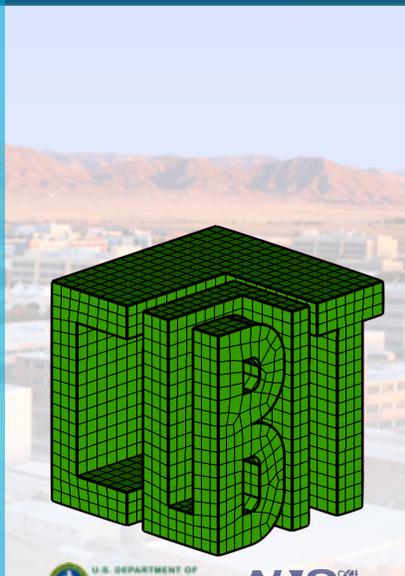
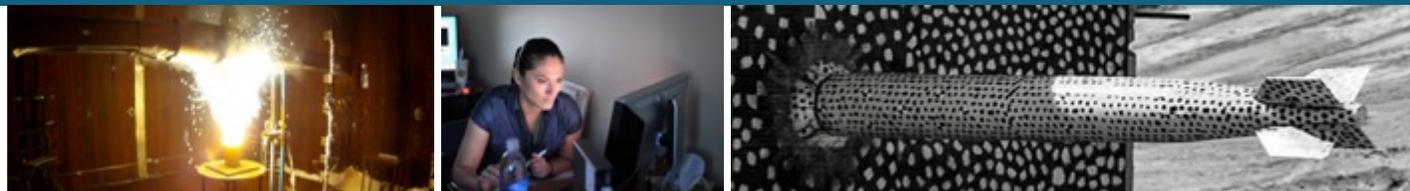




Sandia National Laboratories

# Incremental Interval Assignment for Mesh Scaling



*Presented by*

Scott A. Mitchell  
CCR Center for Computing Research  
Sandia National Laboratories



Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

# Problem: What is Interval Assignment?



Quad and Hex meshing (not simplices)

*Intervals* = number of mesh edges

- Curve, periodic surface crack...

*Assignment* = choose intervals

- Meshing primitive constraints met
- User's desires met (optimization)

