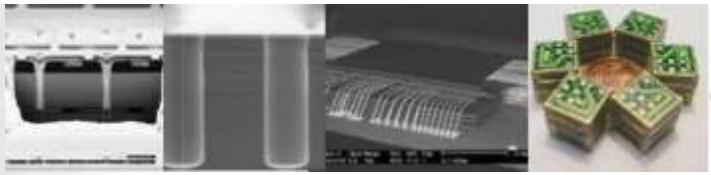
Session 4C

Escape Routing of Mixed-Pattern Signals Based on Staggered-Pin-Array PCBs

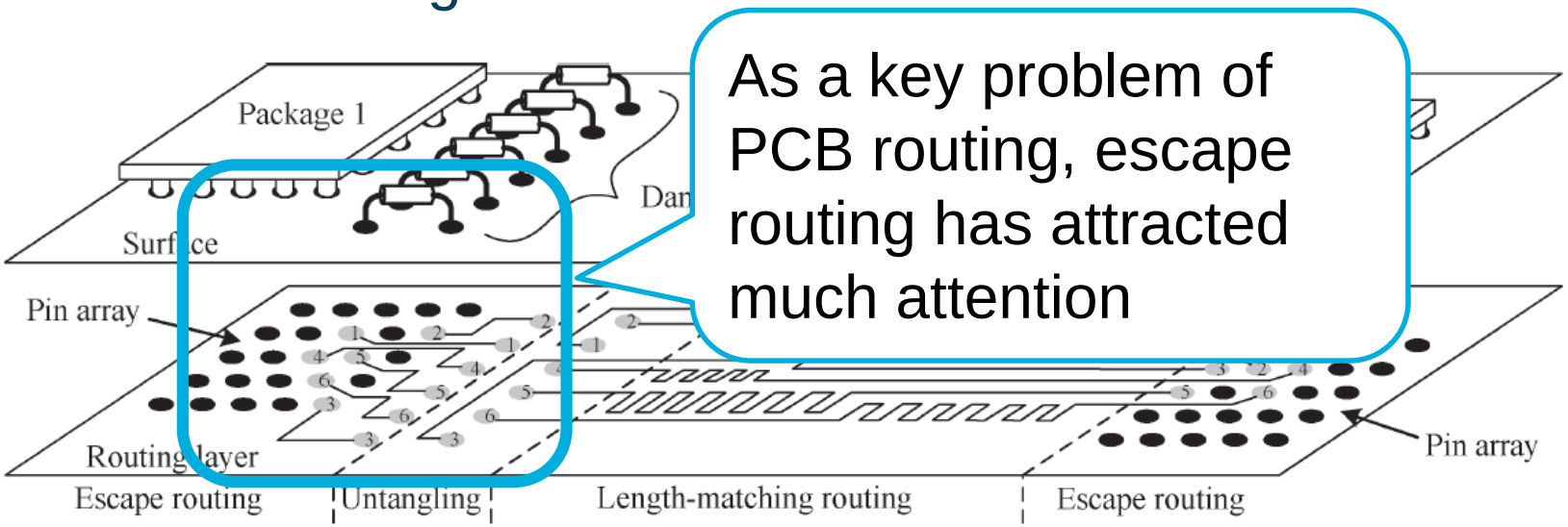


Kan Wang, Huaxi Wang, Sheqin Dong
Tsinghua University, Beijing, P.R.China
E-mail: wangkan09@mails.thu.edu.cn

Outline

Issues of PCB Routing

- Printed circuit board (PCB) routing has become more and more difficult for manual design
  - Due to increasing pin count and decreasing routing resource

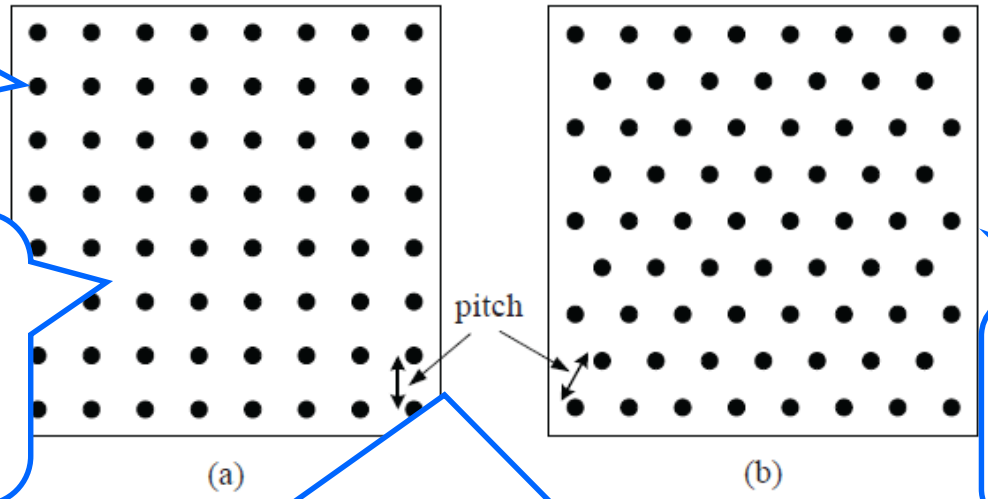


Recent Researches on PCB Routing

- For pin array structure

grid pin array (GPA)

cannot satisfy the demands of the ever-increasing pin number



staggered pin array (SPA) [DAC'06]

- For escap

Compared to GPA, SPA can increase pin density greatly under the similar number of pins and same area [ICCAD'11]

There are still some disadvantages

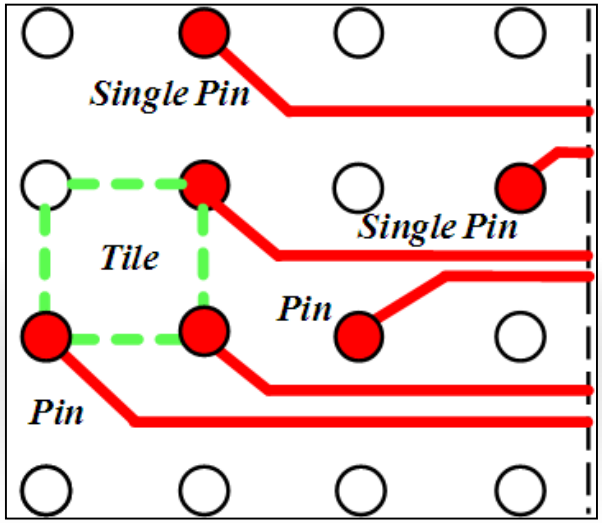
ICCAD'10

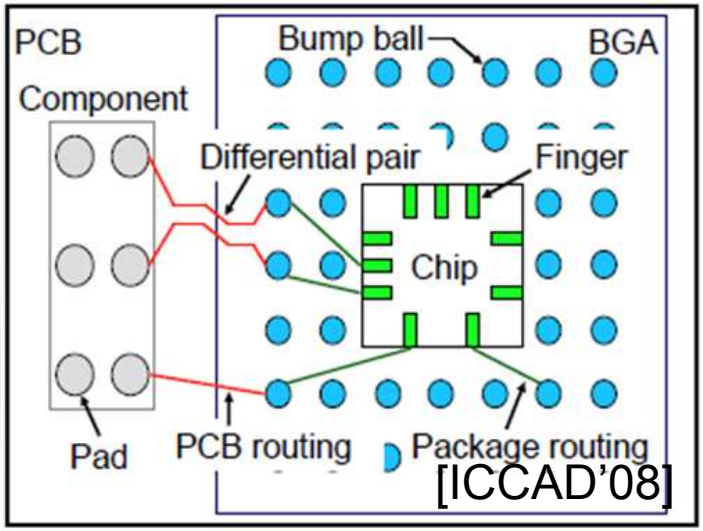
ASPDAC'12

ICCAD'11

Related Work for escape routing

- For GPA
  - Network flow based escape routing algorithms on GPA [ICCAD'96' 06'08]
  - However, they only focused on single-signal
  - A chip-package-board co-design considered escape routing of differential pairs [ICCAD'08]
  - But it paid more attention to co-design



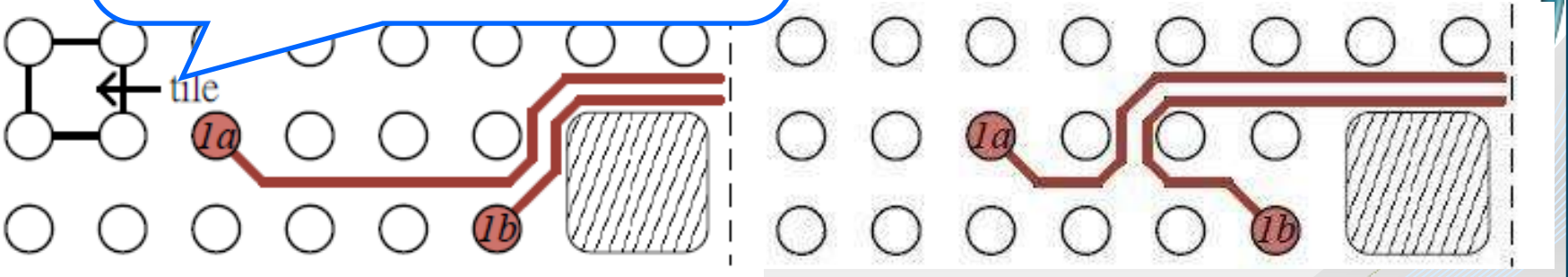


[ICCAD'08]

Related Work for escape routing II

- [ICCAD'10] proposed a negotiated congestion-based differential-pair routing
- But it did not take length-matching rule into account.
- d a five-stage
  length matching

