Any-Angle Die-to-Die Routing for Advanced Packages with Asymmetric Pin Row Structures, Via Constraints, and Shielding-Aware Reservation

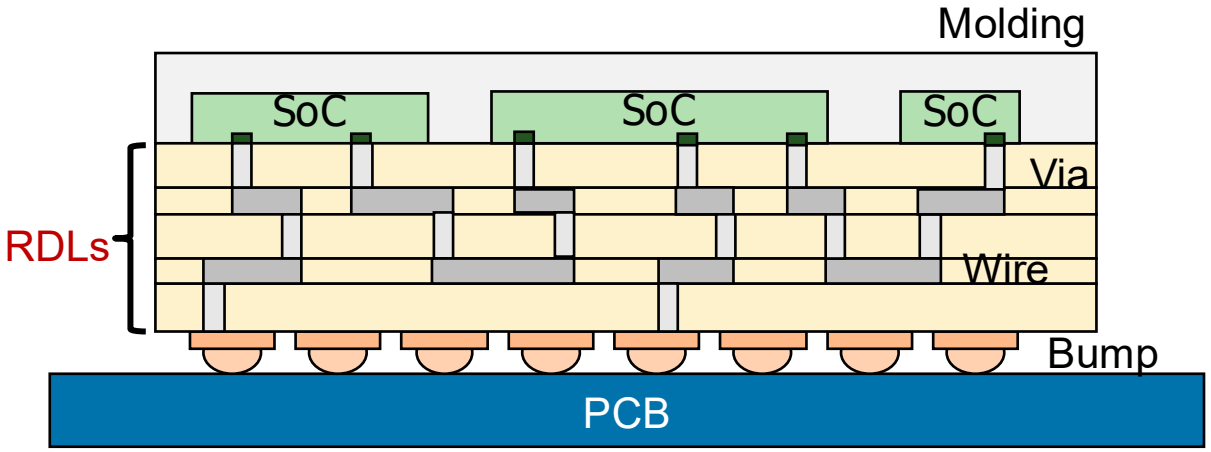
Hsin-Tzu Chang[1], Iris Hui-Ru Jiang[1], Hua-Yu Chang[2], Chun-Hao Lai[1]

Graduate Institute of Electronics Engineering, National Taiwan University[1]

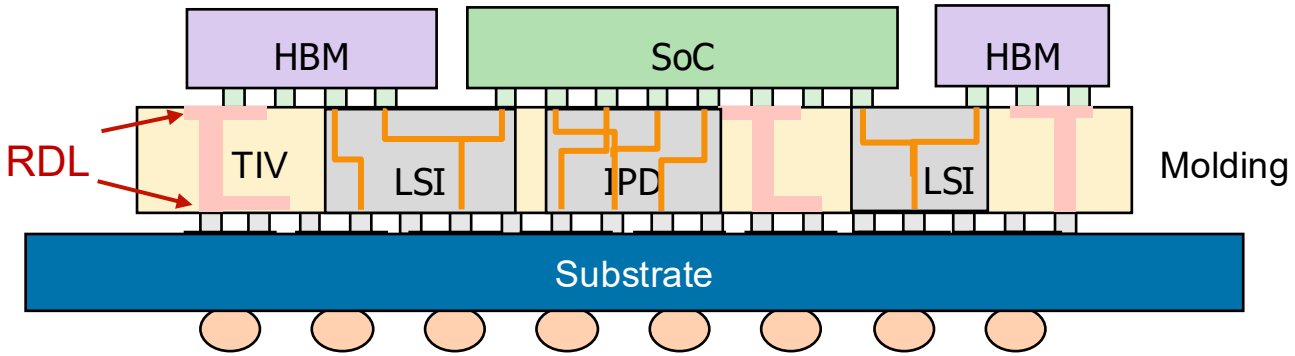
Design Technology Group, Synopsys, Inc., Taipei, Taiwan[2]

Advanced Package

- Heterogeneous Integration
- Interconnections with Redistribution Layers (RDLs)
  - Multi-layer RDL routing and via placement

