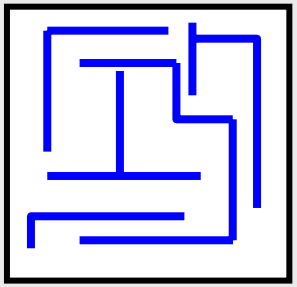
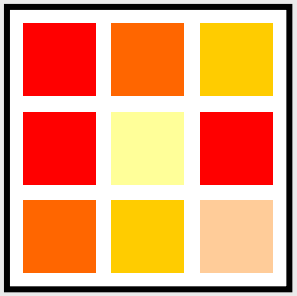
Is Probabilistic Congestion Estimation Worthwhile?

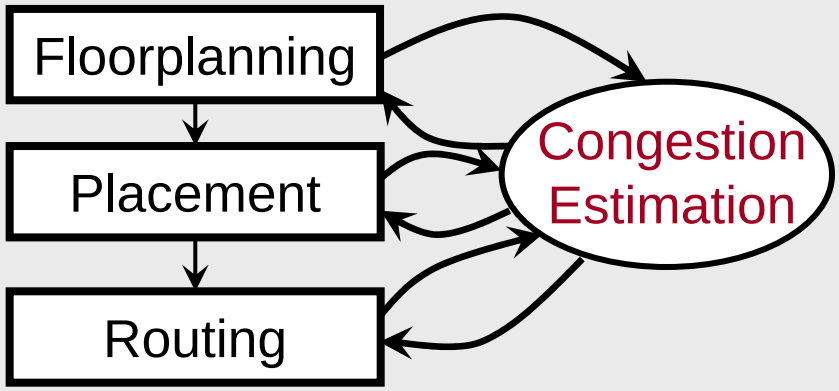
Jurjen Westra and Patrick Groeneveld
Eindhoven University, Fac. Electrical Engineering
ES Group
jwestra@ics.ele.tue.nl

❖ Introduction

- What is (Global) Routing?
  - Matching routing supply and demand.
- What is Congestion Estimation?
  - *Guessing* where matching is difficult!
- Why Congestion Estimation?
  - Prevent routability problems later on!







- Traditionally: Global Routing
  - Accurate / slow
- 2001: Probabilistic Methods
  - Fast / less accurate

new!

❖ Outline

- Motivation
- Implementation of probabilistic method
- Implementation of new GR-based method
- The benchmarks
- Estimation quality
- Results
- Discussion
- Conclusion

