

# CDCTree: Novel Obstacle-Avoiding Routing Tree Construction based on Current Driven Circuit Model

---

---

Yiyu Shi<sup>1</sup>, Tong Jing<sup>2</sup>, Lei He<sup>1</sup>,

Electrical Engineering Department<sup>1</sup>  
UCLA<sup>1</sup>

Los Angeles, California, 90095-1594, USA

Tel: 310-206-2037

Fax: 310-206-4685

e-mail: {yshi, lhe}@ee.ucla.edu<sup>1</sup>

Zhe Feng<sup>2</sup> and Xianlong Hong<sup>2</sup>

Computer Science & Technology Department<sup>2</sup>  
Tsinghua University<sup>2</sup>

Beijing, P.R.China 10084

Tel: +86-10-62785564

Fax: +86-10-62781489

e-mail: {jingtong, hxl-dcs}@tsinghua.edu.cn<sup>2</sup>

***Speaker: Lei He***

# Outline

---

- **Introduction**
  - Circuit Model and Tree Construction
  - Experimental Results
  - Conclusion
-

# Motivation

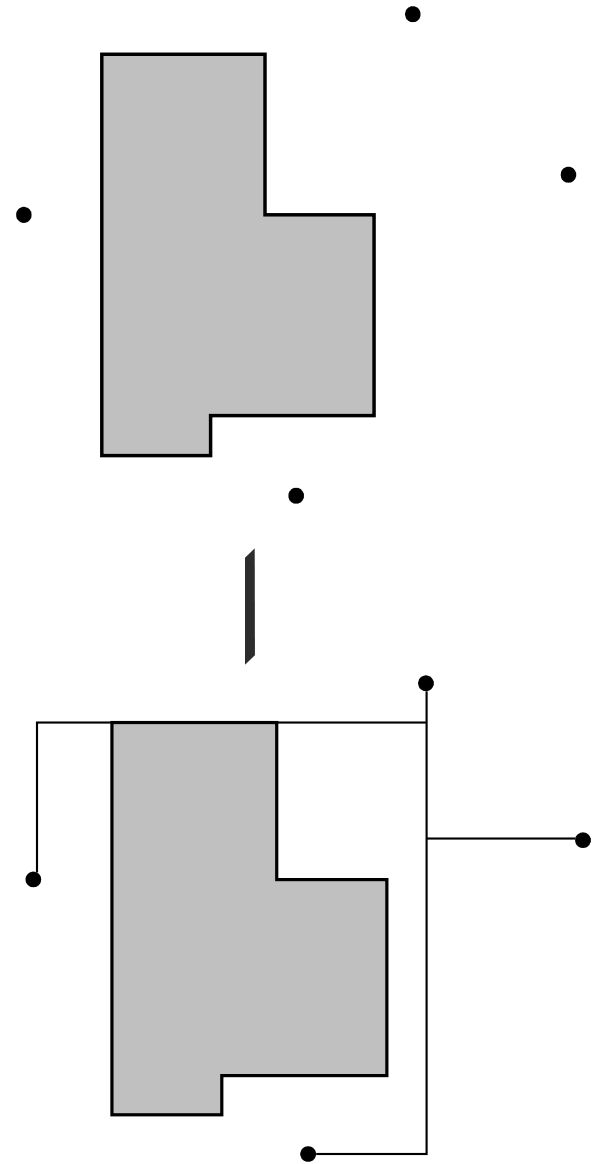
---

- Constructing a rectilinear Steiner minimal tree (RSMT) is a fundamental problem
    - ⦿ For CAD and algorithm development
  - Macro cells, IP blocks, and pre-routed nets are often regarded as obstacles
  - Obstacle-avoiding RSMT (OARSMT) construction is often used to estimate wire length and delay for global routing
  - OARSMT is NP-hard, and there exists a large room to improve existing heuristics
-

