Planning for Local Net Congestion in Global Routing

Hamid Shojaei, Azadeh Davoodi,

and **Jeffrey Linderoth***

Department of Electrical and Computer Engineering

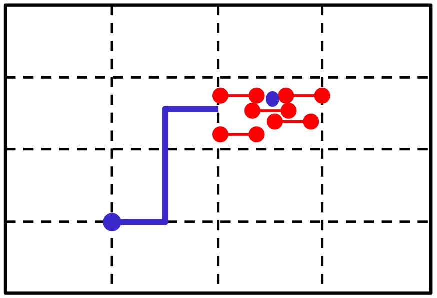
*Department of Industrial and Systems Engineering

University of Wisconsin-Madison

WISCAD
Electronic Design Automation Lab
http://wiscad.ece.wisc.edu

Motivation

- Local net
  - (local) connection of pins which falls completely inside a single global cell (gcell)



- Increase in the number of local nets
  - due to higher pin density, e.g., complex standard cells
  - Observation: in the placement solutions of winners of ISPD11 contest on routability-driven placement, on average 31.20% of (decomposed) nets are local
- Issues with local nets
  - Local nets are not captured during global routing (GR) → create mismatch between GR and detailed routing (DR) stages
  - Local nets consume wire tracks and block access to the pins
    - especially *in combination with* other routability issues such as those captured in the ISPD 11 benchmarks: variations in wire sizes over the metal layers & virtual pins (→ local nets in higher layers?)

Contributions

- Planning for local nets during GR
  1. Reduce the number local nets
     - using *non-uniform binning* (i.e., non-uniform gcell generation from uniform)
  2. Approximate routing usage of local nets in the graph model of GR
     - using *vertex capacity* in addition to edge capacities
- An Integer (Linear) Program *formulation* and GR graph model with the above two planning techniques
  – as well as layer-specific wire size/spacing and virtual pins
- Integration with CGRIP for a practical tool
  – extensions to various stages in CGRIP to accommodate the above planning techniques

Both the formulation and the final tool capture (1) variation in wire size/spacing per layer; (2) virtual pins in higher layers; (3) routing blockage; (4) non-uniform bins; (5) global routing with vertex and edge capacities simultaneously

Putting in Perspective

- Goals of congestion analysis during GR
  1. Model as many factors which contribute to routing congestion
     a. Factors which can fairly accurately be modeled at the GR stage
        – such as variations in wire sizes over the metal layers and virtual pins
     b. Factors which may only be approximated during GR and are only known during DR in a conventional design flow
        – such as local nets

  – This work focuses on goal 1b (while capturing goal 1a)

Putting in Perspective

- Goals of congestion analysis during GR
  2. Fast identification of unroutable regions on the layout for feedback to placement with as high resolution as possible
     - Need point of reference to claim a location is unroutable
       - Unroutable with respect to congestion map created at the DR stage?
         - » Note, this claim only be accurate if both items 1a & 1b are considered
       - Or unroutable with respect to the global router running much longer duration

- Our prior work CGRIP focused on goal 2 (while capturing goal 1a)
  - For a small analysis time budget, CGRIP identifies unroutable regions with "lower resolution"
    - modeled by introducing a new objective for the GR stage (i.e., regional minimization of overflow controlled by an input resolution parameter)

- CGRIP: Shojaei, Davoodi, Lindeorth, "Congestion Analysis for Global Routing Using Integer Programming", ICCAD'11