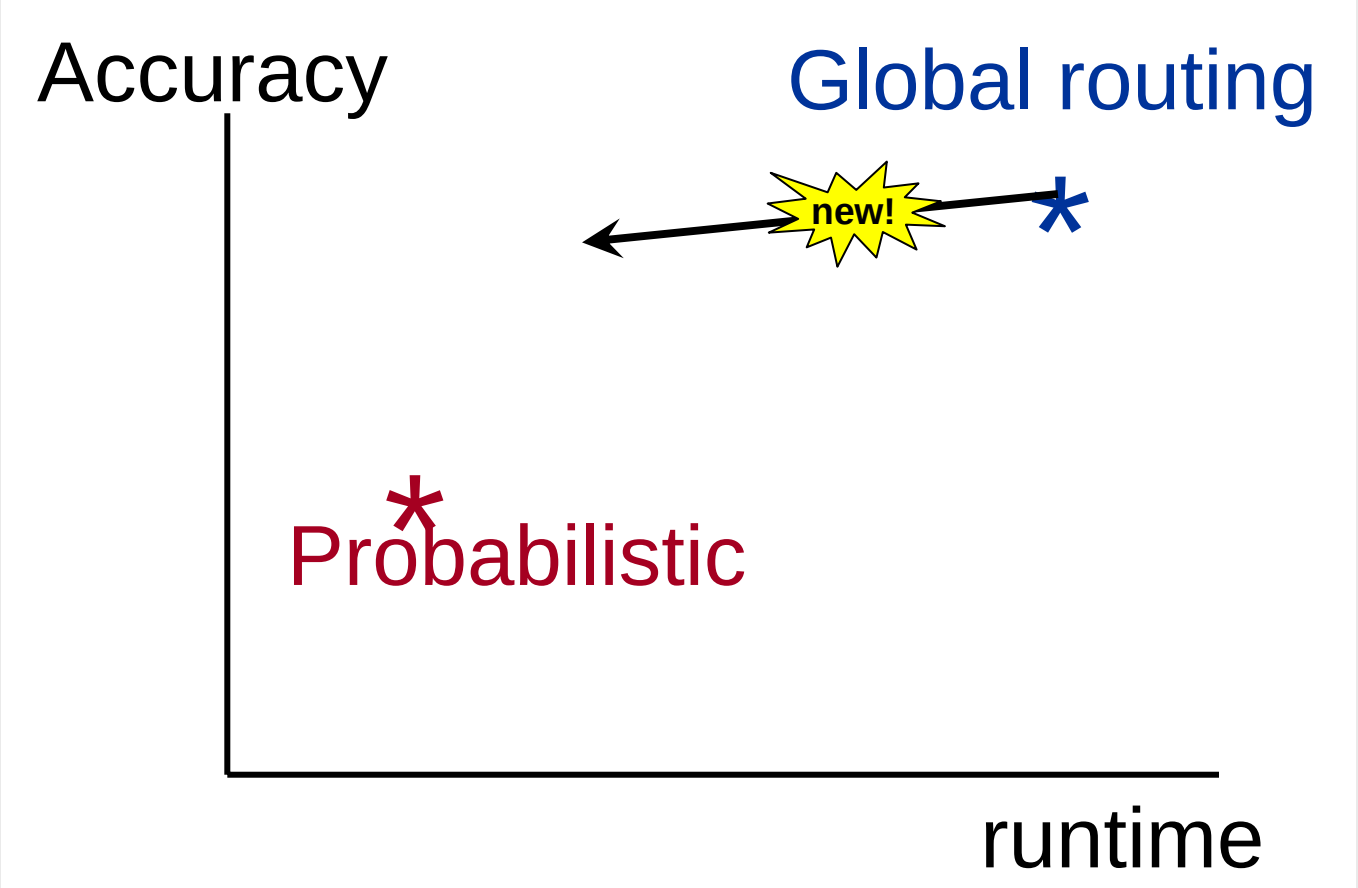
❖ Motivation

- Probabilistic methods differ in some of their observations.
  - Biased towards tool or benchmarks!
- Fundamentally, there is little awareness of congestion during probabilistic analysis.
  - Re-spreading afterwards.

Global routing is tuned towards minimizing congestion.

What if we tune it towards congestion estimation?

❖ Motivation



❖ Implementation of pce (Westra*)

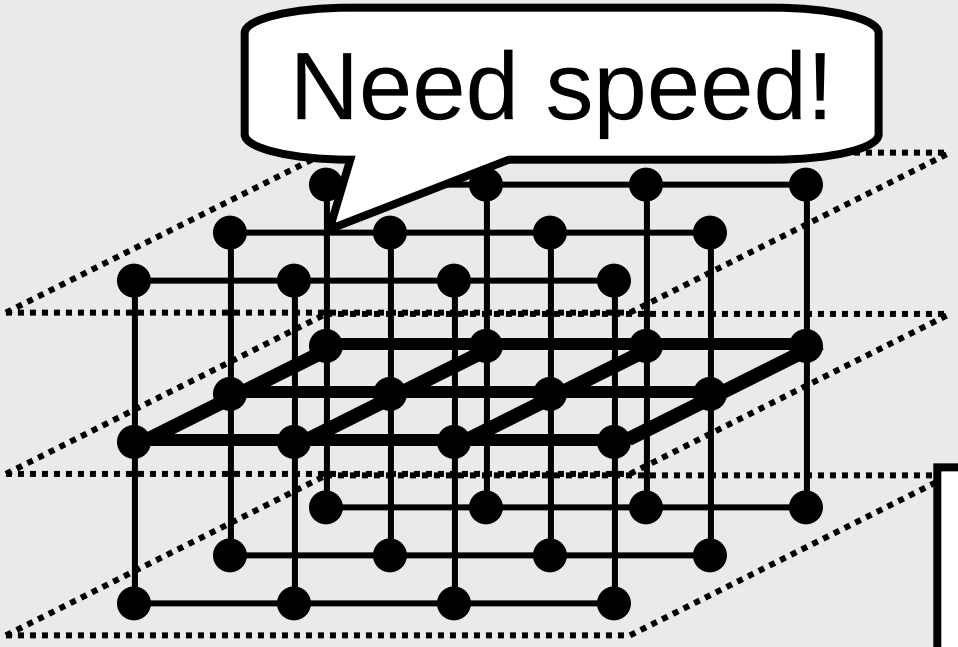
```
create maps h,v
create hashmap stamplib
break up nets with RMST

foreach wire w do
  if stamplib contains w do
    s ← get_stamp(stamplib,w)
    stamp_maps(h,v, w,s)
  else do
    s ← new_stamp(w)
    put_stamp(stamplib,w,s)
    stamp_maps(h,v, w,s)
  end
end
divide h and v by capacities
```

*J. Westra *et al*., “Probabilistic Congestion Prediction”, ispd04

❖ Fast degenerate global routing

- FaDGloR is a maze router tuned towards speed and congestion estimation. (new!)