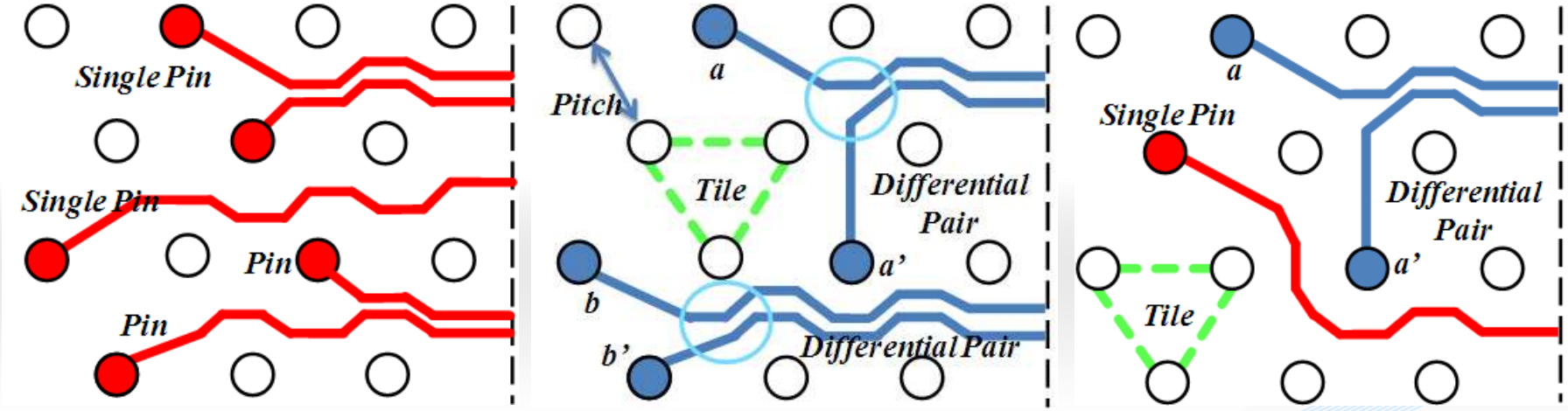
Without considering this, the signal skews will be enlarged, which can lead to degradation of performance



routing of both signals on GPA

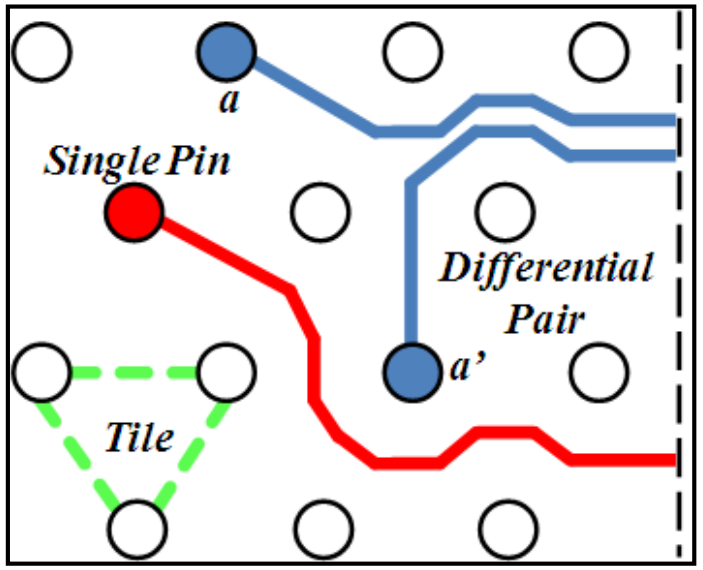
Related Work for escape routing III

- For SPA
  - Only single-signal escape routing was developed [ICCAD'11]
  - There is no work on differential-pair escape routing
  - No work for escape routing of both signals



Motivation of this work

- Because of the high noise immunity and low electromagnetic interference
  - Differential pairs are always used for the high-speed signal transmission on PCB
- Limitation of resources
  - Not all signals are trans
  - The signals of differenti will coexist on board
  - The research on escape will be quite valuable



- The problem of escape routing for mixed-pattern signals

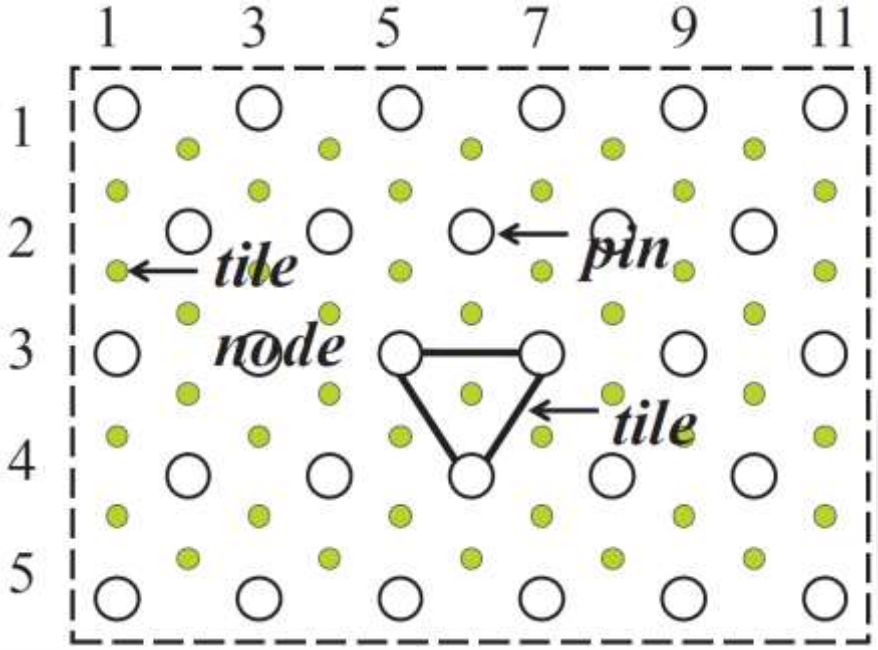
Contributions

- In this paper, a mixed-pattern escape routing algorithm is proposed on staggered pin array
  - 1. The problem of escape routing of mixed-pattern signals is presented for the first time
  - 2. A unified ILP model is formulated for mixed-pattern escape routing problem
  - 3. A slice-based heuristic method is proposed to prune the variables of ILP and speed up the solving

Outline

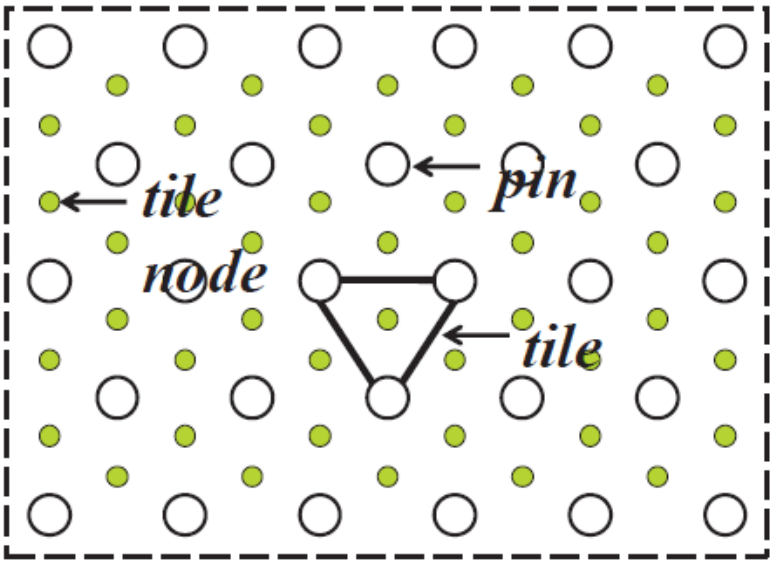
Staggered Pin Array

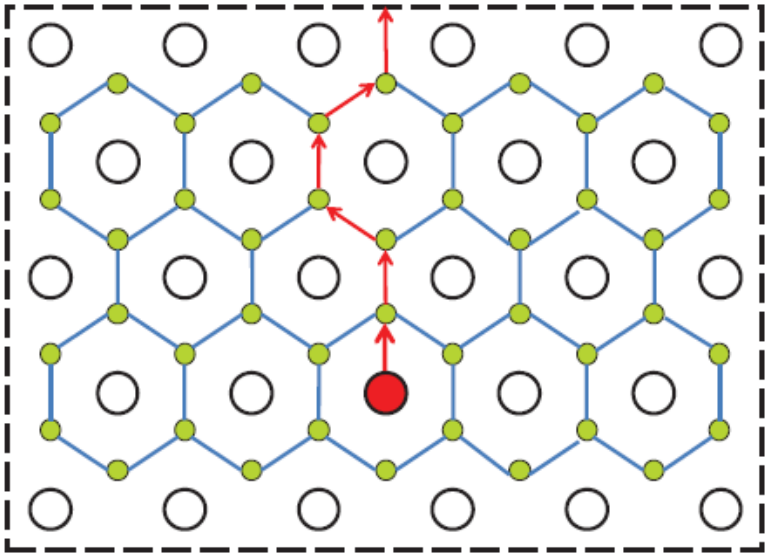
- A *m*×*n* staggered pin array (SPA)
  - composed of *n* rows, and in each row there are *m* (in odd rows) or *m*−*1* pins (in even rows).
  - A triangular tile is composed of three adjacent pins and there is a tile node in each tile



Tile Network

- A tile network is generated by connecting triangular tile nodes with each other in the form of hexagons.
- The edges of tile hexagons will be channels for escape routing and the angle between the routing channels is 120-degree





Problem Definition

The problem of mixed-pattern escape routing (MPER)

- Given:
  - (1) a $m \times n$ staggered pin array
  - (2) a differential pairs and b single signals to be routed to the boundary
  - (3) design rules such as non-crossing rule and wire length matching of differential pairs
  - (4) the constraints such as the limitation of routing resource
- The objective is:
  - Escape all marked pins to the array boundary with minimized total wire length via the tile network and meanwhile no design rule is violated and 100% routability is guaranteed.