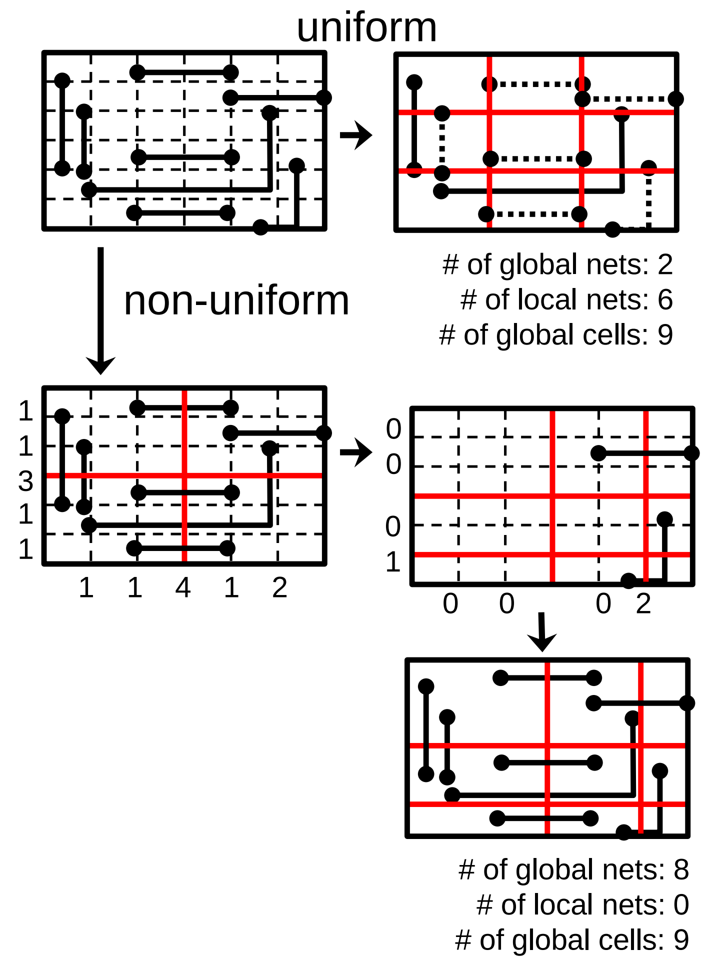
Binning Procedure

- Binning procedure
  - given
    - an existing grid for GR with uniform gcells with certain offset with respect to the placement grid
    - cell pin locations specified with respect to the placement grid graph
  - output
    - GR grid with non-uniform gcells
  - features
    1. trades off increase in global nets with decrease in local nets
       - increases the GR effort but in turn decreases the error associated with approximating or ignoring local nets
       - an input parameter allows controlling this tradeoff
    2. GR grid remains the same
       - as far as size (i.e., number of gcells) and topology of the GR graph (i.e., grid) and the offset with respect to the placement grid
       - changes in the graph model of GR are reflected in the weights of the edges and vertex which relate to capacity *and* wirelength

Binning Procedure

- Step 1:
  - Starting from the uniform grid, visit each cutline (V or H) of the grid and find a new location for it
    1. Each cutline is perturbed with respect to the placement grid within the entire range
    2. The new location results in the closest value to ηN_{max}
       - N_{max}: the maximum number of global nets for that cut when explored over its range of potential locations
       - η: input parameter between 0 and 1
         - » Controls number of global nets introduced at each step

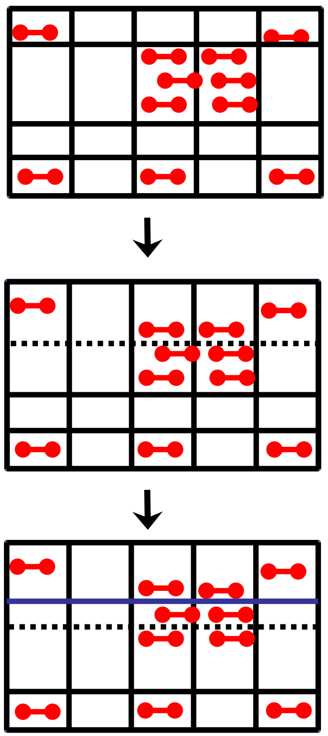
- Example



Binning Procedure

- Step 2:
  - balances local congestion among neighboring gcells
    - maybe good for routability in DR because local nets may be routed inside the corresponding gcells

1. Compute a local congestion ratio for each gcell
   - $LC_{ijl} = \frac{R_{ijl}}{A_{ijl}}$, for gcell located at $(x = i, y = j, z = l)\ \forall i, j, l$ determined by the grid in step 1
     - R_{ijl}: (approximate) routing usage of the gcell's local nets
     - A_{ijl}: area of the gcell
2. Adjust the location of each cutline (on the placement grid) and in a range between its two neighboring (same type) cutlines such that
   - Number of global nets does not change
   - Results in most decrease in sum of deviations among the LC ratios

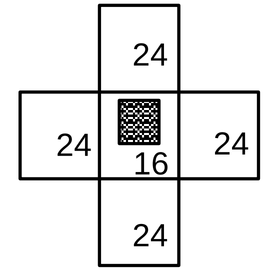


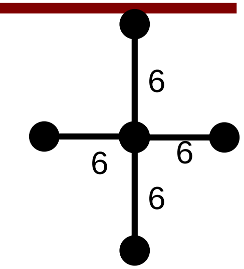
Binning Procedure

- Summary
  - Step 1 is more aggressive than step 2
  - In the presence of many routing blockages step 1 may not be appropriate
    - e.g., in superblue 10 (of ISPD11 benchmarks) over 67% of the first four metal layers are routing blockages
    - Currently, in our framework, we only apply step 1 if the amount of routing blockage compared to the total chip area is below 50%
      - Otherwise step 2 is solely applied
      - This strategy works for ISPD11 benchmarks based on our experiments
        - » For other cases not captured by these benchmarks, e.g., design with more complex routing blockages and containing various sized macros, it would be interesting to apply the non-uniform binning (selectively) to only some “appropriate” regions on the layout