# OARSMT Formulation

---

- **Given:** terminal set  $N$  and obstacle set  $O$  in the Manhattan planar.
- **Find:** rectilinear Steiner trees  $ST$  to connect all terminals in  $N$  while avoiding all obstacles in  $O$ .
- **objective:** minimize total wire length  $L$



# Existing Algorithms

---

- Algorithms (complexity depends on the size of routing area)
  - ⊙ Maze routing [C.Y. Lee, 1961]
  - ⊙ Line search routing [K. Mikami and K. Tabuchi, 1968; D.W. Hightower, 1969]
- Algorithms (complexity depends on the size of cases)
  - ⊙ G3S, B3S, and G4S heuristics [J. L. Ganley and J. P. Cohoon, 1994]
  - ⊙ Exact OAESMT [M. Zachariasen and P. Winter, 1999]
  - ⊙ Based on generalized connection graph [Z. Zhou, C. D. Jiang, L. S. Huang, *et al*, 2003]
  - ⊙ 2-step heuristic [Y. Yang, Q. Zhu, T. Jing, X. L. Hong, *et al*, 2003]
  - ⊙ FORst [Y. Hu, Z. Feng, T. Jing *et al*, 2004]
  - ⊙ An-OARSMAN [Y. Hu, Z. Feng, T. Jing *et al*, 2005]

# Primary Contribution of this Paper

---

- All existing algorithms are based on combinatorial optimization
- We propose to simulate routing problem by current-driven circuit, and called our algorithm as CDCtree
  - ⊙ A new addition to simulated algorithms such as simulated annealing, genetic algorithm, and force-based placement
  - ⊙ Similar idea used in [Y. Shi *et al*, 2004] for traffic flow distribution problem.
- Experiments show that we reduce wire length by up to 11.4%
  - ⊙ Compared to the best existing algorithm

