Incremental Layer Assignment for Critical Path Timing

Derong Liu, Bei Yu, Salim Chowdhury, and David Z. Pan

ECE Dept., University of Texas at Austin; †CSE Dept., Chinese University of Hong Kong; ‡Oracle Corp.

**Motivation and Problem**

- Layer assignment assigns segments to metal layers after 2-D global routing.

**CPLA Algorithms**

**ILP Formulation**

\[
\min \sum_{(i,j) \in E} \sum_{q \in Q} t_{ij}(q) x_{ij} + \sum_{p \in P} \sum_{q \in Q} t_{p}(q) y_{pq}
\]

Binary variables: \( x_{ij} \) represents segment \( i \) assigned on layer \( j \);
\( y_{pq} \) represents the via connecting segment \( p \) from layer \( j \) to layer \( q \), which is equal to the product of \( x_{ij} \) and \( x_{jq} \).

**Constraints**

- Each released segment to be assigned.
  \[
  \sum_{j \in J(i)} x_{ij} = 1, \quad \forall i \in \mathbb{N}(N) \setminus \{s_1\}
  \]

- Edge capacity constraint:
  \[
  \sum_{(i,j) \in E} x_{ij} + x_{jq} \leq \text{cap}(j), \quad \forall j \in \mathbb{N}(J) \setminus \{s_1, s_2, s_3\}
  \]

- Via capacity constraint:
  \[
  \sum_{(p,q) \in V} y_{pq} \leq \text{cap}(p), \quad \forall p \in \mathbb{N}(P) \setminus \{s_1, s_2, s_3\}
  \]

**Overall Algorithm Flow**

Post mapping transfers continuous solutions into discrete assignment result.

**Experimental Results**

- CPLA implemented in C++, Gurobi as the MILP solver and CSDP as the SDP solver.
- Linux machine with 2.9GHz Intel(R) Core and 192GB memory.
- Initial routing and layer assignment result from NCTU-GR and NVM tool.

**Semidefinite Programming Relaxation**

The proposed self-adaptive partition provides an opportunity for further speed-up. Semidefinite programming (SDP) is solvable in polynomial time while providing a theoretically better solution than Linear Programming (LP).

- The objective function:
  \[
  \min \{T(X)\}
  \]

- Matrix \( T \) = (S, L)-dimension symmetric matrix representing timing costs.
- Matrix \( X \) = (S, L)-dimension symmetric matrix representing variables.

\[
T = \begin{pmatrix}
    & X_{11} & X_{12} & \cdots & X_{1L} \\
    & \vdots & \vdots & \ddots & \vdots \\
    & X_{L1} & X_{L2} & \cdots & X_{LL}
\end{pmatrix}
\]

**Semidefinite Programming Example**

Cost matrix \( T \) and solution matrix \( X \) of this example:

\[
T = \begin{pmatrix}
    35 & 58 & 6.7 \\
    58 & 35 & 6.7 \\
    6.7 & 6.7 & 23.9
\end{pmatrix}, \quad X = \begin{pmatrix}
    0.01 & 0 & 0 \cdots 0 & 0.001 & 0.001 & 0.001 & 0.001 & 0.001 \\
    0 & 0 & 0 \cdots 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

Segment \( S2 \) overlaps with other segments, resulting in continuous solutions.

Therefore, post mapping is required to provide integer solutions.

**Conclusion**

This work is supported in part by NSF, Oracle, and CIUHK Direct Grant for Research.