Placement development for the Coriolis toolchain

Gabriel Gouvine

July 9, 2022

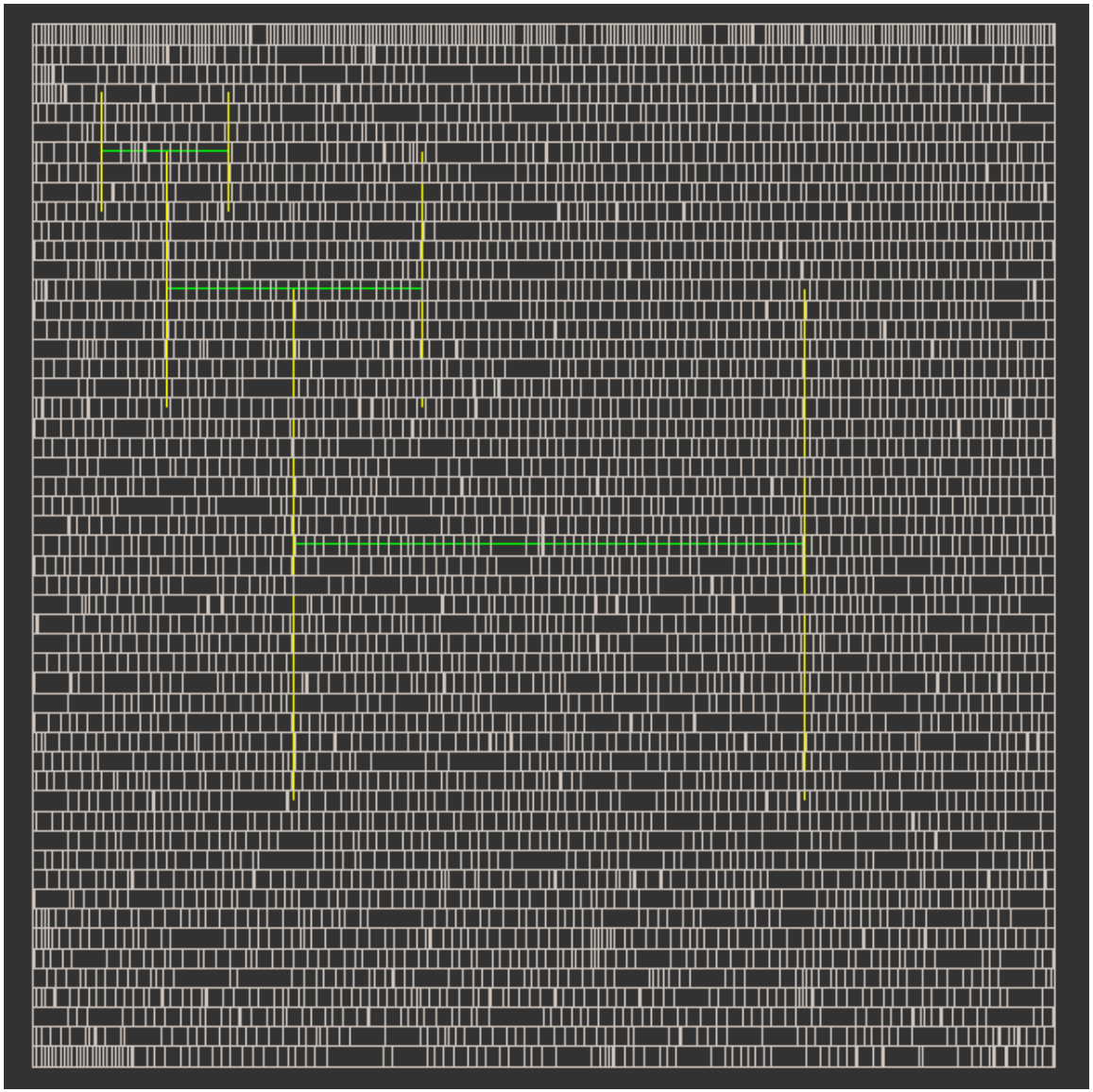
Coriolis/Alliance

Backend flow (Coriolis) + legacy complete flow (Alliance)