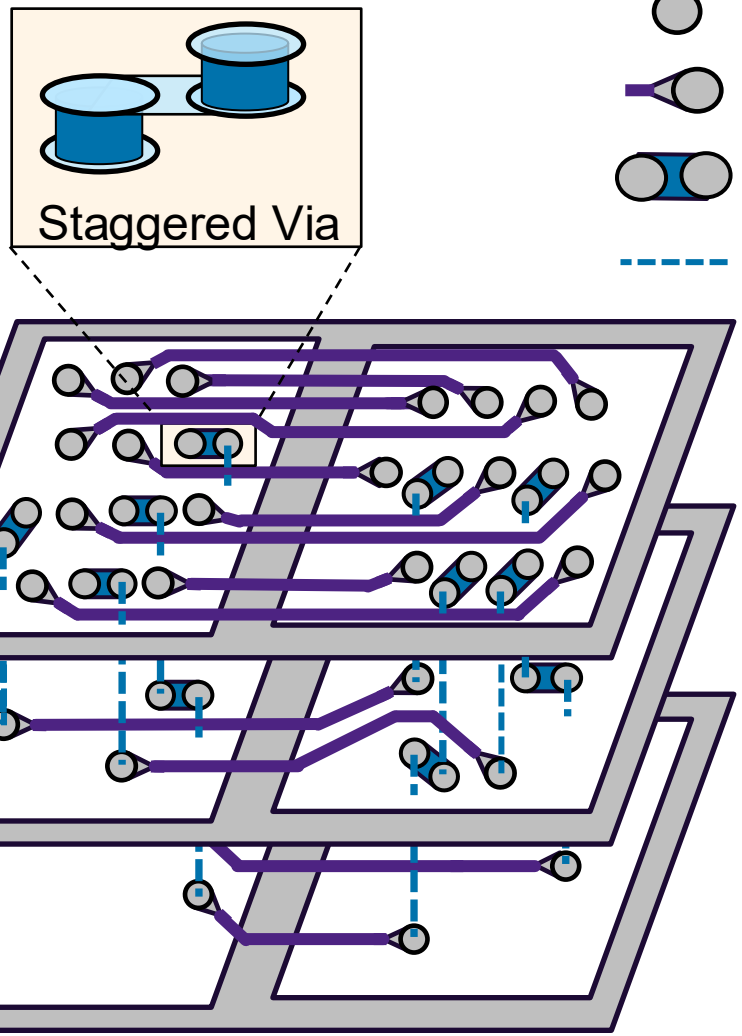


InFO Package

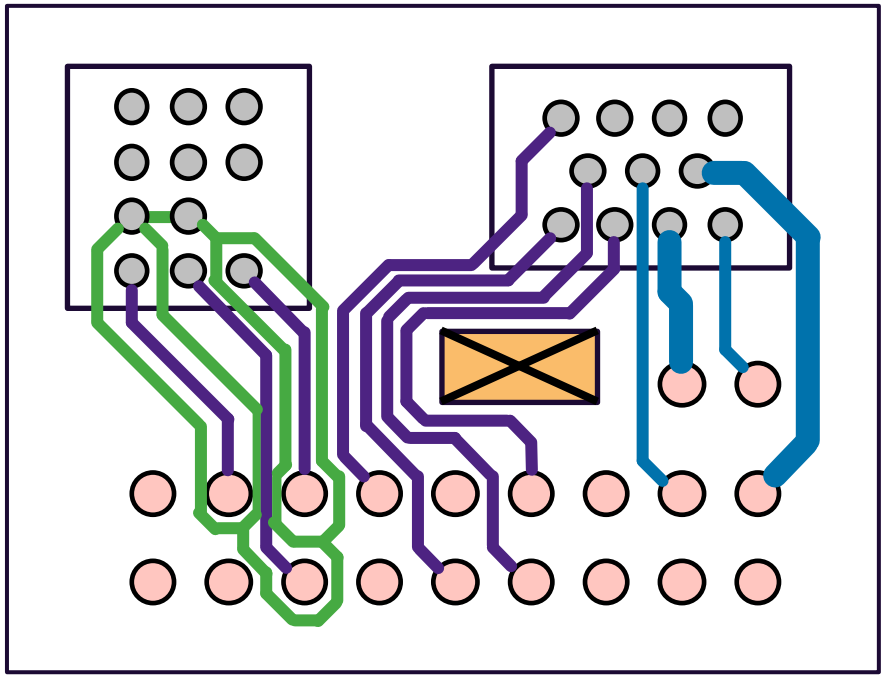


CoWoS®-L Package

D2D vs. D2S Routing



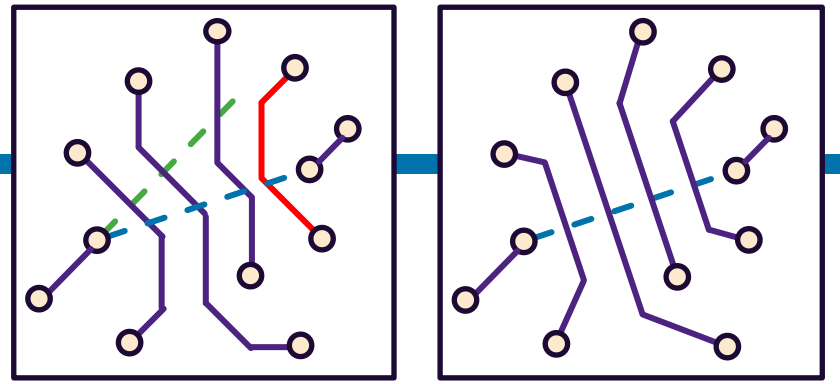
D2D Routing

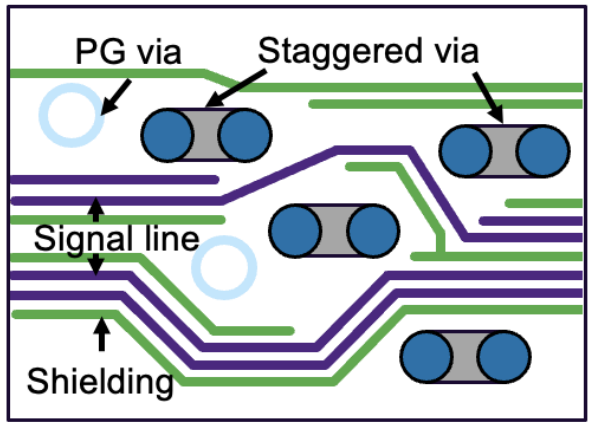


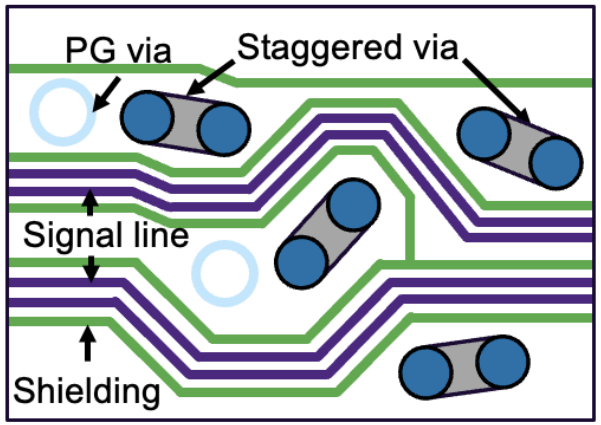
D2S Routing

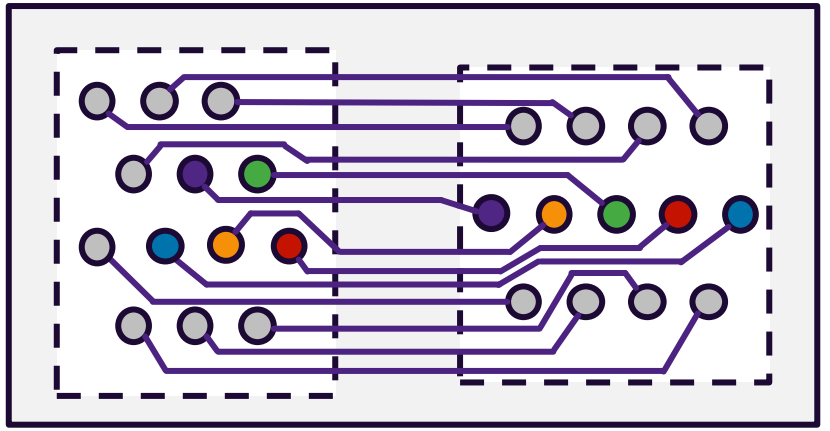
Challenges in D2D Routing

- Ultra-dense D2D connections
- Vias consume routing resources
  - Large via size
  - Staggered vias & teardrops
- Any-angle routing
- Asymmetric pin row structure
  - Arbitrary pin order
  - uneven row counts
- Full shielding
  - Signal integrity
  - Needs space reservation









Comparison of RDL Routing Works for Advanced Package

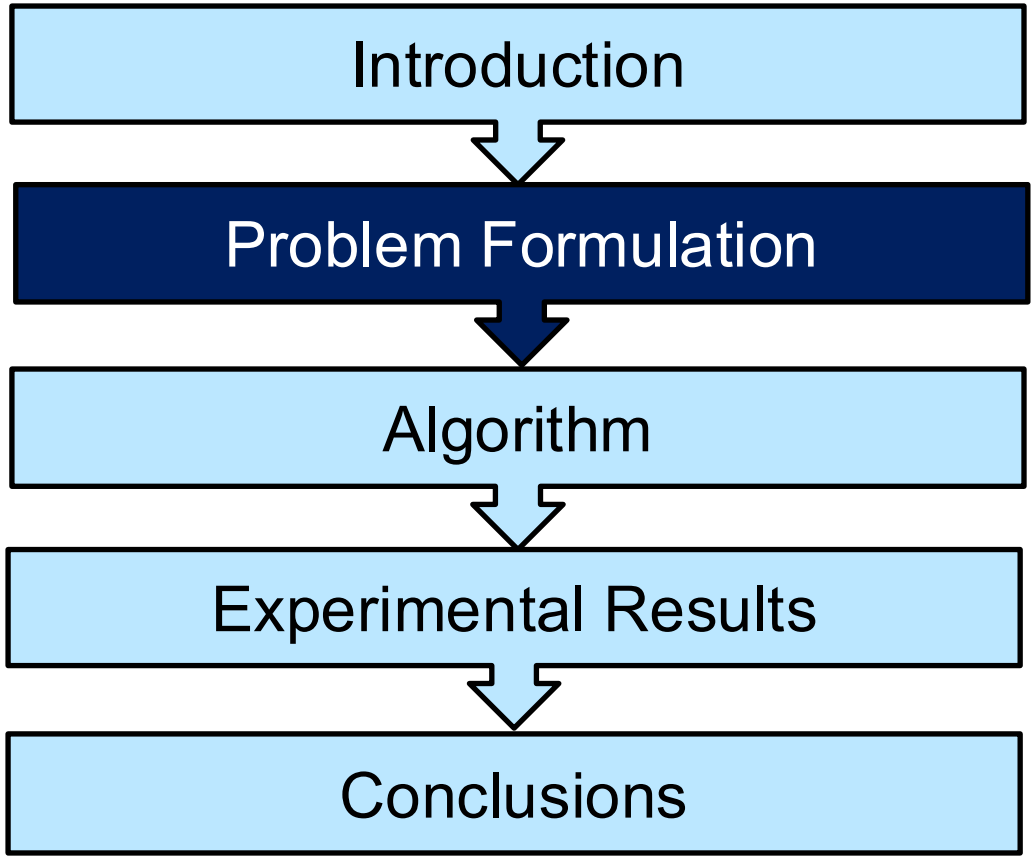
	D2S Routing						D2D Routing	
	Lin+ [ICCAD'16]	Cai+ [DAC'21]	Wen+ [TCAD'22]	Chen+ [ICCAD'22]	Chung+ [DAC'23]	Chuang+ [DAC'24]	Chang+ [ASPDAC'25]	**This Work**
Any-angle	No	No	No	No	Yes	No	Yes	**Yes**
Routing Strategy	Sequential	Sequential	Sequential	Sequential	Sequential	Sequential	Sequential	**Concurrent**
Via Planning	Not Considered	Pre-planned	Pre-planned	Pre-planned	Pre-planned	Dynamic Insertion	Pre-planned Offset Via (1st Layer)	**Routing Co-optimization**
Routing Graph Model	Grid	VD + Convex cells	Fan-out grid	CDT	CDT	Tile-based 3D grid	DT	**Tri-Trap**
Staggered Via / Teardrop	No	No	No	No	No	No	Yes	**Yes**
Asymmetric Pin Row	No	No	No	No	No	No	No	**Yes**

- VD: Voronoi Diagram
- (C)DT: (Constrained) Delaunay Triangulation
- Tri-Trap: Triangle-Trapezoid hybrid model

Contributions

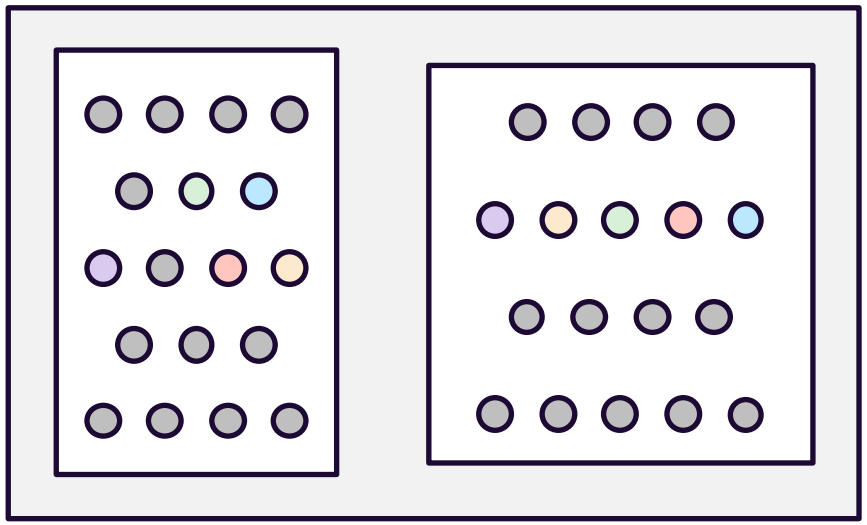
- First general dense D2D routing framework
  - Handles asymmetric pin-row structure, via constraints and shielding reservation
- Globally optimized Concurrent route & Via-routing co-optimization framework
- Efficient acceleration techniques
  - Crossing pair identification & Net grouping to reduce problem size in routing path candidate generation
- High-quality and scalable routing results
  - Overflow-free & DRV-free with short wirelength
  - Efficient empirical time complexity