# Outline

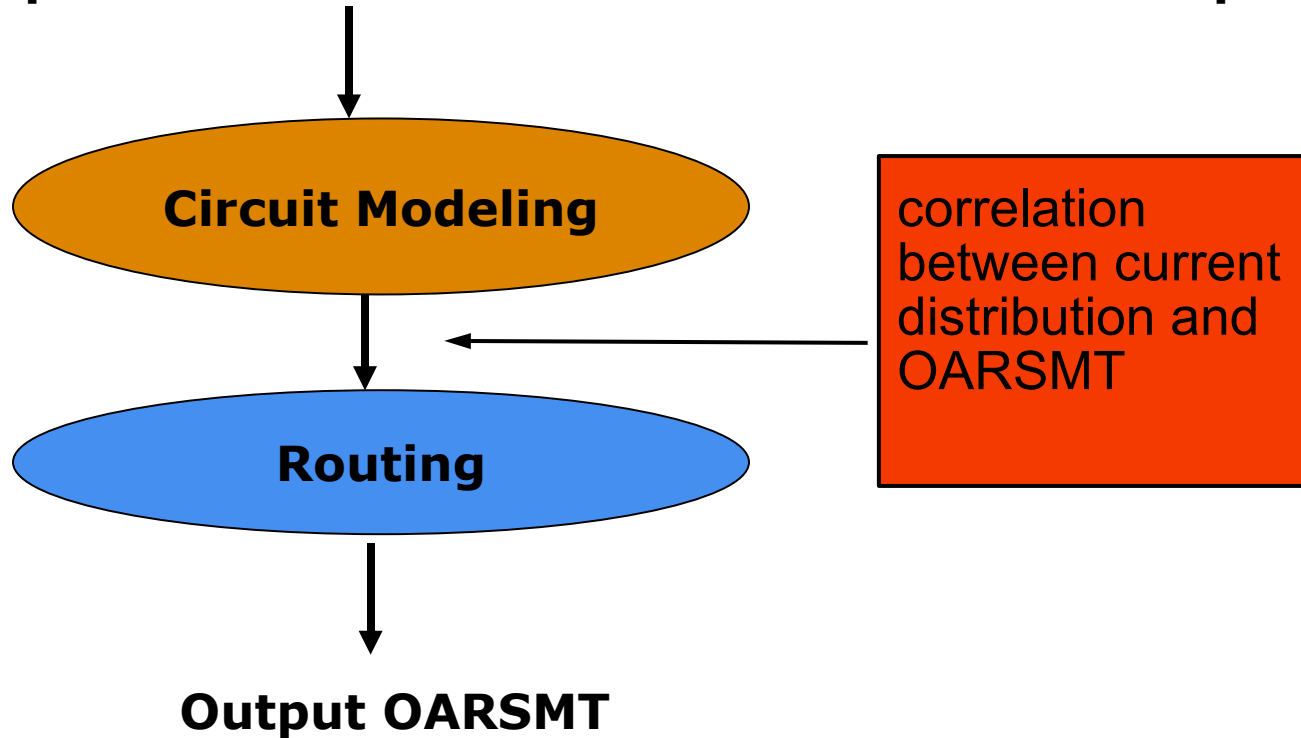
---

- Introduction
  - **Circuit Model and Tree Construction**
  - Experimental Results
  - Conclusion
-

# CDCTree: Overview (Two-step procedure)

---

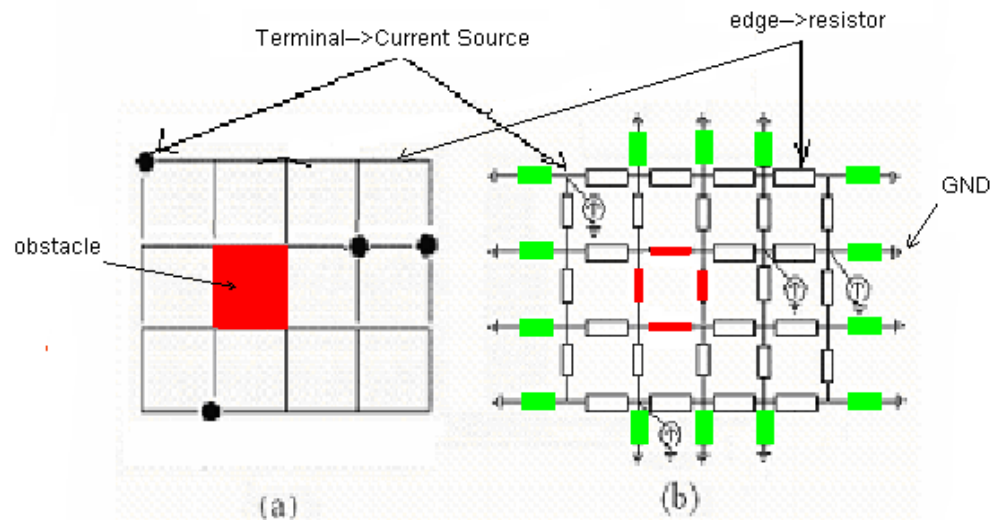
**Input Terminal and Obstacle Location and Shape**





# Circuit Model

- Build escape graph
  - by vertical and horizontal lines from terminals and obstacle edges
- Place a resistor at each edge of the graph
- Add a current source at each terminal
- Connect the circuit to the ground by
  - Either infinite grid (accurate but expensive)
  - Or connecting periphery nodes to *GND* via resistors (approximation)