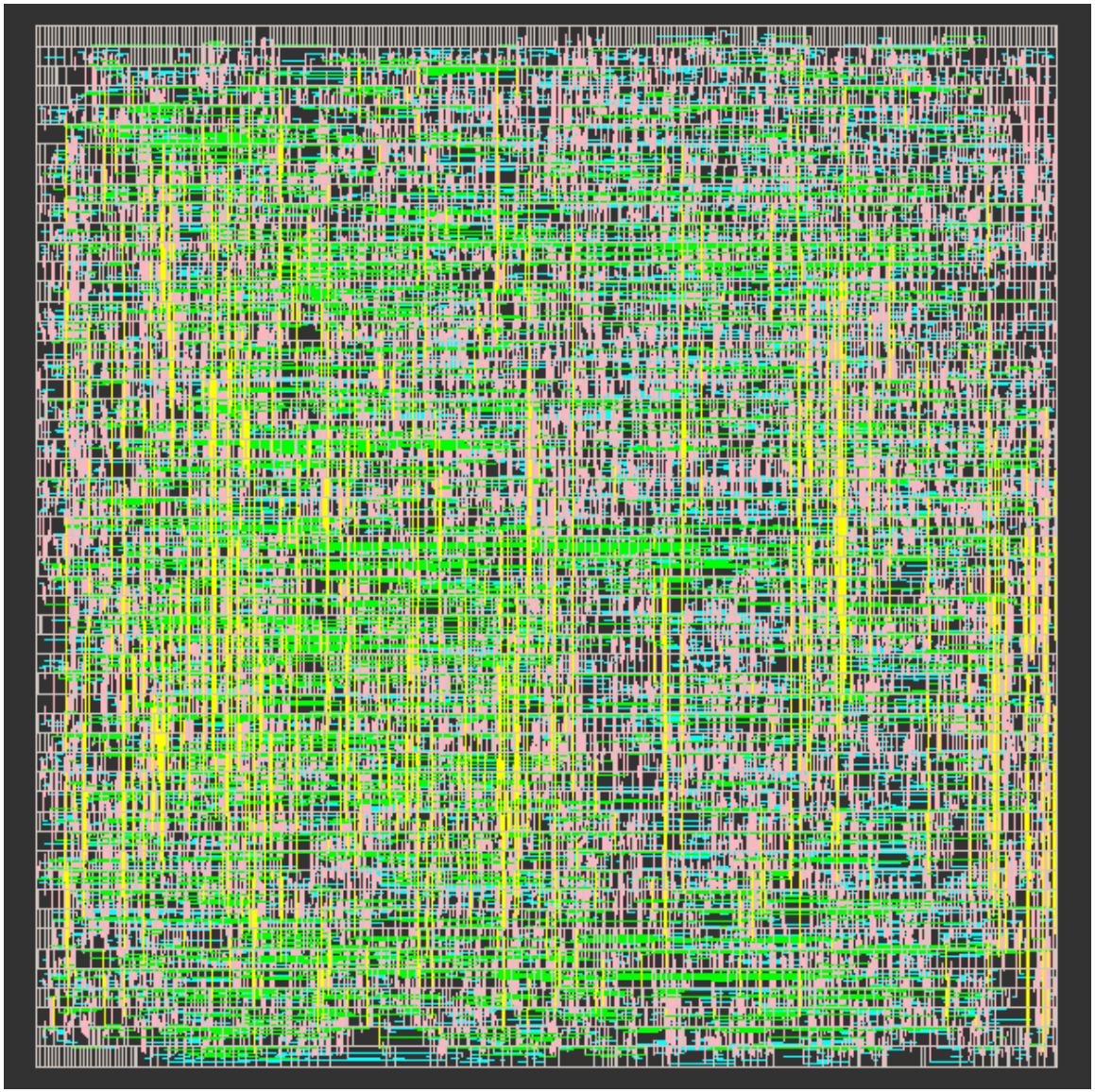
Developed at LIP6 (Sorbonne University)

