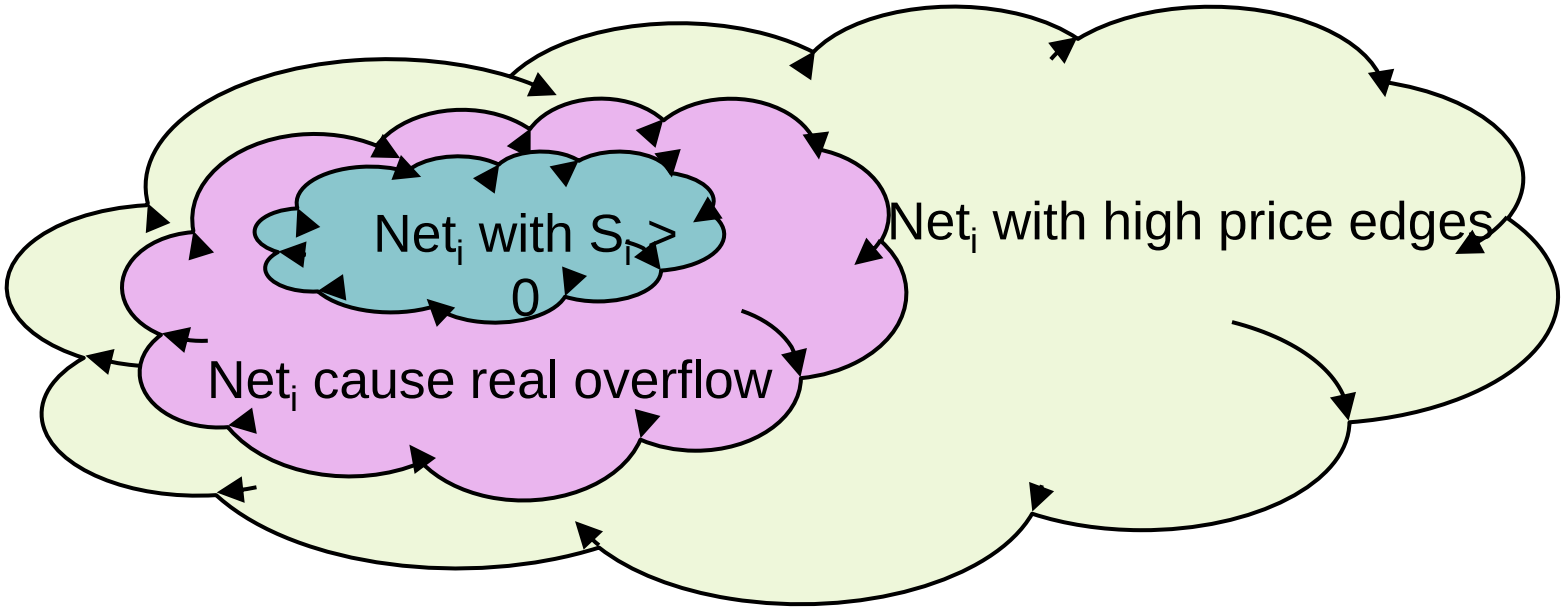


Step3: Select Topologies to Re-route

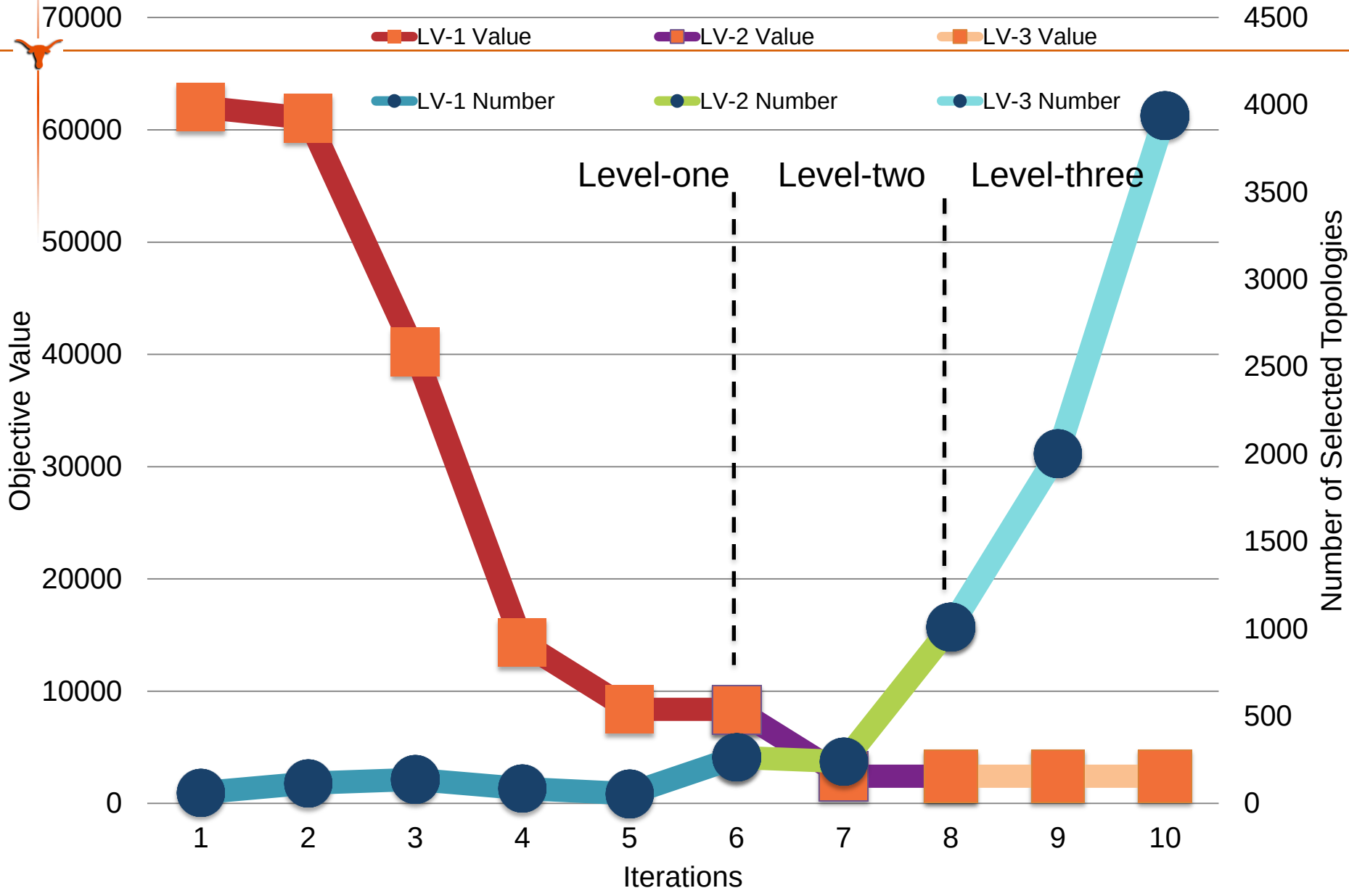
- Dynamic 3-level progressive topology-selecting strategy
- Once current level stops optimization, march to the next level
  - First, un-routable (in LP) inside trees ($S_i > 0$)
  - Then, inside trees with real overflow
    - Solve an ILP to choose topology for each inside tree, then see which inside trees contain real overflow
  - Last, inside trees with high price edge

Step3: Select Re-route Topologies

- First, un-routable (in LP) inside trees ($S_i > 0$)
- Then, inside trees with real overflow
  - Solve ILP to find out inside tree with real overflow
- Last, inside trees with high price edge



Optimization Process



Step 4: RC-constrained A* search

- For any selected topology t
  - Find and sort all positive-price branches
  - For the first branch w(U, V) on t, remove w
  - For each point p in t\w, find RC_p and C_p which are maximum allowed RC and C at p
  - A* search, with pruning by RC^{max} and C^{max}, to reconnect branch w

