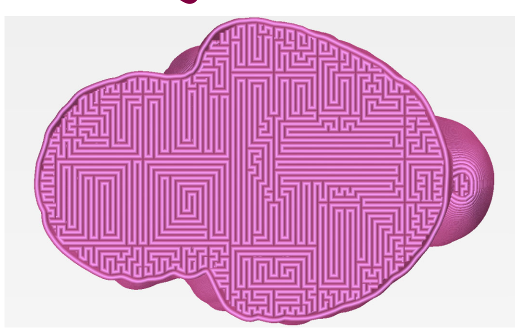
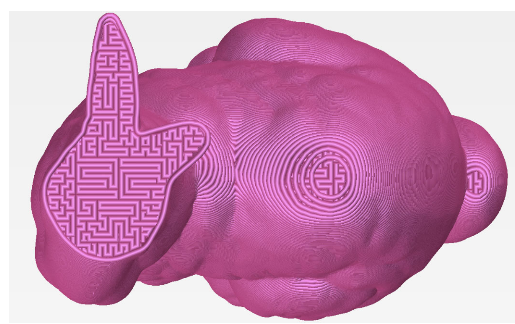
SFCDecomp: Multicriteria Optimized Toolpath Planning in 3D Printing

Prashant Gupta, Yiran Guo, Narasimha Boddeti,
Bala Krishnamoorthy
Washington State University

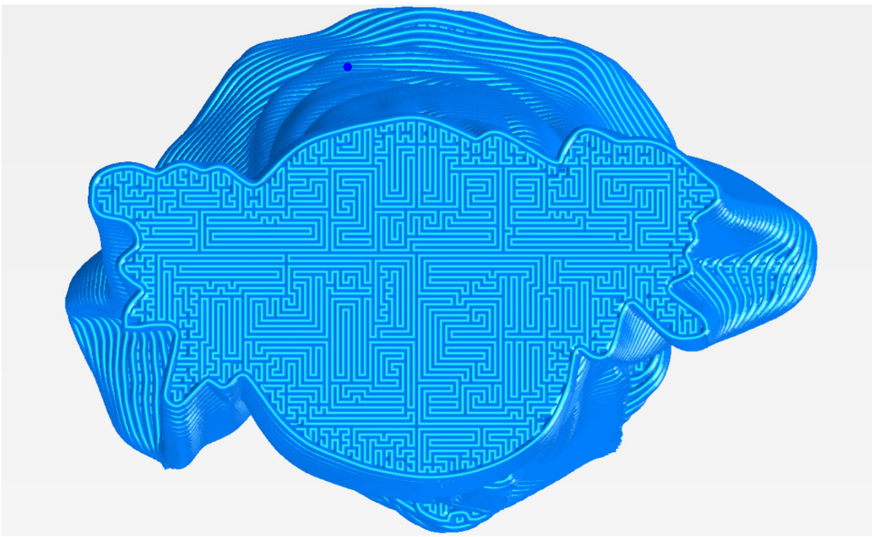


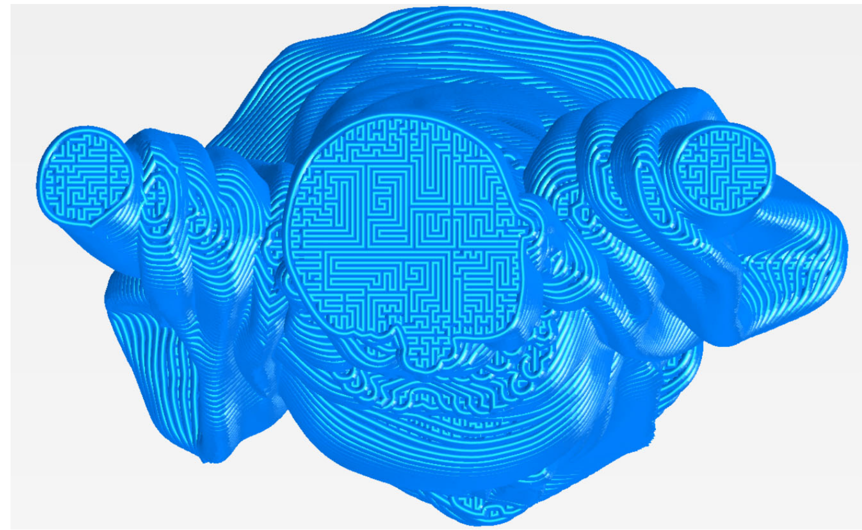


IJCGA
arXiv: 2109.01769

3D PRINTING: TOOL PATH OPTIM.

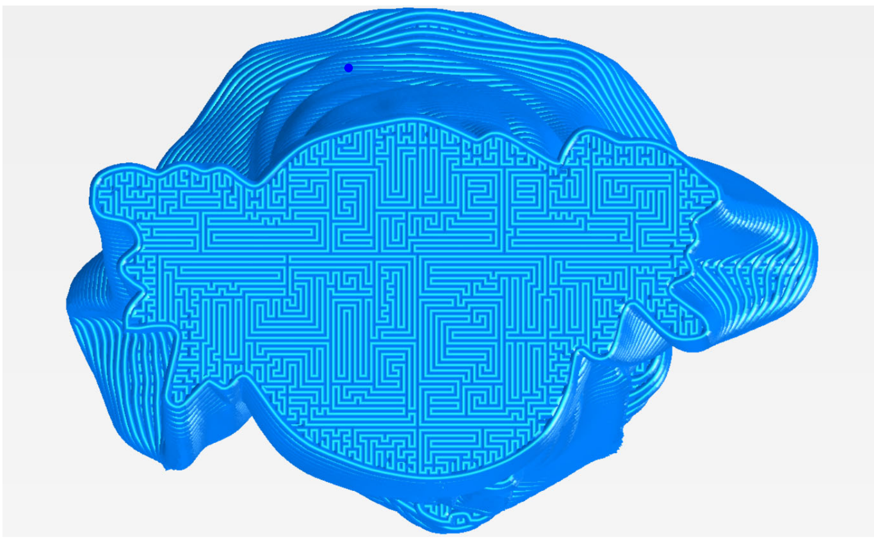
* layer-by-layer, dense infill 3D printing