Chips developed with Yosys + Coriolis or Alliance

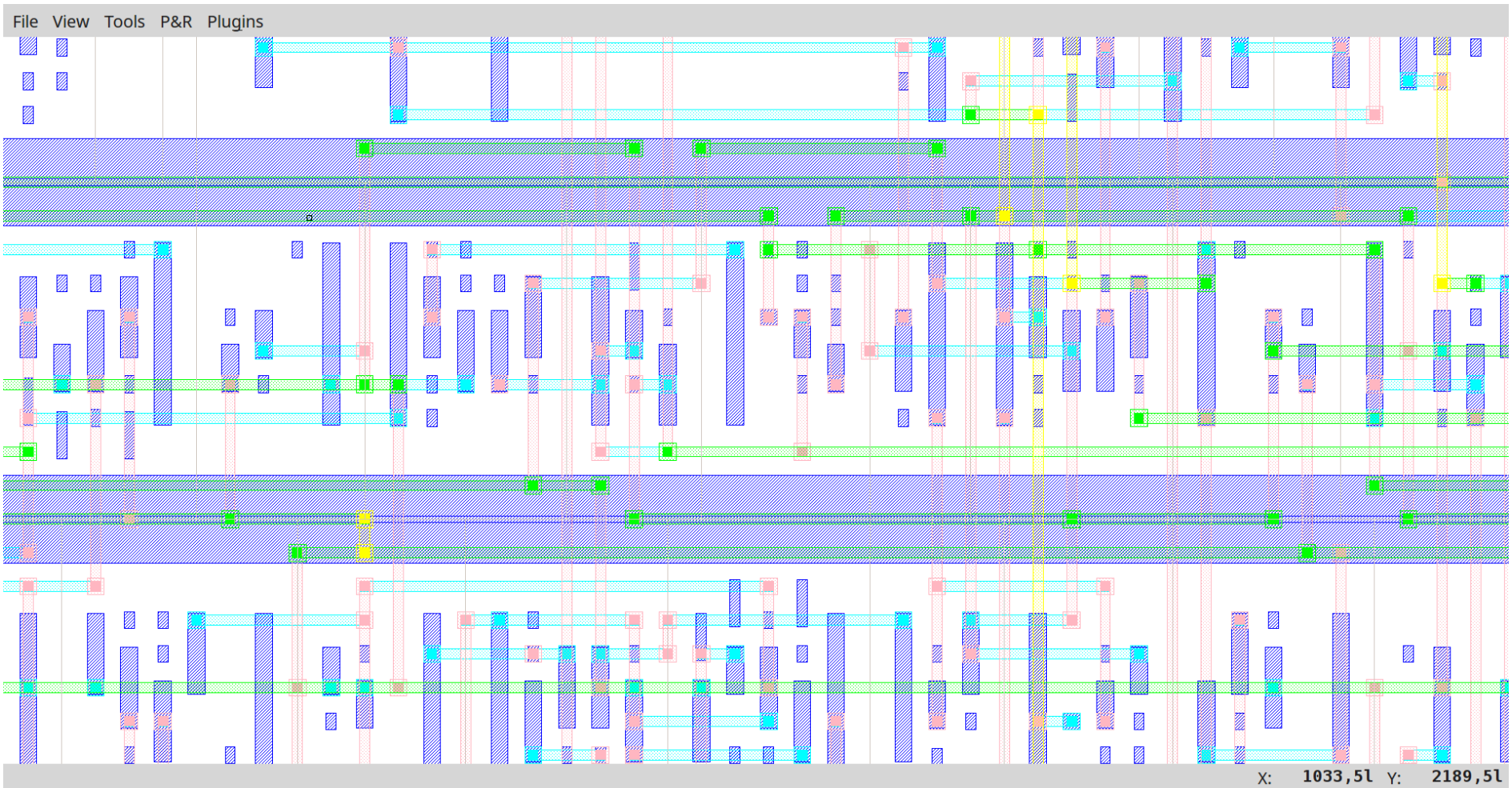
Placement tool: Coloquinte



Routing tool: Kite



Routing tool: Kite



Coloquinte at the moment

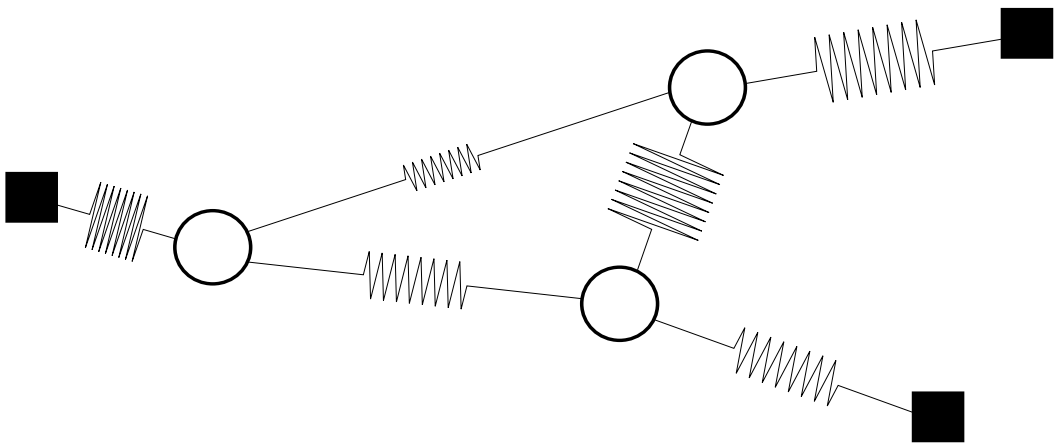
- Written in 2014
- Targets academic benchmarks
- Good quality, but slow