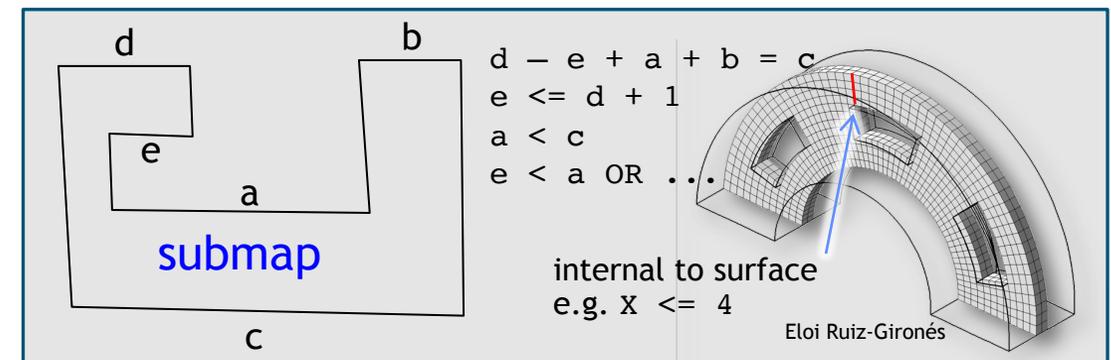
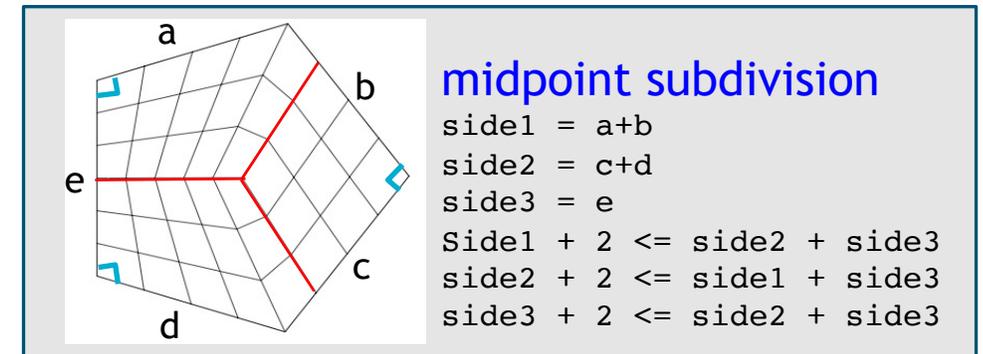
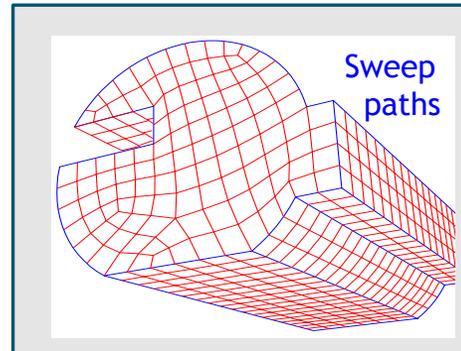
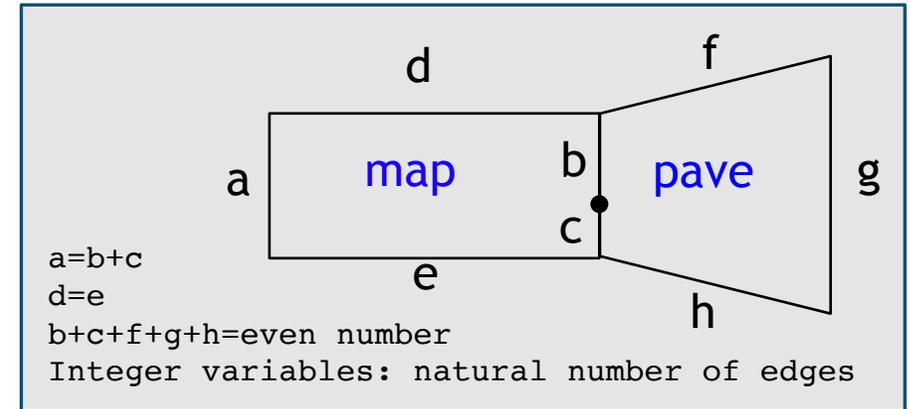
Constraints

$$Ax = b$$

$$1 \leq x \text{ integer}$$

Complications

- $A$  coefficients not always 1  
sum-even variables,  $a_{ij} = 2$
- $b$  not always 0  
midpoint subdivision  
pre-meshed neighbors  
hard-set intervals
- Submapping introduces "or"  
constraints, breaks  $Ax = b$



# Prior Approches to Interval Assignment



## Heuristics

- Start at user goals, add interval somewhere until constraints met

## Linear Programming simplex method for floating point solution

- LIA Objective weighted sum [Tam&Armstrong 1993]
  - Can get integer “for free”
  - Quality may not be great
- MinMax deviation** [Mitchell 1997]
  - Linear Program MinMax  $x(\frac{1}{goal})$** 
    - Good quality, but need expensive Branch & Bound for integer solution, BBIA
  - Non-linear  $\text{Min} \sum (x - goal)^3$  [Mitchell 2013]