Outline

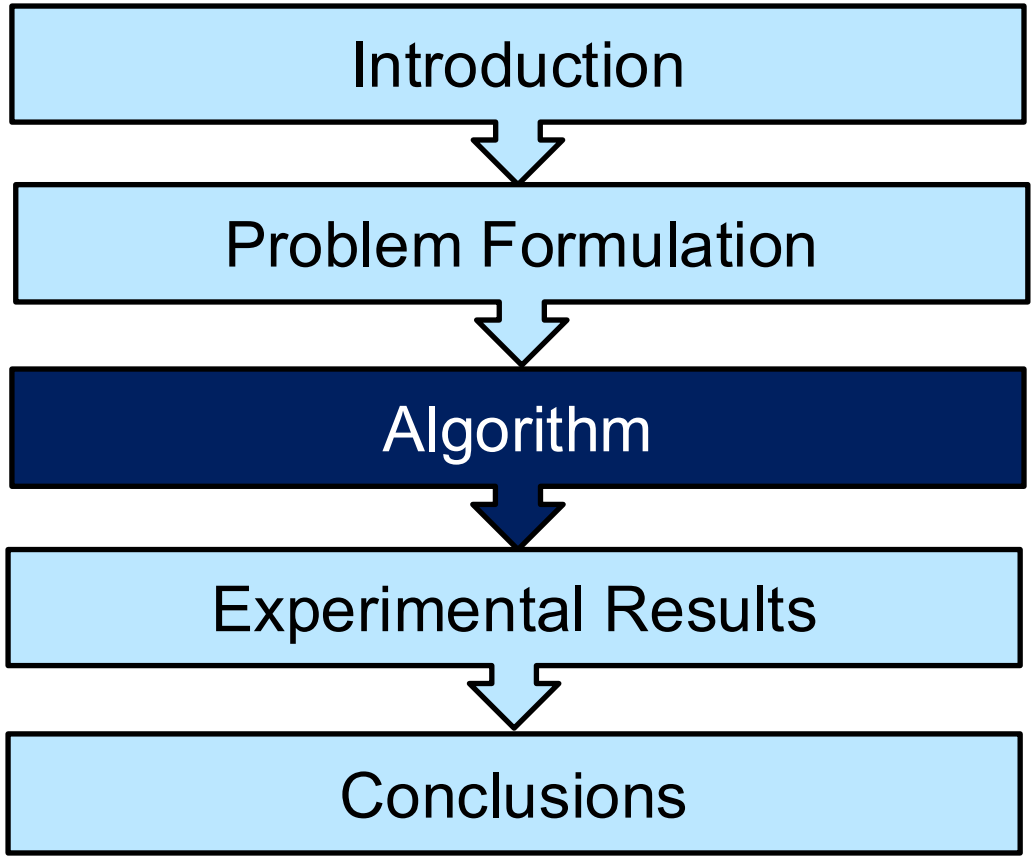

Introduction
Problem Formulation
Algorithm
Experimental Results
Conclusions

Problem Formulation

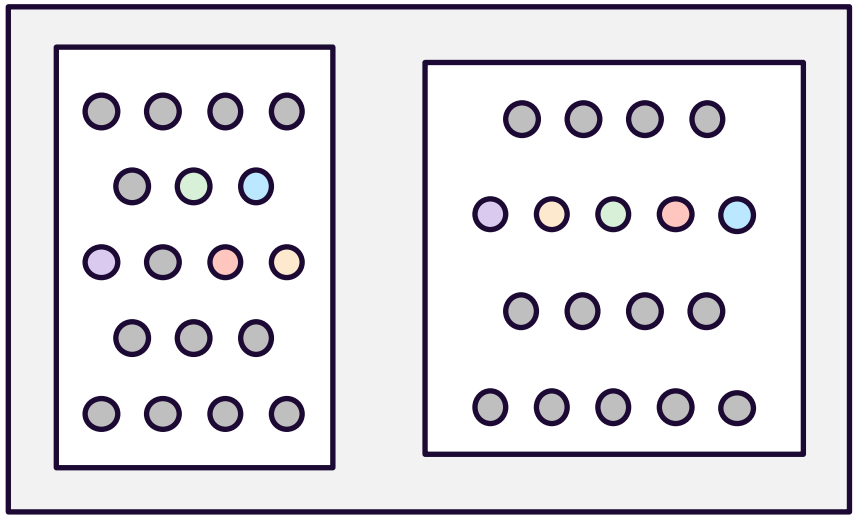
- Given:
  - A two-pin netlist with pre-placed pins of D2D design
  - Routing boundary
  - A set of design rule constraints
    - Spacing constraints (wire-wire, wire-via, via-via)
    - Maximum allowed consecutive stacked vias
    - Teardrop length and staggered via length
- Output:
  - Overflow-free & DRV-free any-angle routing result
- Objective:
  - Minimize routing layer usage
  - Minimize total wirelength