# Values of Resistors

- **GND Resistor**

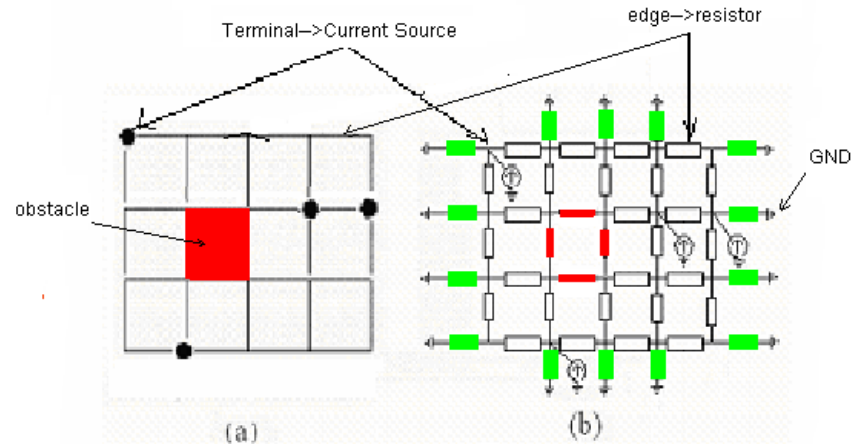
- ⊙ *Constant  $R = 1\text{ohm}$*

- **EDGE Resistor**

- ⊙ Longer edge  $\Leftrightarrow$  smaller current
- ⊙ Set  $R = 10 - \log(L)$ , where 10 is used to guarantee  $R > 0$

- **Obstacle Resistor**

- ⊙ usually have larger current (current flow congestion effect)
- ⊙ Set  $R = 20 - \log(L)$  to offset the effect



# Circuit Simulation

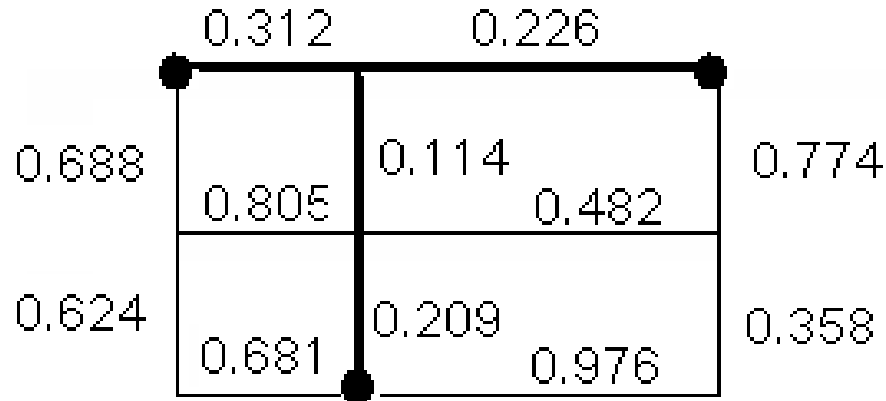
---

- The current through each edge can be solved by using nodal analysis (NA) method .
  - The equations can be written in matrix form as  $Ax=b$ , where  $A$  is a positive definite diagonally dominant sparse matrix.
  - The equations can be solved efficiently.
-

# Current versus Routing (Key of this work)

---

- Edges with smaller currents construct the tree.

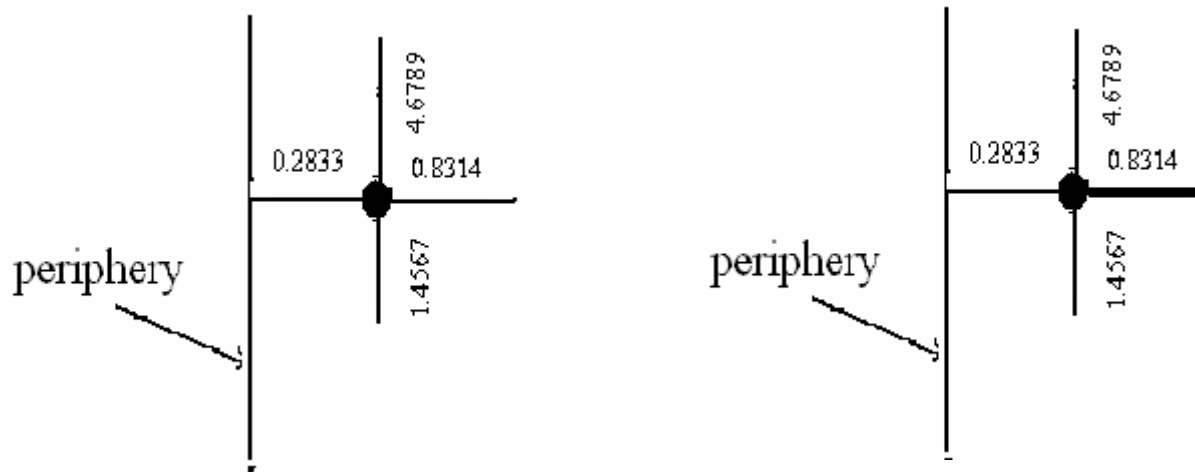


- In general, short-cuts between terminals have stronger compelling forces from current sources at terminals, and therefore have smaller current.