Outline

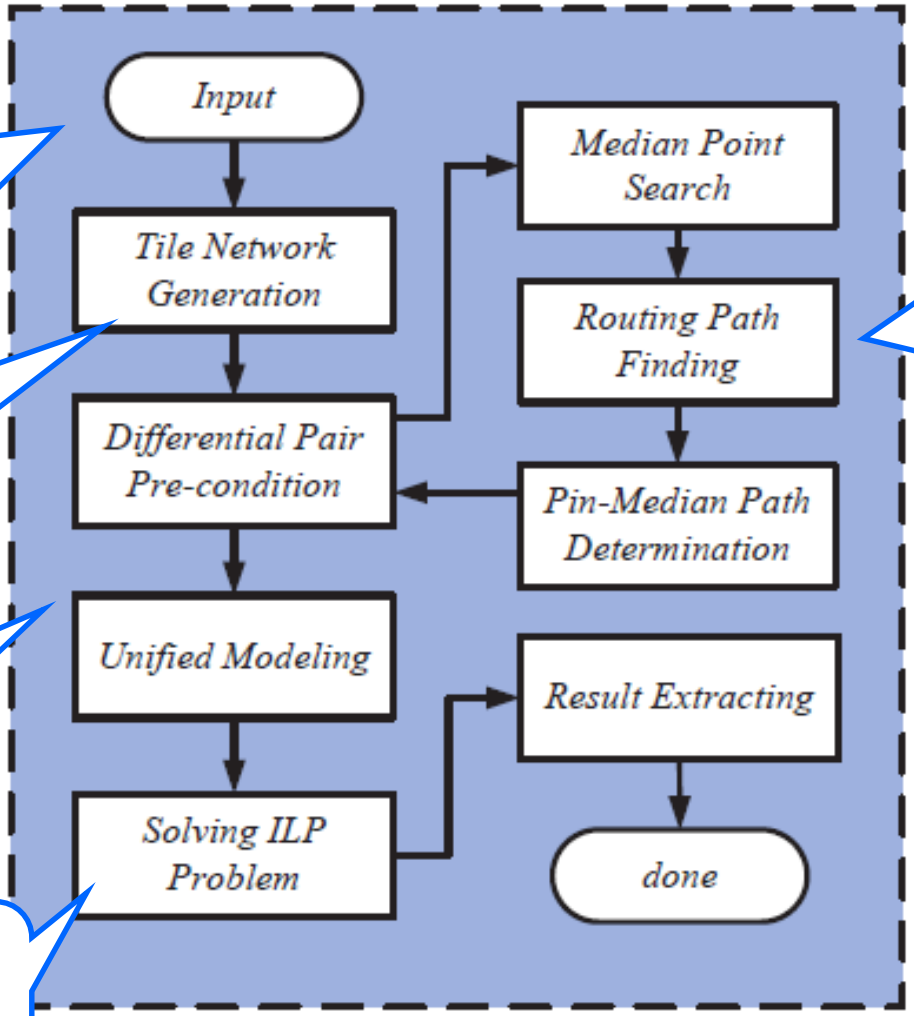
Overview flow of MPERA

Inputs of pin array, signals and constraints

Tile network generation

ILP based Unified Modeling

a slice-based heuristic algorithm to solve the ILP



Differential Pair Pre-condition

Constraints in *MEPR*

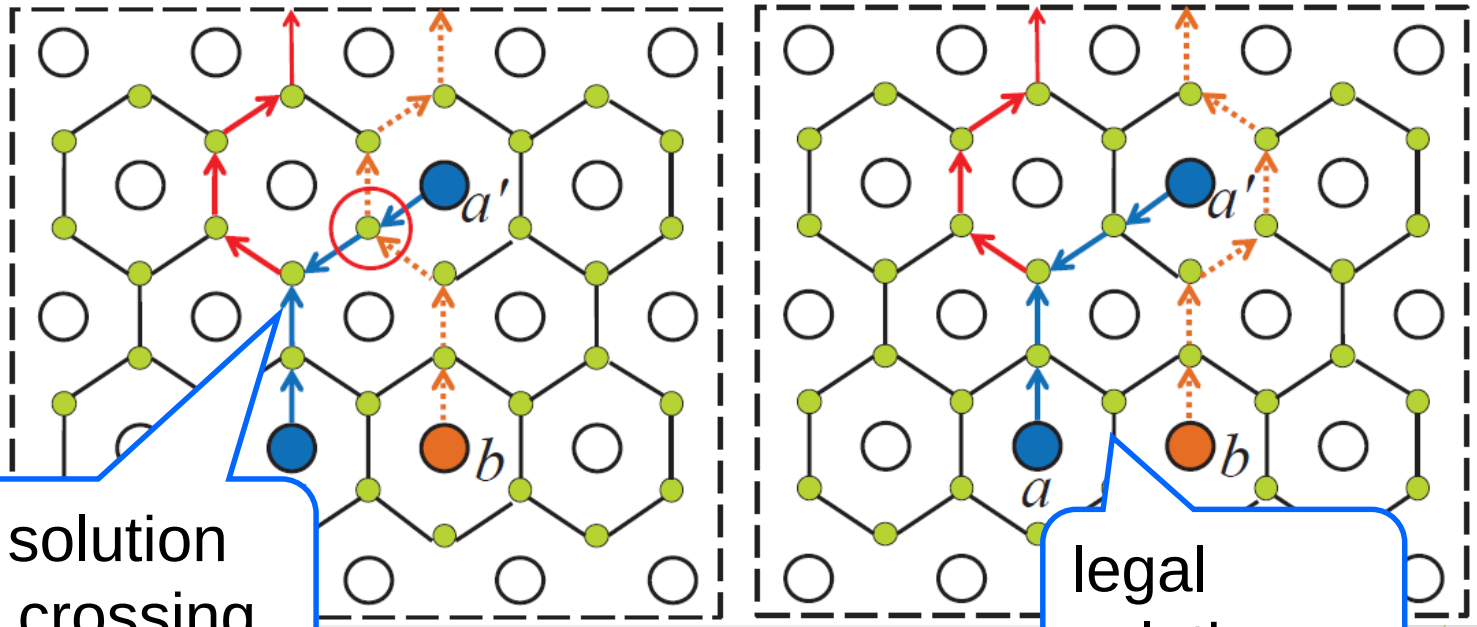
- Constraints:
  - Differential-pair protection constraint
  - Differential-pair length matching rule
  - Non-crossing rule
  - Routing resource
  - Wire width constraint
  - Acute-angle constraint

Used for differential pairs

Used for both signals

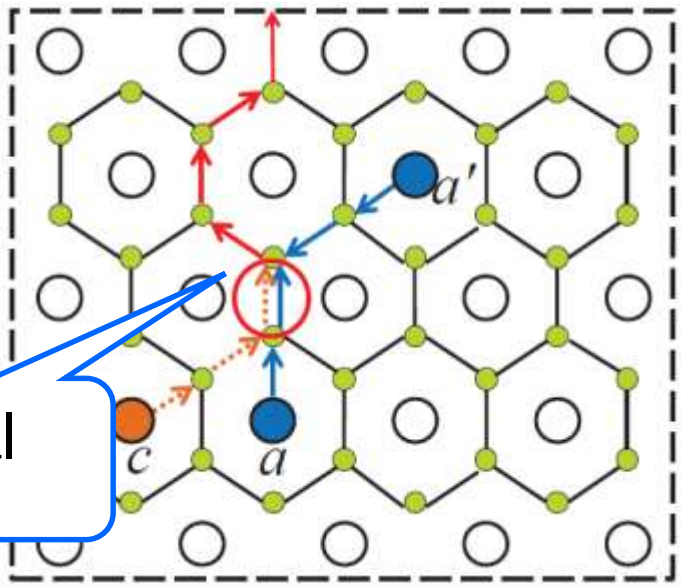
Constraints in *MEPR*

- Non-crossing rule
  - The routing paths between two signals are not allowed to be crossed

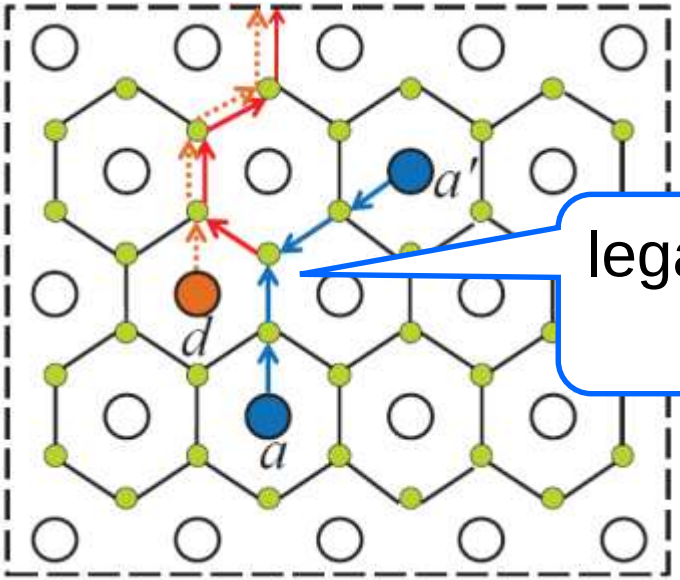


Constraints in *MEPR*

- Differential-pair protection constraint
  - In order to avoid signal crosstalk, before the two signals of differential pair meet with each other, no other signal is allowed to be close to



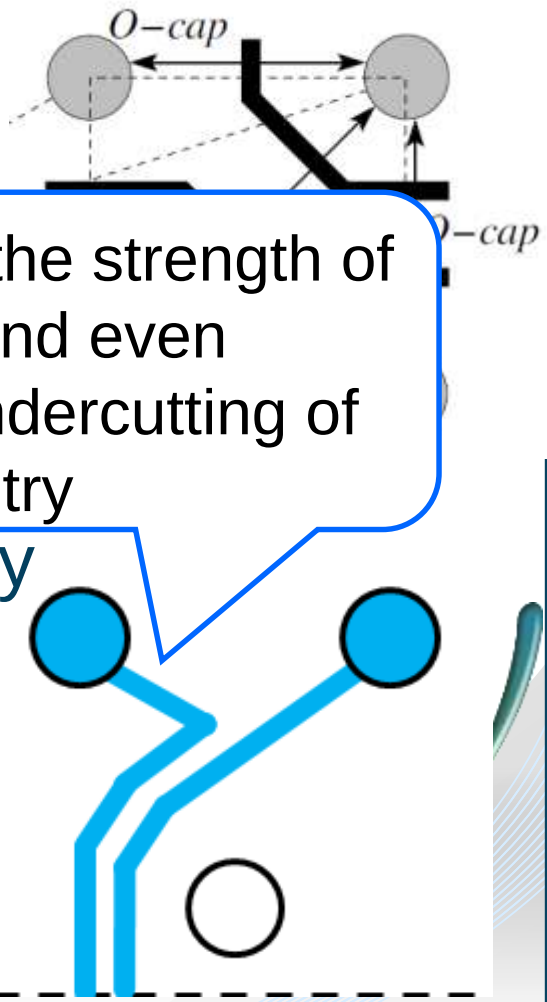
Illegal



legal

Constraints in *MEPR*

- Routing resource constraint
- Wire width constraint
- Acute-angle avoidance constraint
  - For SPA, acute-angle may possibly generated, especially for differential pairs
  - It is necessary to avoid the acute-angle routing



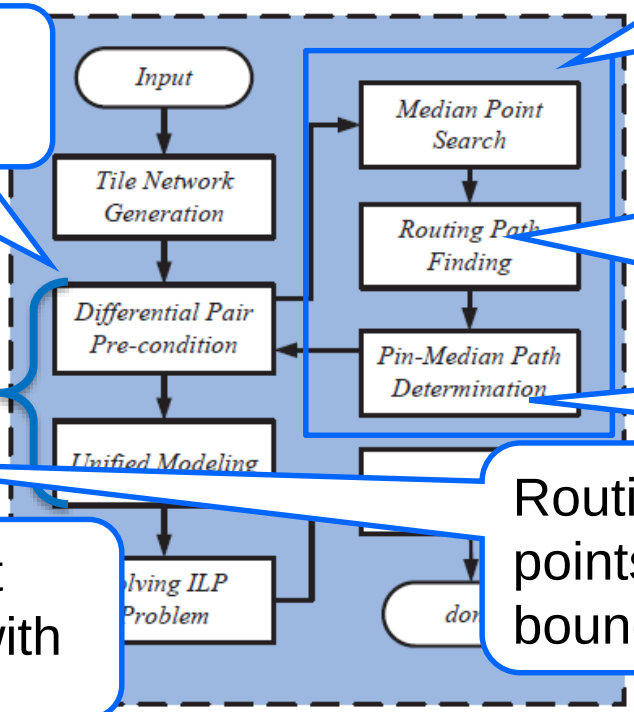
Constraints in *MEPR*

- Difficult to take the wire length-matching rule together with others as it is only for differential pairs
  - The constraint will be solved sep