Three big steps

- Global placement
- Legalization
- Detailed placement

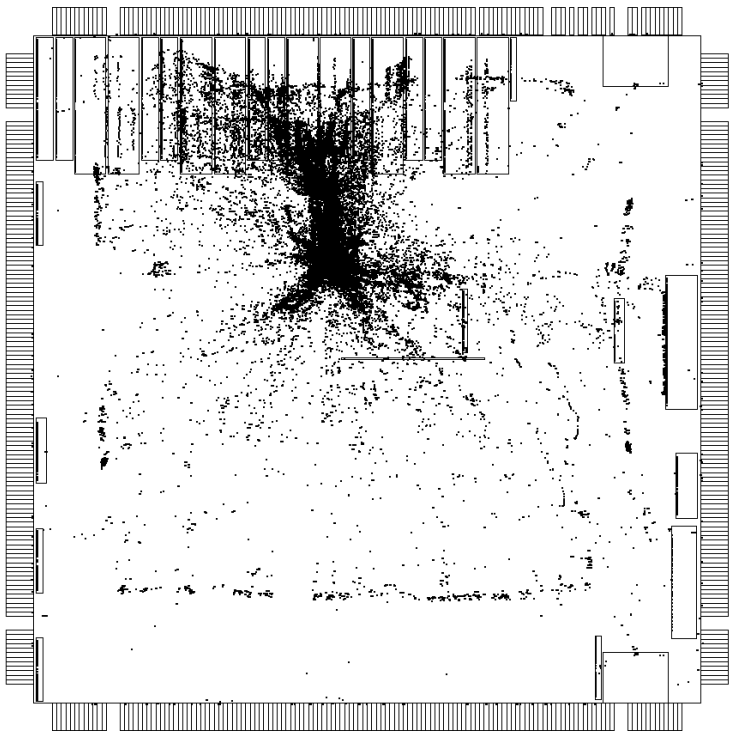
Global placement: quadratic placer

Model wires as springs