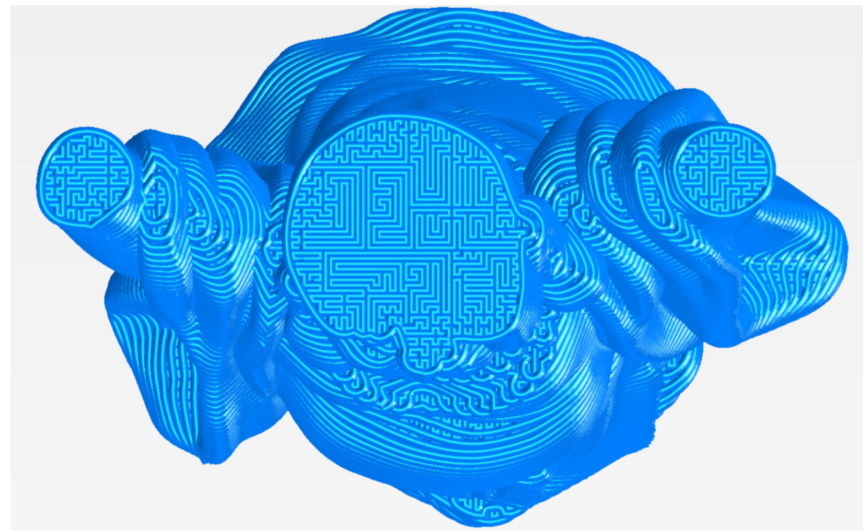




3D PRINTING: TOOL PATH OPTIM.

* layer-by-layer, dense infill 3D printing



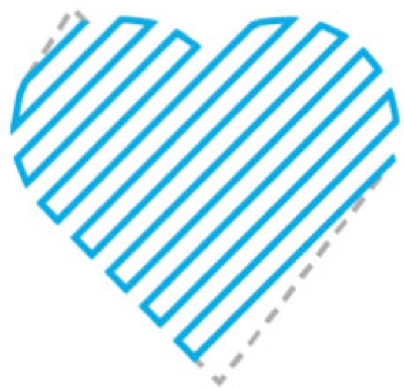


* optimize sequence of extruder movement (tool path) for
  - continuity
  - turn costs
  - smoothness
  - local orientation

  ...

TOOL PATH GEOMETRY

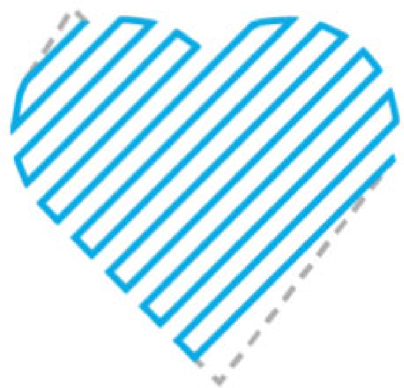
* standard : zigzag , contour parallel



img: formizable.com

TOOL PATH GEOMETRY

* standard : zigzag , contour parallel

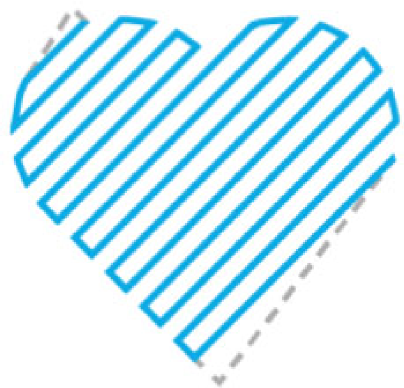


img: formizable.com

✗ suffer from directional biases

TOOL PATH GEOMETRY

* standard : zigzag , contour parallel

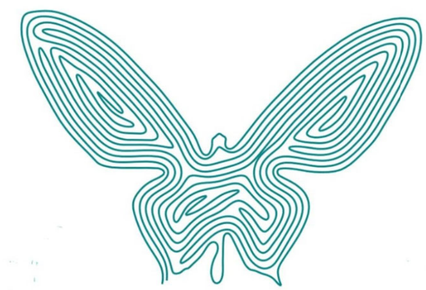


img: formizable.com

- ✗ suffer from directional biases
- ✗ cannot crossweave easily across layers

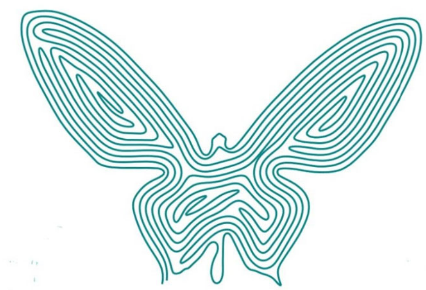
TOOL PATH GEOMETRY

* Zhao et al. (2016): Fermat spiral
  - smooth toolpath optimized for continuity
  - curved path approximated by PL line segments in practice

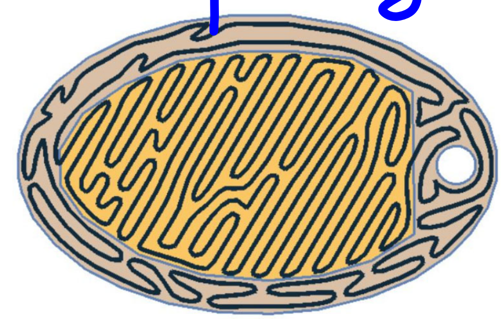


TOOL PATH GEOMETRY

* Zhao et al. (2016) : Fermat spiral
  - smooth tool path optimized for continuity
  - curved path approximated by PL line segments in practice