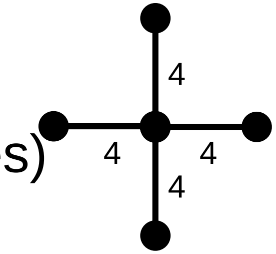
GR Graph Model: Motivation

- GR graph model with local nets
  - assuming routing resource usage of local nets inside a gcell is somehow approximated*

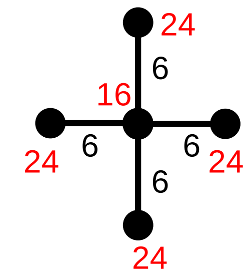




    1. Reduce the corresponding edge capacities (i.e., number of routes passing from the gcell boundaries)



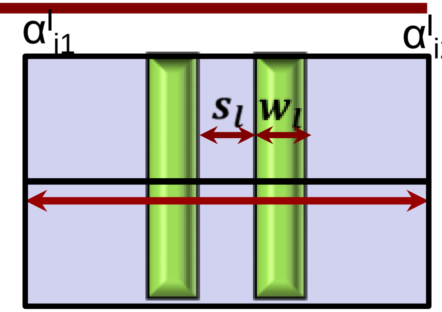
    2. **Our proposal**: Use a vertex capacity reflecting the *reduced* routing resource available *inside* each gcell along with *unreduced* edge capacities

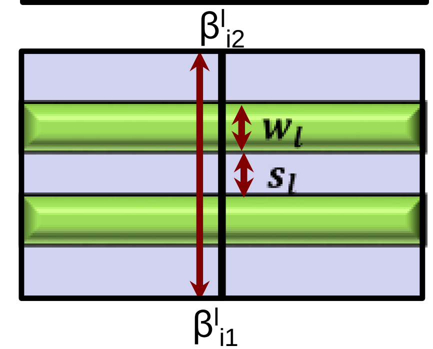


  - Comparison
    - Approach 1 restricts the search space and may yield to suboptimal solutions while approach 2 does not result in any restrictions
      - we show it does not add further complexity to a standard rip-up and reroute process

