A Parallel Integer Programming Approach to Global Routing

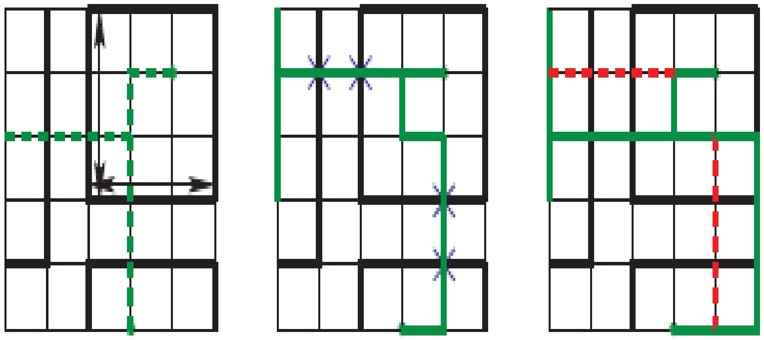
T-H. Wu, A. Davoodi and J. Linderoth
University of Wisconsin, Madison

GRIP:IP-based global router

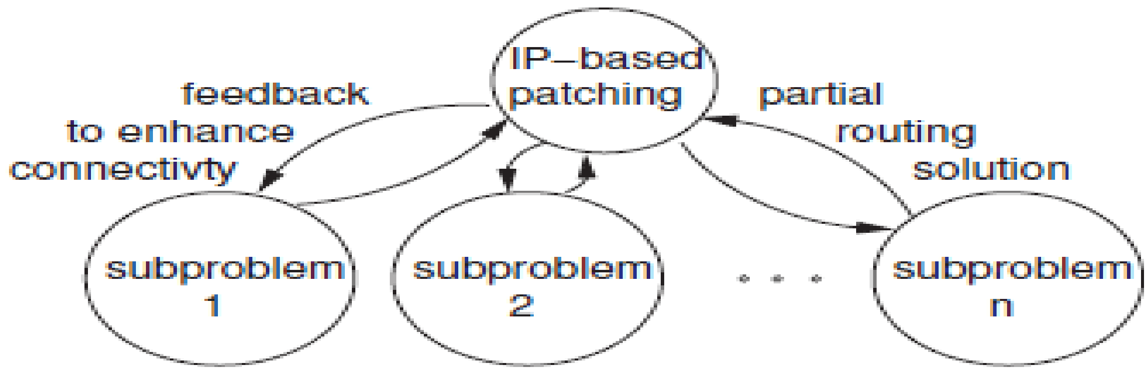
$$\min_{x,s} \sum_{i=1}^{N} \sum_{t \in \mathcal{T}(T_i)} c_{it} x_{it} + \sum_{\forall e \in E} Q_e o_e \qquad \text{(ILP-GR)}$$

$$\begin{cases} \sum_{t \in \mathcal{T}(T_i)} x_{it} = 1 & \forall i = 1, \ldots, N \\ \sum_{i=1}^{N} \sum_{t \in \mathcal{T}(T_i)} a_{te} x_{it} \le u_e + o_e & \forall e \in E \\ x_{it} = \{0, 1\} & \forall i = 1, \ldots, N, \forall t \in \mathcal{T}(T_i), \\ o_e \ge 0 & \forall e \in E. \end{cases}$$

- Solved separately for rectangular subregions by price and bound.
- Limited parallelism since only non-neighboring subproblems can be concurrently solved.



Parallel Router:Overview



Key Steps:

- Sub-problem definition
- Initial pricing
- Patching
- Repricing
- Parallel reconnection of neighboring sub-problems