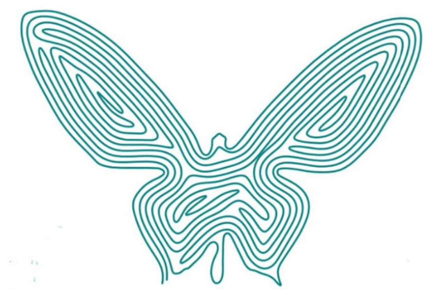


* Bedel et al. (2022): space-filling curve optimized for local orientations

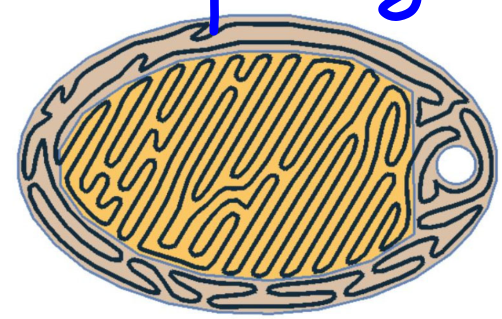


TOOL PATH GEOMETRY

* Zhao et al. (2016): Fermat spiral
  - smooth tool path optimized for continuity
  - curved path approximated by PL line segments in practice



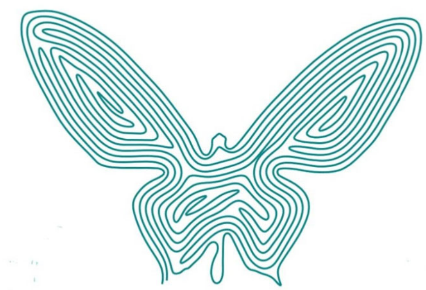
* Bedel et al. (2022): space-filling curve optimized for local orientations



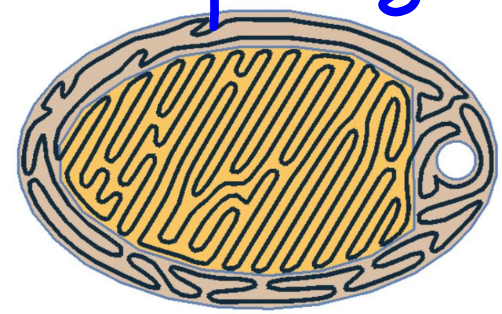
✗ hard to crossweave across layers

TOOL PATH GEOMETRY

* Zhao et al. (2016): Fermat spiral
  - smooth toolpath optimized for continuity
  - curved path approximated by PL line segments in practice



* Bedel et al. (2022): space-filling curve optimized for local orientations



✗ hard to crossweave across layers

✗ cannot apply to subdomains

QUESTIONS

? Optimize toolpaths by decomposing into subdomains?

QUESTIONS

? Optimize toolpaths by decomposing into subdomains?

? Process subdomains in parallel ?

QUESTIONS

? Optimize toolpaths by decomposing into subdomains?

? Process subdomains in parallel ?

? Optimize for multiple criteria ?

- continuity
- turn costs
- maximize rectilinear movement
- edge overlap across layers?

OUR RESULTS

✓ min turn 3DPP is NP-hard

OUR RESULTS

✓ min turn 3DPP is NP-hard

✓ SFCDecomp

- decompose domain into smaller parts
- optimize subdomain tool paths in parallel (for continuity, turn cost, ...)
- link subpaths using space-filling curves (SFC)

OUR RESULTS

- ✓ min turn 3DPP is NP-hard
- ✓ SFCDecomp
  - decompose domain into smaller parts
  - optimize subdomain tool paths in parallel (for continuity, turn cost, ...)
  - link subpaths using space-filling curves (SFC)
- ✓ optimize edge overlap between layers i & $i+1$
  - mechanical testing.

NP-HARDNESS OF 3DPP

* 3DPP related to milling, lawn mowing

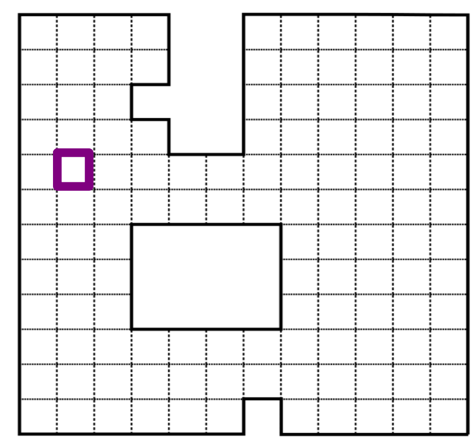
NP-HARDNESS OF 3DPP

* 3DPP related to milling, lawn mowing
  - min length milling, lawn mowing: NP-hard (Arkin et al., 2000)
  - min turn cost milling: NP-hard (Arkin et al. 2005)

NP-HARDNESS OF 3DPP

* 3DPP related to milling, lawn mowing
  - min length milling, lawn mowing: NP-hard (Arkin et al, 2000)
  - min turn cost milling: NP-hard (Arkin et al. 2005)
* we study integral orthogonal 3DPP