- Degenerate global routing model



No layer-assignment

➔

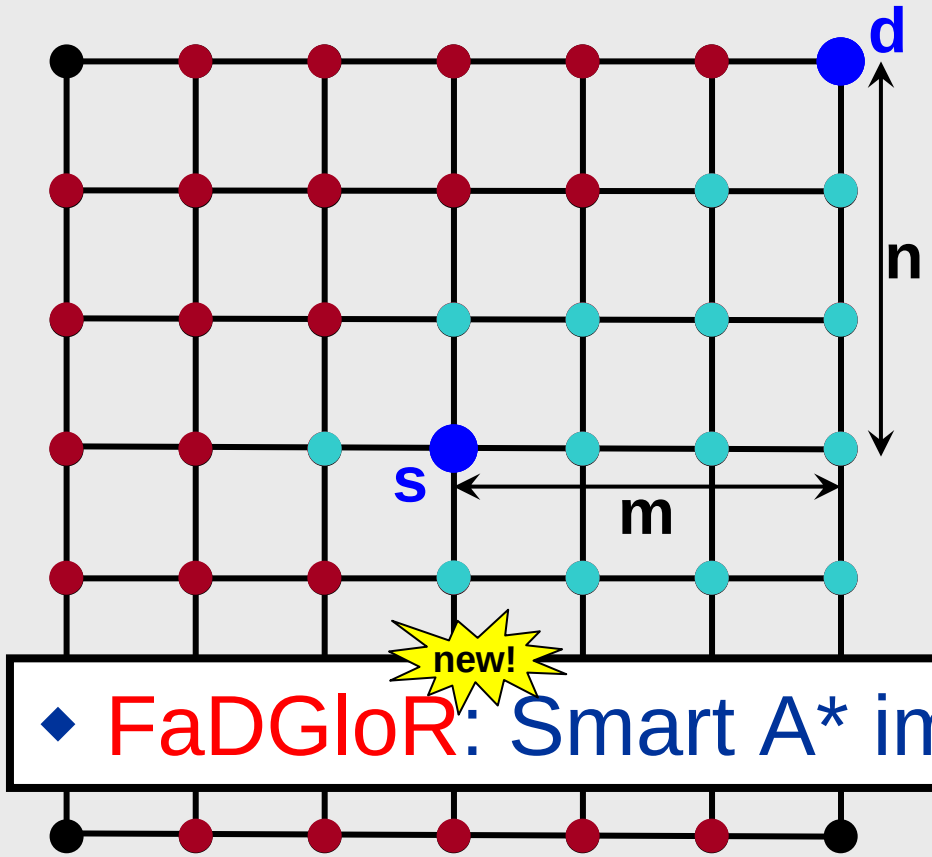
Lower complexity

Common in academic routers.

Minor effect on estimation!

❖ Fast degenerate global routing

- Common in GR: Dijkstra's algorithm.
- Faster: A* algorithm.



- Visited nodes:
  - Dijkstra: $O((m+n)^2)$
  - A*: $O(m+n)$

If uncongested!

- FaDGloR: Smart A* implementation!

❖ Fast degenerate global routing

- GR: ripup-and-reroute (R&R).
  - First, allow overflow, then spread congestion
  - Time consuming

Wire is routed X times!

new!

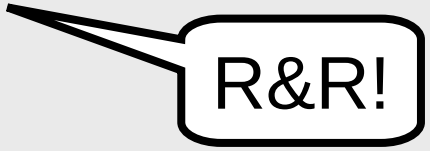
- FaDGloR: ~~R&R~~.
  - Make the tradeoff between wire length and congestion in one shot!

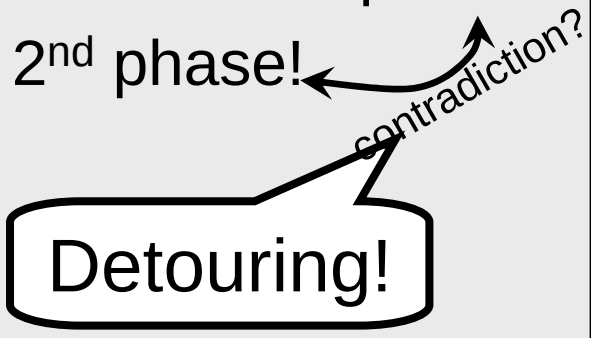
new!

- FaDGloR: 2 step strategy:
  - First route with no overflow ➔ unrouted wires.
  - Route remaining wires for minimum overflow.

❖ Fast degenerate global routing

- Wire ordering: common in GR: shortest first
  - Literature: little effect
  - Experiments:
    - Longest first ➔ Less unrouted distance after 1st phase!
    - Shortest first ➔ Less overflow after 2nd phase!



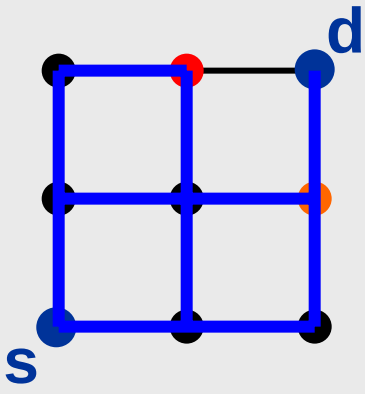


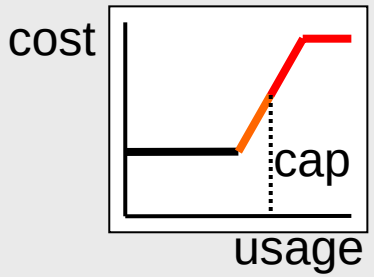
- FaDGloR: Shortest wires first.

❖ Fast degenerate global routing

- GR: Cost-functions: path segments have cost
  - Penalize distance and local congestion
  - Increase cost of congestion during R&R

- CFs effective during R&R.
  - FaDGloR: no congestion information available!
- CFs ➔ more nodes visited!





Longest wires last
➔
problem worse!

new!