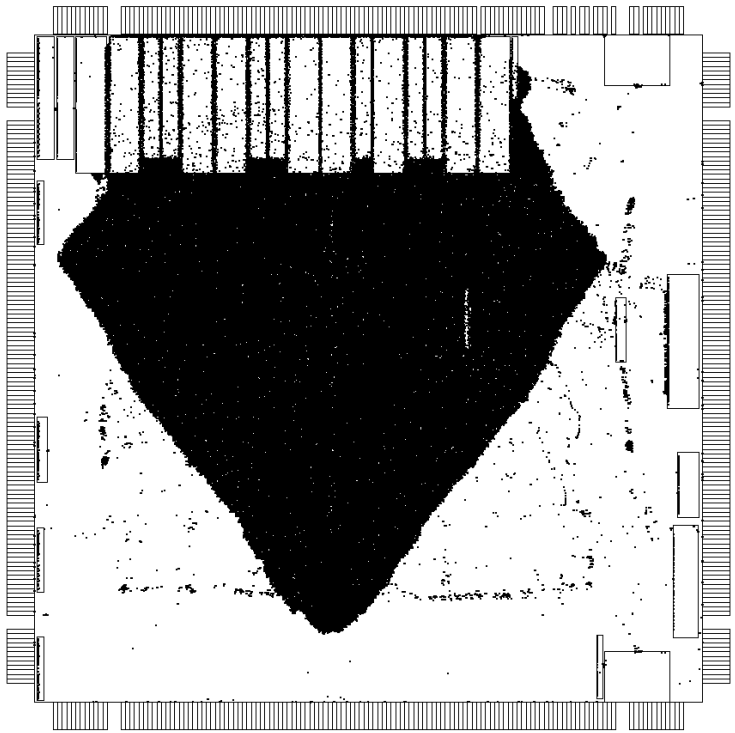


Easy to solve: sparse symmetric linear system

Global placement: density limits

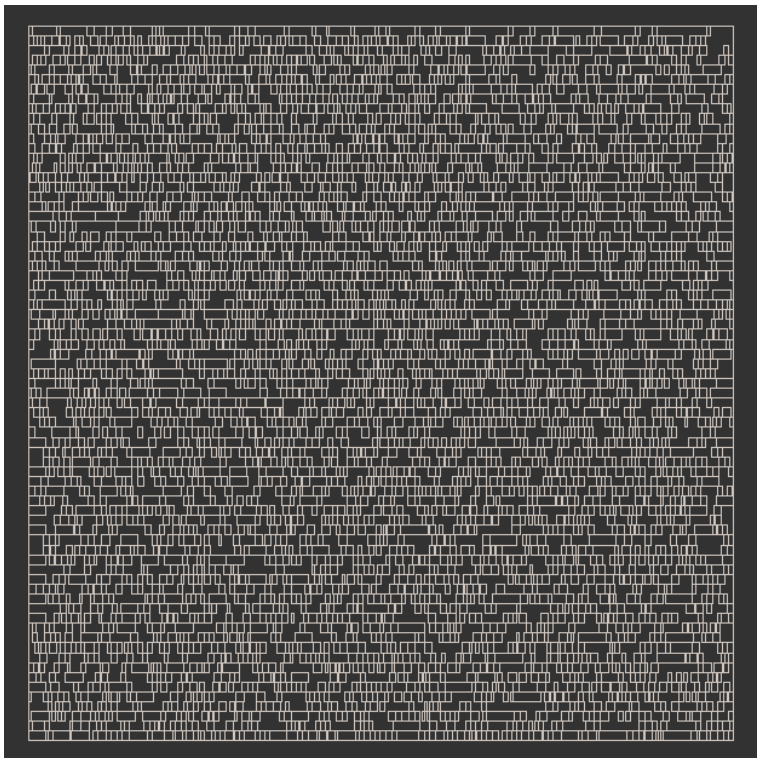


Wirelength optimization