## Network flows

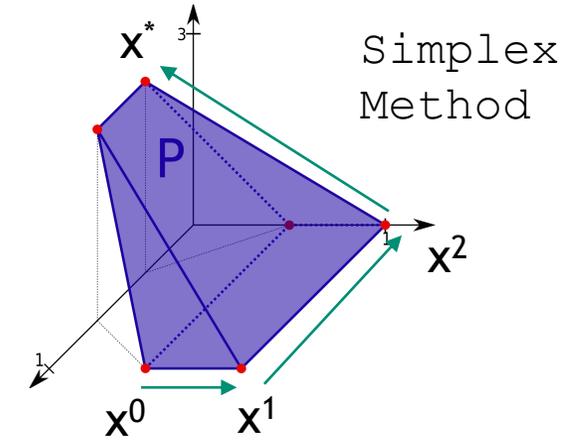
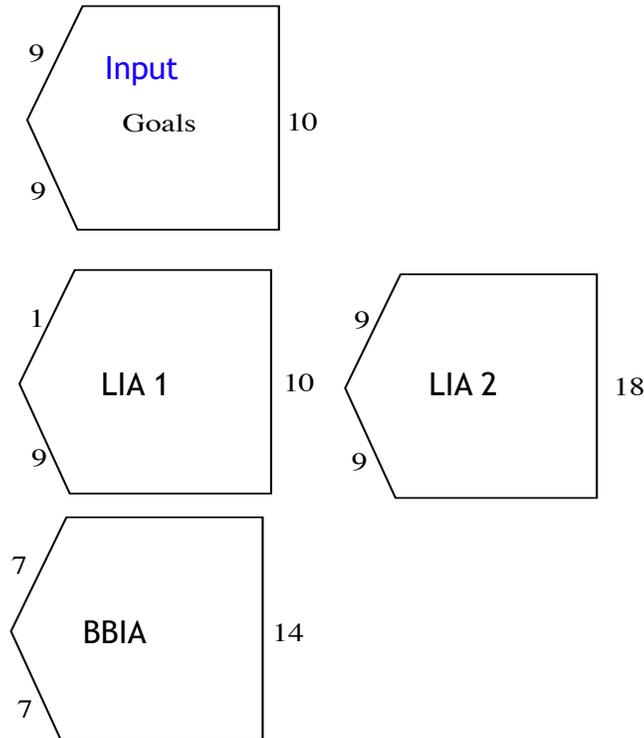
+solves selecting templates

[Möhring, Müller-Hannemann & Wiehe 1997]

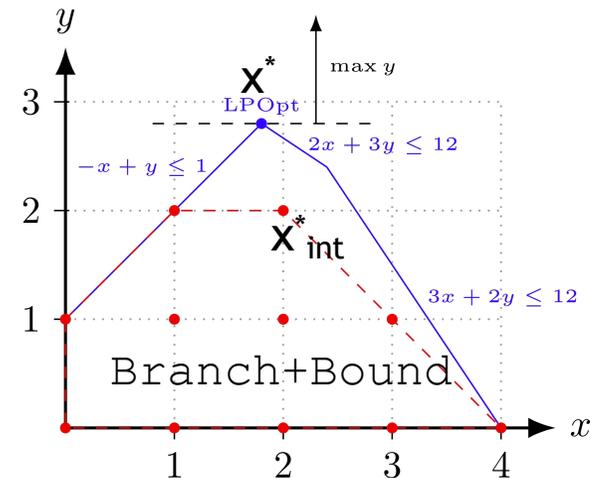
## Frame fields

+solves picking corners of templates

- Quadratic programming for surface meshing [Bommes 2010]



Images courtesy wikipedia

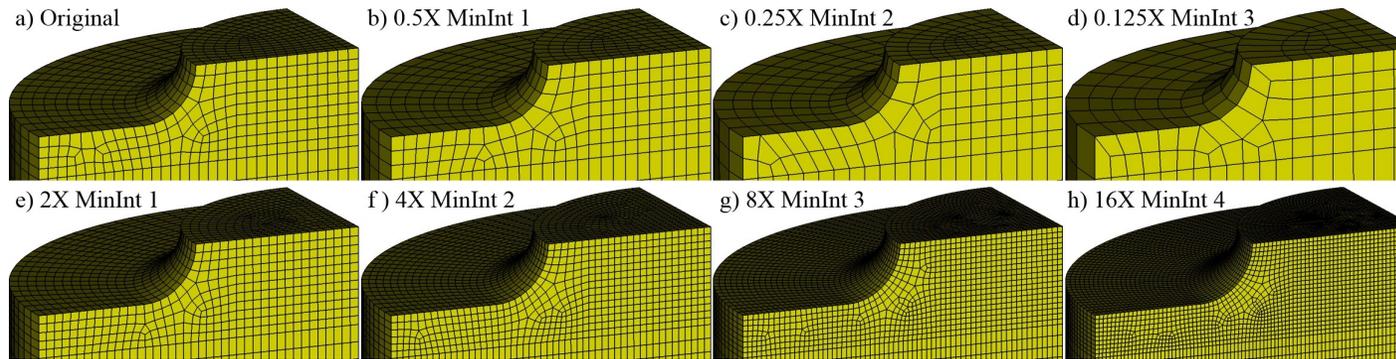


# Problem: What is Mesh Scaling?



Verification studies, is solution converged?

- Initial mesh, refine it, solve again. If solution unchange it might be converged.