- Focus of "GLARE: global and local wiring aware routability evaluation", DAC12

GR Graph Model

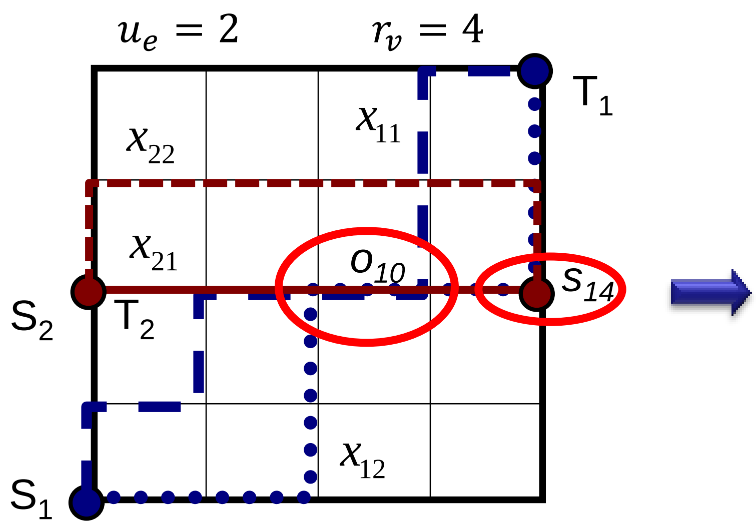




- Edge capacity with layer-specific wire size
  - (normalized) capacity edge $e = (i_1, j_1, l), (i_2, j_2, l)$
    - (EW) edge = $\frac{\alpha^l{}_{i2} - \alpha^l{}_{i1}}{w_l + s_l}$, (NS) edge = $\frac{\beta^l{}_{j2} - \beta^l{}_{j1}}{w_l + s_l}$
- Vertex capacity
  1. Each local net $T_n = \{(x_1^n, y_1^n), (x_2^n, y_2^n)\}$ is routed using its half-parameter bounding box on layers 2 and 3
     - its area usage is computed considering its wire size $d_n = \begin{cases} w^l |x_1^n - x_2^n| \; if \; l = 3 \\ w^l |y_1^n - y_2^n| \; if \; l = 2. \end{cases}$
  2. Summation of the areas of the local nets inside a gcell is subtracted from the gcell area
     - $r_v = \frac{A_{ijl} - R_{ijl}}{\gamma_{ij}}$ $\quad R_{ijl} = \begin{cases} \sum_{n \in \delta_{ij}} d_n^l \; if \; l = 2,3 \\ 0 \; otherwise. \end{cases}$ $\quad \gamma_{ijl} = \begin{cases} \propto_{i2} - \propto_{i1} \; if \; l = 2 \\ \beta_{i2} - \beta_{i1} \; if \; l = 3. \end{cases}$

     flexible to incorporate other models of routing usage of local net
- Other features of GR graph remains same
  - i.e., grid-graph size and offset with respect to the placement grid

Integer Program Formulation



$$\min[(8x_{11} + 8x_{12} + 6x_{21} + 4x_{22} + M1(o_1 + \cdots + o_{40}) + M_2(s_1 + \cdots + s_{20})]$$

$$\begin{cases} x_{11} + x_{12} = 1 \\ x_{21} + x_{22} = 1 \end{cases}$$

$$\begin{cases} x_{11} + x_{12} + x_{21} \le 2 + o_{10} \\ \vdots \end{cases}$$

$$\begin{cases} x_{12} + x_{21} + x_{22} \le 4 + s_{14} \\ \vdots \end{cases}$$

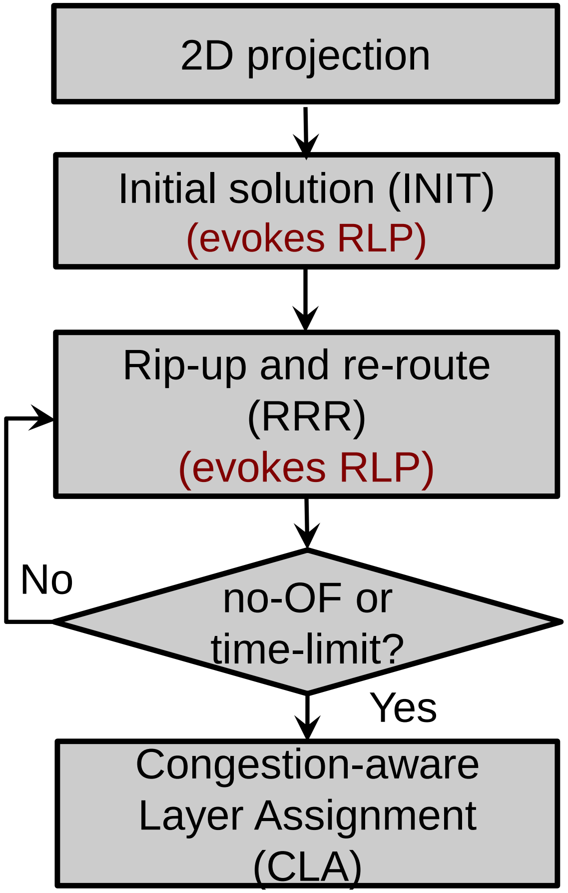
$$\min(\sum_{n \in N} \sum_{t \in T_n} p_t x_t + M_1 \sum_{e \in E} o_e + M_2 \sum_{v \in V} s_v$$

$$\begin{cases} \sum_{t \in T_n} x_t = 1 & \forall i = 1, \ldots, N \\ \sum_{n \in N} \sum_{t \in T_n} a_{et} x_t \le u_e + o_e & \forall e \in E \\ \sum_{n \in N} \sum_{t \in T_n} z_{vt} x_t \le r_v + s_v & \forall v \in V \end{cases}$$

- This IP is extension of the one in CGRIP when using maximum resolution
- (Shown for uniform binning, only for demonstration but IP handles generic (non-uniform) case)