- FaDGloR:
  - ~~Cost-function~~
  - Detour-bounding

❖ The benchmarks

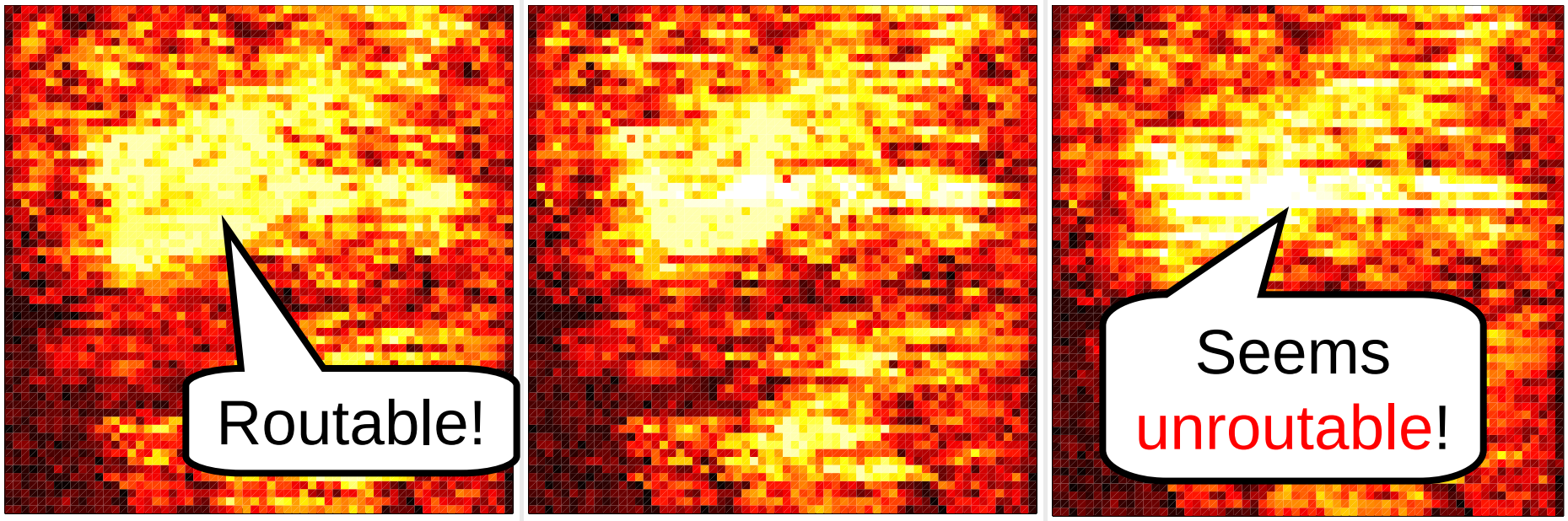
- The Labyrinth benchmarks
  - Only commonly used GR benchmarks (?)
  - From real designs.
- But they are all difficult?!?!?!
  - In reality very easy to impossible designs!

Chip	Grid	Nets	wires	Chip	Grid	Nets	wires
ibm01	64x64	12k	27k	ibm06	128x64	33k	79k
ibm02	80x64	18k	53k	ibm07	192x64	44k	105k
ibm03	80x64	22k	44k	ibm08	192x64	48k	128k
ibm04	96x64	26k	52k	ibm09	256x64	50k	124k
ibm05	128x64	28k	90k	ibm10	256x64	64k	175k

new!

- FaDGloR: Added capacity -5…+10.

❖ Estimation quality

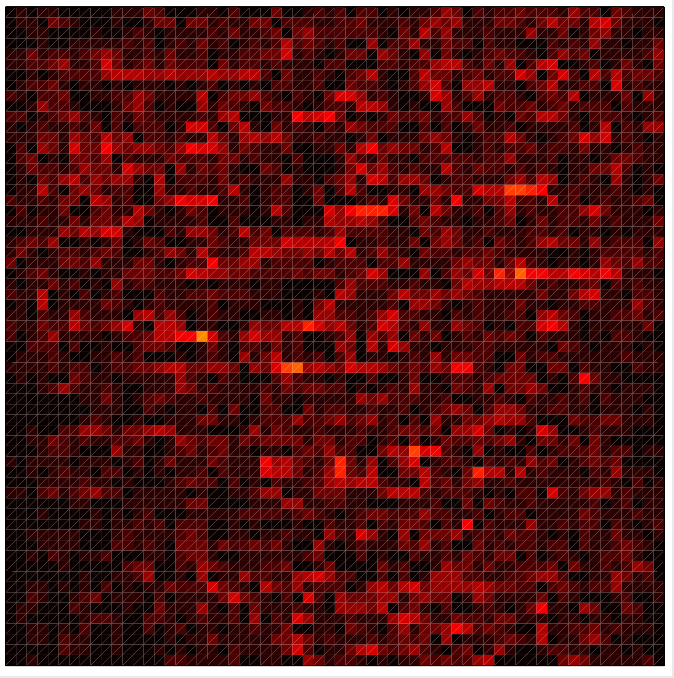


Labyrinth FaDGloR pce

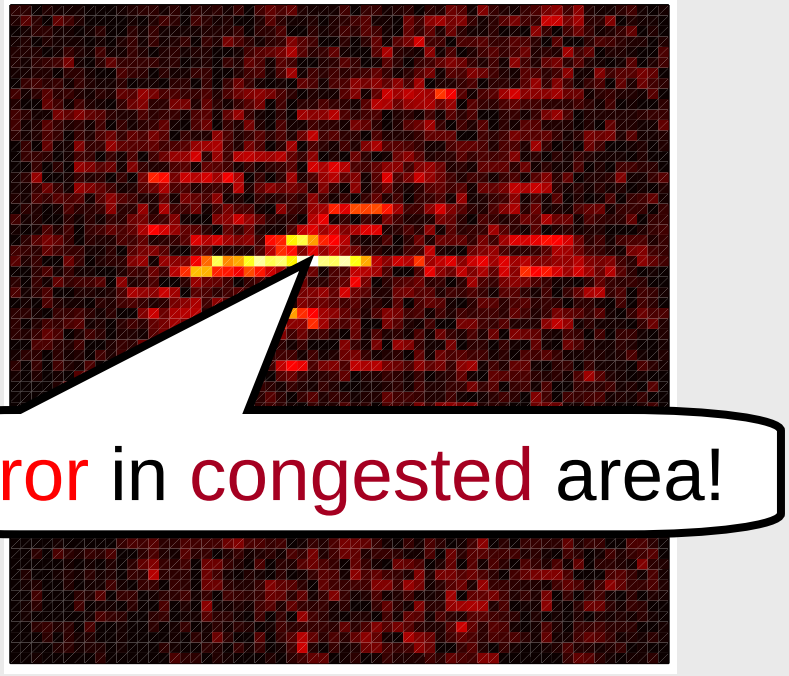
- Interested in ‘wrongly congested’.
  - c(i,j) > 1.1 $\wedge$ $C(i,j) \leq 1.1$
- Less interested in ‘wrongly uncongested’.
  - c(i,j) < 0.9 $\wedge$ $C(i,j) \geq 0.9$