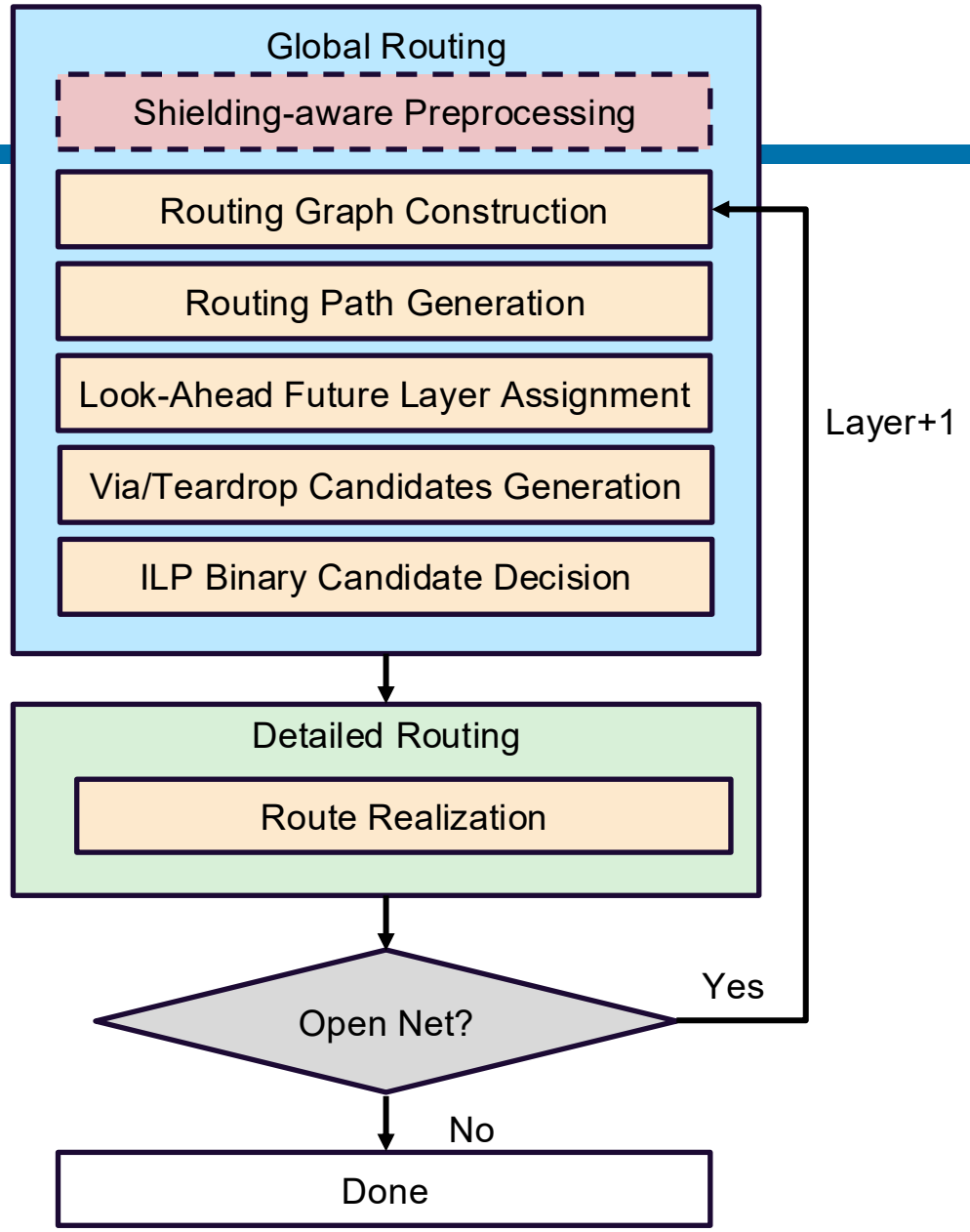


Outline

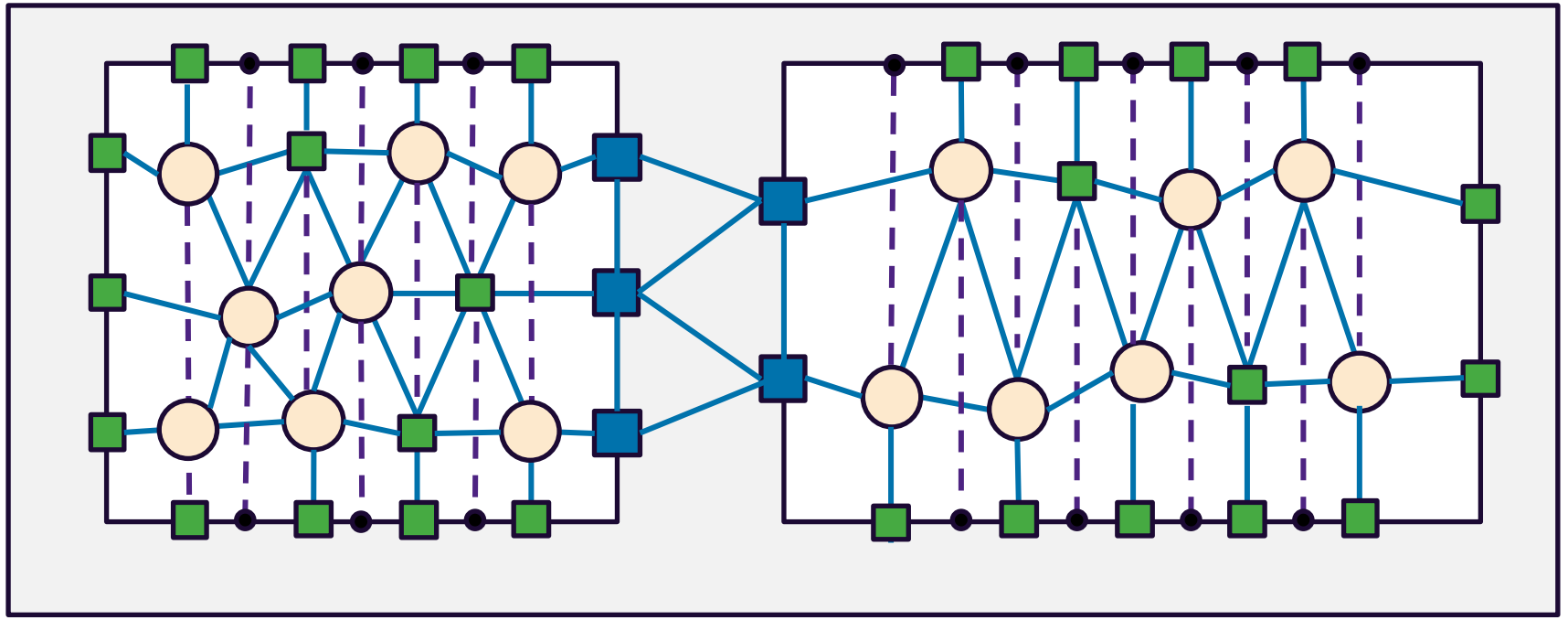

Introduction
Problem Formulation
Algorithm
Experimental Results
Conclusions

Overall Framework





Routing Graph Construction



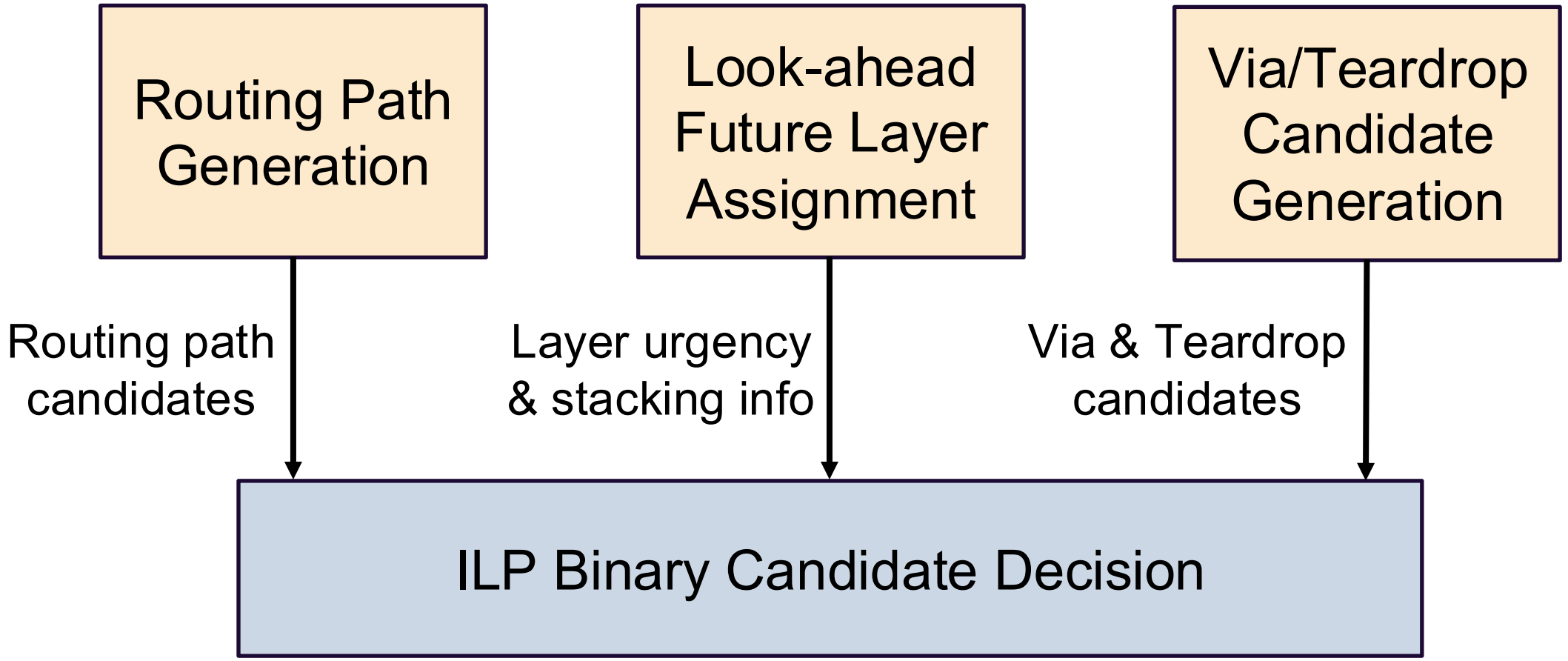
Pin/Via

Pseudo-vertex

Routing graph edge

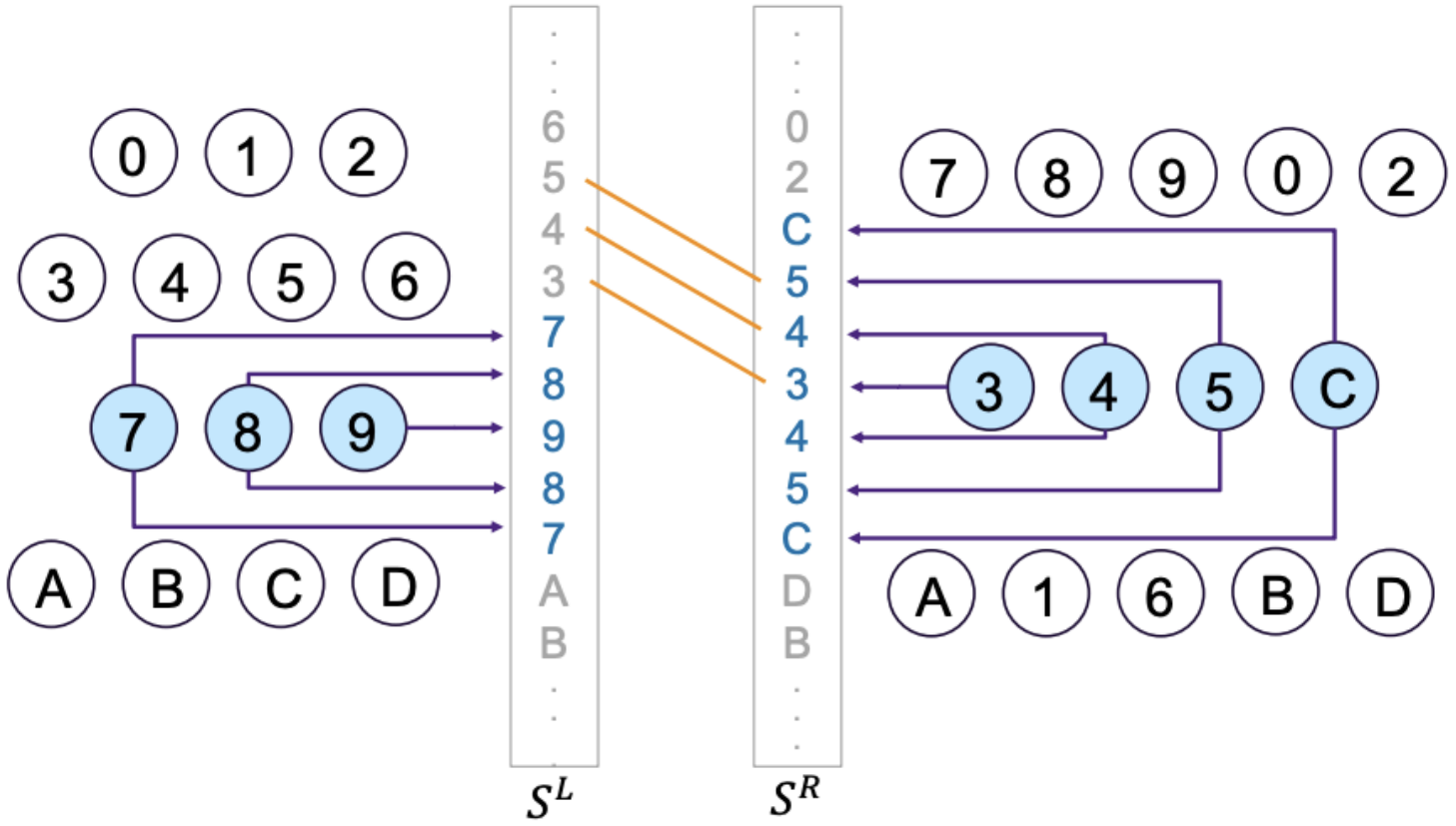
Cut-edge

Candidate Generation for Concurrent Optimization

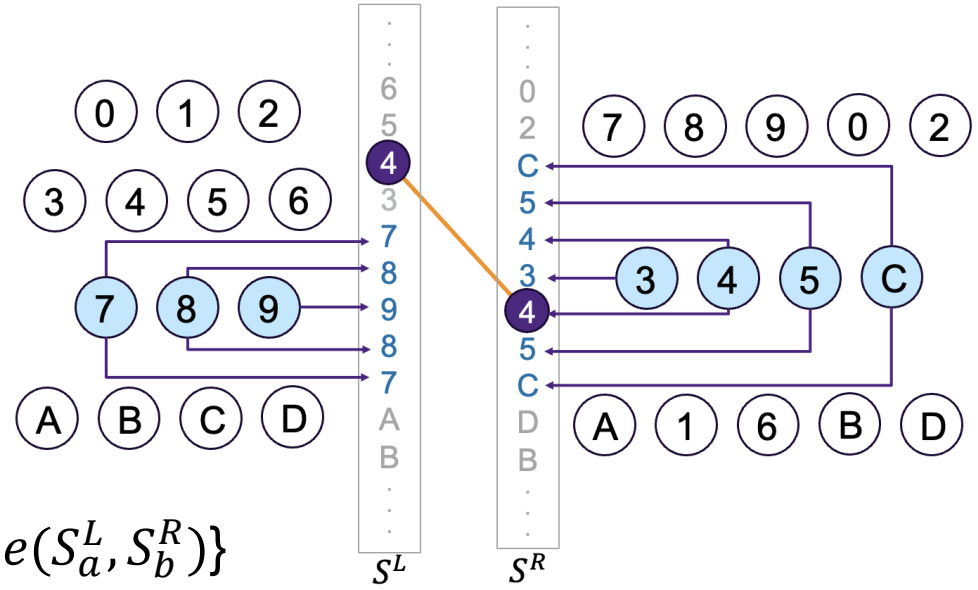


Routing Path Generation

- Generate corresponding state sequences according to pin topology



Dynamic Programming Process

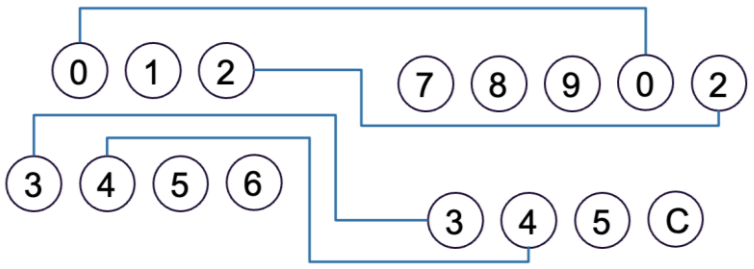


- $T(a,b)$ = state considering $S^L[0:a]$ and $S^R[0:b]$
- Transition (for matched pins):

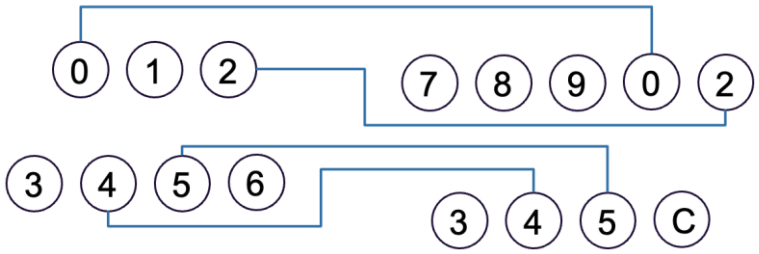
$$\mathrm{T}(a,b) = \max\{\mathrm{T}(a-1,b), \mathrm{T}(a,b-1), \mathrm{T}(a-1,b-1) + score(S_a^L, S_b^R)\}$$

