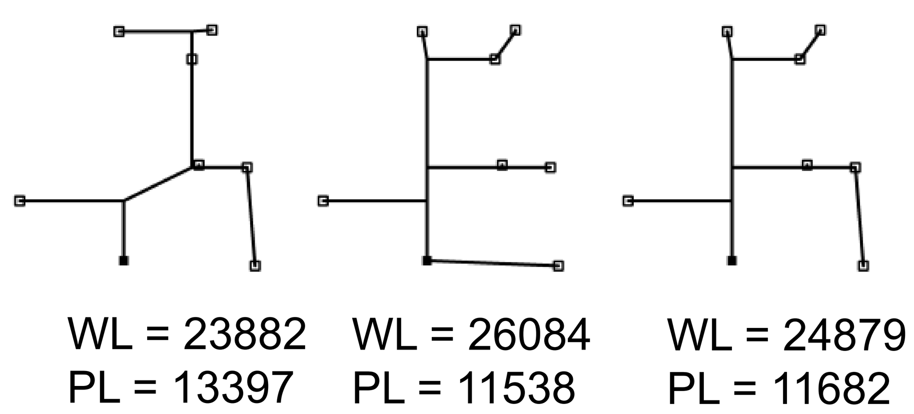
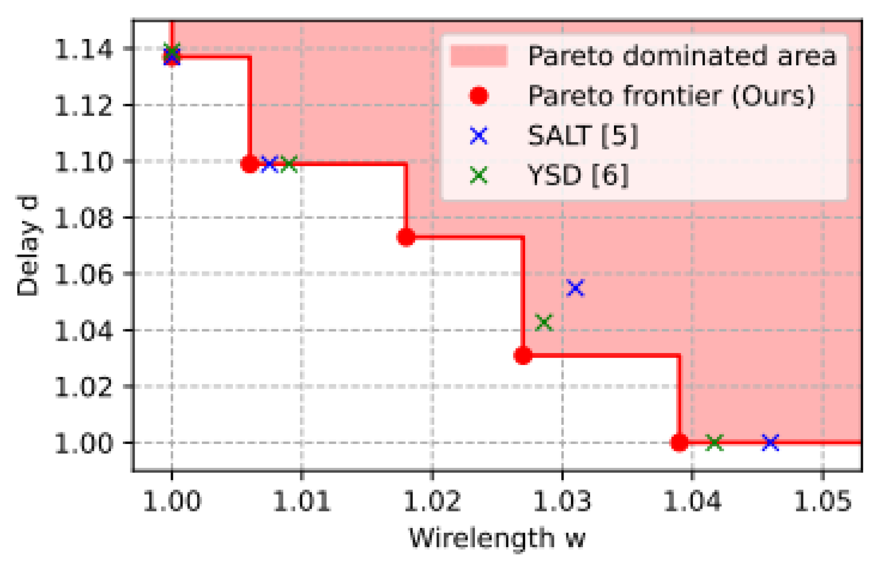
PatLabor: Pareto Optimization of Timing-Driven Routing Trees

Zhiyang Chen, Hailong Yao, Xia Yin
Tsinghua University, University of Science and Technology Beijing

Introduction





- Timing is a critical performance metric for modern VLSI design
- As technology advances, wire delay becomes a significant factor
- Timing-driven routing trees: Given a set of pins, construct a rectilinear Steiner routing tree that minimizes:
  - Wirelength: the total weight of the routing tree
  - Pathlength: the maximum path length from the source to all sinks

Previous Work

- Prim-Dijkstra II [Alpert et al., ISPD'18]
- SALT [Chen & Young, TCAD'19]
- Neural network method [Yang, Sun & Ding, ICCAD'23]

- Previous work balances wirelength and pathlength via
  - optimizing a weighted sum of two objectives;
  - greedy heuristics.
- Thus, previous methods
  - produce a single solution for one configuration of the parameter (requires parameter tuning);
  - have no optimality guarantees.

Why Pareto Optimization?

- Given two different solutions x & x', x is better than x' (in the Pareto sense) if
  - both WL(x) ≤ WL(x') & PL(x) ≤ PL(x')
- The optimal set of solutions is known as the Pareto frontier

- Can we directly obtain the Pareto frontier of timing-driven routing?
  - Compute the Pareto set using only one run, no parameter tuning;
  - Recent work on global routing [Li et al., DAC'24] shows that selecting net topologies from a candidate set may improve the global routing performance;
  - Overcome the weakness of weighted sum methods that only computes convex Pareto curves.

