

EDA (CS286.5b)

Day 8
Placement
(Constructive)

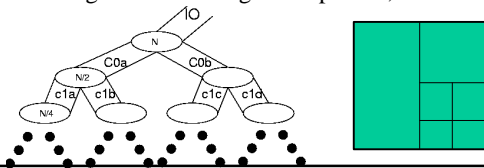
N.B. assignment #1 due Friday

Today

- Constructive Placement
- Divide and Conquer
- Estimation
- Refinement

Basic Idea

- Partition (bisect) to define halves of chip
– minimize wire crossing
- Recurse to refine
- When get down to single component, done



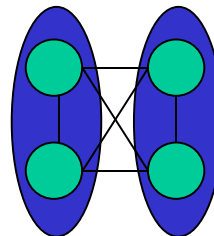
Adequate?

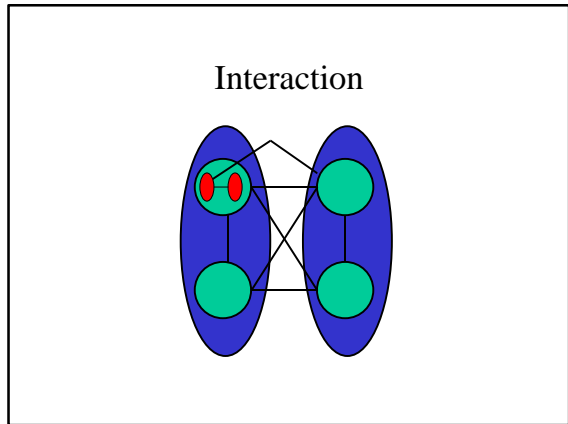
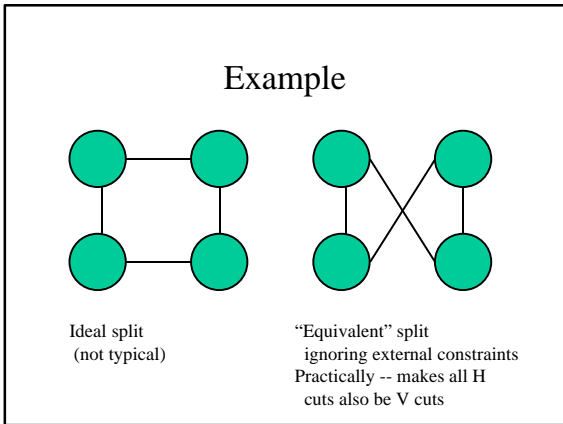
- Does recursive bisection capture the primary constraints of two-dimensional placement?

Problems

- Greedy, top-down cuts
– maybe better pay cost early?
- **Two-dimensional problem**
– (often) no real cost difference between H and V cuts
- Interaction between subtrees
– not modeled by recursive bisection

Interaction

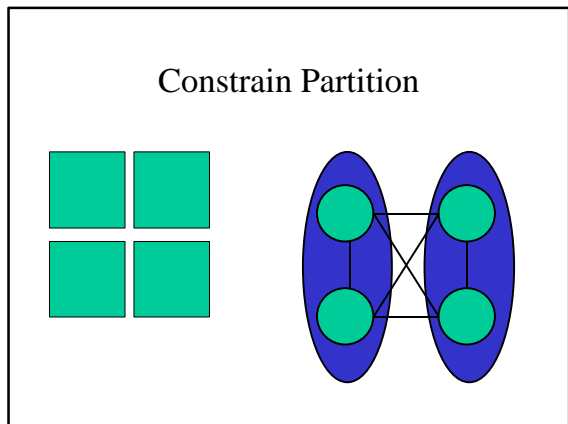




- ### Problem
- Need to keep track of where things are
 - outside of current partition
 - include costs induced by above
 - Don't necessarily know where things are
 - still solving problem

- ### Improvement: Ordered
- Order operations
 - Keep track of existing solution
 - Use to constrain or pass costs to next subproblem
 - Flow cut
 - use existing in src/sink
-