❖ Estimation quality

Estimation errors



FaDGloR



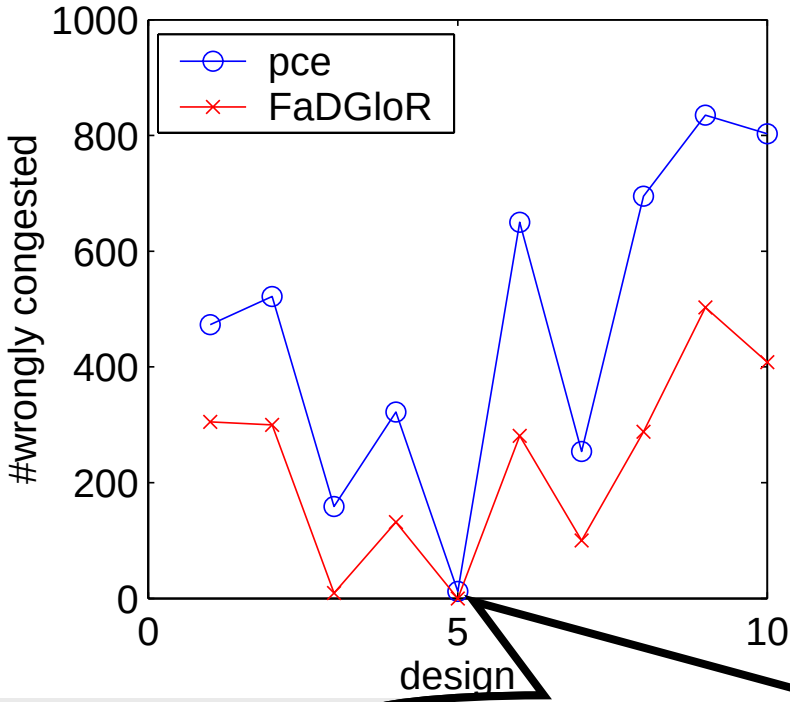
pce

new!

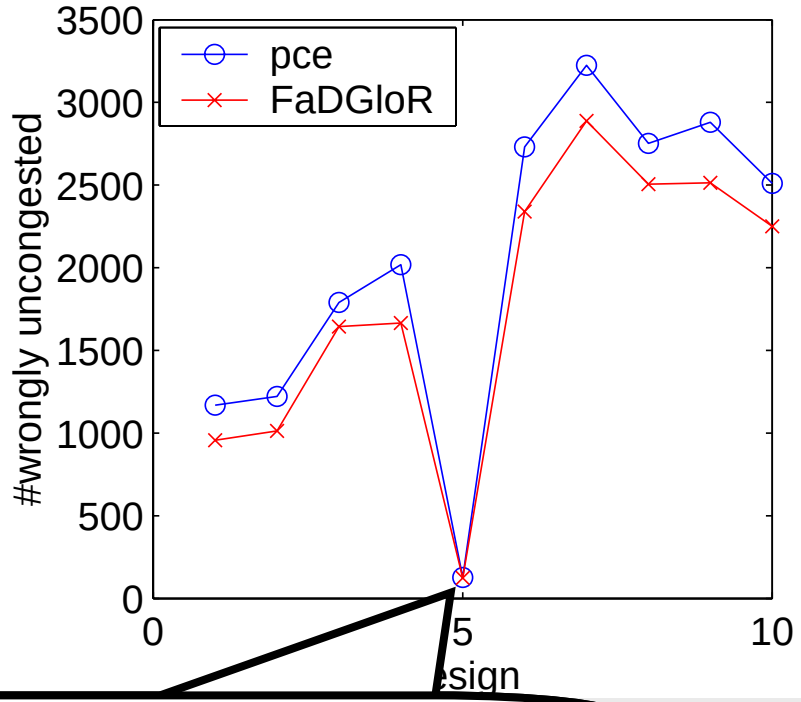
- FaDGloR: Errors more noise-like!
  - Error independent of congestion level.