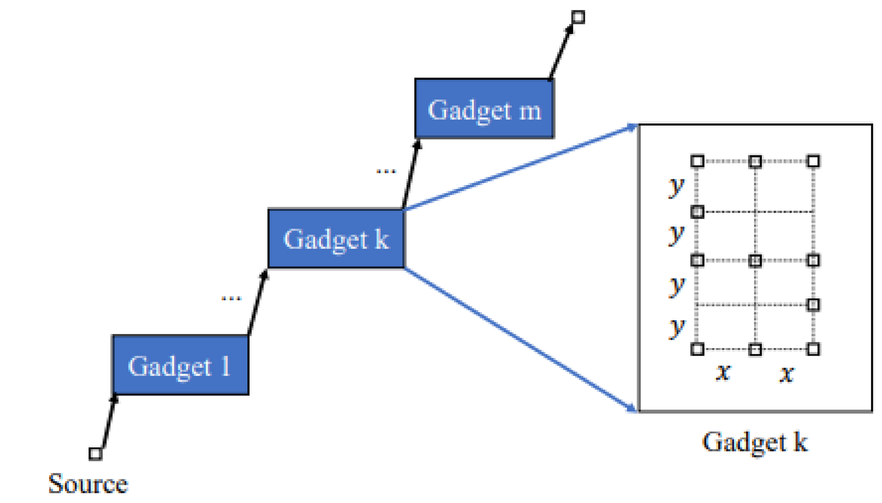
Summary of Our Contributions

- **PatLabor** (Pareto Optimization with Lookup tables and local search)
  - is the first work that directly optimizes the Pareto curve of timing-driven routing trees (parameter-free);
  - guarantees optimality for nets with ≤ 9 pins (motivated by lookup tables in FLUTE [Chu & Wong, TCAD'08]);
  - provably approximates the Pareto frontier for nets with > 9 pins;
  - obtains tighter Pareto curves than previous methods in experimental results.

Theoretical Analysis

- Pareto optimization is good. But is it tractable?

- Bad news
  - **Theorem**: *There exists a timing-driven routing instance such that the Pareto frontier size is* $2^{\Omega(n)}$.
  - In the worst-case, no polynomial-time algorithm even if P=NP!



The bad instance construction

- Good news
  - Pins in real-world designs contains some randomness (a.k.a., smoothed analysis in algorithms [Spielman & Teng, JACM'09]).
  - **Theorem**: *If the pin positions are randomly perturbed by κ-bounded noises, the expected solution number is only* $O(n^3\kappa)$.
  - In practice, the solution number is mild (only polynomial)!

Our Algorithm

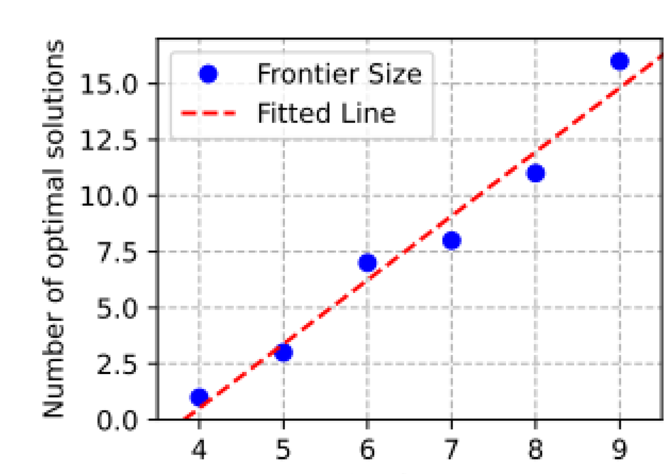
PatLabor:

- For small-degree nets, we design a dynamic-programming-based algorithm.

$$S_{v,Q} = \text{Pareto}\begin{cases} \cup_u \{S_{u,Q} + \|u - v\|_1\}, \\ \cup_{Q_1 \subseteq Q} \{S_{v,Q_1} \oplus S_{v,Q \setminus Q_1}\}. \end{cases}$$

  - Time complexity $\boldsymbol{O}^*(\boldsymbol{3^n}|\boldsymbol{S}|)$, where |S| is the Pareto frontier size.
- Further acceleration?
  - We use this algorithm to generate all possibly optimal solution net topologies for n≤9 (lookup tables).
  - Some advanced pruning techniques to improve efficiency.
- For large-degree nets, we design a local search heuristic.
  - Use lookup tables to regenerate routes for pins with large path lengths.
  - $\boldsymbol{O}(\sqrt{\boldsymbol{n}/\log \boldsymbol{n}})$-approximation in $\widetilde{\boldsymbol{O}}(\boldsymbol{n}^2|\boldsymbol{S}|^2)$ time.