Median points searching

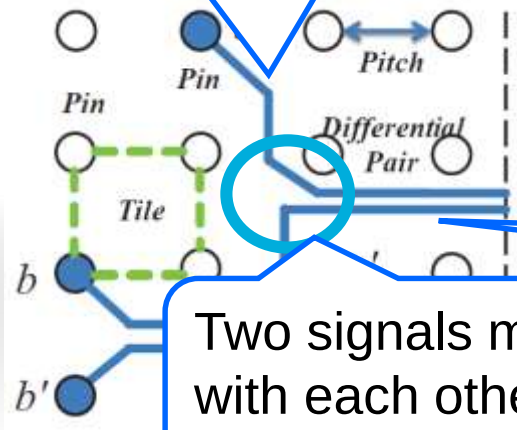
Routing f
pins to m
points

Differential Pair Pre-Condition



Routing from pins to median points with length-matching

Path determination

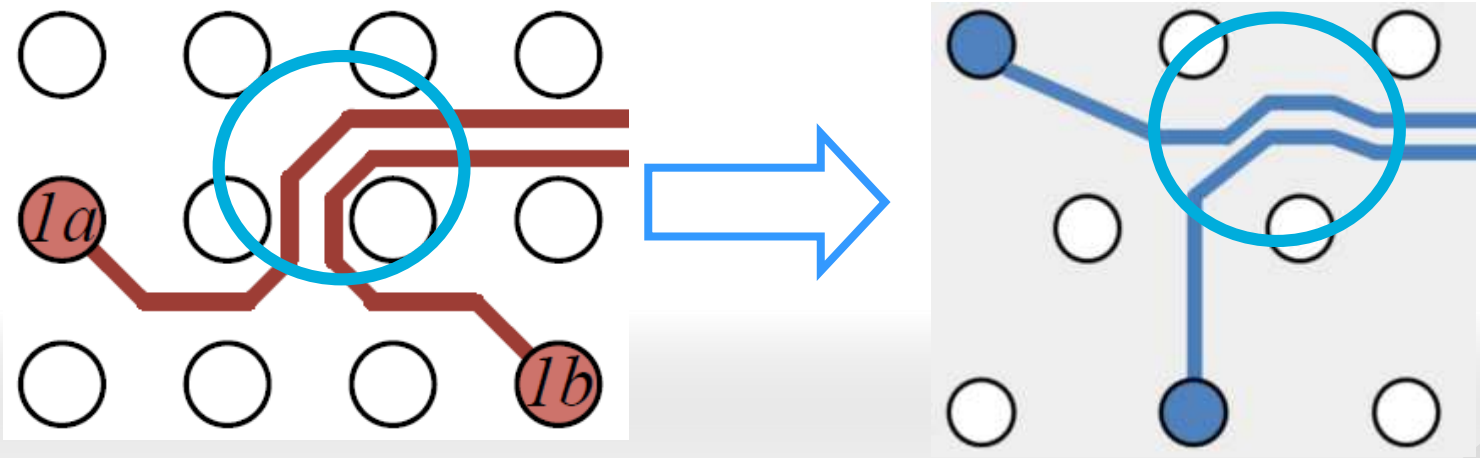


Routing for both median points and single signals to boundary

Two signals meet with each other with same length

Median Points Searching

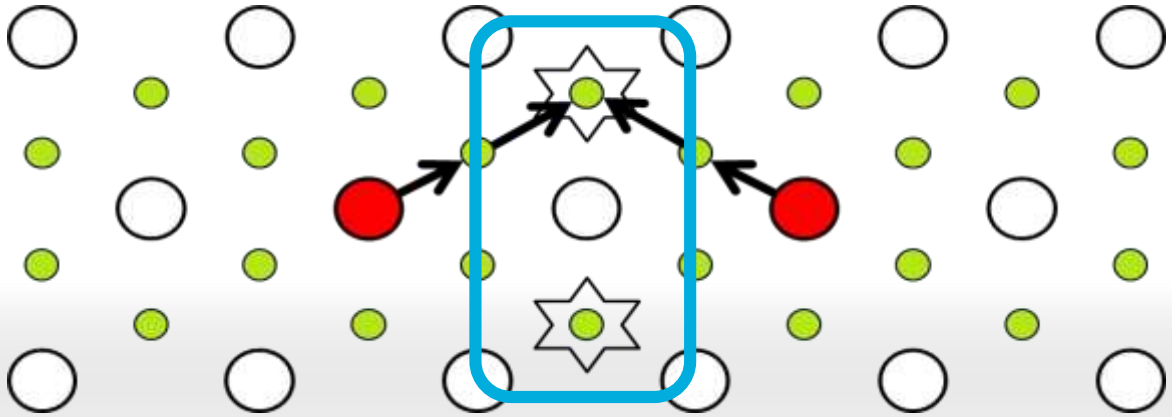
- Min-cost Median Points
  - An effective method was proposed to find median point candidates for each pair [ASPDAC'12]
    - However, it is based on GPA



median point searching algorithm for SPA

Median Points Searching- simple cases

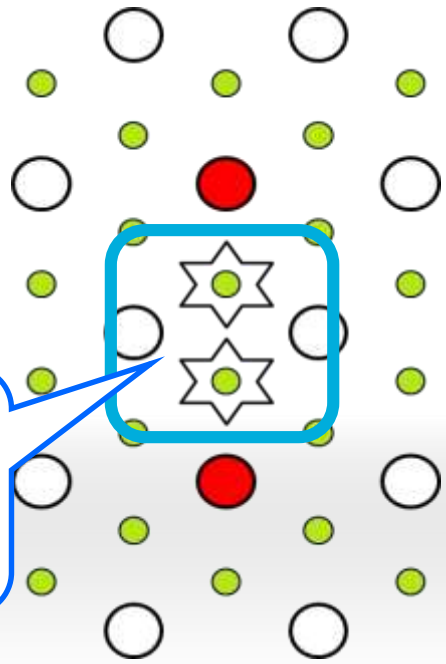
- Let $(x_a{}^p, y_a{}^p)$ and $(x_b{}^p, y_b{}^p)$ be the coordinates of two pins of differential pair
- Case 1. $y_a{}^p = y_b{}^p$:there are two min-cost median point candidates, which lie on the **mid-perpendicular** between pins



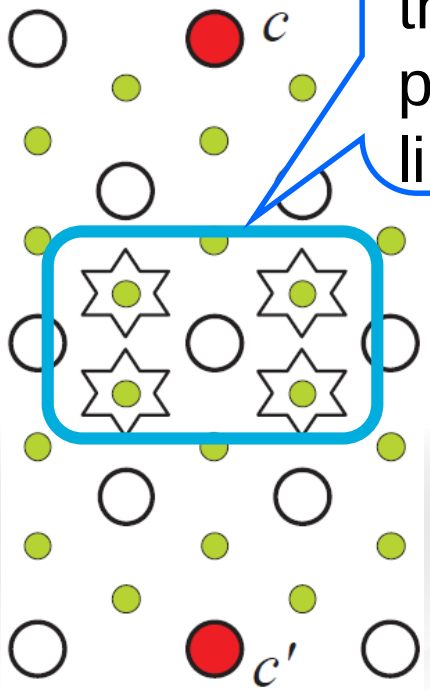
Median Points Searching- simple cases

Case 2: ${x_a}^p = {x_b}^p$ and $|{y_a}^p - {y_b}^p|$ is not multiple of 4: There are two min-cost median point candidates

Case 3: ${x_a}^p = {x_b}^p$ and $|{y_a}^p - {y_b}^p|$ is multiple of 4: There are four min-cost median point candidates



Lie on the line of two pins



Lie around the middle pin of the line

Median Points - complicated cases

- For complicated cases:

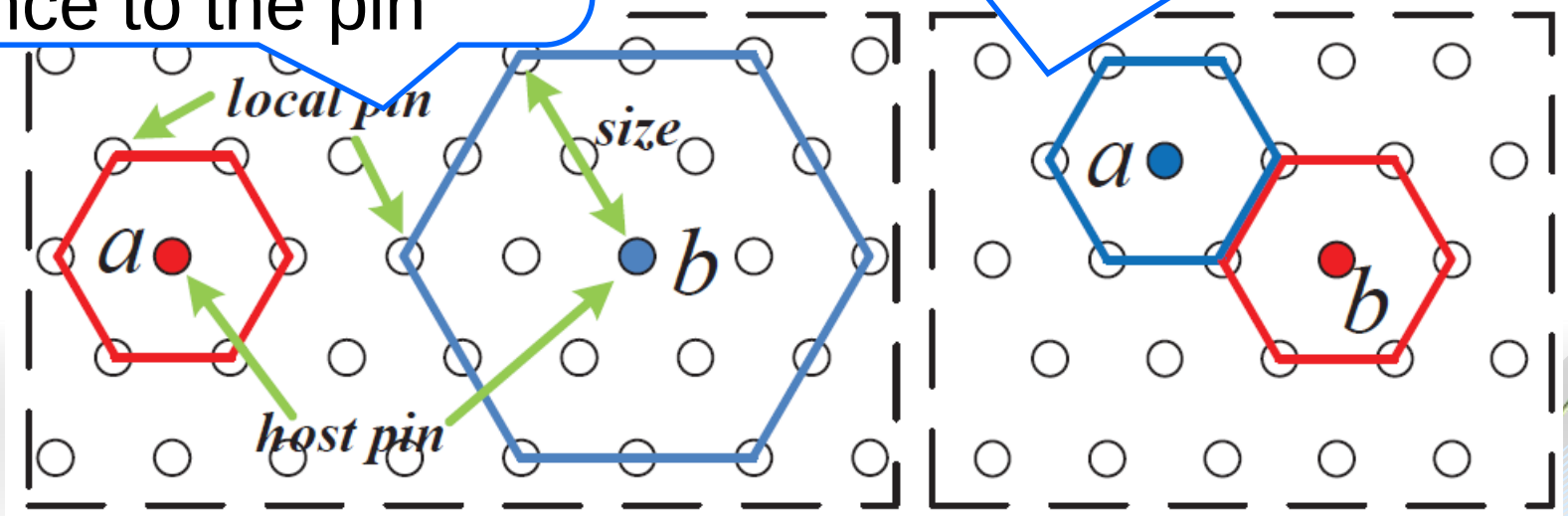
$$x_a{}^p \neq x_b{}^p,\ y_a$$

gon ar

hexago

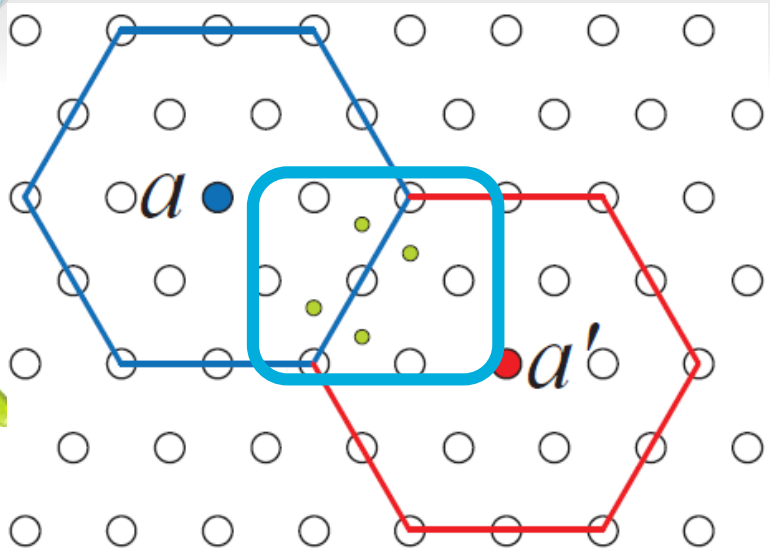
Composed of certain pins with the same distance to the pin