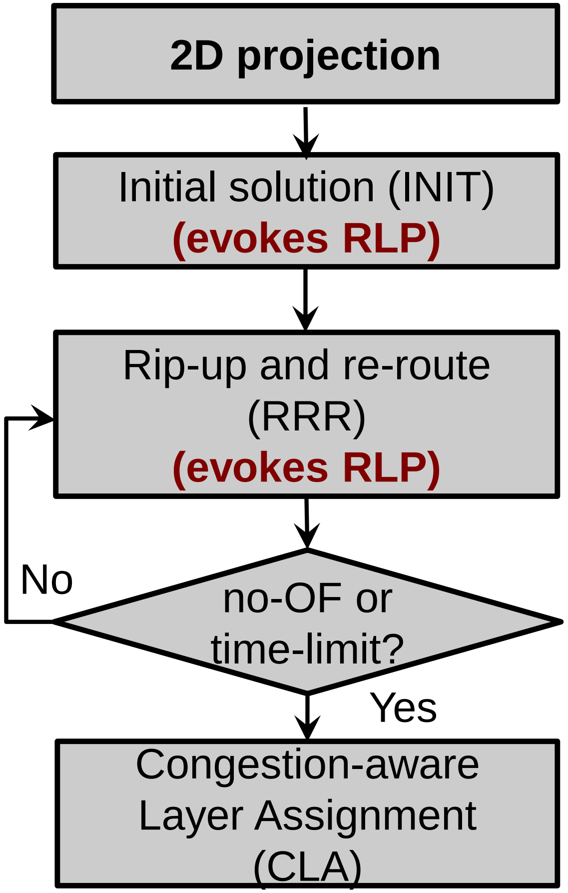
CGRIP: Overview of Framework

1. Solves RLP
   - a reduced-sized and relaxed version of Integer Program formulation
   - *Significant* amount of reduction in overflow of CGRIP is due to RLP
2. Integration of RLP in a standard rip-up and reroute framework
   - RLP is formed by selection of very small subset of “critical” variables
     - Selection changes at each iteration of RRR based on the latest RRR routing solution



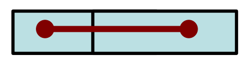
Integration with CGRIP

- 2D projection
  - creates $G_{2D} = (V_{2D}, E_{2D})$
  - compute normalized vertex and edge capacities
  - $u_e = \sum_{\forall l} u_{(i_1,j_1,l),(i_2,j_2,l)}\,; \forall e = (i_1, j_1, l), (i_2, j_2, l) \in E_{2D}$
  - $r_v = \sum_{\forall l} r_{i,j,l}\,; \forall (i,j) \in V_{2D}$
- RLP
  - Select critical edges (just as in CGRIP)
    - critical vertices are then the vertices connected to the critical edges
  - Solve reduced IP presented in this work which includes vertex capacity

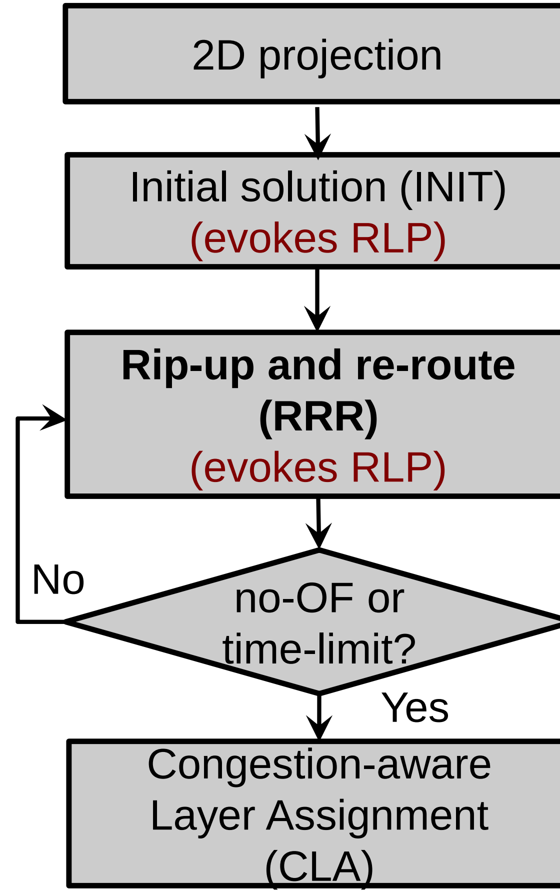


Integration with CGRIP

- RRR
  - Solve shortest path with both edge and vertex capacity
  - edge weight $l_e + f(\frac{g_e}{u_e})$; $\forall e \in E_{2D}$
    - l_e: wirelength of edge *e*