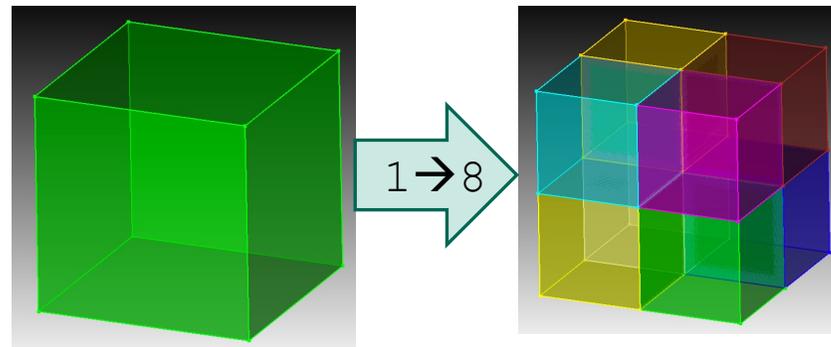


How many elements can you afford?

- 1 hex to 8 = 8, 64, 512, 4096, 32768...
- Does 4000x fit on machine?

Need

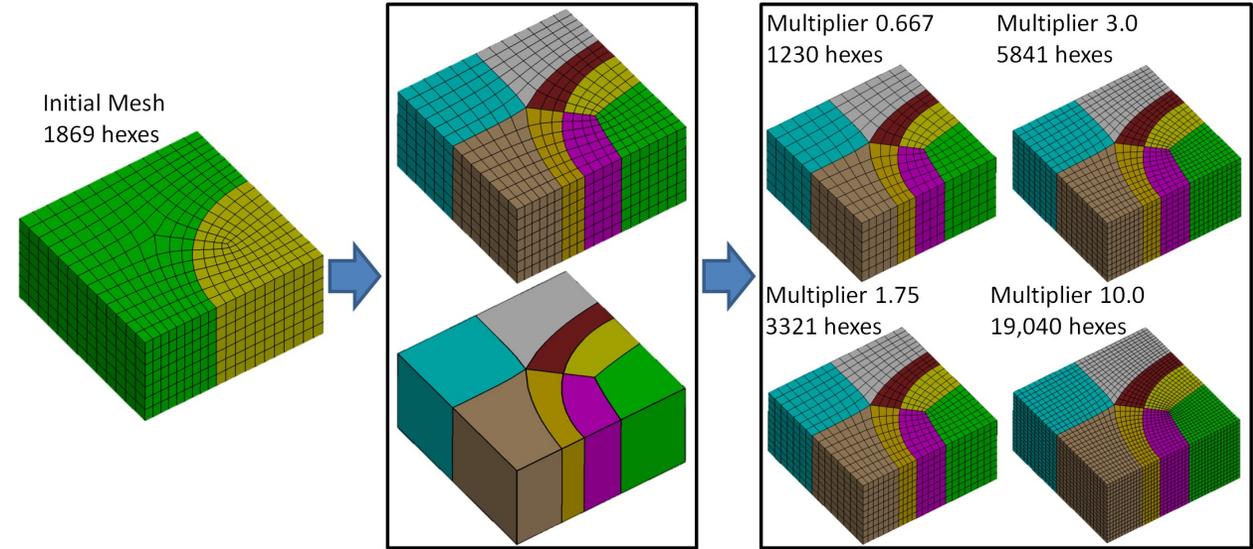
- Finer mesh, but not  $8 \rightarrow 1$ .
- 1.5x elements = 1.5, 2.3, 3.4, 5.1
  - 5x probably fits
- Some refinement everywhere



# 5 Mesh Scaling Algorithm

## Mesh Scaling (one variant)

- Decompose mesh into structured blocks
- Delete mesh
- For each refinement level
  - Solve IA (Interval Assignment) to choose more intervals
  - Remesh blocks