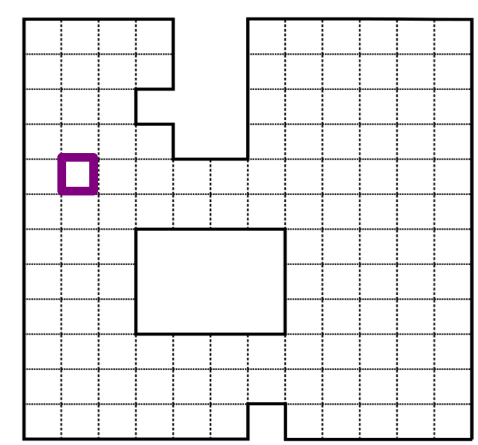
extruder: unit square



NP-HARDNESS OF 3DPP

* 3DPP related to milling, lawn mowing
  - min length milling, lawn mowing: NP-hard (Arkin et al, 2000)
  - min turn cost milling: NP-hard (Arkin et al. 2005)
* we study integral orthogonal 3DPP

extruder: unit square



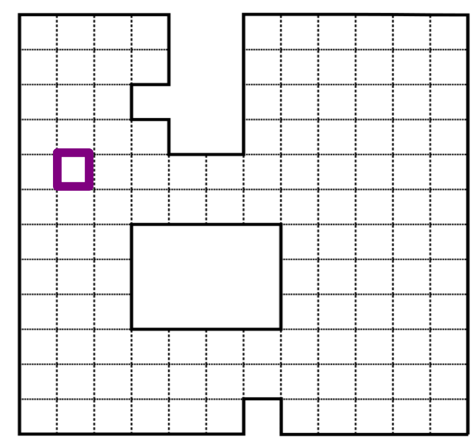
Lemma Min length 3DPP is NP-hard ($\approx$ lawn mowing)

NP-HARDNESS OF 3DPP

* 3DPP related to milling, lawn mowing
  - min length milling, lawn mowing: NP-hard (Arkin et al, 2000)
  - min turn cost milling: NP-hard (Arkin et al. 2005)

* we study integral orthogonal 3DPP

extruder: unit square

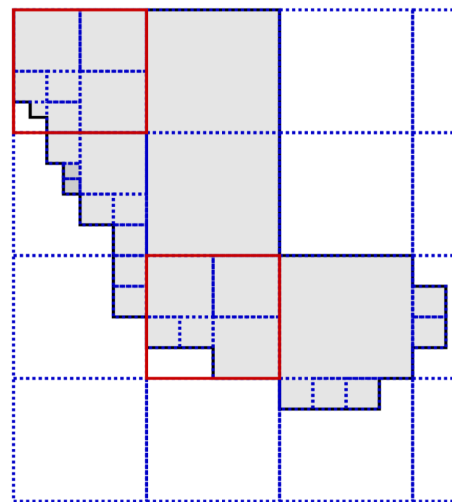


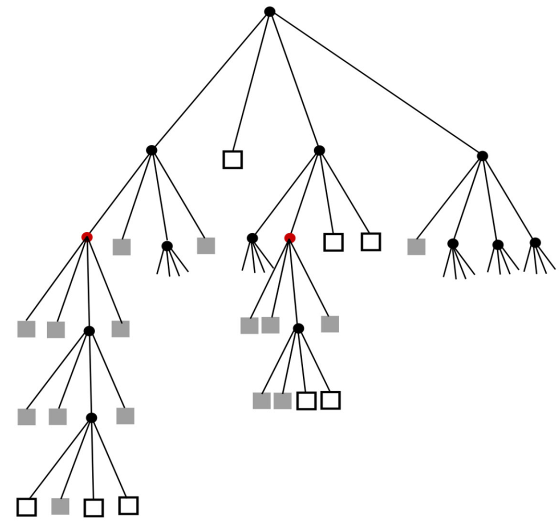
<u>Lemma</u> Min length 3DPP is NP-hard
(≈ lawn mowing)

<u>Theorem</u> Min turn cost 3DPP is NP-hard.

SFCDecomp: Domain Decomposition

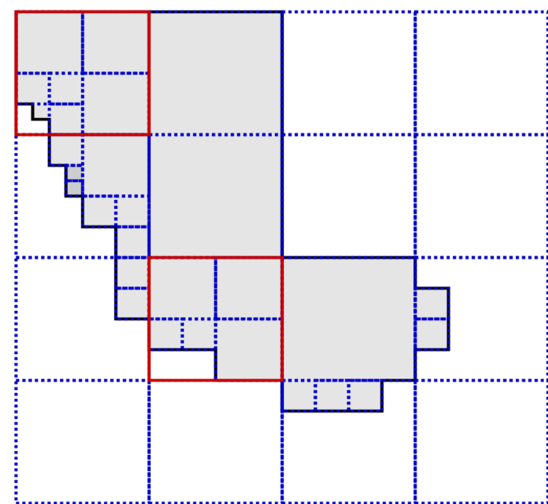
* quadtree representation of print domain
  - leaf squares in domain are kept
  - solve leaf cells in parallel