❖ Results

Wrongly congested



Wrongly uncongested

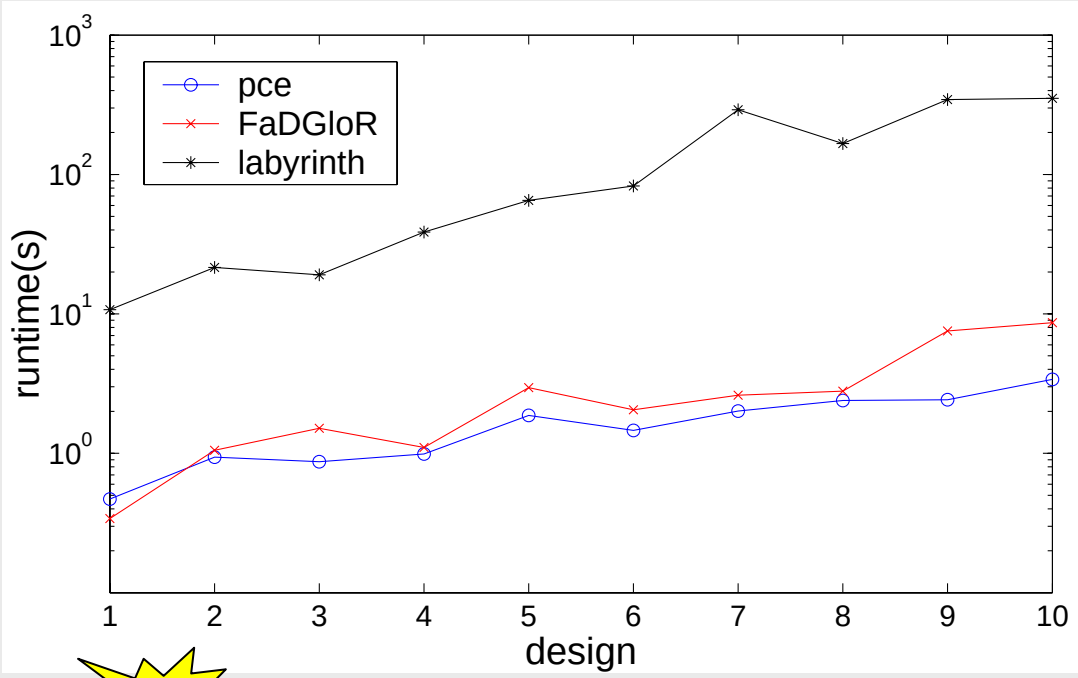


new!

- FaDGloR: Half the wrongly congested!
- Slightly less wrongly uncongested!

❖ Runtimes

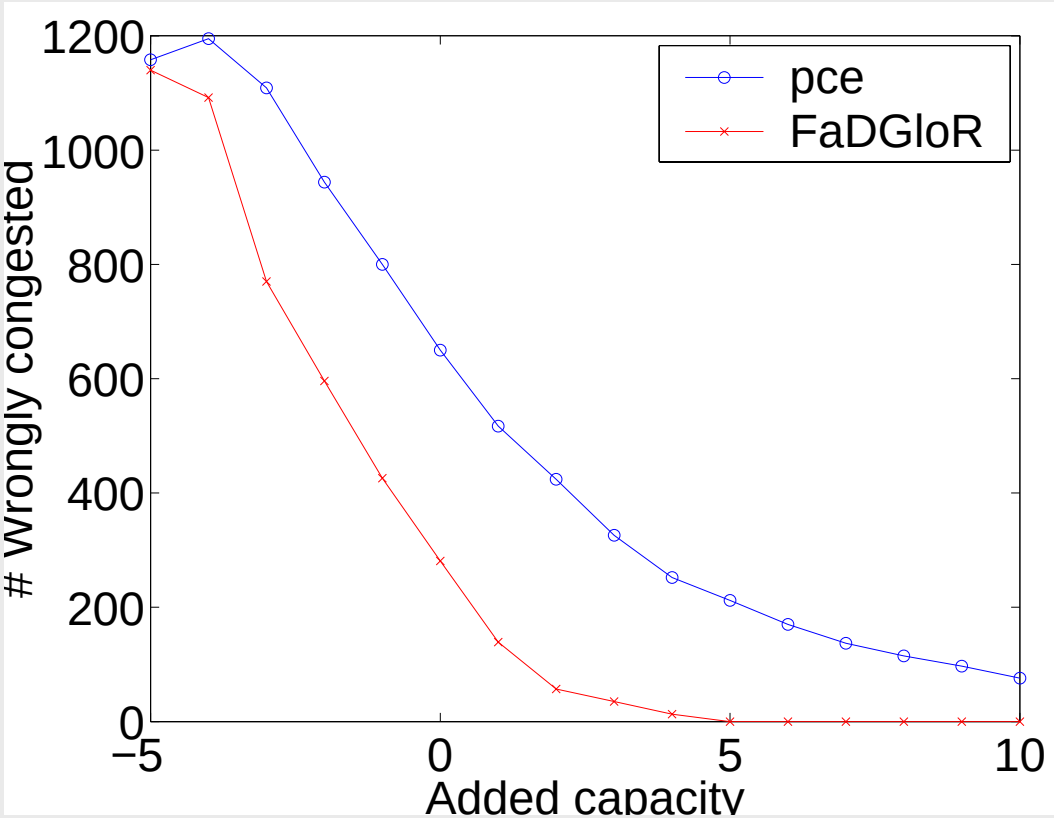
- Capacity dependence ➔ First routable
  - Still difficult designs!



new!

- FaDGloR: Only slightly slower than pce, but order of magnitude faster than Labyrinth!