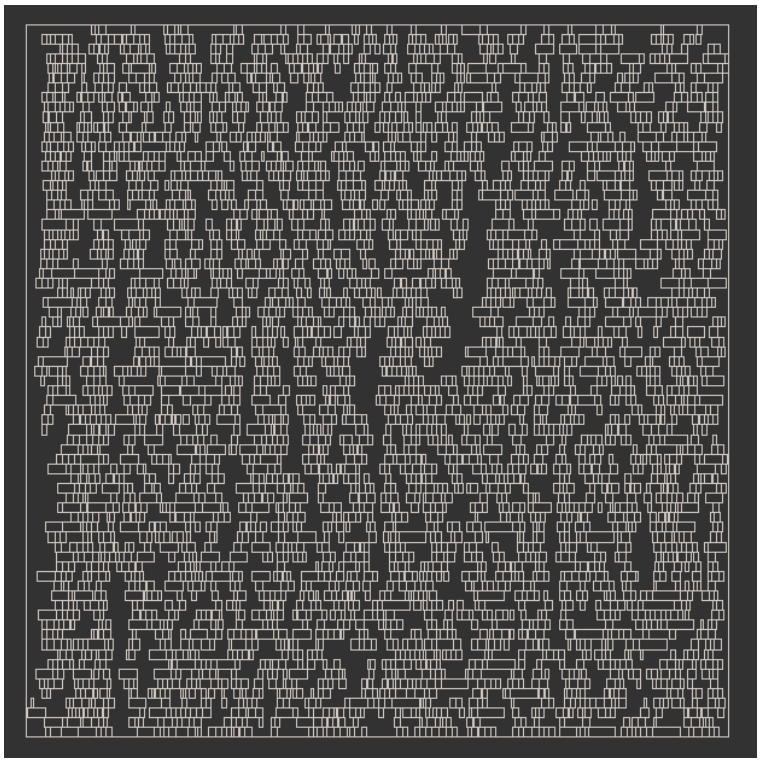


Density handling

Detailed placement: refinement



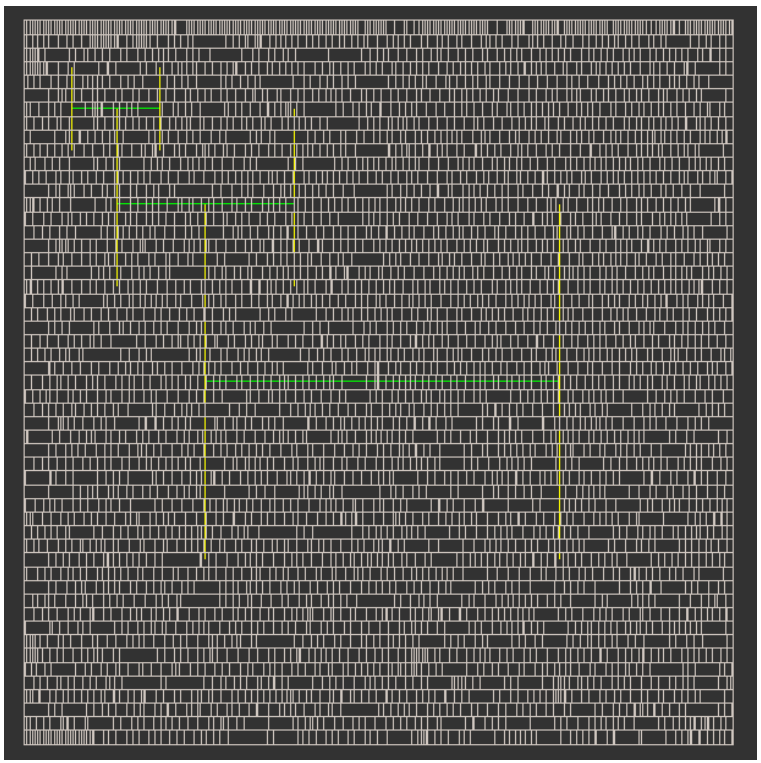
Legalized



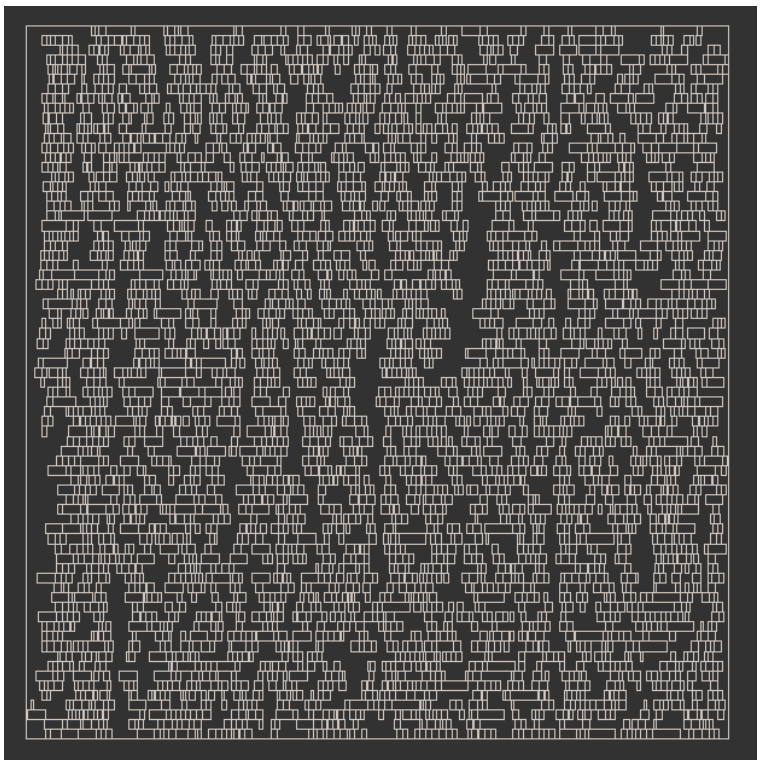