are two adjacent hexagons with the same minimum size



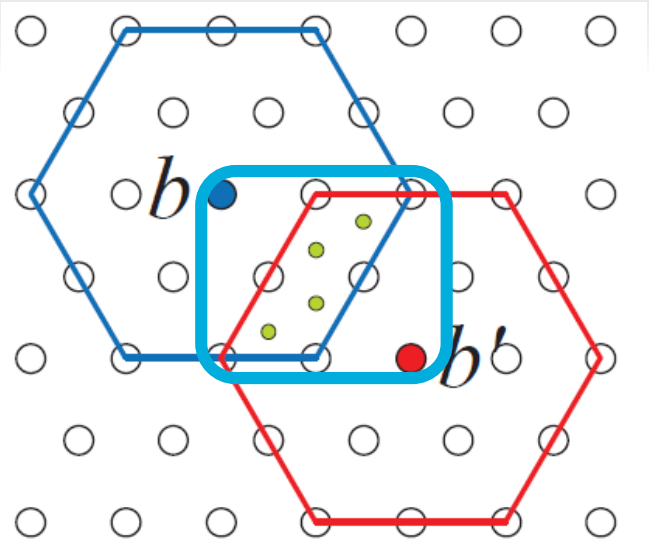
The pin hexagon Minimum intersection hexagons

Median Points - complicated cases



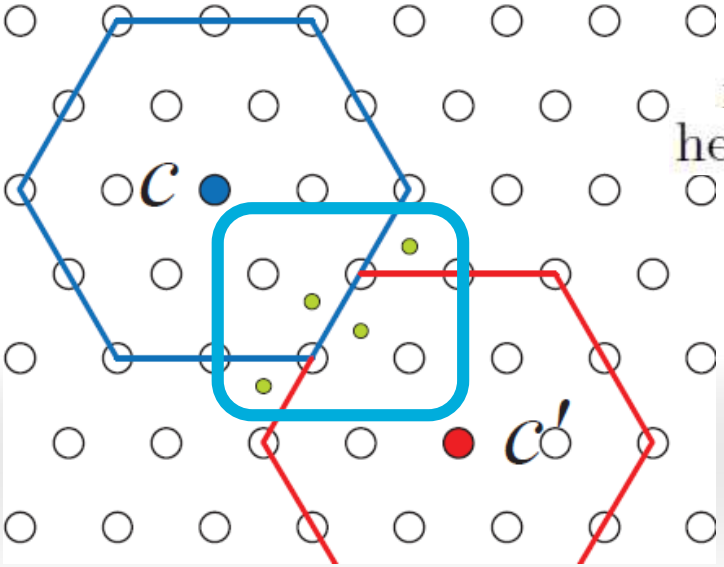
Case 4

minimum intersection hexagons are adjacent



Case 5

minimum intersection hexagons are intersectional



Case 6

minimum intersection hexagons are partly adjacent

Path Candidates Generation

清華大學 Tsinghua University

Algorithm 1 Dynamic Programming b

Require: $pin1$, $pin2$, $mpoint$, $maxLe$

```
1: calMaxLength(maxLength);
2: adjTiles1 = calPinAdjTiles(pin1);
3: adjTiles2 = calPinAdjTiles(pin2);
4: removeReTiles(adjTiles1, adjTile
5: region = calRoutingRegion(pin1, p
6: isdone = false;
7: while isdone do
8:    for each t1 ∈ adjTiles1 do
9:       path1.push_back(t1);
10:      findPath(pathSet1, path1, t1, r
         region, maxLength);
11:   end for
12:   for each t2_i ∈ adjTiles2 do
13:      path2.push_back(t2);
14:      findPath(pathSet2, path2, t2, mpoint,
         region, maxLength);
15:   end for
16:   if pathSet1 is not null and pathSet
17:      mergePaths(pathSet1, pathSet2);
18:      isdone = true;
19:   else
20:      isdone = false;
21:      maxLength++;
22:   end if
23: end while
24: outputAllPaths();
```

Calculate the acceptable maximum length of pa

Find adjacer tiles around two pins

For each pos entry node, calculate pat length of *maxlength*

The same for the other pin

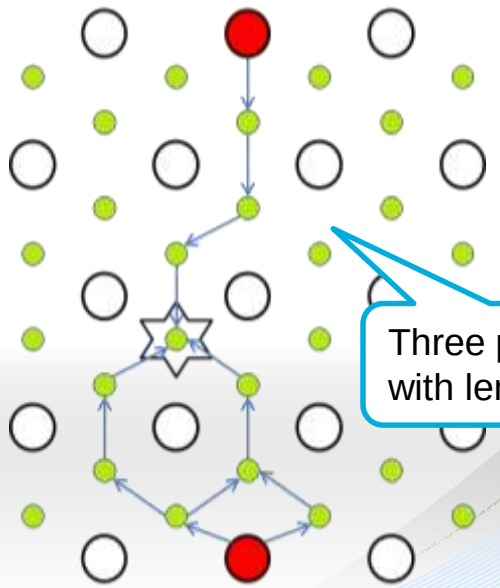
Merge the two path sets and generate the final paths from one pin to another via median point

Algorithm 2 $findPath(pSet, path, tile, mp, region, l)$

Require: $pSet$, $path$, $tile$, mp, $region$, l;

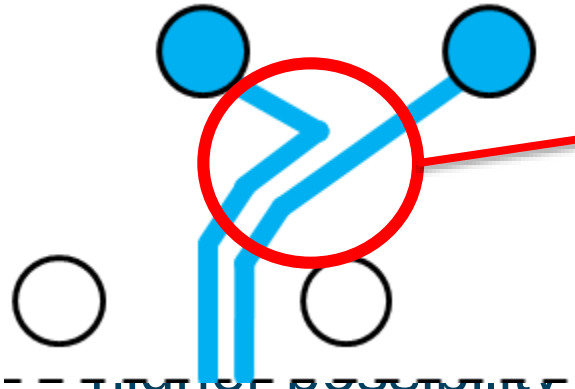
```
1: if length ≤ 1 then
2:    if tile == mpoint then
3:       pSet.push_back(path);
4:    end if
5: else
6:    adjTiles = calTileAdjTiles(tile);
7:    for each t ∈ adjTiles do
8:       if tile ∈ region then
9:          path.push_back(tile);
10:         findPath(pSet, path, tile, mp, region, l − 1);
11:      end if
12:   end for
13: end if
```

Find the paths with length of



Three paths with length of 6

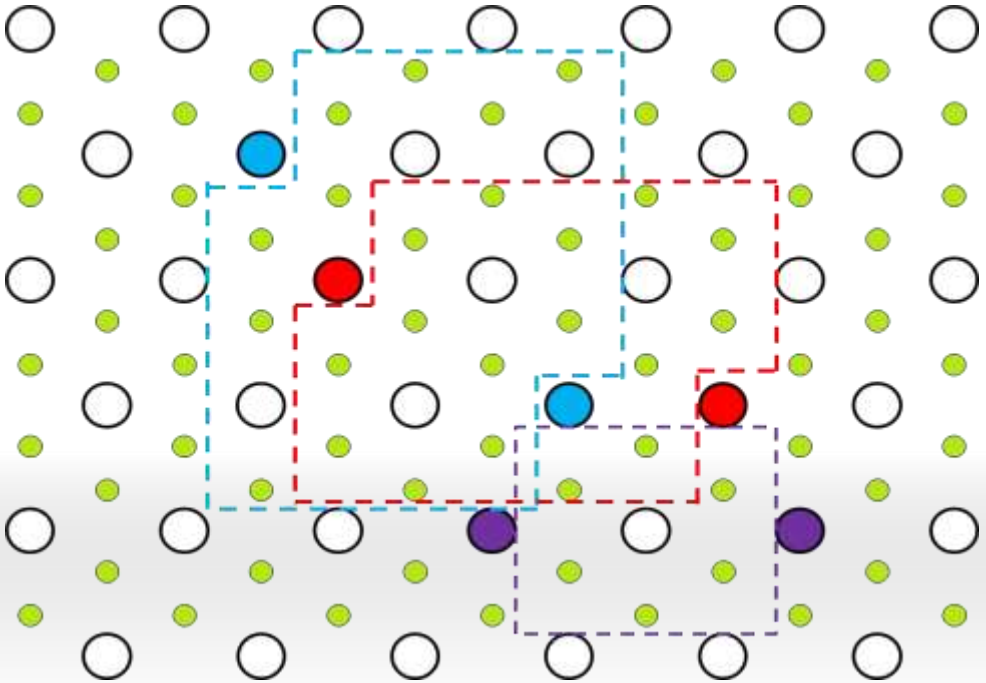
Acute-angle Avoidance for Differential-pair

- Acute-angle Avoidance
  - 
    ath can
    t acute-a
    e
    orities
    y a path possesses, the
    it will be selected as a routing solution
  - For the rest unavoidable cases
    - The 60-degree angle can be split into double 120-degree angles by adding an additional segment with a little wire length sacrificed

Group Dividing Method

Determine the path to be actually used

- Differential pairs are classified into *K* groups according to the crossing possibility
  - *K* is the maximum value that makes the paths in groups without crossing with each other



Median Point Determination

For group G_k

- Objective:

Total length of paths selected

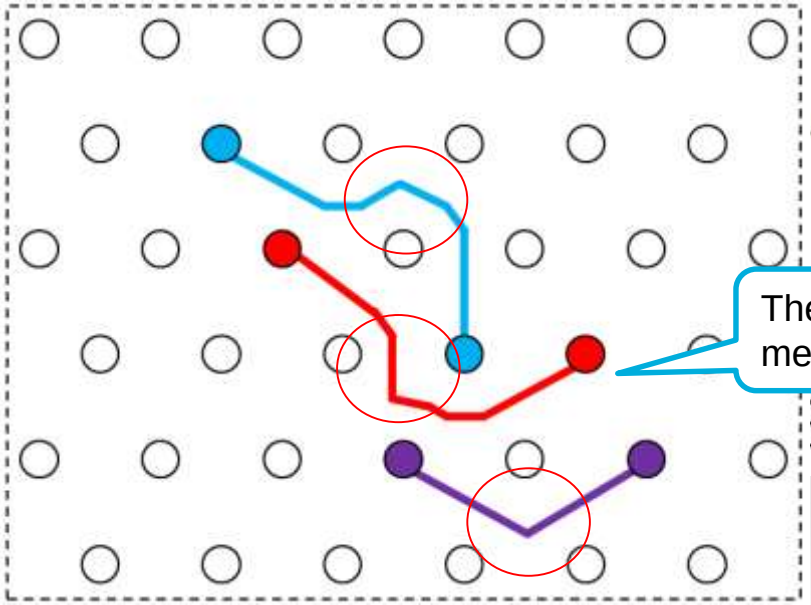
$$Min \sum_{i=1}^{a_k} \sum_{p=1}^{n_{ki}} x_{ip} \cdot l_p$$

Number of path candidates in *Gk*

the sum of the length of path *p* from differential pair pins to median point and the distance of median point to the nearest boundary

If path *p* is selected for differential pair *i,*

Where:



$$x_{ip} = 0 \ or \ 1, \forall 1 \le i \le N$$

The path connecting median point to pins

For each differential pair, one and only one path is assigned

$$\le 1, (ip, jq) \in PCC_k$$

the path crossing cluster for group *Gk*

Step II: Unified Modeling for MPER

Total length of all routing paths

- Objective:

$$Min\ \alpha \times \sum_{e_{i,j} \in E} l(e_{i,j}) \times f_d(e_{i,j}) +$$

Guarantees the resource constraint and differential pair protection constraint.