- Scoring function $score(\cdot)$
  - Prefer near-center, close-row nets
  - Prioritize rejected nets

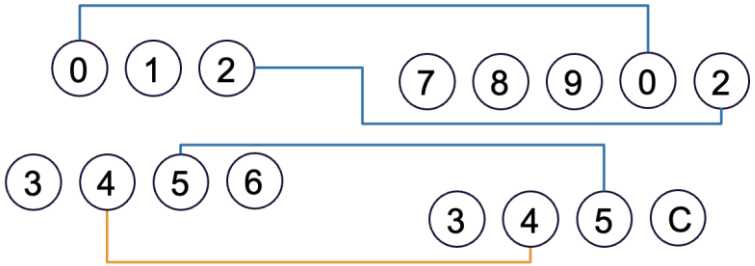
L\R	…	3	4	5	…
…					
5		C	A		
4		B			
3					
…					



Case A

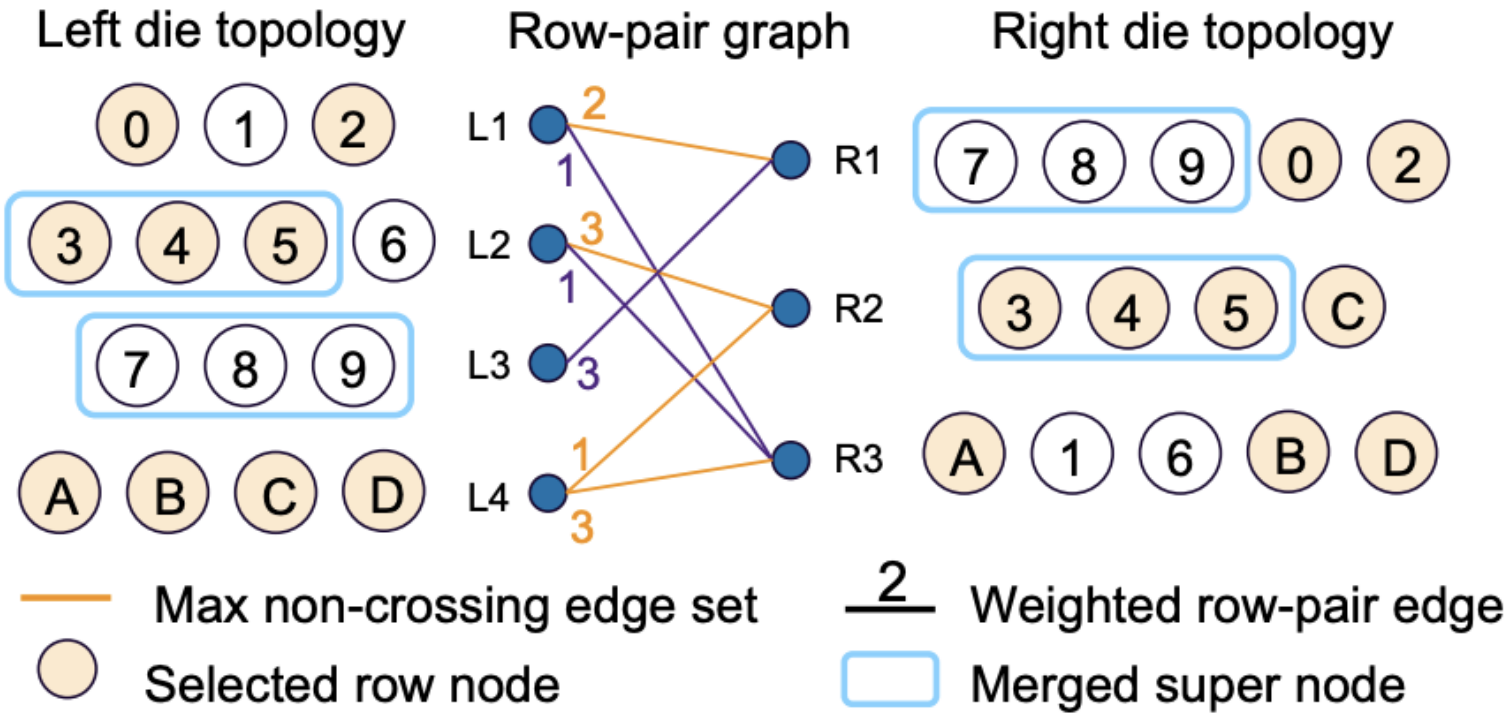


Case B



Case C

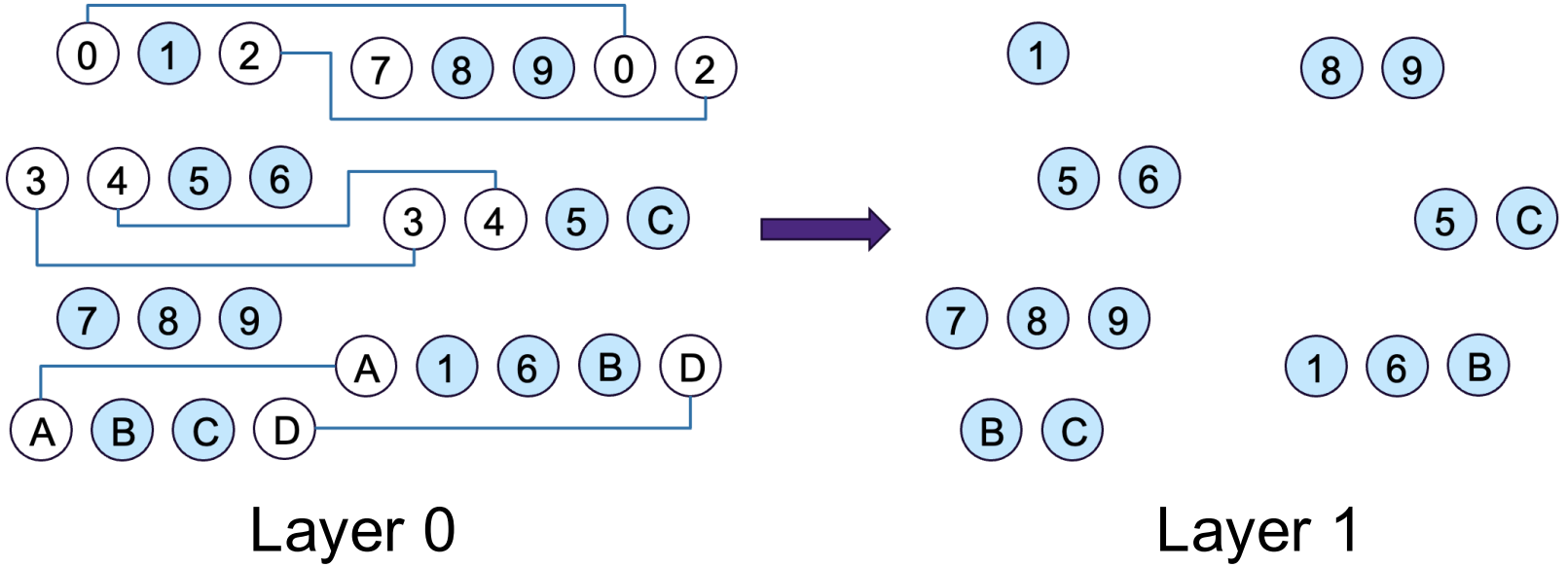
Acceleration: Crossing Pair Identification & Net Grouping



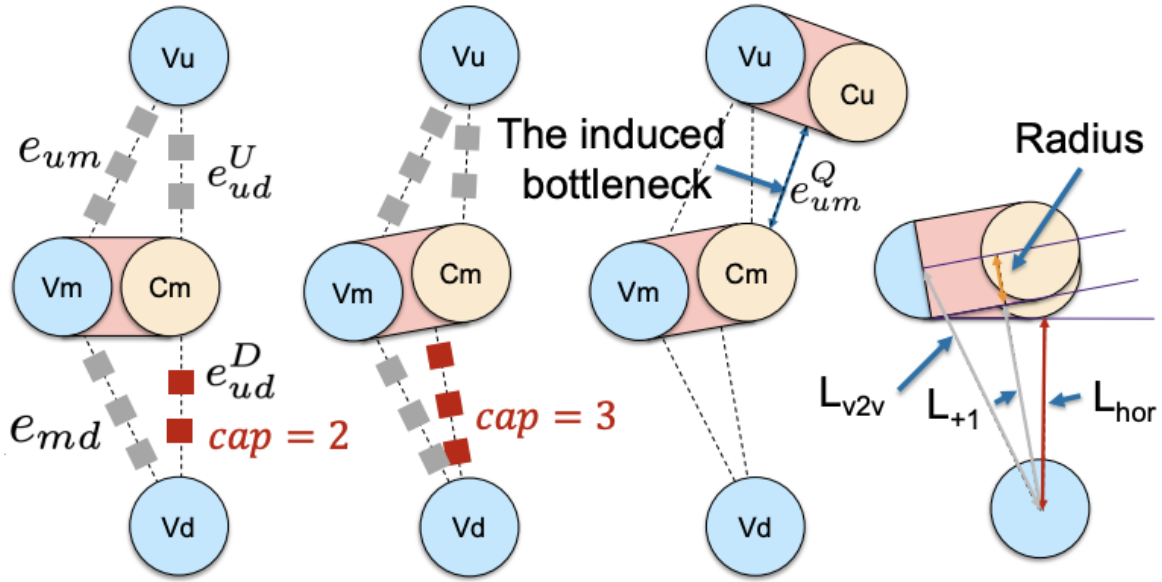
- Reduced sequence:
  - L: 0203*ABCDCBA
  - R: 202C3*CDBABD

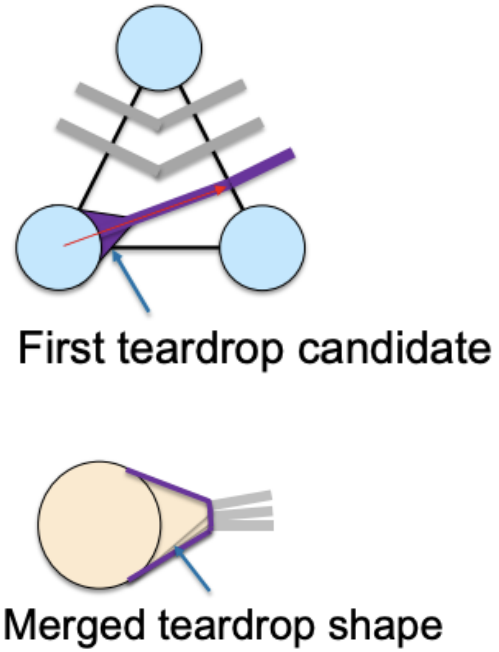
Look-Ahead Future Layer Assignment

- Help decide stacked or staggered vias
- Define $RemainStk(v)$ as number of remaining allowable stacked vias
- Apply DP for layer assignment