# Tree Construction

---

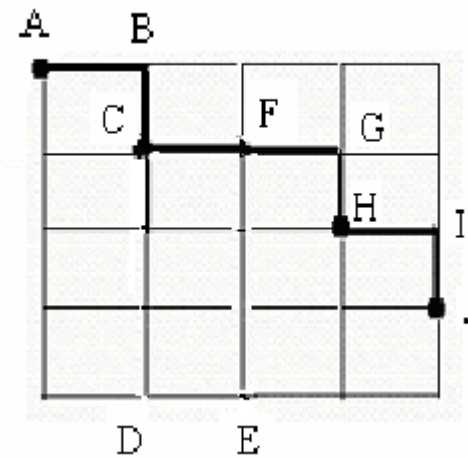
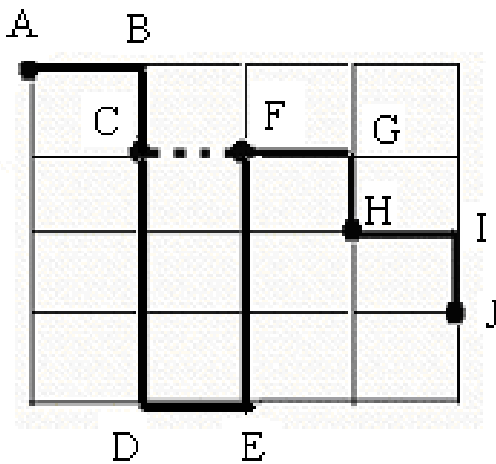
- Simultaneously start from each terminal and always try to select the edge with the minimum current
- To compensate for the effect that *GND* resistors draw large current
  - ⊙ May not move to the periphery node



# Tree Construction

---

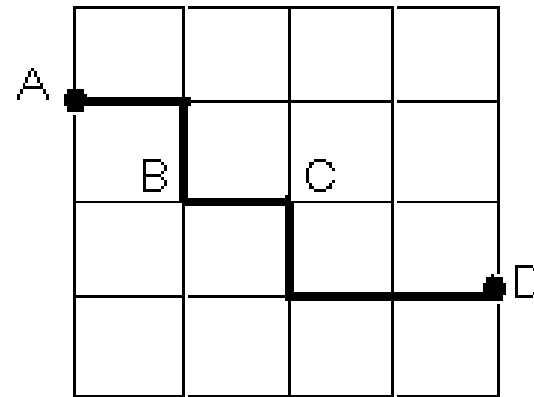
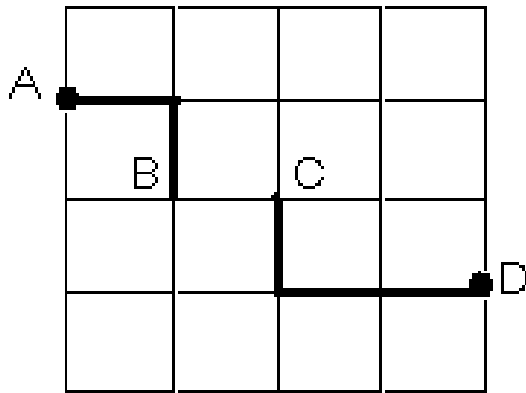
- When a neighbor is already visited by the same track, perform U-reduction