Optimized

Limitation: routing

Fixed placement density ⇒ all or nothing behaviour

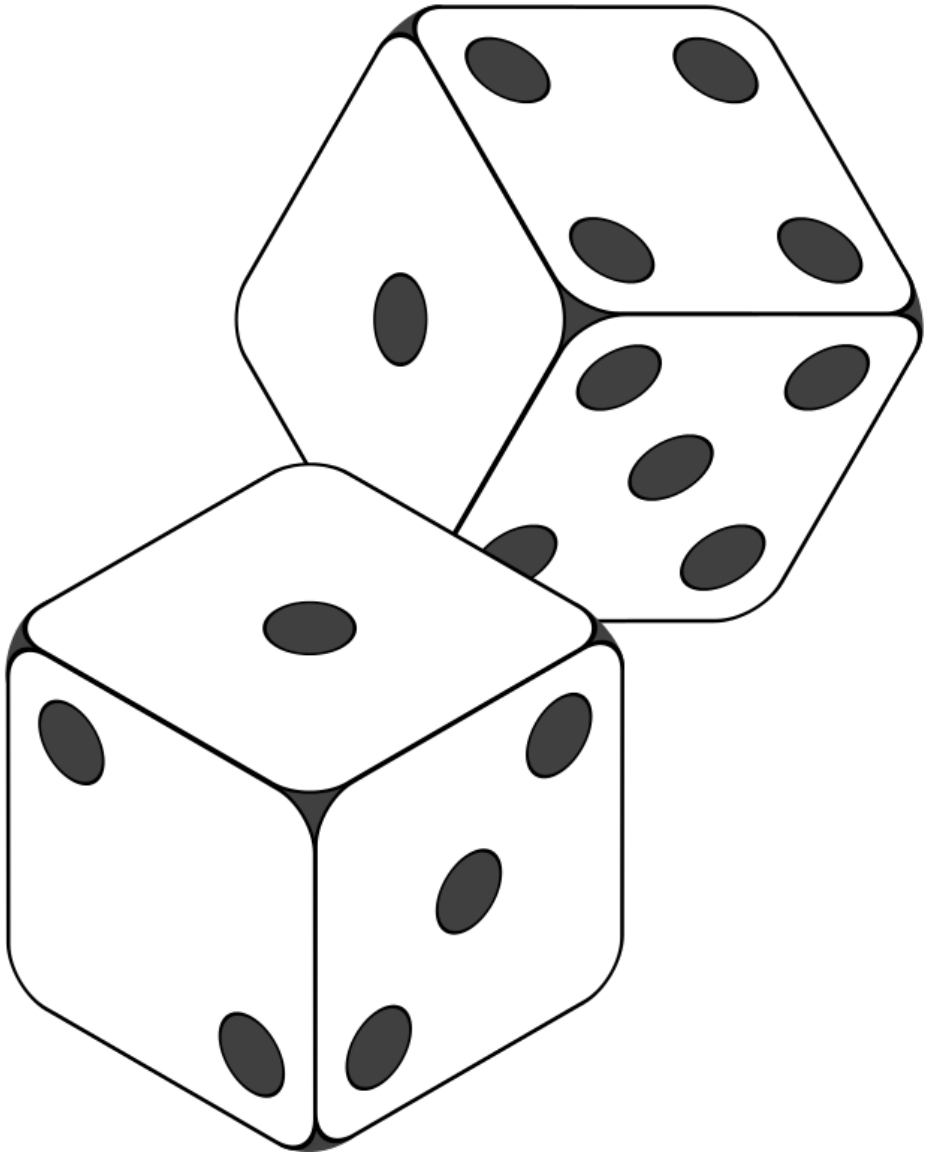


99% density



50% density

Limitation: timing



Good news!