$$RC^{max} = \max\{RC_p, p \in \{t \backslash w\}\}$$

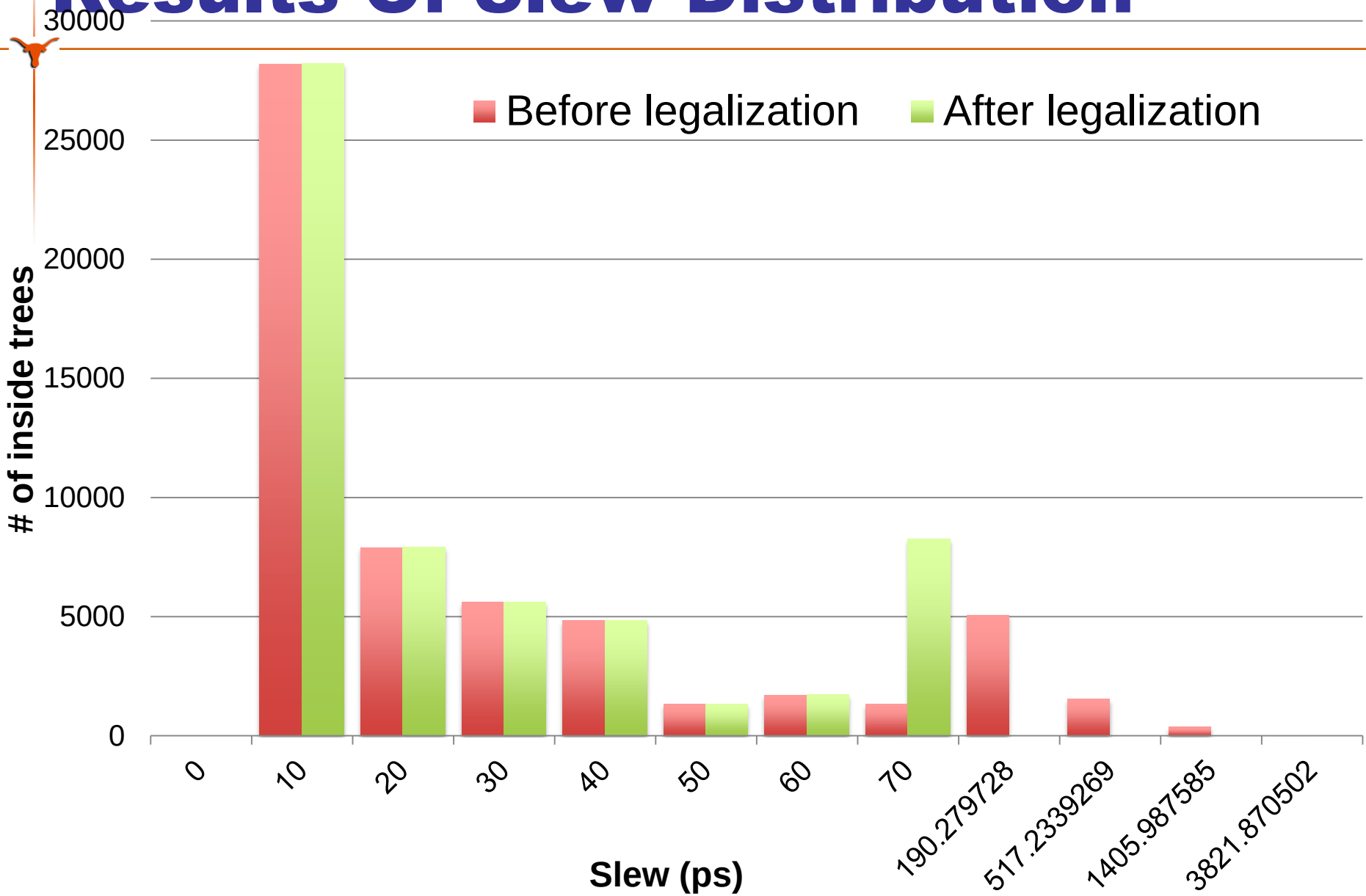
$$C^{max} = \max\{C_p, p \in \{t \backslash w\}\}$$

  - The connected point p will check RC_p and C_p
- This guarantee new topology satisfy slew

Outline

- Background, Motivation and Difficulties
- BOB-Router
  - Formulation
  - Four steps
- Experimental Results
- Conclusion