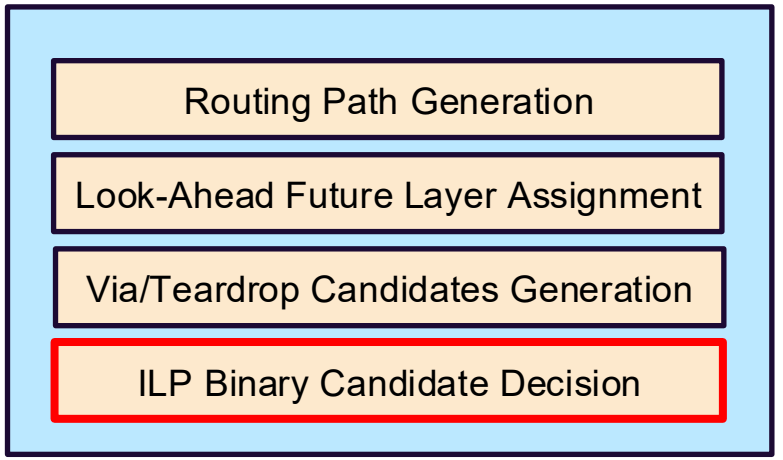
Via/Teardrop Candidate Generation





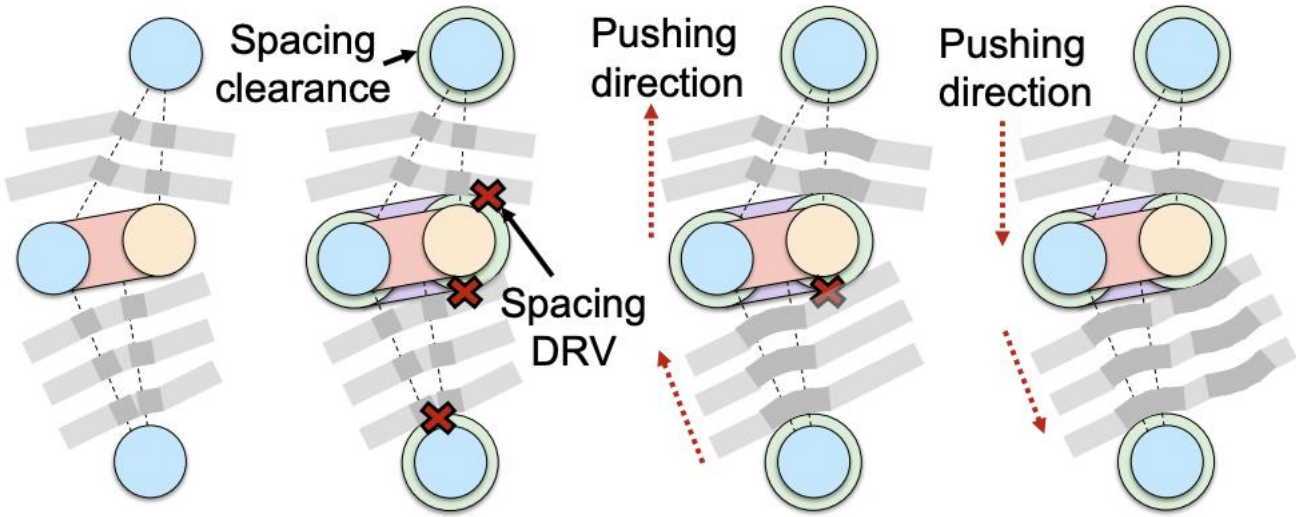
ILP Binary Decision Making

- Given the favorable routing paths and via arrangement from previous steps, apply ILP to globally optimize the overall decision
  - Each candidate would be a binary variable (Accept/Reject)
  - This formulation is very efficient for ILP solving
- Binary variables
  - Path, Stacked via, Staggered via, Teardrop candidates
- Connectivity constraints
  - Both sides should be aligned
- Capacity constraints
  - Path capacity, cut-edge capacity, induced bottleneck
- Objective:
  - Maximize weighted sum of selected candidates
    - Path > Stacked via > Staggered via

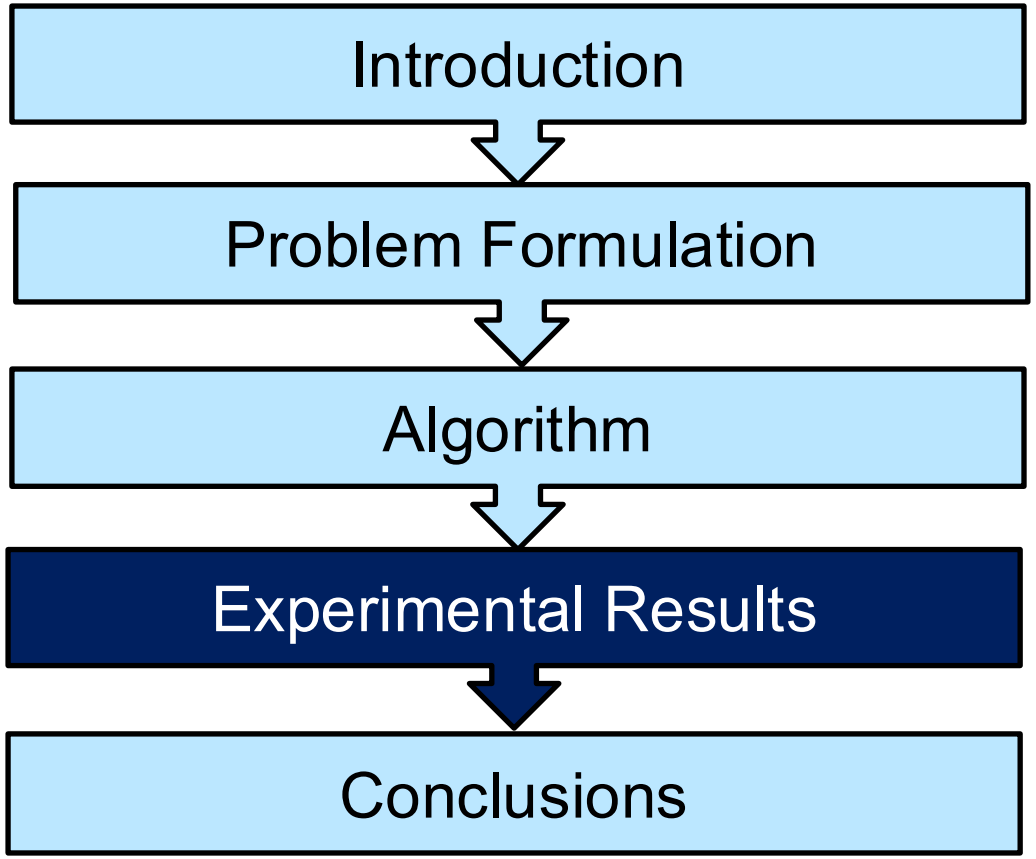


Route Realization

- Since sufficient space is reserved by previous planning step, we can always find a feasible solution
  - Place the selected routing paths according to routing graph and spacing constraints
  - Select an appropriate teardrop angle



Outline


Introduction
Problem Formulation
Algorithm
Experimental Results
Conclusions

Experimental Setup

- Language
  - C++
- Platform
  - MacBook Pro with M2 chip, 8-core CPU and 16GB RAM
- ILP solver
  - Gurobi v12.0.2
- Geometry computation
  - Boost Library
- Benchmark
  - 16 industrial-derived test cases
    - 8 asymmetric pin row cases
    - 8 aligned pin row cases
- Comparison
  - AAR [Chung+, DAC23]
    - Initial iteration, Final iteration ($max\_iter = 100$)
  - SR [Chang+, ASPDAC'25]
    - Aligned pin row cases only

	AAR [DAC'23]	Dynamic Via Insertion [DAC'24]	SR [ASPDAC'25]
Any-angle	Yes	**No**	Yes
Asymmetric pin row structure handling	Partial	Partial	**No**
Foundation	General	X-architecture	Row-based

Benchmark

- $P2P\ Dist$: total pin-to-pin Euclidean distance
  - **Very loose lower bound for wirelength**
- $\#Stack$: Maximum number of allowed consecutive stacked vias
- $\#Layer$: Maximum number of routable layers

Case	#Nets	Aligned-row (c_al)		Asymmetric-row (c_as)		#Stack	#Layer
		#Rows	P2P Dist	#Rows	P2P Dist		
c1	51	4/4	86700	4/3	91834	0	3
c2	70	4/4	147560	4/3	157658	1	4
c3	111	12/12	143412	12/9	153442	0	2
c4	162	12/12	342144	12/9	374616	1	3
c5	246	12/12	518568	12/9	576198	1	5
c6	246	12/12	518568	12/9	576198	1	5
c7	489	24/24	1028570	24/21	1075522	1	5
c8	1230	60/60	2592840	60/55	2671244	2	5

Routability and Runtime Comparison -- Asymmetric

Case	AAR				Ours	
	Initial Iteration		Final Iteration			
	Rout.(%)	RT(s)	Rout.(%)	RT(s)	Rout.(%)	RT(s)
c1_as	54.9	0.21	62.7	7.52	100	2.79
c2_as	37.1	0.32	54.3	14.04	100	3.86
c3_as	32.4	0.29	53.2	13.31	100	4.21
c4_as	54.3	0.88	59.9	73.29	100	5.86
c5_as	39.8	4.75	46.7	369.5	100	11.45
c6_as	41.5	4.26	47.2	344.3	100	13.44
c7_as	37.2	16.15	39.9	1397.8	100	27.48
c8_as	34.8	73.18	37.2	6247.1	100	68.52
Avg.	41.5	12.51	50.0	1058.4	100	17.20
Ratio	0.415	0.727	0.500	61.528	1.000	1.000

- Rout: routability, RT: runtime

- Sequential route methodology is not the best approach for dense D2D routing
  - Limited improvement after 100 iterations of rip-up and reroute