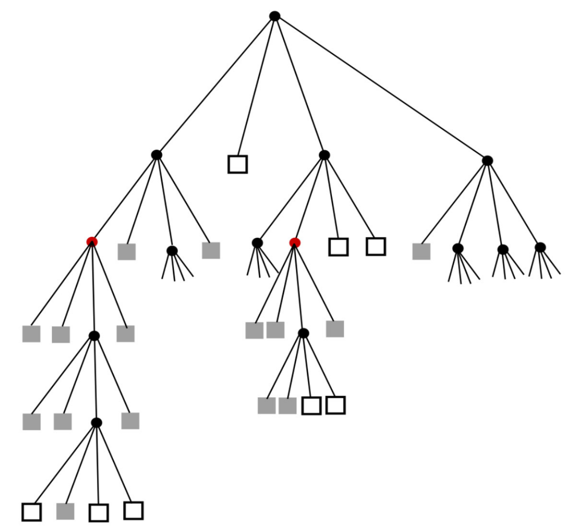




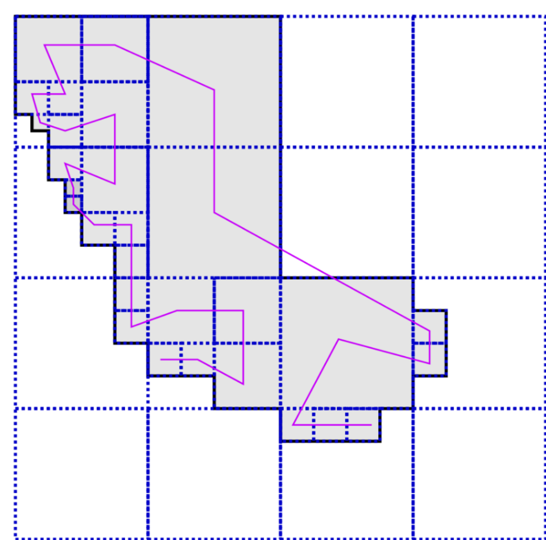
SFCDecomp: Domain Decomposition

* quadtree representation of print domain
  - leaf squares in domain are kept
  - solve leaf cells in parallel



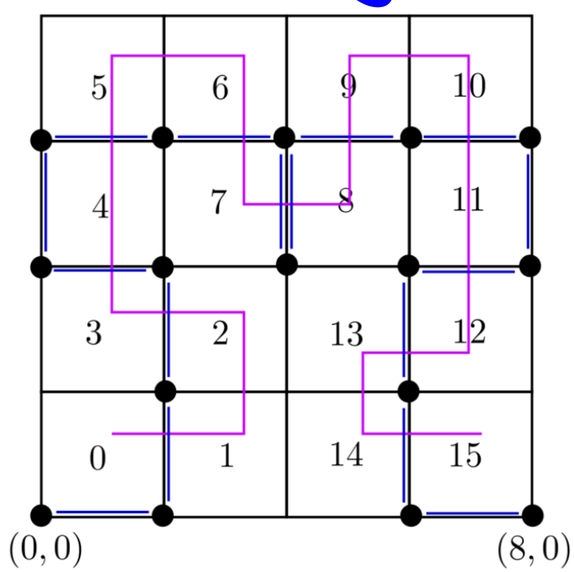


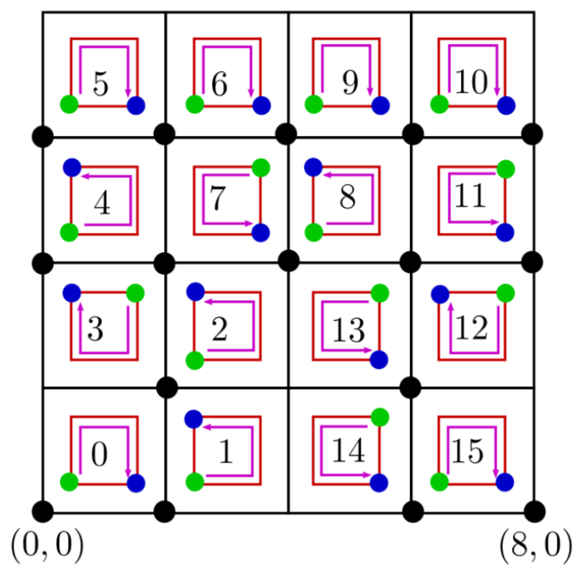
* connect leaf domains using a Hilbert SFC