Parallel Router: Initial Steps

- Sub-problem Definition

  - Generate candidate sols using Flute, solve ILP-GR with randomized rounding

  - Bi-partitioning to balance nets

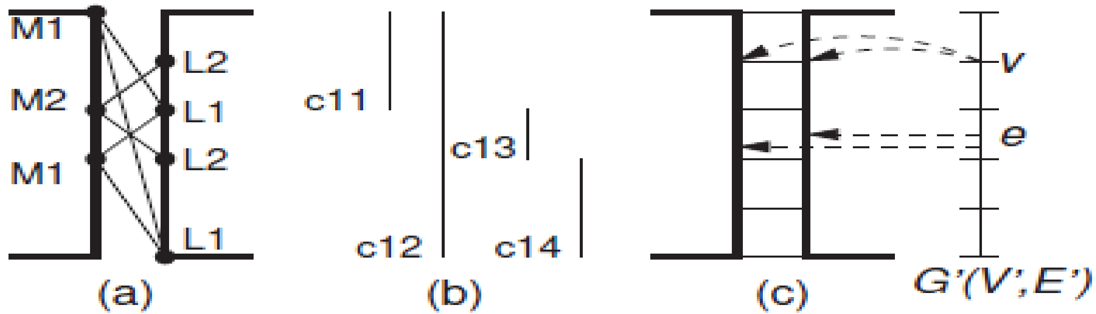
  - Detouring of high overflow nets

- Initial Pricing

  - Inter-region nets connected anywhere on subproblem boundary

  - Larger overflow penalty at boundary.

Parallel Router: Patching



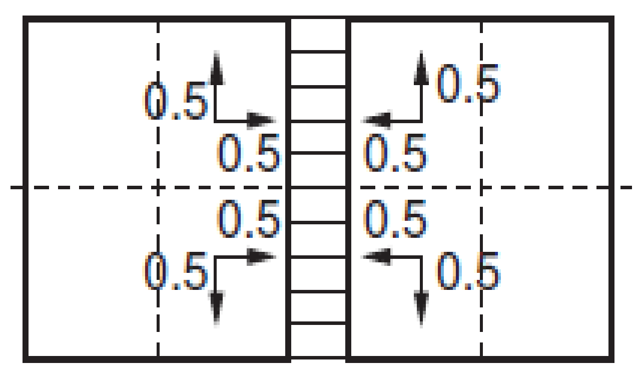
$$\min_{x} \sum_{i=1}^{N} \sum_{t=1}^{|L_i|\times|M_i|} c_{it}x_{it} + \sum_{i=1}^{N} Qs_i \qquad \text{(ILP-PATCH)}$$

$$\begin{cases} \sum_{t=1}^{|L_i|\times|M_i|} x_{it} + s_i = 1 & \forall i = 1, \ldots, N \\ \sum_{i=1}^{N} \sum_{t=1}^{|L_i|\times|M_i|} a_{te}x_{it} \le u_e & \forall e \in E'_v \\ x_{it} = \{0, 1\} & \forall i = 1, \ldots, N, \forall t = 1, \ldots, |L_i||M_i|. \end{cases}$$

- Q set high to keep slack variable zero
- If si > 0, net i hard to route and given entire window

Parallel Router: Final Steps

- Re-pricing
  - Candidate routes generated with specified window constraints
- Parallel conneting of sub-problems
  - Fix nets inside subproblems and inter-region net "backbones"
  - Connect "backbones" using IP-based price and bound