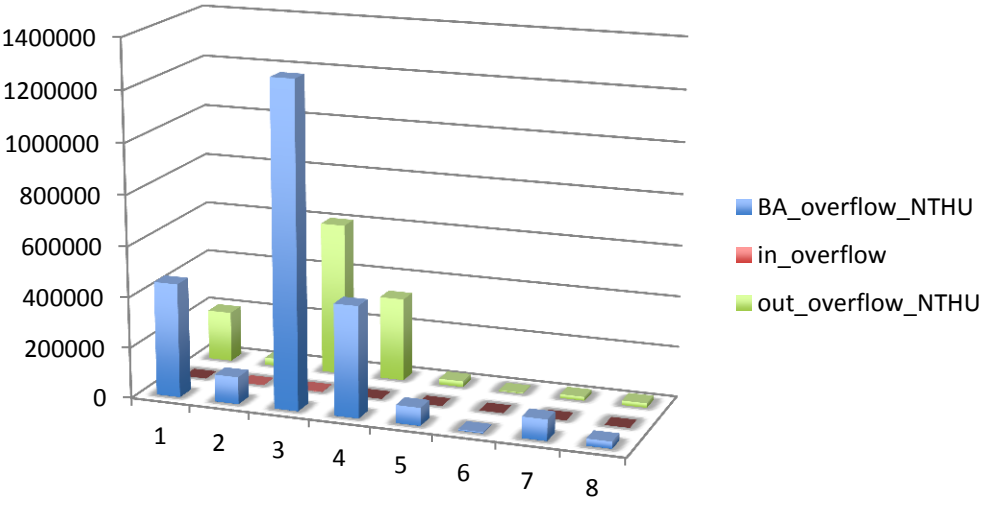
Results Of Slew Distribution



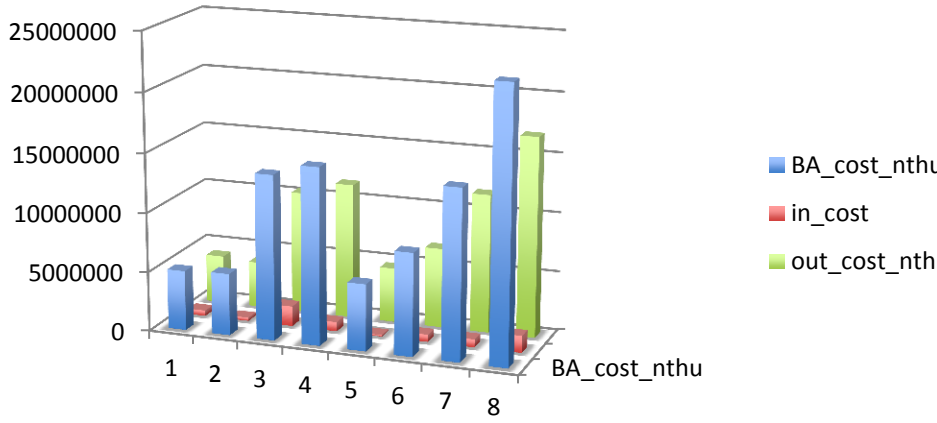
Results Of TOF, Wirelength, Vias

- Compare results in terms of TOF, cost (WL+via)
  - Benchmarks adaptec1~5, bigblue1~4
  - In inside part, 6 of 8 benchmarks have 0 TOF

TOF



WL+VIAs



Result of Bob-Router

SLEW DISTRIBUTION OF INSIDE TREES

Benchmarks	# nets	# inside trees	max slew	average slew
adaptec1	219794	57852	1713.8	36.9
adaptec2	260159	34769	494.4	28.5
adaptec3	466295	105137	23785.5	141.6
adaptec4	515304	86199	3986.7	65.8
bigblue1	282974	18763	380.1	22.1
bigblue2	576816	117259	69.9	4.0
bigblue3	1122340	79659	2025.1	22.1
bigblue4	2228930	234692	631.1	5.0

COMPARISONS BETWEEN OUR PROPOSED BOB-ROUTER AND OA-ROUTER

Bench	over-the-block				outside-the-block				overall				OA-Router			
-marks	WL	Vias	TOF	cpu(s)	WL	Vias	TOF	cpu(s)	WL	Vias	TOF	cpu(s)	WL	Vias	TOF	cpu(s)
adaptec1	431886	138207	0	5690	2733837	1344218	199565	1421	3165723	1482425	199565	7111	3317320	1724765	450300	3463
adaptec2	261957	57838	265	4523	2615068	1258131	28847	1038	2877025	1315969	29112	5561	3371453	1836853	107498	4577
adaptec3	1235721	154123	1333	100210	8355049	2849048	639049	16527	9590770	3003171	640382	116737	10100613	3740726	1276779	18845
adaptec4	836840	105953	0	32718	8831370	2580484	329221	13202	9668210	2686437	329221	45920	11326871	3498262	438954	13455
bigblue1	98044	42090	0	55	3248498	1367350	22612	1637	3346542	1409440	22612	1692	3637249	1967568	70853	2232
bigblue2	258699	350385	0	520	3730497	2985365	3795	1131	3989196	3335750	3795	1651	4799773	3800398	5145	1346
bigblue3	522841	141885	0	2119	7800699	3847139	15148	2621	8323540	3989024	15148	4740	8961863	5267470	83416	8603
bigblue4	575639	731836	0	303	9358521	7489968	5266	2266	9934160	8221804	5266	2569	12363167	10444398	27939	5784
average	0.08	0.06	0.00	0.51	0.92	0.94	1.00	0.49	1.00	1.00	1.00	1.00	1.13	1.28	3.07	1.00

Outline

- Background, Motivation and Difficulties
- BOB-Router
  - Formulation
  - Four steps
- Experimental Results
- Conclusion

Conclusion

- We study an important new class of Global Routing problem
- We propose an effective and efficient algorithm which can properly use over-the-IP-block routing resources satisfying slew constraints
  - Make use of over-the-IP-block routing resources
  - Satisfy slew constraints
  - Provide less TOF and WL+VIAs

Thank you!