Routability and Runtime Comparison -- Symmetric

Case	AAR								SR										Ours				
	Initial Iteration				Final Iteration				Limited Layer					Unlimited Layer									
	Rout.	RT	WL	#Via	Rout.	RT	WL	#Via	Rout.	RT	WL	#Layer	#Via	Rout.	RT	WL	#Layer	#Via	Rout.	RT	WL	#Layer	#Via
c1_al	45.1	0.24	109965	102	62.7	7.77	111476	102	70.6	1.58	109732	3	112	100	3.02	87931	5	216	100	2.07	87781	3	93
c2_al	30.0	0.24	176238	144	42.9	13.90	179166	144	72.8	2.04	185125	4	266	100	3.58	149618	5	266	100	3.06	149562	4	178
c3_al	46.8	0.26	230430	114	55.0	13.02	227205	114	47.1	1.47	227813	2	96	100	3.82	146281	4	358	100	3.16	145890	2	116
c4_al	49.4	0.96	496689	324	55.6	69.40	517553	324	50.6	3.88	559675	3	232	89.5	6.91	396265	5	718	100	5.02	351578	3	245
c5_al	51.2	4.32	731054	990	52.4	359.8	772982	990	67.3	8.90	723881	5	816	97.6	15.24	540583	8	1692	100	11.35	524866	5	824
c6_al	52.0	4.44	726071	990	54.1	326.9	722096	990	66.8	8.56	729989	5	1002	94.0	17.82	589302	6	1486	100	10.71	529406	5	808
c7_al	51.5	13.52	1920057	1980	51.7	1241.8	1917628	1980	77.6	14.13	1577002	5	1604	97.5	26.39	1085412	8	3414	100	22.44	1042037	5	1583
c8_al	49.8	69.93	7069073	4680	51.3	6058.5	7126192	4680	68.9	51.75	4957213	5	4046	97.6	68.59	2811079	10	8726	100	59.66	2626981	5	4152
Avg.	46.98	11.74	1432447.13	1165.50	53.21	1011.38	1446787.25	1165.50	65.21	11.54	1133803.75	4.00	1021.75	97.03	18.17	725808.88	6.38	2109.50	100.00	14.68	682262.63	4.00	999.88
Ratio	0.470	0.380	1.590	1.127	0.532	30.669	1.611	1.127	0.652	0.719	1.475	1.000	1.086	0.970	1.318	1.048	1.623	2.248	1.000	1.000	1.000	1.000	1.000

- Rout: routability, RT: runtime, WL: total wirelength

Wirelength Comparison & Analysis -- Asymmetric

Case	AAR						Ours		
	Initial Iteration			Final Iteration					
	WL	R. avg	R. std	WL	R. avg	R. std	WL	R. avg	R. std
c1_as	114588	1.299	0.268	114754	1.257	0.267	103892	1.060	0.180
c2_as	189212	1.301	0.290	184825	1.186	0.263	168201	1.080	0.101
c3_as	259204	1.510	0.581	246460	1.316	0.381	183928	1.090	0.122
c4_as	532051	1.338	0.334	553672	1.291	0.349	402525	1.060	0.090
c5_as	820355	1.522	0.405	867669	1.638	0.522	621402	1.090	0.104
c6_as	843378	1.586	0.480	854292	1.640	0.441	630928	1.110	0.110
c7_as	2286788	1.807	0.553	2330771	1.730	0.566	1215346	1.070	0.110
c8_as	8675541	1.710	0.660	8570552	1.719	0.567	3366012	1.090	0.090
Avg.	1715139.75	1.509	0.446	1715374.38	1.472	0.420	836529.38	1.081	0.113
Ratio*	1.509	1.395	3.937	1.517	1.360	3.700	1.000	1.000	1.000

- Total WL: Unrouted nets' wirelength set to HPWL of D2D corridor
- R. avg & R. std only consider routed nets

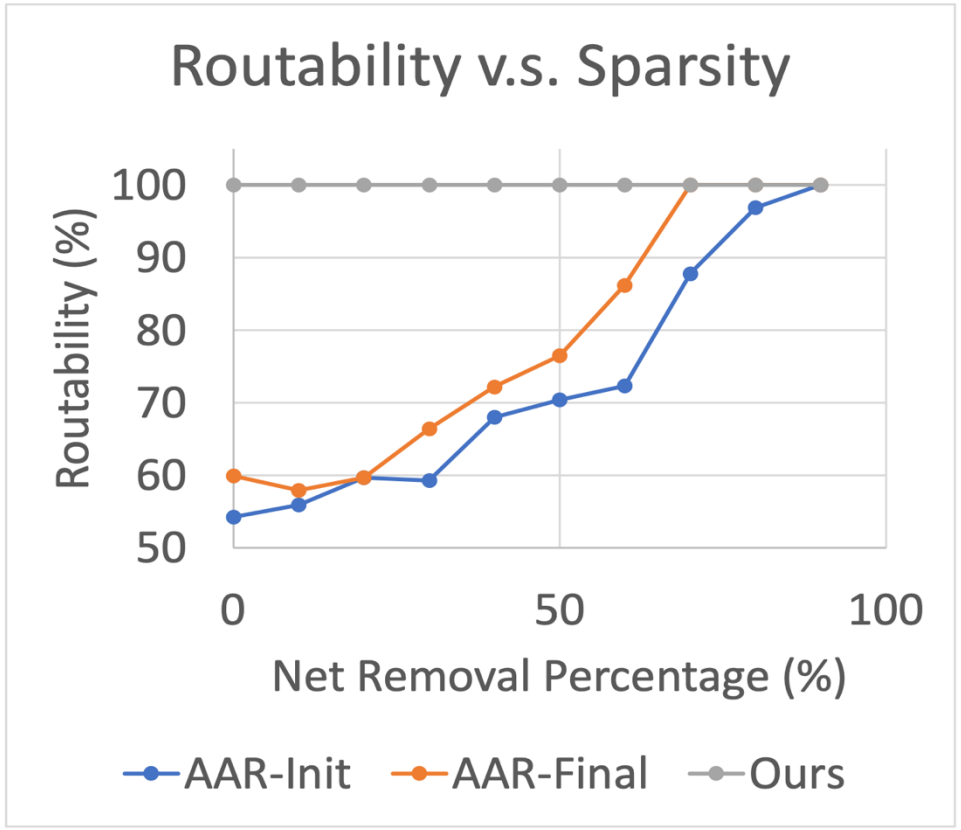
Wirelength Comparison & Analysis -- Symmetric

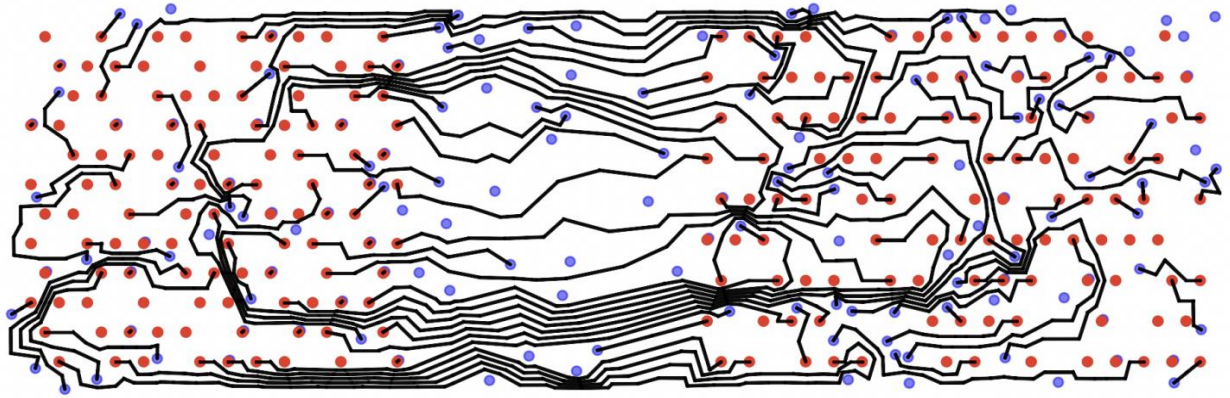
Case	AAR						SR						Ours		
	Initial Iteration			Final Iteration			Limited Layer			Unlimited Layer					
	WL	R. avg	R. std	WL	R. avg	R. std	WL	R. avg	R. std	WL	R. avg	R. std	WL	R. avg	R. std
c1_al	109965	1.406	0.439	111476	1.363	0.407	109732	1.014	0.000	87931	1.015	0.000	87781	1.013	0.000
c2_al	176238	1.356	0.264	179166	1.333	0.373	185125	1.015	0.000	149618	1.015	0.000	149562	1.014	0.000
c3_al	230430	1.291	0.268	227205	1.338	0.420	227813	1.020	0.010	146281	1.020	0.010	145890	1.017	0.000
c4_al	496689	1.407	0.434	517553	1.308	0.402	559675	1.030	0.130	396265	1.030	0.000	351578	1.028	0.120
c5_al	731054	1.457	0.446	772982	1.453	0.474	723881	1.020	0.060	540583	1.018	0.060	524866	1.012	0.010
c6_al	726071	1.450	0.427	722096	1.401	0.372	729989	1.043	0.150	589302	1.042	0.130	529406	1.021	0.060
c7_al	192057	1.525	0.483	1917628	1.507	0.481	1577002	1.028	0.040	1085412	1.030	0.050	1042037	1.013	0.010
c8_al	7069073	1.516	0.523	7126192	1.366	0.467	4957213	1.061	0.050	2811079	1.059	0.040	2626981	1.013	0.010
Avg.	1432447.13	1.430	0.410	1446787.25	1.380	0.420	1133803.75	1.013	0.060	725808.88	1.030	0.060	682262.63	1.020	0.030
Ratio*	1.590	1.403	15.638	1.611	1.362	16.171	1.475	1.012	2.095	1.048	1.012	1.905	1.000	1.000	1.000