# Tree Construction

---

- When a neighbor is already visited by other tracks, merge them



# Outline

---

- Introduction
  - Circuit Model Construction and Tree Algorithm
  - **Experimental Results**
  - Conclusion
-



# Experiment Setting

---

- Experiment platform
  - ⊙ Hardware: sun V880 fire workstation
  - ⊙ Software: gcc2.9.1, solaris5.8
- Benchmark data
  - ⊙ Randomly generate terminals with several kinds of rectilinear polygon obstacles
    - rectangle, L-shaped, cross-shaped, etc.
  - ⊙ Compare to the best existing algorithm:
    - An-OARSMAN [Hu *et al*, 2005]

# Comparison with An-OARSMAN

Terminal #	Obs #	An-OARSMAN		CDCTree		
		Wire Length	Running Time (s)	Wire Length	Running Time (s)	
					I	II
3	3	2350	<0.01	2350	<0.01	<0.01
5	3	4380	0.01	4350	0.01	<0.01
7	5	9610	0.04	9610	0.02	0.01
10	5	11340	0.06	10980	0.08	0.02
20	7	14790	0.28	13110	0.84	0.13
30	10	21220	0.77	19970	1.38	0.25
40	10	28600	1.98	27300	4.32	0.77
50	15	31330	3.22	31310	10.25	1.02

CDCtree reduces the wire length by up to 11.4%

# Comparison with An-OARSMAN

Terminal #	Obs #	An-OARSMAN		CDCTree		
		Wire Length	Running Time (s)	Wire Length	Running Time (s)	
					I	II
3	3	2350	<0.01	2350	<0.01	<0.01
5	3	4380	0.01	4350	0.01	<0.01
7	5	9610	0.04	9610	0.02	0.01
10	5	11340	0.06	10980	0.08	0.02
20	7	14790	0.28	13110	0.84	0.13
30	10	21220	0.77	19970	1.38	0.25
40	10	28600	1.98	27300	4.32	0.77
50	15	31330	3.22	31310	10.25	1.02

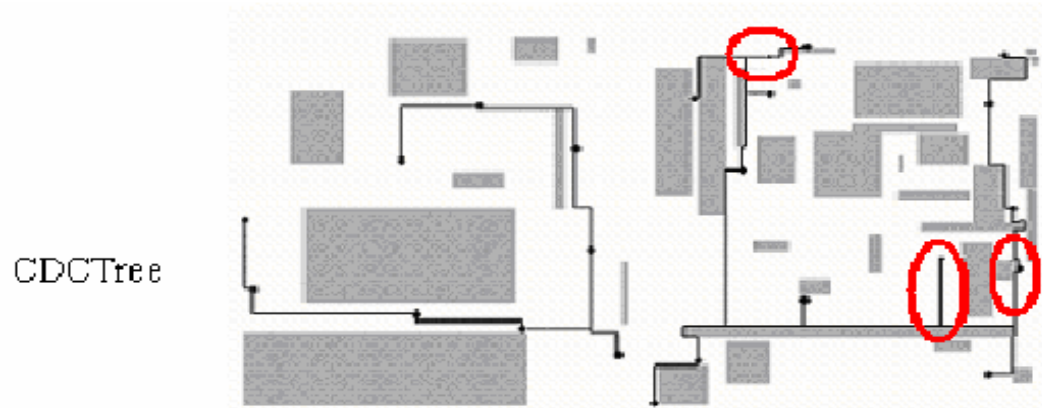
Runtime I : solve linear equations (dominance); Runtime II: construct RSMT

Gaussian-Siedel iteration is currently used and over 1000x speedup can be achieved to make runtime in Part I negligible.