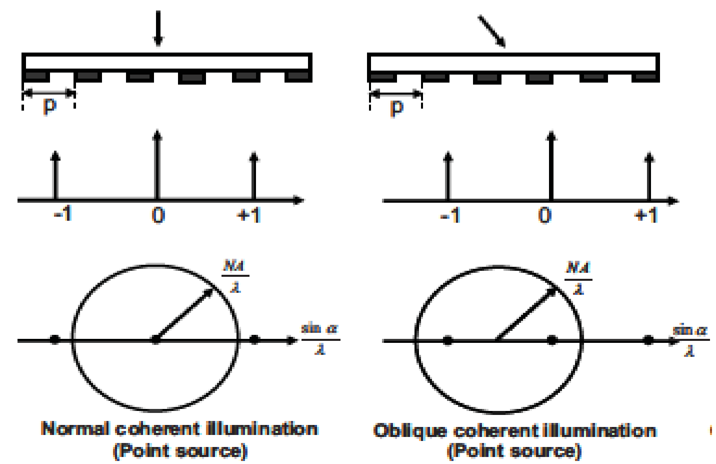


Frequency Domain Decomposition of Layouts for Double Dipole Decomposition

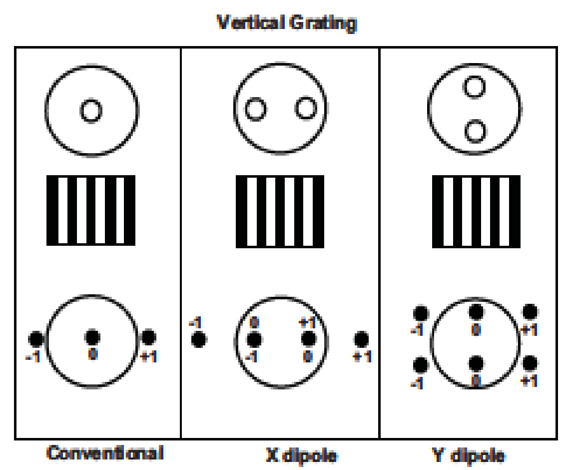
Kanak Agarwal
IBM

Dipole Illumination

- OAI offers better resolution by shifting the diffraction orders
  - Need to capture at least 1st order



- Dipole illumination improves resolution for patterns perpendicular to dipole axis

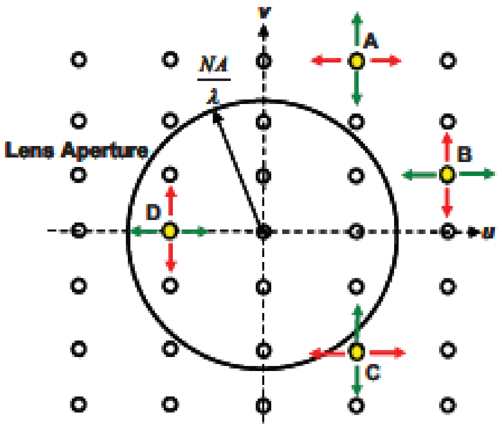


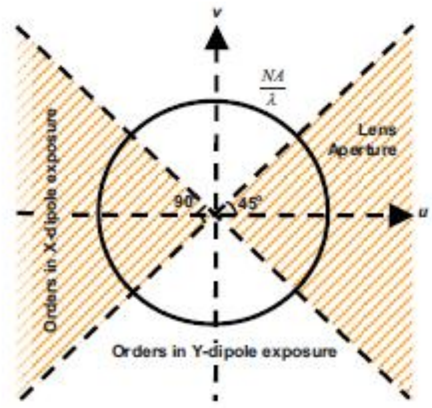
Double Dipole Lithography

Split layout to vertical+horizontal features -> X dipole exposure for vertical -> Y dipole exposure for horizontal

- Incurs cost of 2 masks like DPL
- But, only one etch step, unlike LELE DPL
    - -Intensities from two exposures can interact
- Layout decomposition seems trivial but jogs, line ends hard to classify
- This paper proposes frequency domain decomposition.

DDL Decomposition





$$M(u,v) = F\{m(x,y)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} m(x,y)e^{-j2\pi(ux+vy)}dxdy$$

$$H(u,v) = \begin{cases} 1 & \text{if } \sqrt{u^2+v^2} < \dfrac{NA}{\lambda} \\ 0 & \text{otherwise} \end{cases}$$

$$A(x,y) = F^{-1}\{M(u,v).H(u,v)\}$$

$$A_X(x,y) = F^{-1}\{M(u-u_0,v).H(u,v)\}$$

$$A_Y(x,y) = F^{-1}\{M(u,v-v_0).H(u,v)\}$$

$$m_X(x,y) = F^{-1}\{M(u,v).H_X(u,v)\}$$

$$m_Y(x,y) = F^{-1}\{M(u,v).H_Y(u,v)\}$$

Fixing non-binary splits

- For each pixel, magnitude of the two IFTs compared and each pixel is then assigned an exposure accordingly
- If equal, assigned to both exposures
- Aerial image simulation results indicate more robust decomposition with lesser hotspots