Subject to

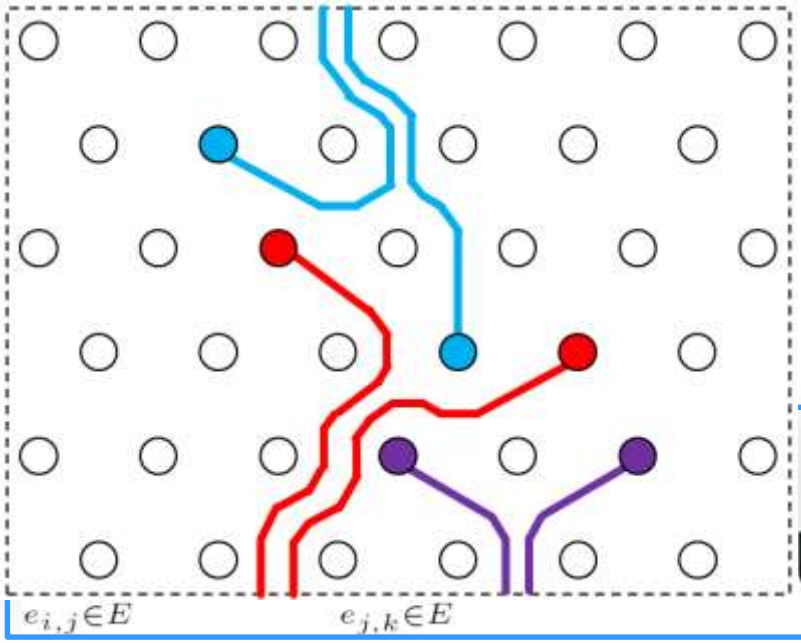
$$\sum_{e_{i,j} \in E} f$$

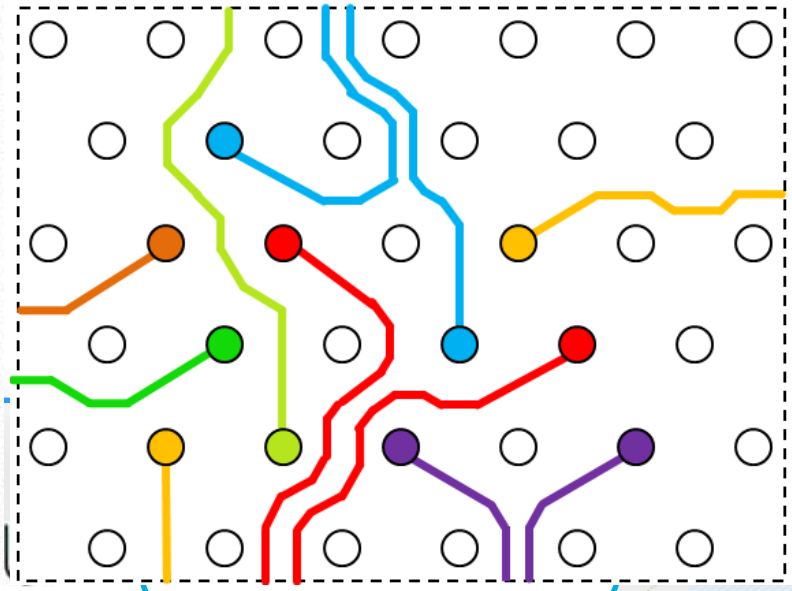
$$\times f_d(e_{i,j}) + f_s(e_{i,j}) \le c(e_{i,j})$$

$$\sum_{e_{i,j} \in Path_{DP}} f_s(e_{i,j}) \le 0$$

Adjust the relative weighting between differential pairs and single signals, in this paper, it is set to 1

Ensure the 100% routa… each





ve

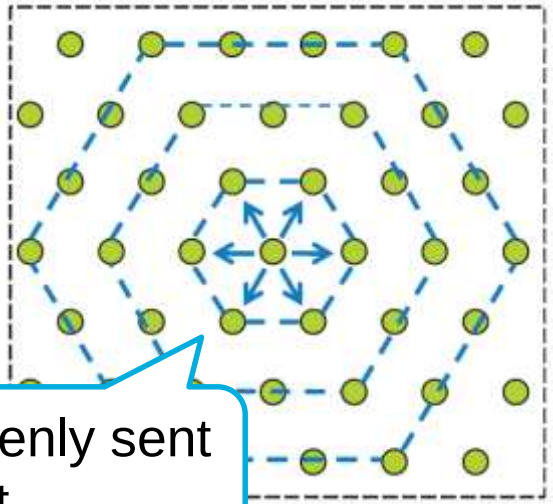
NP-hard Problem

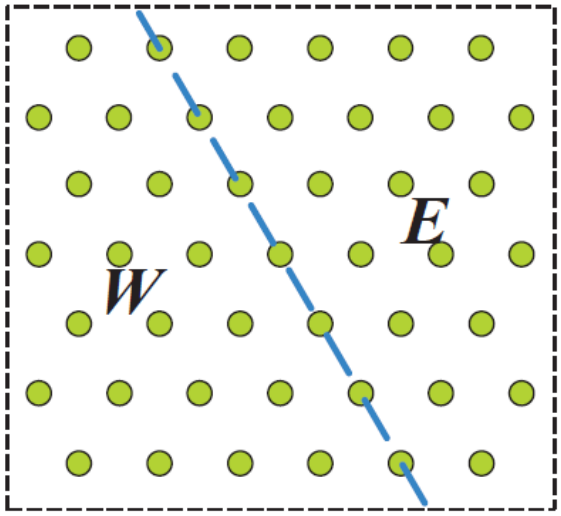
- For single signal, the problem can be transformed into LP problem and solved in polynomial time [ICCAD'11]
- However, for MPER, the situation will be different.
  - First, there are two kinds of input sources
  - Second, as the two kinds of signals take different network resources, more constraints will be brought in to distinguish them.
  - As a result, the problem becomes a **multi-commodity problem** which has been proven to be **NP-hard**.

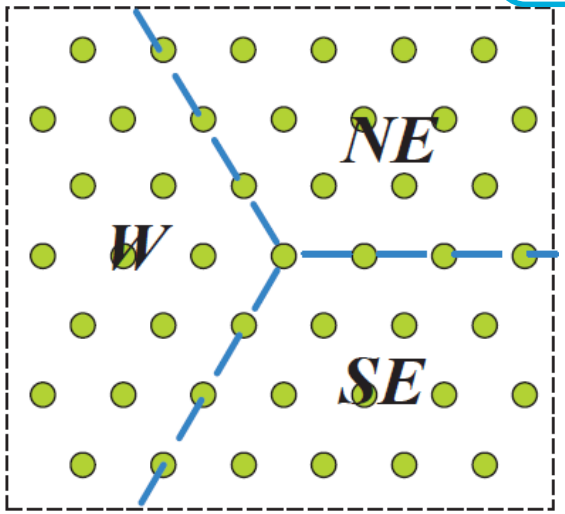
Outline

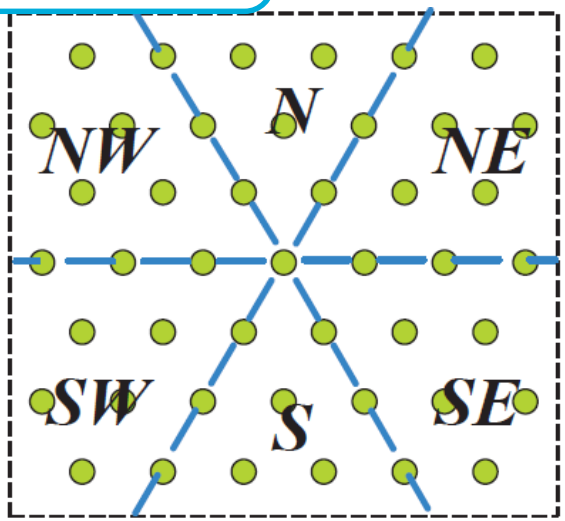
Divergence Property of Escape Routing

- Phenomenon: when a water-drop falls down to the ground, the droplets will spread all around and if the ground is smooth, the spread will be even enough









Slice-based MPER Algorithms

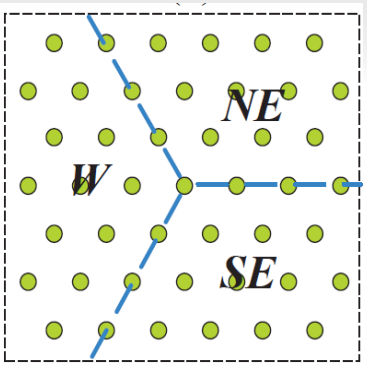
Algorithm 1 Variable Pruning for ILP

Require: *width*, *height*, *sinpins*, *dpairs*, *apartNum*;

```
1: generatePinArray();
2: generateTileArray();
3: createRegions(apartNum);
4: isdone = false;
5: while isdone do
6:    for each i ∈ [1, apartNum] do
7:       initalization(i);
8:       generateDpair(i);
9:       generaterSinpins(i);
10:      IDCoorMapping(i);
11:      createEdges(i);
12:      ILPEscapeRouting(i);
13:      resultsExtracting(i);
14:   end for
15:   if ILP is optimized solved for all regions then
16:      isdone = true;
17:   else
         redistributeFailurePoints();
      end if
   end while
   mergeResults();
   outputAllPaths();
```

each region will solve independently

The chip is partitioned into *apartNum* regions



Tile network is generated and edges are set up for the network flow based ILP formulation

Then an ILP solver is performed to solve the problem

If the ILP is optimized solved for all regions then go

Otherwise, the failed signals in congested regions will be redistributed heuristically into nearby region which has the most routing resource