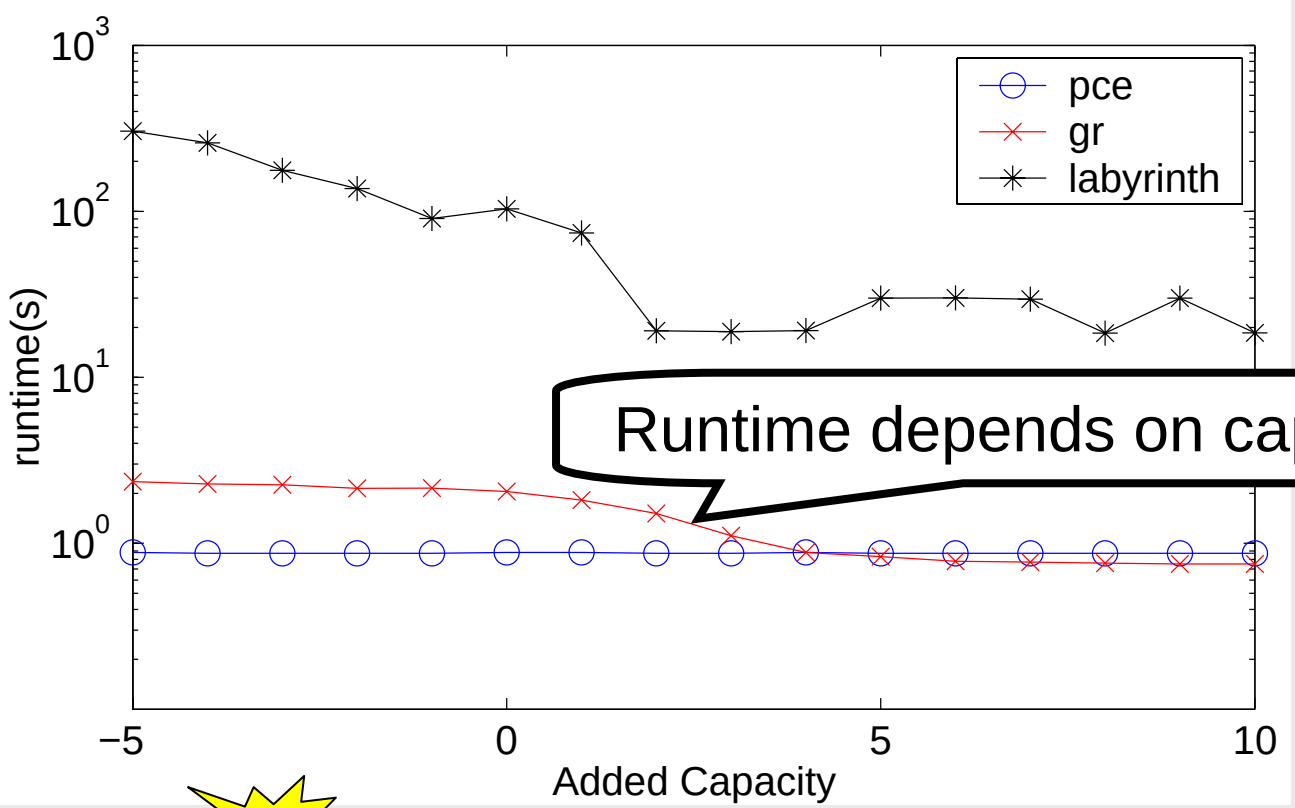
❖ Adding capacity - Quality



new!

- FaDGloR: Much less wrongly congested at all capacities!

❖ Adding capacity - Runtimes



new!

- FaDGloR: Faster with more capacity!
  - High cap: sometimes faster, sometimes slower than pce!

❖ Discussion

- FaDGloR much better on ‘wrongly congested’ new!
  - Slightly congestion-driven
  - Detouring allowed
- FaDGloR only slightly better on ‘wrongly uncongested’ new!
  - Labyrinth detours much more ➔ more congestion
  - uncongested ➔ many legal realizations

❖ Discussion

- FaDGloR only slightly slower than pce (new!)
  - pce: stamps ➔ m·n entries copied
  - FaDGloR: uncongested ➔ O(m+n) nodes visited
- Is pce a fast implementation?