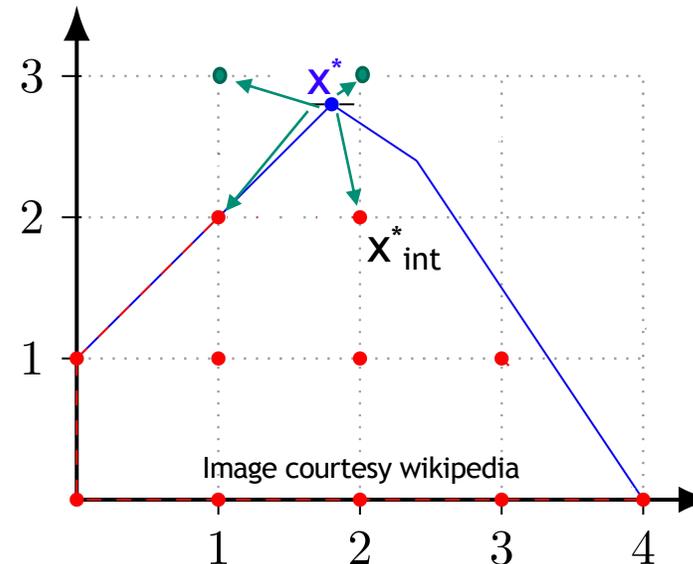
“Mesh scaling for affordable solution verification” IMR2016  
Matt Staten, Brian Carnes, Corey McBride, Clint Stimpson, Jim Cox.

## Problem

- BBIA (Branch&Bound IA) on 1000 blocks
- Run overnight, fail, no solution
- Branch&Bound is exponential search in worst-case
  - Round 1.8 to 2? If that doesn't work, round 1.8 to 1
  - Do for every variable!  $2^n$  choices

## But it's just blocks!

- Freedom is +/- number of edges on each block side
- Changes propagate simply



# IIA Incremental Interval Assignment

## User Input

- **Multiplier** : target #elements
- **Min Increment** for every curve/group (can be 0)

## IIA algorithm

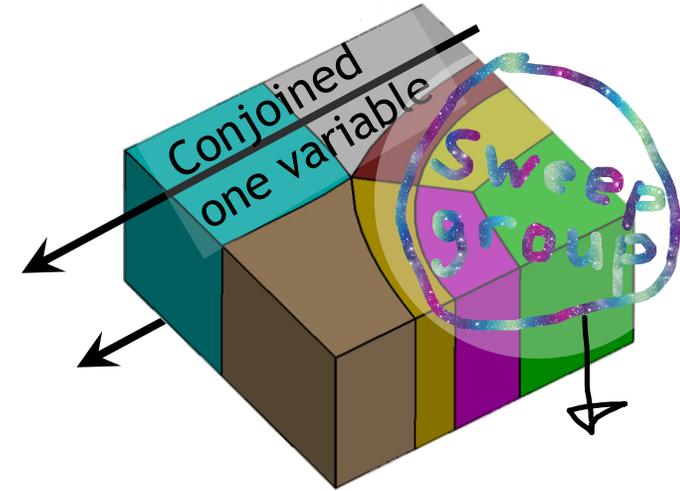
### Given initial intervals

- Increment
  - by one interval on one set of block-curves
- Until target #elements reached

### Block-curve priority-queue function

- Initial and current intervals
- Increase in number of elements
- Spread out the change
  - Group nearby small blocks

Multiple passes with different priority functions



Variant: Re-pave and sweep  
sweep curves in groups  
Other curves in no group  
Little bit of change by group

**Priority functions** for next curve (group) to increment. 8 different queues.

If min increment, then do once

- **Min Simple** : increment every curve not in a group
- **Min Groups** : pick group, pick curve in group
- **Decrement Redundant** : decrement curve if group still has min increment (curves in more than one group)

Else repeat until no progress

- **At Least One** : increment curve so each group has at least one increment
  - But stay below  $\text{target\#elements}$
- **Decrement Redundant** : decrement curve if another curve of same group is incremented

Final steps

- **Reach Goal** : pick curve with minimum ratio  $(\text{new} : \text{old}) < \text{multiplication factor}$ .
  - But stay below  $\text{target\#elements}$
- **Reach Count** : pick curve that adds the most elements
  - But if  $\text{target\#elements}$  exceeded, pick curve exceeding it least
- **Reduce Count** : decrement curve if still  $\text{\#elements} > \text{target}$ , and still min increment



## IIA, new

- Priority queue
- Start integer, stay integer
- Scales with **#intervals**

## BBIA, old

- Simplex method, then Branch&Bound
- Non-integer solution, LP floating point
- Exponential in **#variables**, B&B search

## IIA Features

- Output quality better than BBIA
  - #elements closer to the desired multiplication factor
  - Changes better spread out
- Runtime is a non-issue.
  - Solves all mesh scaling problems in < 1 second
  - BBIA failed on large models after running for days.
  - Efficiency depends on monotonicity of priority function as other block-curves change.