The results in each region are merged and final routing results are stored

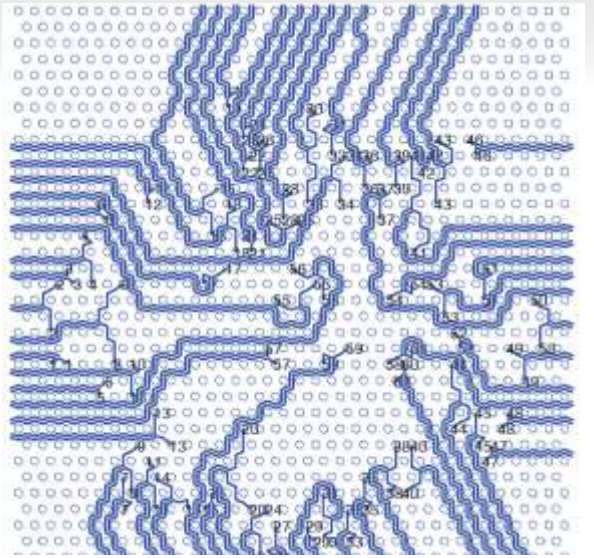
Outline

Experimental results

- Benchmark:
  - Experiment I: effect of differential-pair escape routing on SPA
  - Experiment II: effect of the proposed method
- Workstation:
  - Implemented in *C++*
  - Intel Xeon 2.40GHz CPU and 12GB physical memory
- Tool:
  - *lp_solve* used for ILP and LP solving
- The capacity of edge on tile is set to 2 with only one routing layer considered

Experimental results

- Effect of MPERA on differential-pair escape routing on SPA
  - Compared to grid pin array, the proposed method can reduce wire length by about 13.8% on average and chip area by 13.4%



Escape routing results of case10

Table 2: Effect of escape routing algorithm on staggered pin array

Benchmark information			[5]			Ours			
Benchmark	Dpair	Array size	Avg. l	Run-time (s)	Area	Pin Array len.	Avg. len.	Run-time (s)	Area
		11x8	2.7	<1	70	2.67	2.31	<1	60.62
			4.0	<1	126	4.00	3.46	<1	109.12
		2	4.8	<1	187	4.74	4.10	<1	161.95
		1x3	2.0	<1	20	1.33	1.15	<1	17.32
		4x3	2.6	<1	26	1.76	1.52	<1	22.52
		7x6	4.1	<1	80	4.18	3.62	<1	69.28
		7x6	3.0	<1	80	3.04	2.63	<1	69.28
		x16	3.5	<1	120	3.50	3.03	<1	103.92
		x15	3.3	<1	98	3.30	2.86	<1	84.87
		35x35	12.2	<1	1156	13.50	11.68	2	1001.13
Ratio			1	0.91	1	0.996	0.862	1	0.866

A differential pair escape routing on GPA [ASPDAC'12]

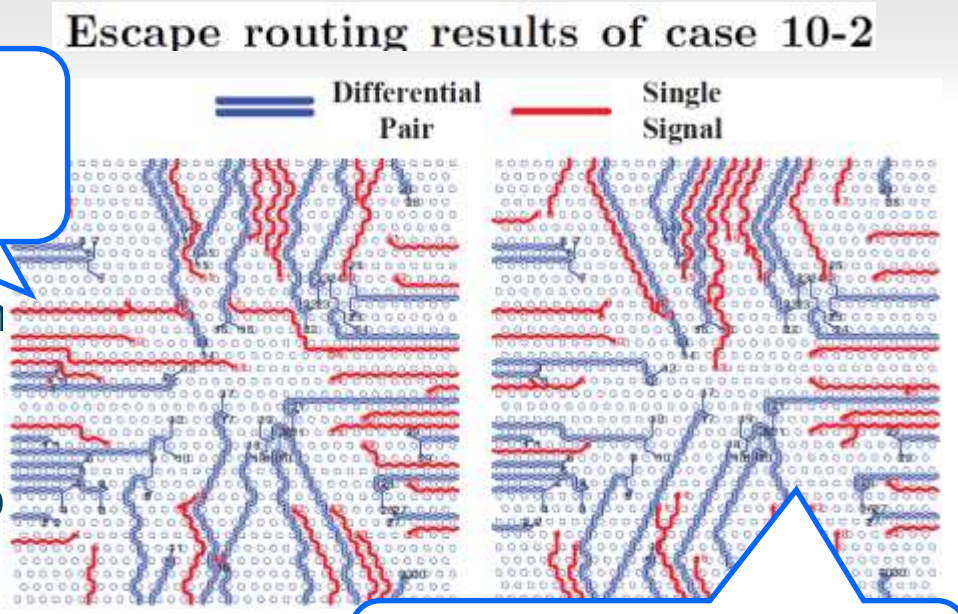
Experimental resultsII

- Effect of M
  Routing
  - Compared to two-stage m
    increase the routability by
  - Reduce the average wire
    average and 22.0% at mo

Two-stage method

Solve the escape routing of differential pairs and single-signals respectively

Our method based on three-division



f the proposed mixed-pattern escap

				Two-stage method							
				DPair length	Sin. length	Routability (%)	Runtime (s)	DP len			
				8.8	8.5	80%	<1	8.8	8.2	100%	<1
				5.78	6.1	88%	<1	5.78	5.67	100%	<1
			2	10.44	7.5	88%	<1	10.44	7.33	100%	<1
				2.4	2.6	90%	<1	2.4	2.2	100%	<1
			x3	3.67	2	100%	<1	3.67	2	100%	<1
case6-2	6	7	17x6	7.17	5.67	85%	<1	7.67	4.43	100%	<1
case7-2	9	9	17x6	4.67	4.88	88%	<1	4.78	4.55	100%	<1
case8-2	10	10	9x16	6.4	7	90%	<1	6.4	6.5	100%	<1
case9-2	10	10	8x15	5.7	6.63	80%	<1	5.8	6.1	100%	<1
case10-2	30	36	35x35	18.33	16.12	88%	2	18.4	14.52	100%	3
case11-2	50	66	35x35	15.92	12.61	69%	2	16.0	11.36	100%	13
Ratio				1	1.093	1	-	1.010	1	1.163	-

Outline

Conclusion

- Staggered pin array has become more and more popular for PCB.
- At the same time, differential-pair is a good method to increase noise immunity of signal for high-speed signal transmission
- In this paper, an algorithm for escape routing of both differential-pair and single signals is proposed on staggered pin array based PCB.
- Experimental results show that the proposed method can solve both single-pattern and mixed-pattern effectively.