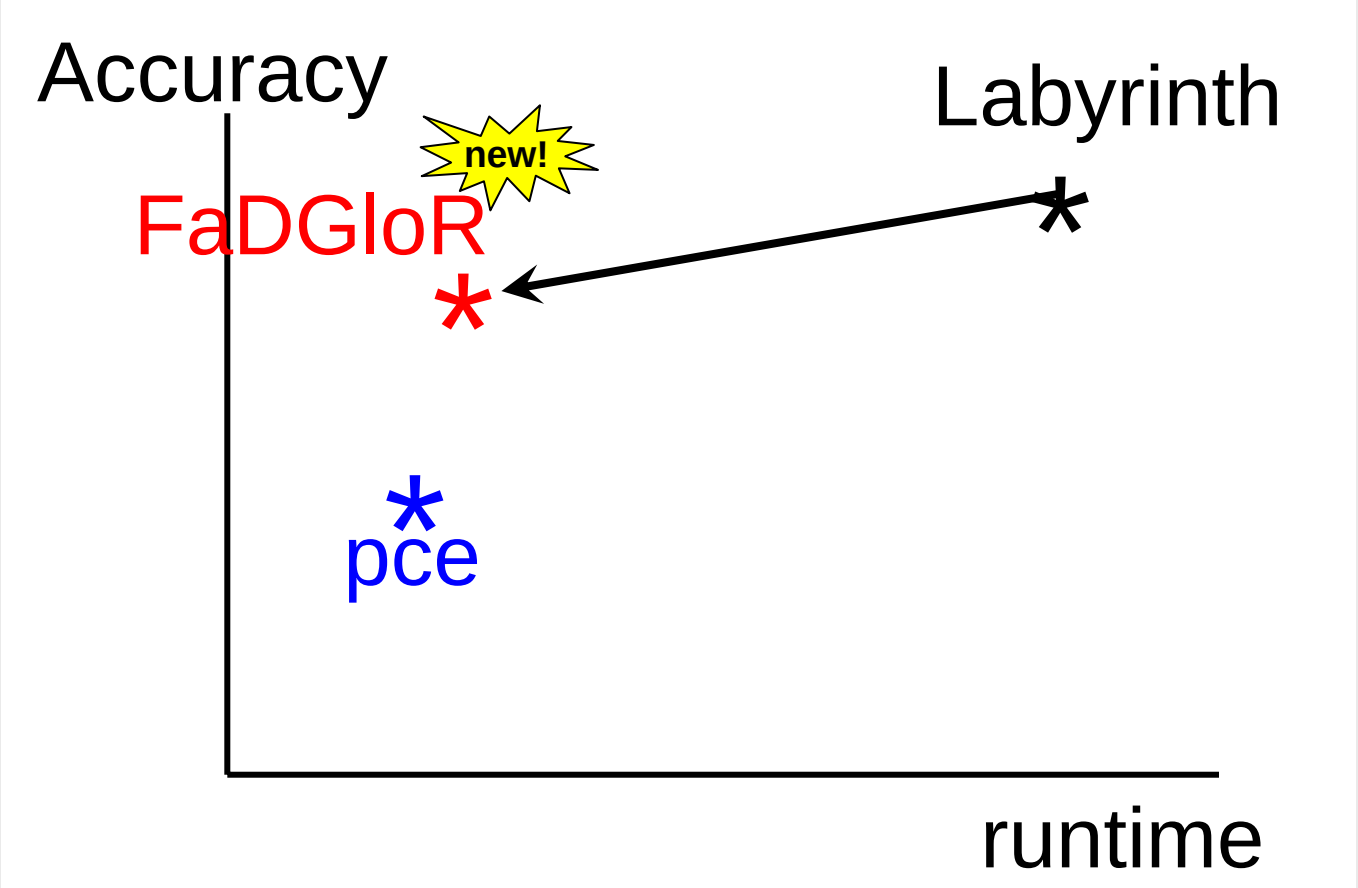
Method	Runtime per net		cpu
		1st routable	
Lou	250us		???
Kahng	330us		2.4 GHz
Pce I	50us		1.0 GHz
Pce II	14us		1.4 GHz
FaDGloR I	88us	77us	1.0 GHz
FaDGloR II	40us	35us	1.4 GHz

  - Ongoing improvements… at the price of runtime!

❖ Discussion

- FaDGloR: one shot solution (new!)
  - Highly tuned: interplay between 2-step strategy, detour-bounding and wire ordering
- FaDGloR probably better with blockages than pce (new!)
- FaDGloR allows refinement (new!)
  - R&R certain windows
  - FaDGloR results are seeds for other applications

❖ Discussion



❖ Conclusion

Is Probabilistic Congestion Estimation Worthwhile?

NO!

Global routing-based methods more flexible and promising!

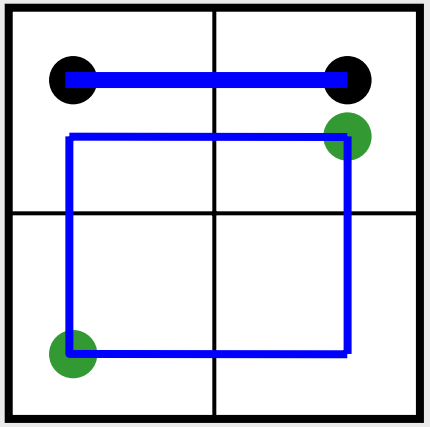
Thank you!

Questions?

❖ Problem with stamping

- Stamping: no awareness of congestion!

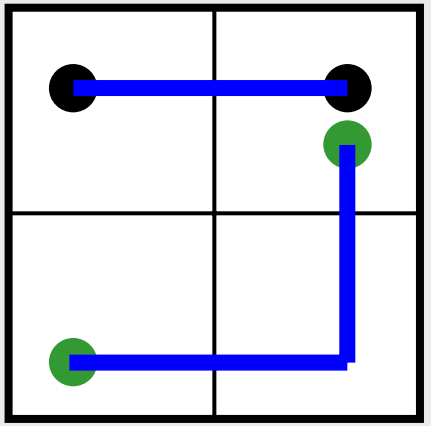
Capacity = 1



1½

½

pce



1

1

FaDGloR

- Solution: re-spreading afterwards ➔ slow!

❖ Motivation – previous work

- Lou *et al.*: "Estimating Routing Congestion using Probabilistic Analysis" ispd 2001
  - Tiles have fixed capacity
  - Spread wire over detour-free paths with equal probability
- Kahng *et al.*: "Accurate Pseudo-Constructive Wire
  - Us
  - Co
- West
  - L- and Z-shapes only
  - No detours

p 2003

rediction"

Not consistent!

Biased towards tool and/or benchmarks!