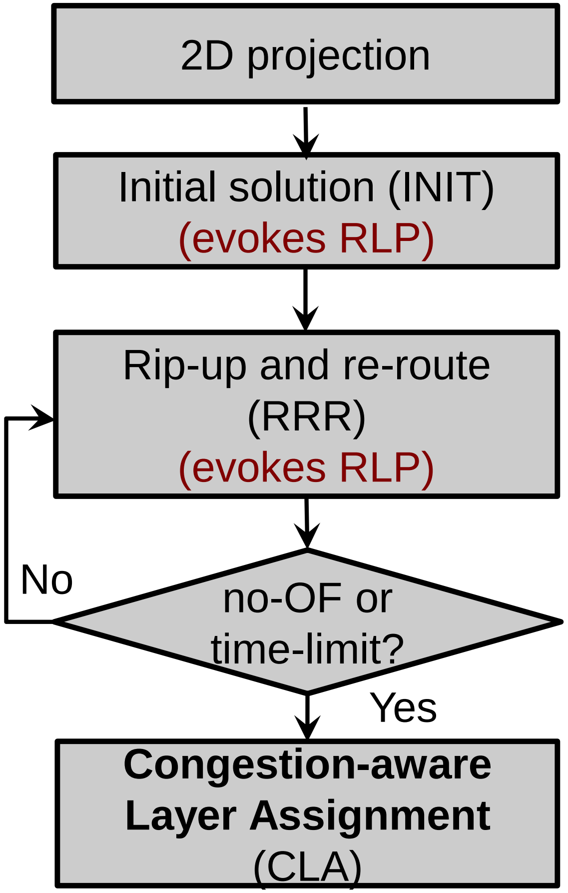


      - computed as distance of centers of two corresponding gcells
      - may be different among the edges due to non-uniform binning
    - g_e: utilization of edge *e*
      - estimated same way as CGRIP
    - u_e: capacity of edge *e*
  - vertex weight $f(\frac{h_v}{r_v})$; $\forall v \in V_{2D}$
    - h_v and r_v utilization and capacity of vertex *v*, respectively



Integration with CGRIP

- CLA
  - In CGRIP the routes in G_{2D} are converted into routes on G
    - using a congestion-aware layer assignment greedy procedure
      - while accounting for wire sizes, routing blockages and virtual pins
  - Extensions:
    1. use updated edge capacities accounting for local congestion using our model
       - assuming the local nets are routed at the two lowest layers
    2. computing the utilization is extended to account for non-uniform gcell dimensions
       - the routing resource of edge $e \in G$ is computed using an estimated length l_e and the corresponding wire size for layer l



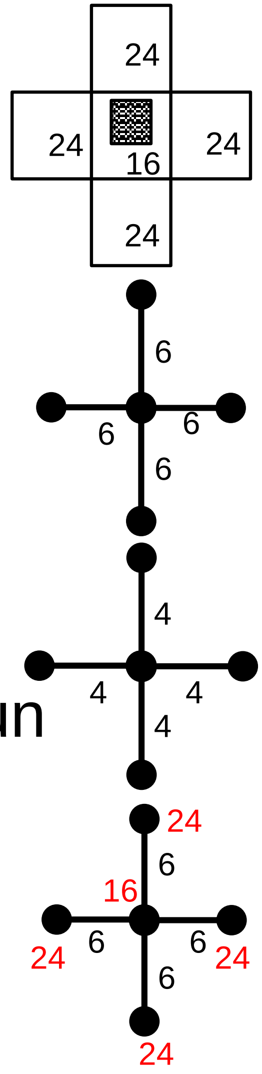
Simulation Setup

- Experimented with the ISPD 2011 benchmarks
  - Have large number of local nets when using the winning solutions from ISPD 2011 contest on routability-driven placement

Design	X x Y	# Nets	# 2T-Nets	%LC	
				Uniform	Non-uniform
superblue1	704x516	822744	2038444	30.8	14.1
superblue2	770x1114	990899	2237446	28.9	13.5
superblue4	467x415	567607	1316401	35.2	16.8
superblue5	774x713	786999	1713307	29.4	12.2
superblue10	638x968	1085737	2579974	34.1	34.0
superblue12	444x518	1293436	3480633	28.6	14.6
superblue15	399x495	1080409	2736271	34.4	15.8
superblue18	381x404	468918	1395388	28.2	15.0
average				**31.2%**	**17.0%**