- ### Improvement: Constrain
- Partition once
 - Constrain movement within existing partitions
 - Account for both H and V crossings
 - Partition next
 - (simultaneously work parallel problems)
 - easy modification to FM

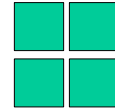


Improvement: Quadrisect

- Solve more of problem at once
- Quadrisect:
 - partition into 4 bins simultaneously
 - keep track of costs all around

Quadrisect

- Modify FM to work on multiple buckets
- k-way has:
 - $k(k-1)$ buckets
 - from*to
 - quad $\rightarrow 12$
- reformulate gains
- update still $O(1)$

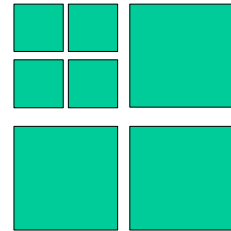


Quadrisect

- Cases (15):
 - (1 partition) x 4
 - (2 part) x 6 = (4 choose 2)
 - (3 part) x 4 = (4 choose 3)
 - (4 part) x 1

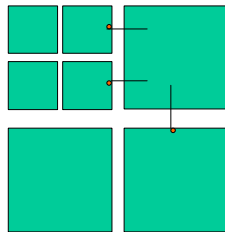
Recurse

- Keep outside constraints
 - (cost effects)
- Don't know detail place
- Model as at center of unrefined region



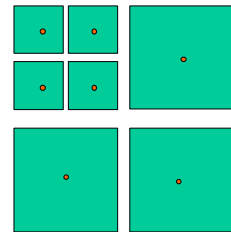
Option: Terminal Propagation

- Abstract inputs as terminals
- Partition based upon
- Represent cost effects on placement/refinement decisions



Option: Refine

- Keep refined placement
- Use in cost estimates

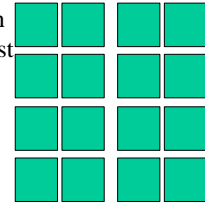


Problem

- Still have ordering problem
- Earlier subproblems solved with weak constraints from later
 - (cruder placement estimates)
- Solved previous case by flattening
 - ...but in extreme give up divide and conquer

Iterate

- After solve later problems
- Relax solution
- Solve earlier problems again with refined placements (cost estimates)
- Repeat until converge

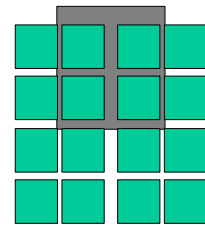


Iteration/Cycling

- General technique to deal with phase-ordering problem
 - what order do we perform transformations, make decisions?
 - How get accurate information to everyone
- Still basically greedy

Refinement

- Relax using overlapping windows
- Deal with edging effects
- Khang etc. claim 10-15% improve
 - cycle
 - overlap



Possible Refinement

- Allow unbalanced cuts
 - most things still work
 - just distort refinement grip
 - allowing unbalance using FM quadrisection looks a bit tricky

Runtime

- Each gain update still $O(1)$
 - (bigger constants)
 - so, FM partition pass still $O(N)$
- $O(1)$ iterations expected
- assume $O(1)$ overlaps exploited
- $O(\log(N))$ levels
- Total: $O(N \log(N))$
 - very fast compared to typical annealing

Uses

- Good by self
- Starting point for simulated annealing
 - speed convergence
- With synthesis (both high level and logic)
 - get a quick estimate of physical effects
 - (play role in estimation/refinement at larger level)
- Early/fast placement
 - before willing to spend time looking for best
- For fast placement where time matters
 - FPGAs, online placement?

Summary

- Partition to minimize cut size
- Additional constraints to do well
- Quadrisection
- Keep track of estimated placement
- Relax/iterate/Refine

Today's Big Ideas:

- Divide-and-conquer
- Successive Refinement
- Phase ordering: estimate/relax/iterate