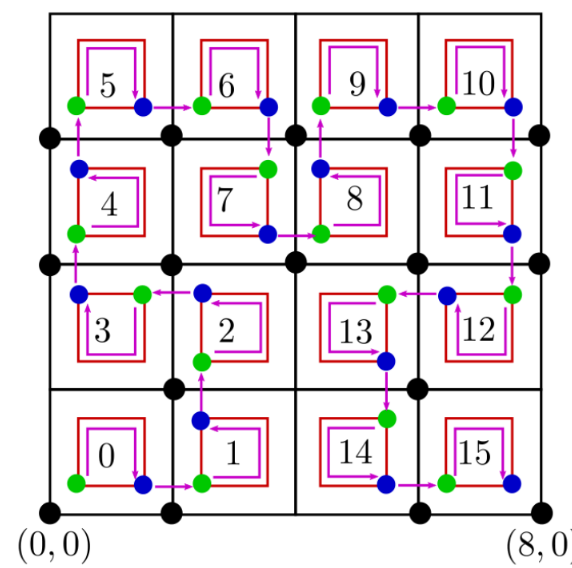


SFCDecomp: Domain Decomposition

* can merge square cells into bigger subdomains

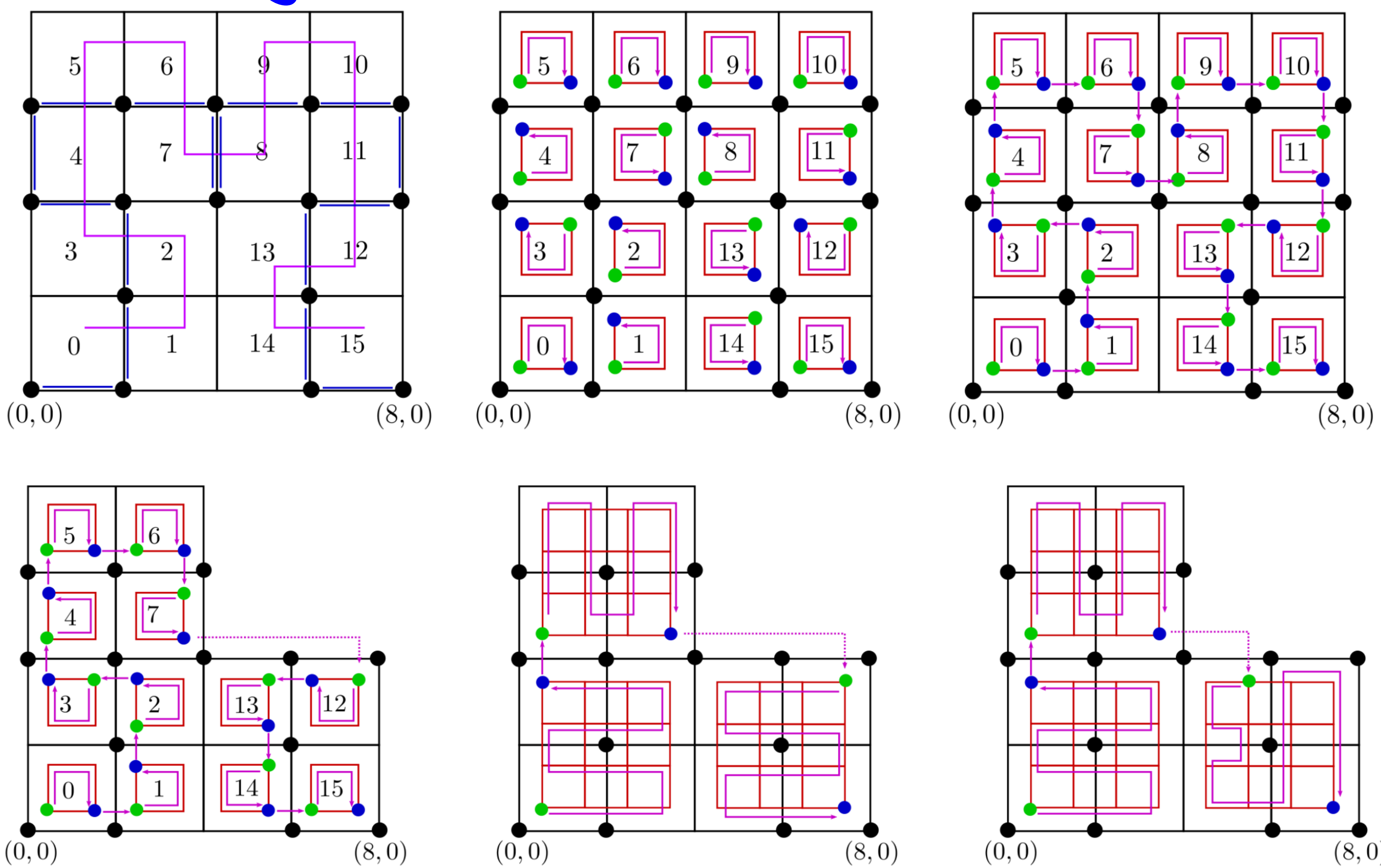






SFCDecomp: Domain Decomposition

* can merge square cells into bigger subdomains



SFCDecomp: Hamiltonian Path

Lemma Hamiltonian path is guaranteed to exist

SFCDecomp: Hamiltonian Path

<u>Lemma</u> Hamiltonian path is guaranteed to exist

- idle movement can be minimized

SFCDecomp: Hamiltonian Path

<u>Lemma</u> Hamiltonian path is guaranteed to exist

- idle movement can be minimized
- if subdomains are solved to optimality, any discontinuities (idle movements) occur only at boundary of print domain

SUBDOMAIN TOOL PATH : MIP

* Miller-Tucker-Zemlin (MTZ)-type formulation
  - turn cost: c_j at node j:

$$c_j \geq A_{ijk}(x_{ij} + x_{jk} - 1), \quad c_j \geq 0$$

A_{ijk}: 1 if $(i,j) \perp (j,k)$

SUBDOMAIN TOOL PATH : MIP

* Miller-Tucker-Zemlin (MTZ)-type formulation
  - turn cost c_j at node j:

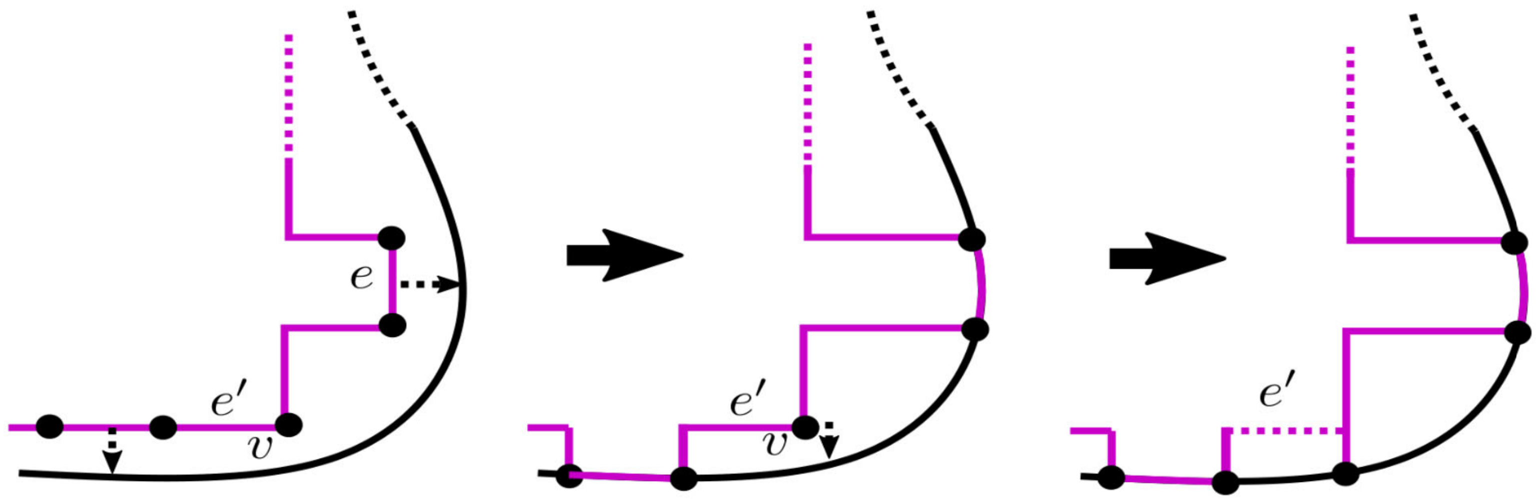
$$c_j \geq A_{ijk}(x_{ij} + x_{jk} - 1), \quad c_j \geq 0$$