Effectiveness of Acceleration Techniques

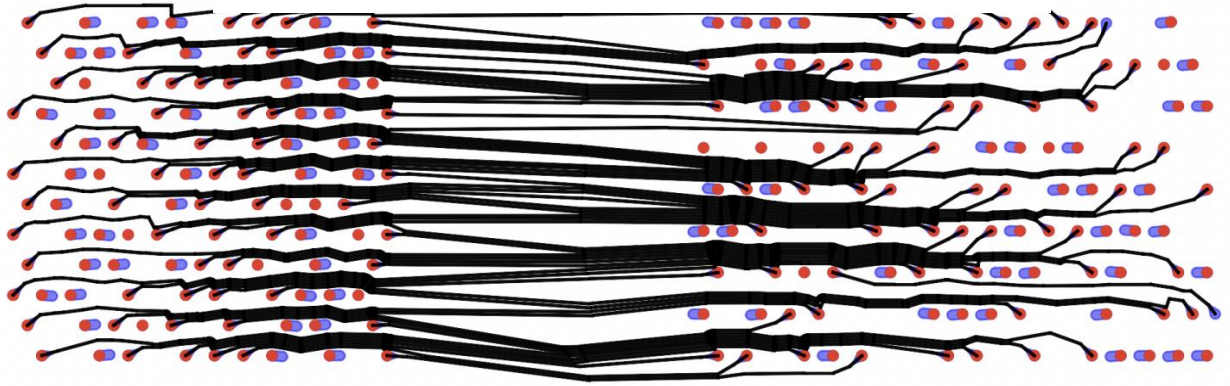
Case stat.	Runtime (s)		
#Net, #Row(L,R)	Original DP	+ Crossing Pair	+ Net Grouping
8, (3,2)	17.2	9.7	9.7
12, (3,2)	29.1	21.8	10.2
24, (5,3)	56.2	24.1	18.6
36, (6,4)	108.5	37.0	26.7
50, (7,5)	207.8	73.9	38.2
Total	418.8	166.5	103.4
Ratio	4.05	1.61	1.00

Routability with Different Sparsity





AAR (30% removal)

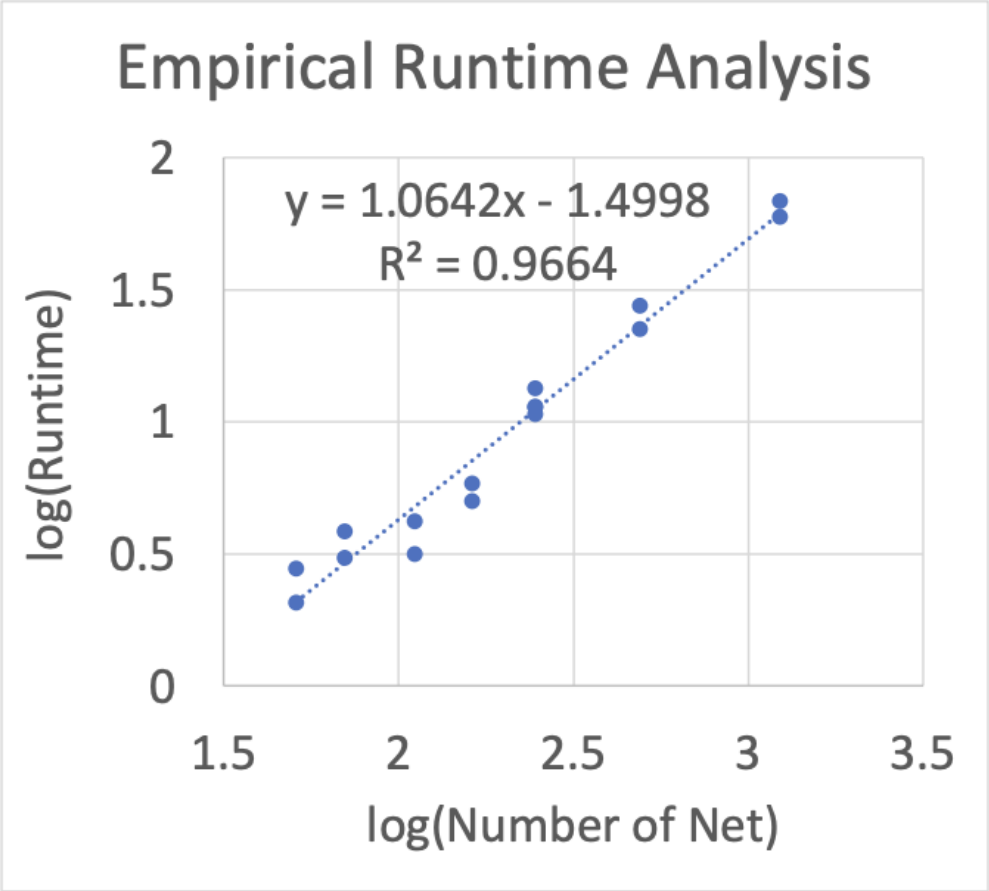


Ours (30% removal)

- AAR couldn't achieve 90% routability even after removing 60% of nets

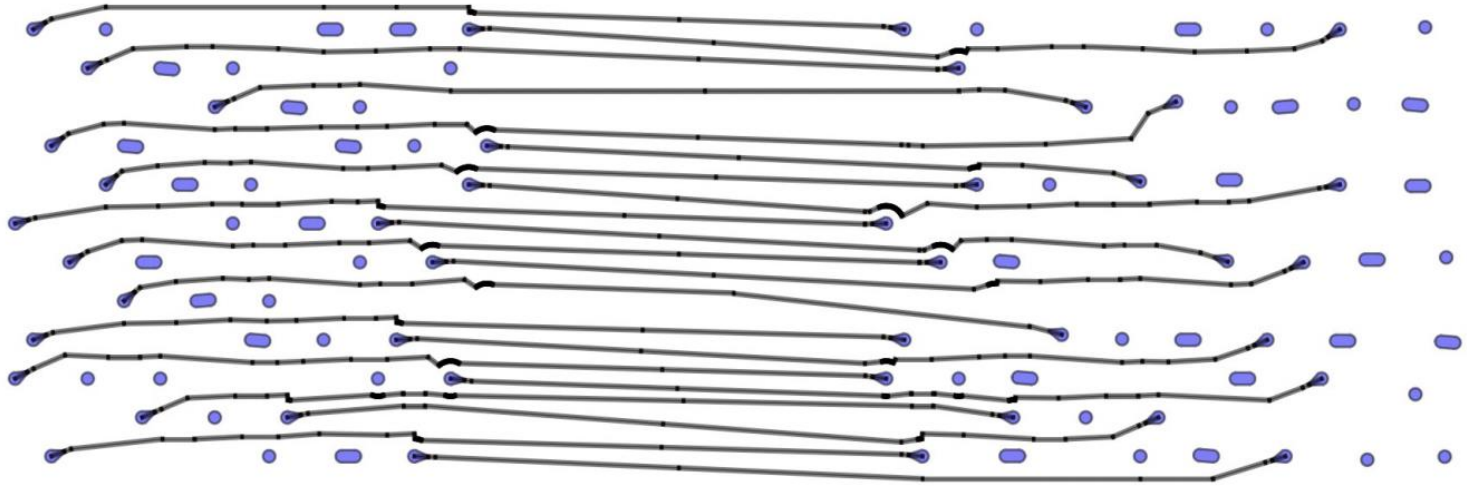
Runtime Analysis

- Empirically, our algorithm can generate the routing result in $O(n^{1.06})$ time, where n represents the number of nets in an instance

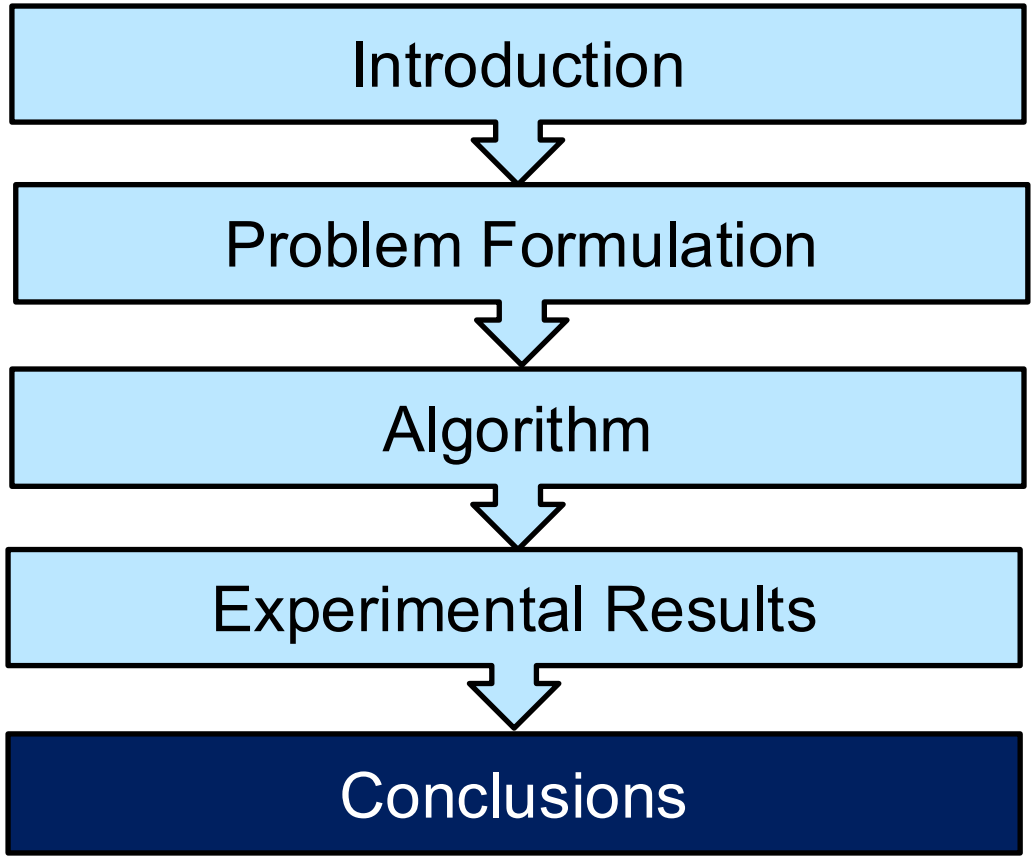


Shielding-aware Routing Results

- Routing resource reserved through design rule remapping



Outline


Introduction
Problem Formulation
Algorithm
Experimental Results
Conclusions

Conclusions

- Propose a general D2D routing framework with concurrent route and via-routing co-optimization
  - Generalizes to asymmetric pin row structure
  - Considers shielding space reservation
- Propose efficient acceleration techniques to enhance scalability
  - Crossing pair identification for early conflict pruning
  - Net grouping to further reduce problem size in path candidate generation
- Experimental results show that our algorithm can produce high-quality routing results efficiently
  - 100% routability across all test cases, near-optimal wirelength
  - $O(n^{1.06})$ empirical time complexity

Thank you