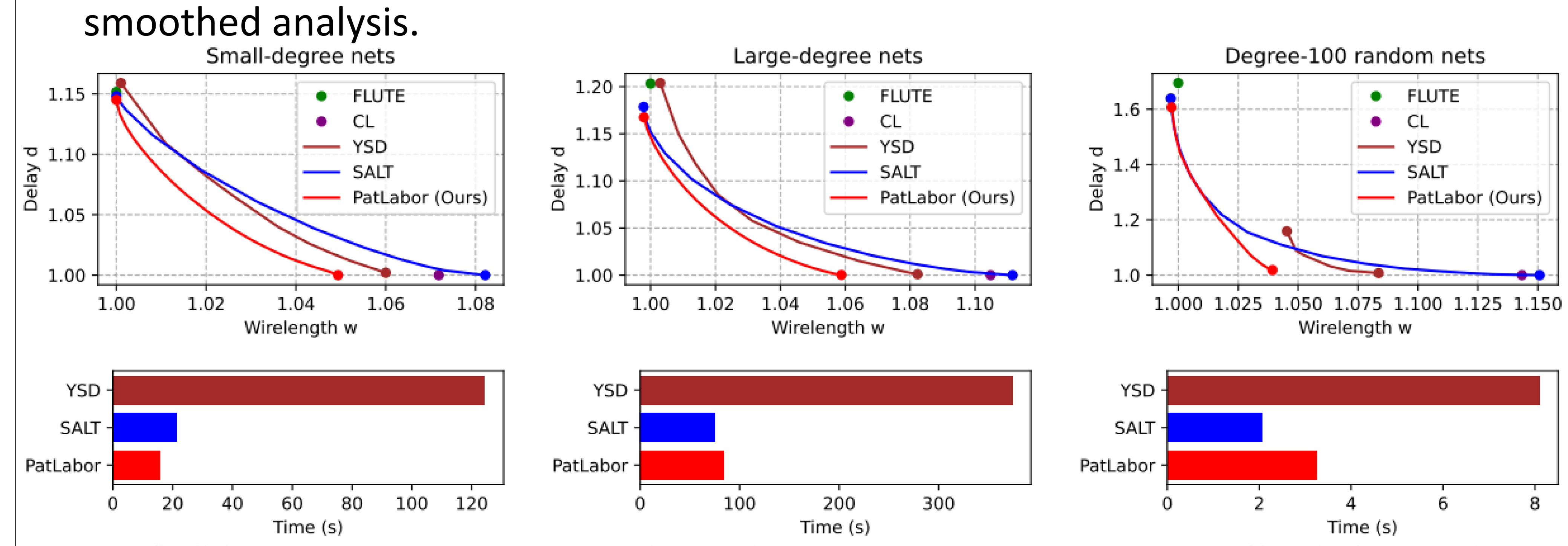
Experiments



Number of optimal solutions:
Mild growth compared with the worst-case exponential bound.
Verifying the effectiveness of our smoothed analysis.

Degree	#Index	#Topo	Size (MB)	Time
4	24	1.67	< 0.01	0s
5	220	4.6	< 0.01	0s
6	1008	10.67	< 0.01	0s
7	5824	32.52	0.19	4.9s
8	46880	107.05	6.23	276s
9	429516	378.05	240	4.68h
Total	483472	-	246	4.76h

Lookup Table generation:
Generate lookup tables ~246MB in <5 hours.
Over 1.7×10^8 topologies in total for 483k indices.



(a) Small-degree nets ($n \le 9$) (b) Large-degree nets ($n > 9$) (c) Degree-100 randomly generated nets

Pareto curve comparison:
Evaluations on ICCAD'15 benchmarks and synthesized degree-100 nets.
Tighter curves compared with previous methods.