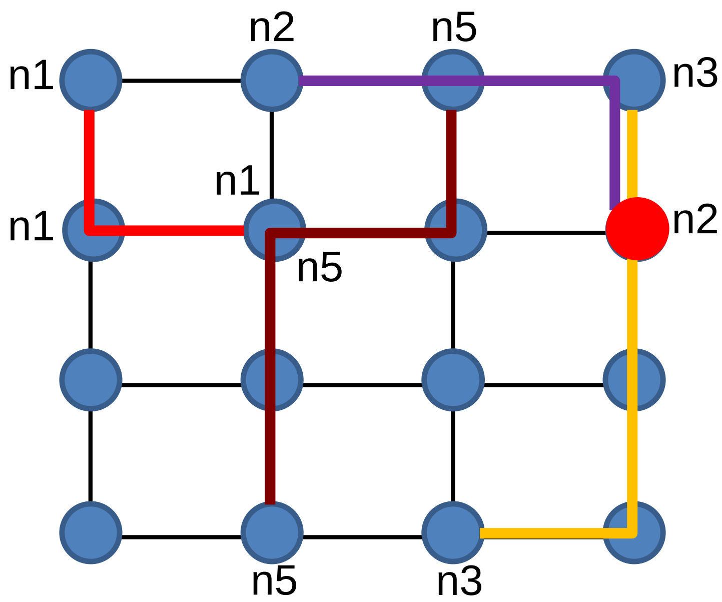
Simulation Setup

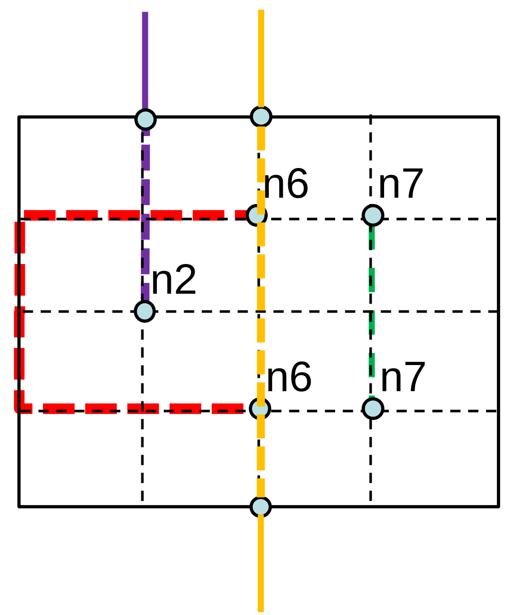
- Experimented with the following variations
  - **U-E (CGRIP)**: Uniform grid with Edge capacity only
  - **U-AE**: Uniform grid with Adjusted Edge capacity
    - no vertex capacity
  - **U-AV**: Uniform grid with Adjusted Vertex capacity
    - with (unreduced edge capacity)
  - **NU-AV (LCGRIP)**: Non-Uniform grid with Adjusted Vertex capacity
    - similar to U-AV but with non-uniform global cells
- For each variation we allow our framework to run until there is no further overflow improvement



Simulation Setup

- Detailed routing emulator
  - There is no way to pass a GR solution as input to a commercial DR
  - We implemented our own detailed routing *emulator* for validation
    - visits the gcells sequentially on a 2D projected model
    - applies <u>one</u> iteration of RRR to route the nets inside each gcell
      - reflects immediate picture right at the beginning of the DR stage





Simulation Setup

- Evaluation metrics
  - **GR-OF**: overflow of GR with un-adjusted capacities
  - **DR-OF**: total overflow computed by our detailed routing emulator
  - **GR-WL**
    - In NU-AV, the wirelength is computed while accounting for non-uniform gcells for fair comparison
      - an edge in NU-AV which is twice than an edge in U-E due to non-uniform gcells is counted as 2 units of wirelength

Simulation Results: Overflow

GR-OF

Design	U-E (CGRIP)	U-AE	U-AV	NU-AV (LCGRIP)
superblue1	0	0	0	0
superblue2	3168	14496	10526	7756
superblue4	228	2024	880	420
superblue5	0	322	0	448
superblue10	124	4502	872	766
superblue12	0	274	302	12232
superblue15	0	1022	846	2582
superblue18	0	0	0	0
average	**1.0X**	**6.4X**	**3.8X**	**6.9X**

DR-OF

Design	U-E (CGRIP)	U-AE	U-AV	NU-AV (LCGRIP)
superblue1	23142	23020	12740	806
superblue2	18880	18506	13154	9140
superblue4	28696	27476	13296	888
superblue5	10878	9256	2588	1032
superblue10	84842	73862	66780	65232
superblue12	44556	44416	36414	15566
superblue15	29982	29800	18886	7922
superblue18	11406	11184	558	444
average	**1.0x**	**0.9x**	**0.7x**	**0.4x**

The GR-WL of U-AE, U-AV, NU-AV are up to 1% larger than U-E

Simulation Results: Runtime (min)