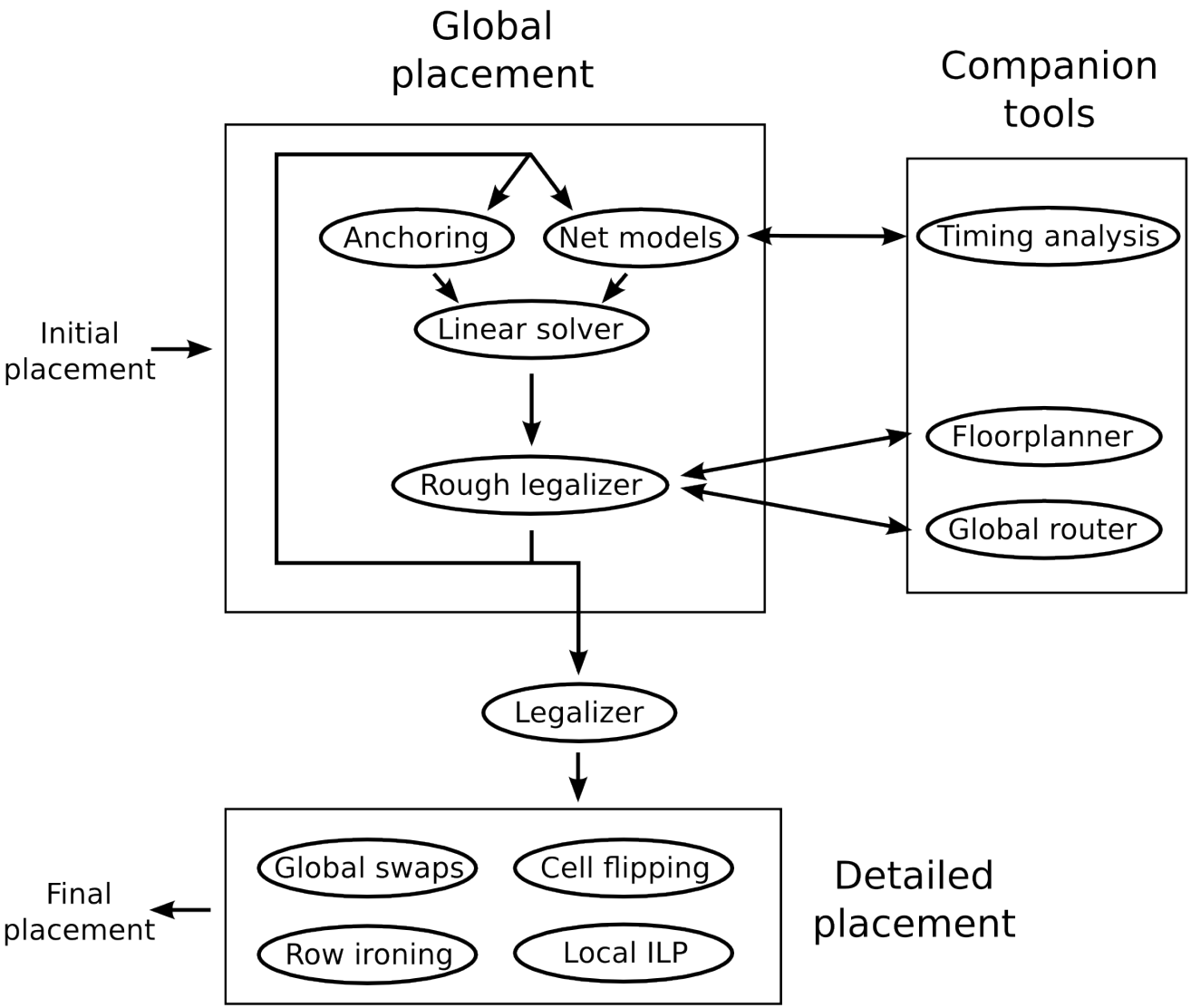
Routing $\Rightarrow$

- Local density limits to handle congestion
- Change net model to reflect chosen path

Timing $\Rightarrow$

- Penalize predicted critical path
- Route critical pins first

Flow


Global placement
Companion tools
Anchoring
Net models
Timing analysis
Linear solver
Initial placement
Floorplanner
Rough legalizer
Global router
Legalizer
Global swaps
Cell flipping
Final placement
Detailed placement
Row ironing
Local ILP

In practice

Timing analysis

- Donated sources from Avertec: Hitas/Yagle
- Reimplementation in Coriolis
- Provide cell load + Elmore delay

Placement

- Reimplement Coloquinte
- More modular (needs routing/timing callbacks)
- Improve benchmarks

Questions