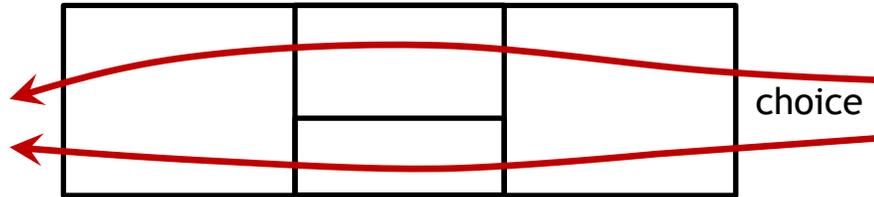
# Future: Can we Increment for the general interval problem?



Optimize  $x : Ax = b$

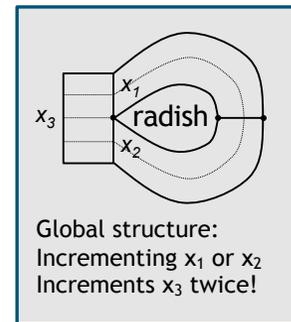
Challenges vs. mesh scaling

- No initial solution
- More shapes, not just blocks
  - More freedom on how an increment propagates
    - How to choose? Quickly?
- Different priority functions



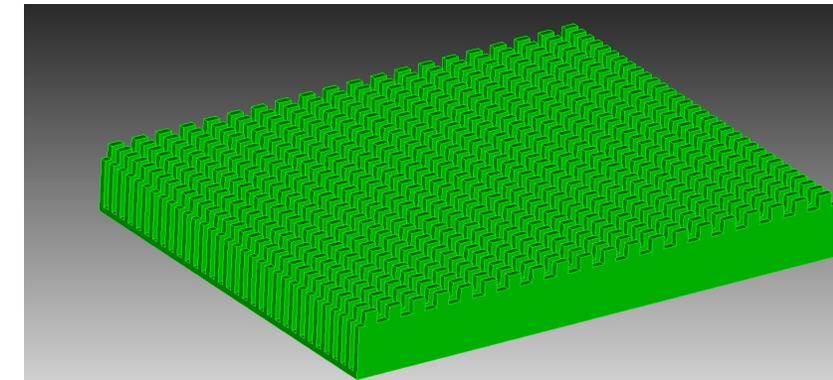
$Ax = b$  complications

- $A$  coefficients not always 1. Sum-even variables,  $a_{ij} = 2$ 
  - Guarantees of “totally unimodular” matrices does not apply
- $b \neq 0$
- Submapping introduces “or” constraints, breaks  $Ax = b$

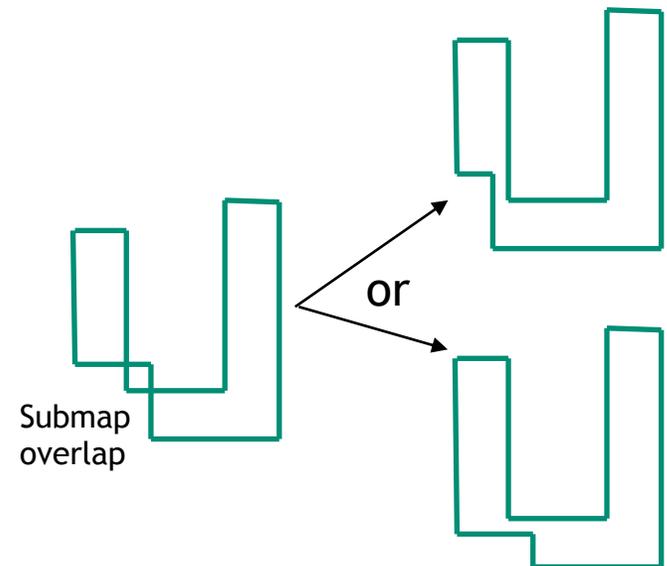


Status

- Solves entire Cubit COMMIT and NIGHTLY test suite (200+ tests)
  - But probably fails on some assemblies
- Fast, but still too slow on heat sink



Heat-sink mockup, symmetries →  
BBIA takes half an hour!  
Goal : IIA < 1 second





Thanks