GR-T

Design	U-E (CGRIP)	U-AE	U-AV	NU-AV (LCGRIP)
superblue1	3	7	5	7
superblue2	352	321	303	381
superblue4	180	60	201	62
superblue5	135	184	164	220
superblue10	251	341	329	342
superblue12	238	360	309	302
superblue15	212	269	259	221
superblue18	10	20	16	10
average	**1.0X**	**1.1X**	**1.1X**	**1.1X**

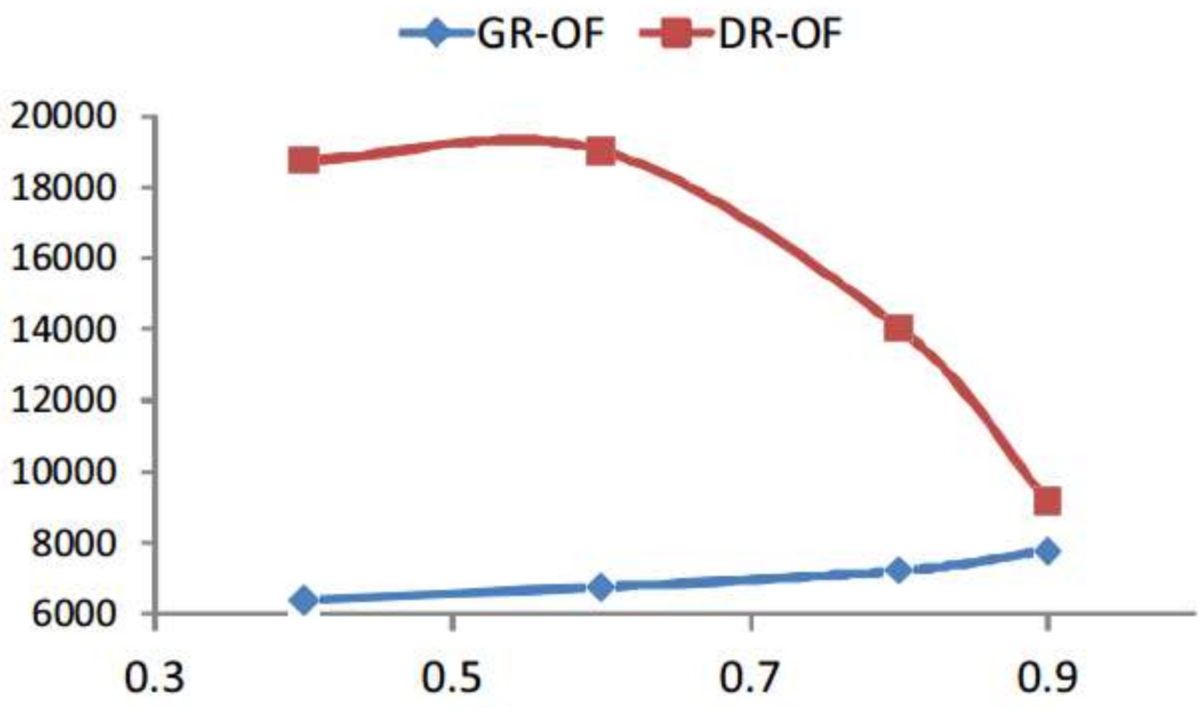
DR-T

Design	U-E (CGRIP)	U-AE	U-AV	NU-AV (LCGRIP)
superblue1	28	21	18	7
superblue2	22	17	17	16
superblue4	39	25	25	10
superblue5	42	33	24	4
superblue10	62	51	32	33
superblue12	41	42	37	22
superblue15	34	24	19	11
superblue18	32	20	15	9
average	**1.0X**	**0.8X**	**0.6X**	**0.4X**

Increase in effort in GR results in significant decrease in (single iteration) DR

Tradeoff with η

- Tradeoff in DR-OF and GR-OF with input parameter η in Superblue2



Conclusions

- Proposed two techniques for considering local nets
  - reducing the number of local nets by non-uniform binning
  - approximating their routing usage by adding a vertex capacity
    - Our work can consider other models of local net routing usage in a gcell as long as it can be translated into a single usage number
      - this modeling was not the main focus of our work
- Showed significant reduction in the detailed routing effort using our emulator
  - traded off with increase in effort in the global routing stage
- Other factors contributing to unroutability (i.e. wire sizes, routing blockages, virtual pins) also important
  - should be considered along with local nets
  - (See paper for details of related experiment)

Thank You

Visit CGRIP's page for more information:

http://homepages.cae.wisc.edu/~adavoodi/gr/cgrip.htm

or google "CGRIP"