$A_{ijk} \to 1$ if $(i,j) \perp (j,k)$

* solve large instances (approx):
  - relax subtour constraints
  - megre subtours using 2-opt + min. span. forest.
  - found Hamiltonian path in all 10,000+ instances except one

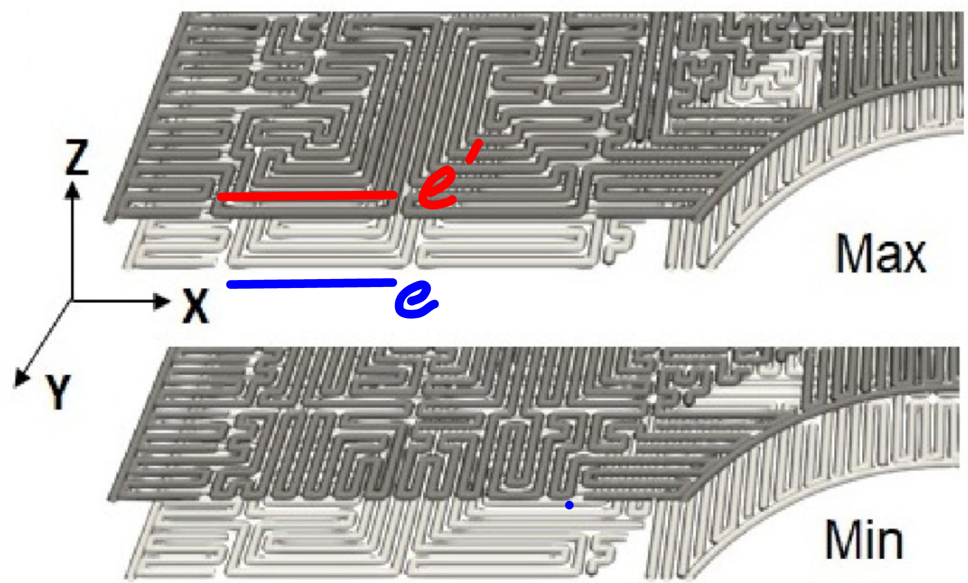
GENERAL GEOMETRY

* project print boundary vertices of IOD to the general polygon ($\supset$ IOD)