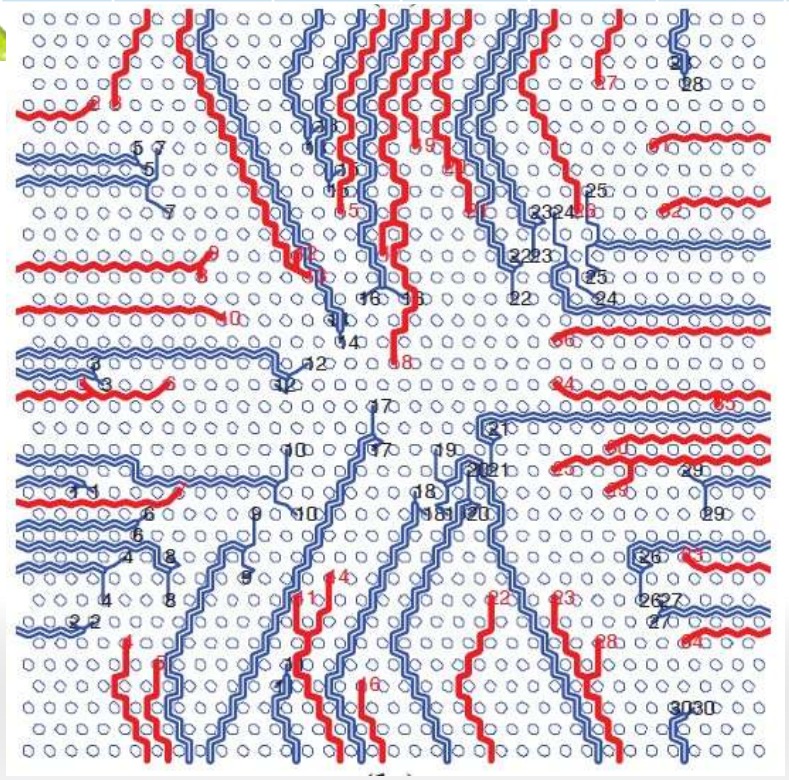
Thanks

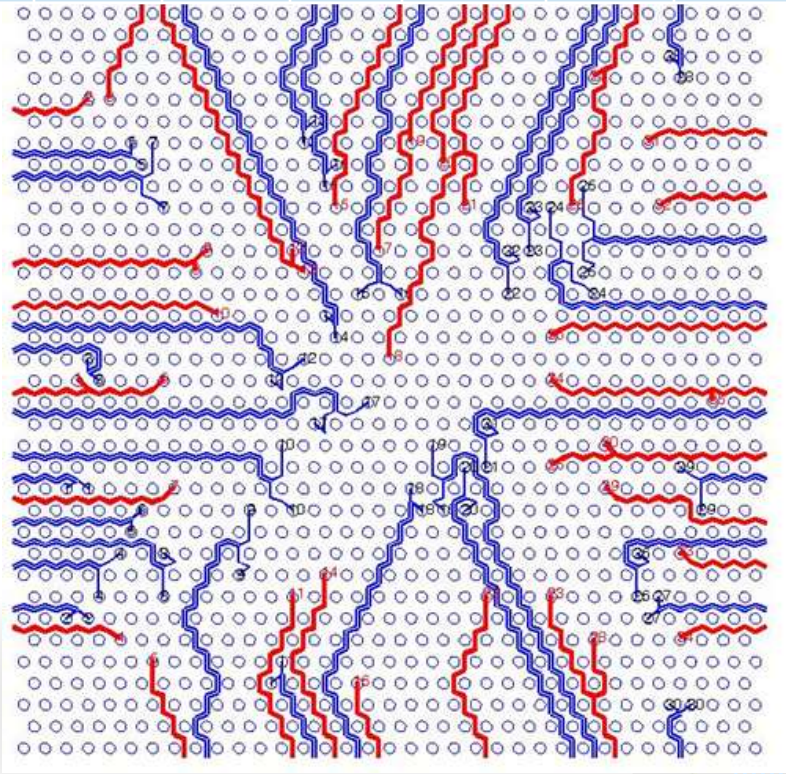
Q&A

Email: wangkan09@mails.thu.edu.cn

PCB Routing Results

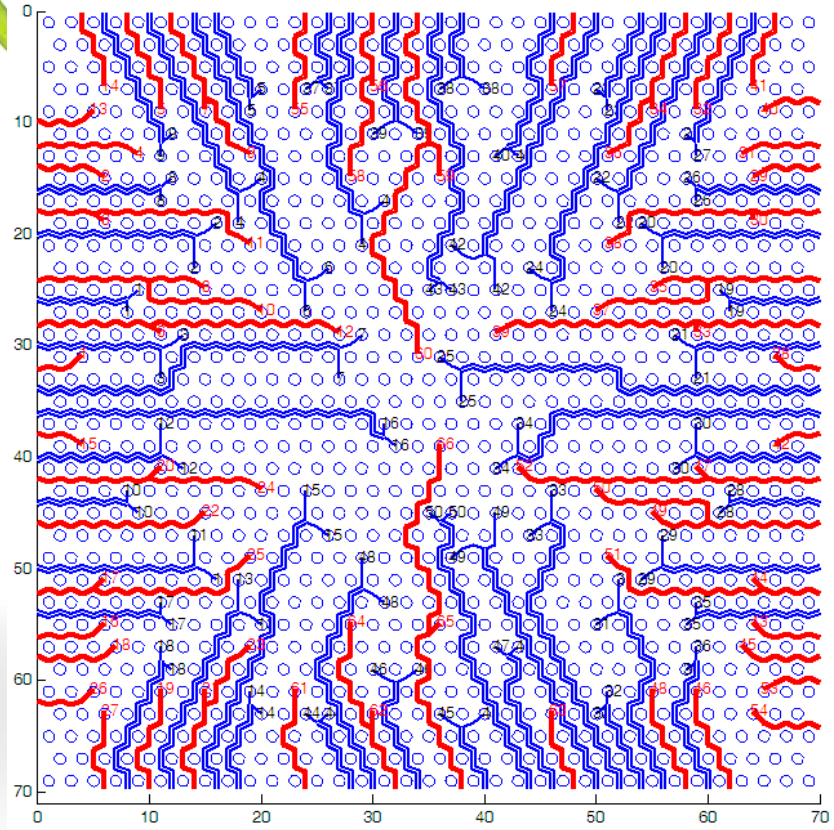
Diff. pair #	Sin. Pin#	array size	Diff. Pair length	Sin. Signal length	Routability (%)	Run-time (s)	Diff. Pair length	Sin. Signal length	Routability (%)	Run-time (s)	Diff. Pair length	Sin. Signal length	Routability (%)	Run-time (s)
30	36	35x35	18.33	16.12	86%	2	18.40	14.52	100%	6	18.47	14.53	100%	2



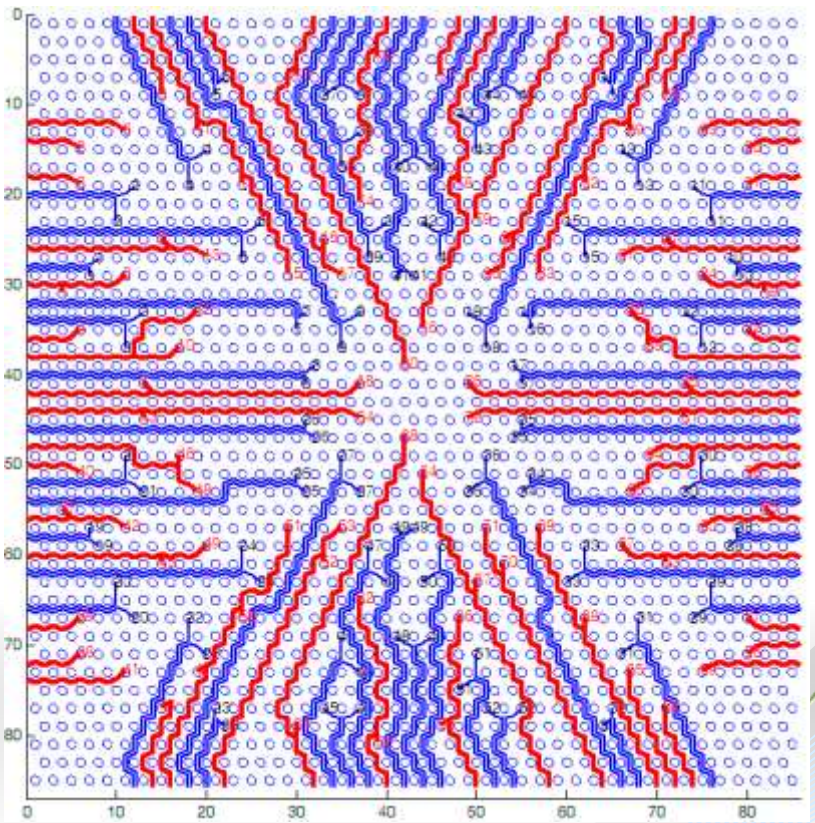


For large cases

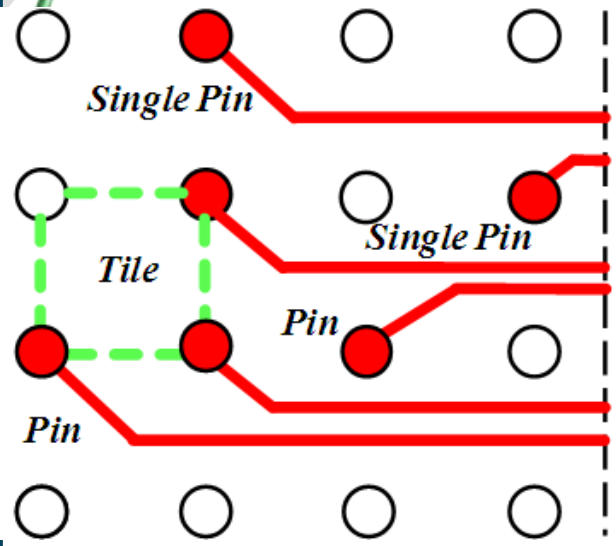
Diff. pair #	Sin. Pin#	array size	Run-time (s)
50	66	35x35	13

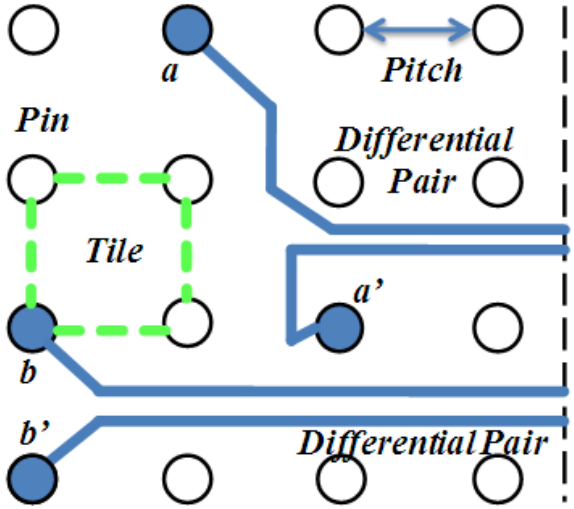


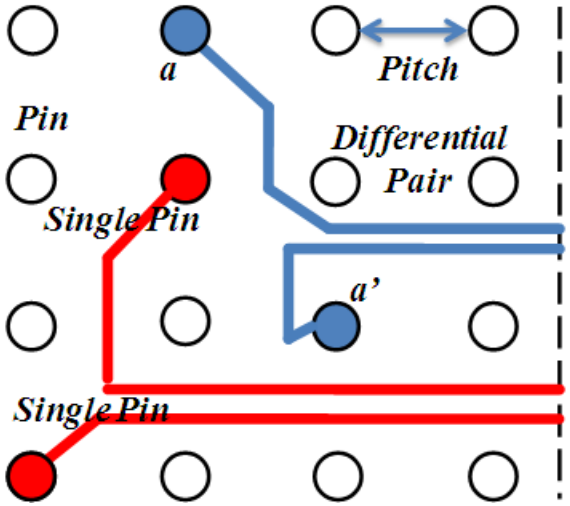
Diff. pair #	Sin. Pin#	array size	Run-time (s)
52	88	43x43	37

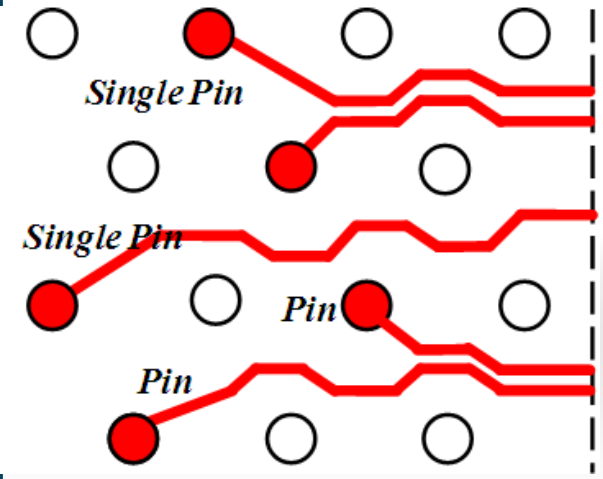


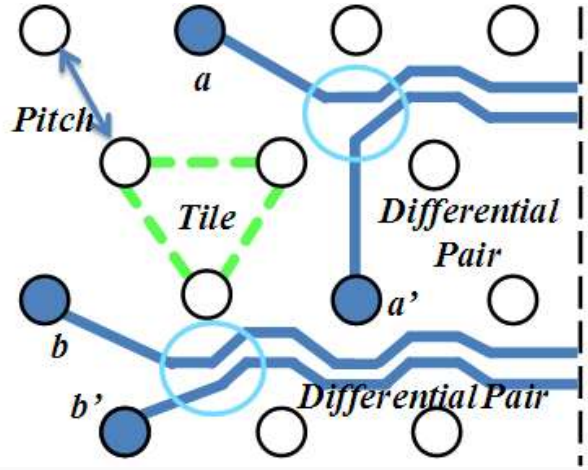
Related Work III

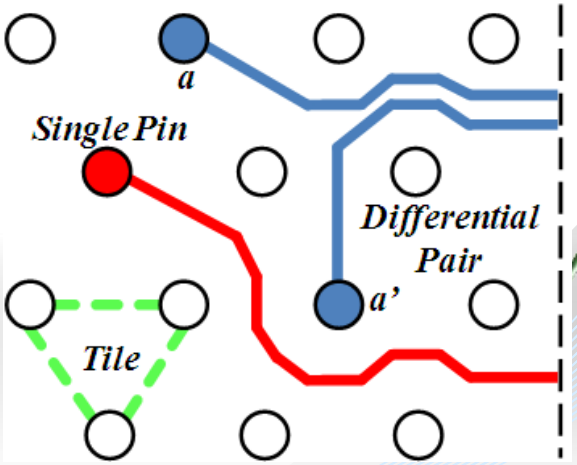












Two transformations

- Modified MPERA for single-pattern escape routing
  - the problem will be transformed to the LP problem in [ICCAD'11]
- MPERA considering crosstalk between single signals:

$$f_d(e_{i,j}) + f_s(e_{i,j}) \leq 1$$

- MPERA without considering the crosstalk between single signals
  - *Slice-based MPERA*

Issues of PCB Routing

- High speed printed circuit board (PCB) routing has become more and more difficult for manual design
  - Due to increased pin count and dwindling routing resource