# Routing with Convex Obstacles

---

Examples from [Hu *et al*, 2005]



Wirelength

11280

12280



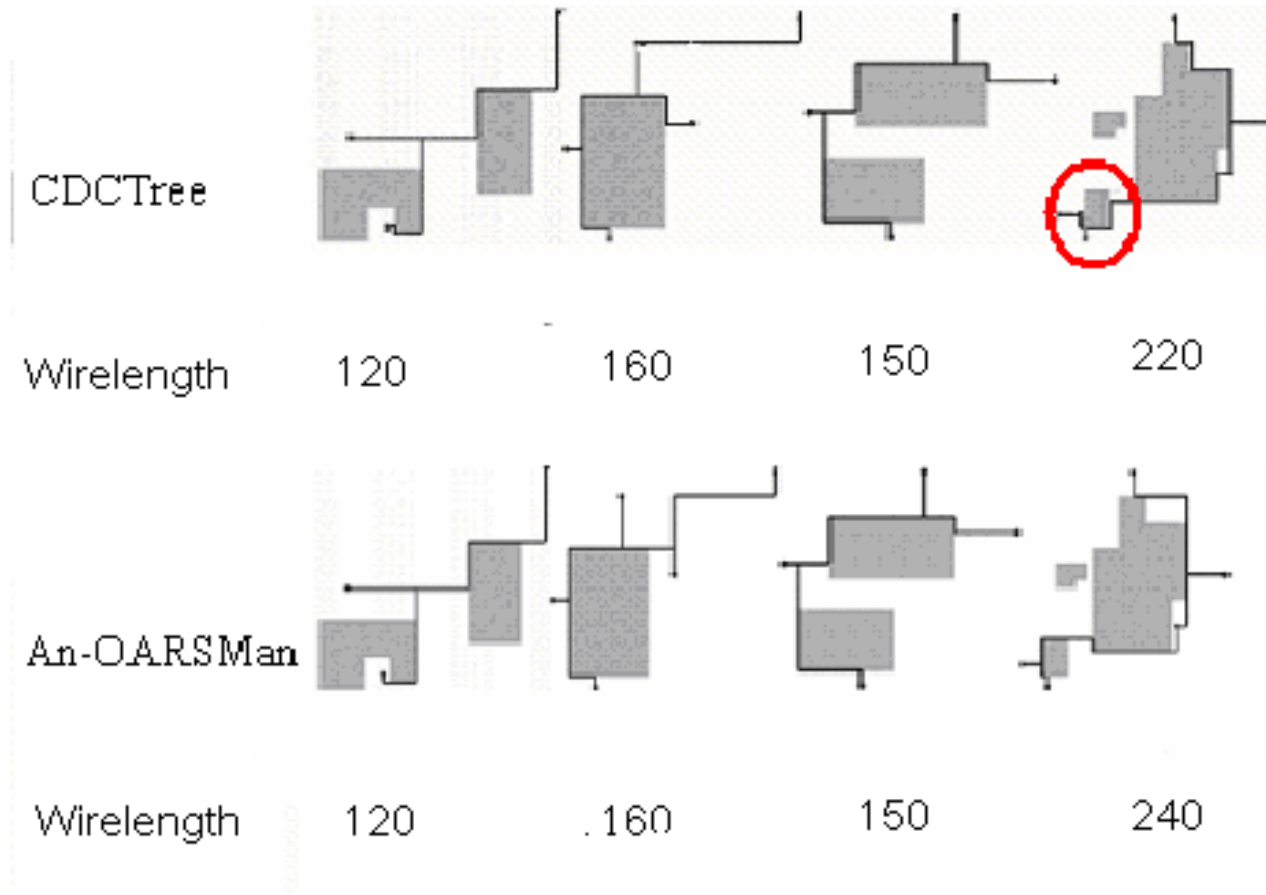
Wirelength

11280

13790

# Routing with Concave Obstacles

Examples from [Hu *et al*, 2005]



# Outline

---

- Introduction
  - Circuit Model Construction and Tree Algorithm
  - Experimental Results
  - **Conclusion**
-

# Conclusion

---

- A new simulation based CAD algorithm to simulate routing by current-driven circuit (CDCtree)
  - CDCtree reduces wire length by up to 11.4% compared to best existing algorithm
  - Our newest results able to reduce runtime by 100x and with up to 15% more reduction of wire length
-