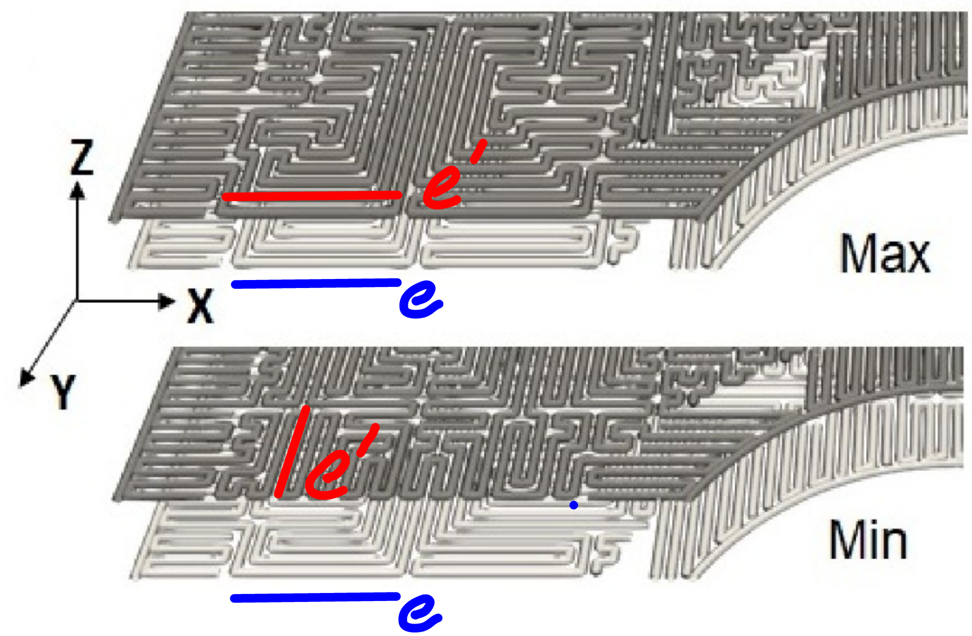
OPTIMIZING EDGE OVERLAPS

* overlap between edges e and e' is
  - max if $e = e'$



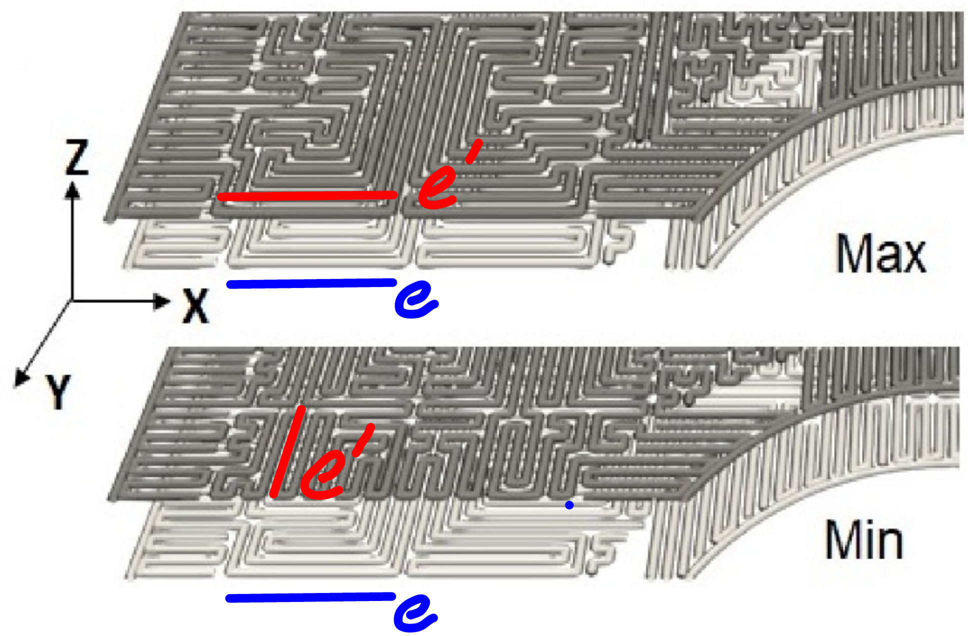
OPTIMIZING EDGE OVERLAPS

* overlap between edges e and e' is
  - max if $e = e'$
  - min if $e \cap e'$ is a single point



Optimizing Edge Overlaps

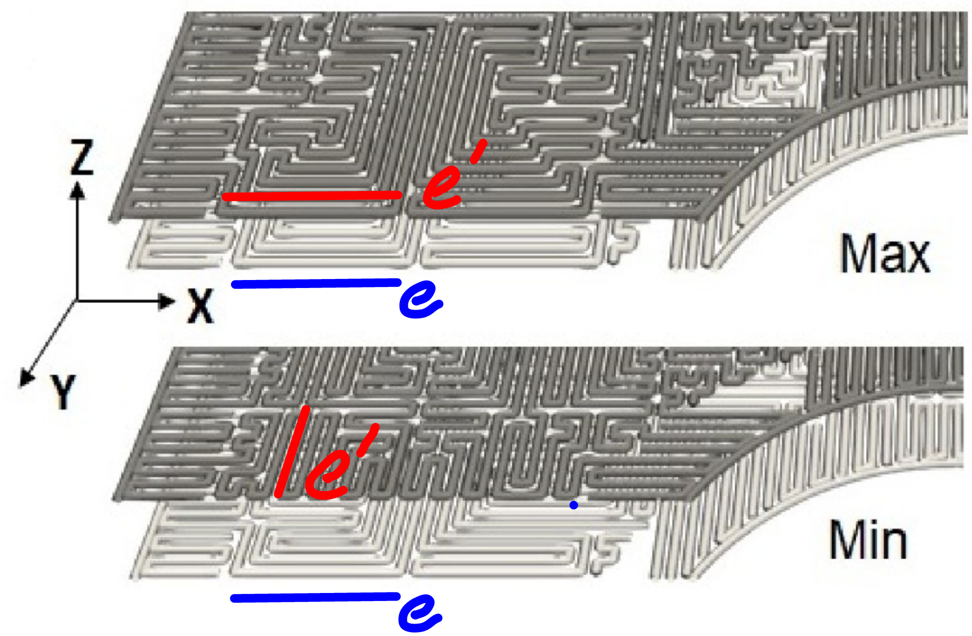
- overlap between edges e and e' is
  - max if $e = e'$
  - min if $e \cap e'$ is a single point



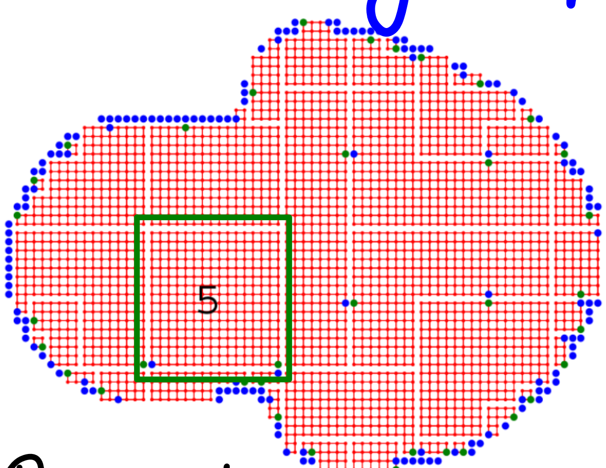
- set weight of e' lower (higher) for max (min) overlap

OPTIMIZING EDGE OVERLAPS

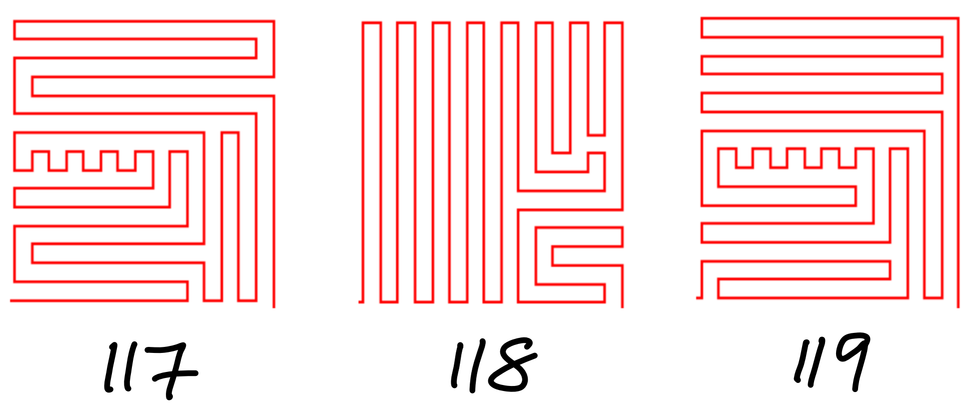
* overlap between edges e and e' is
  - max if $e = e'$
  - min if $e \cap e'$ is a single point



* set weight of e' lower (higher) for max (min) overlap

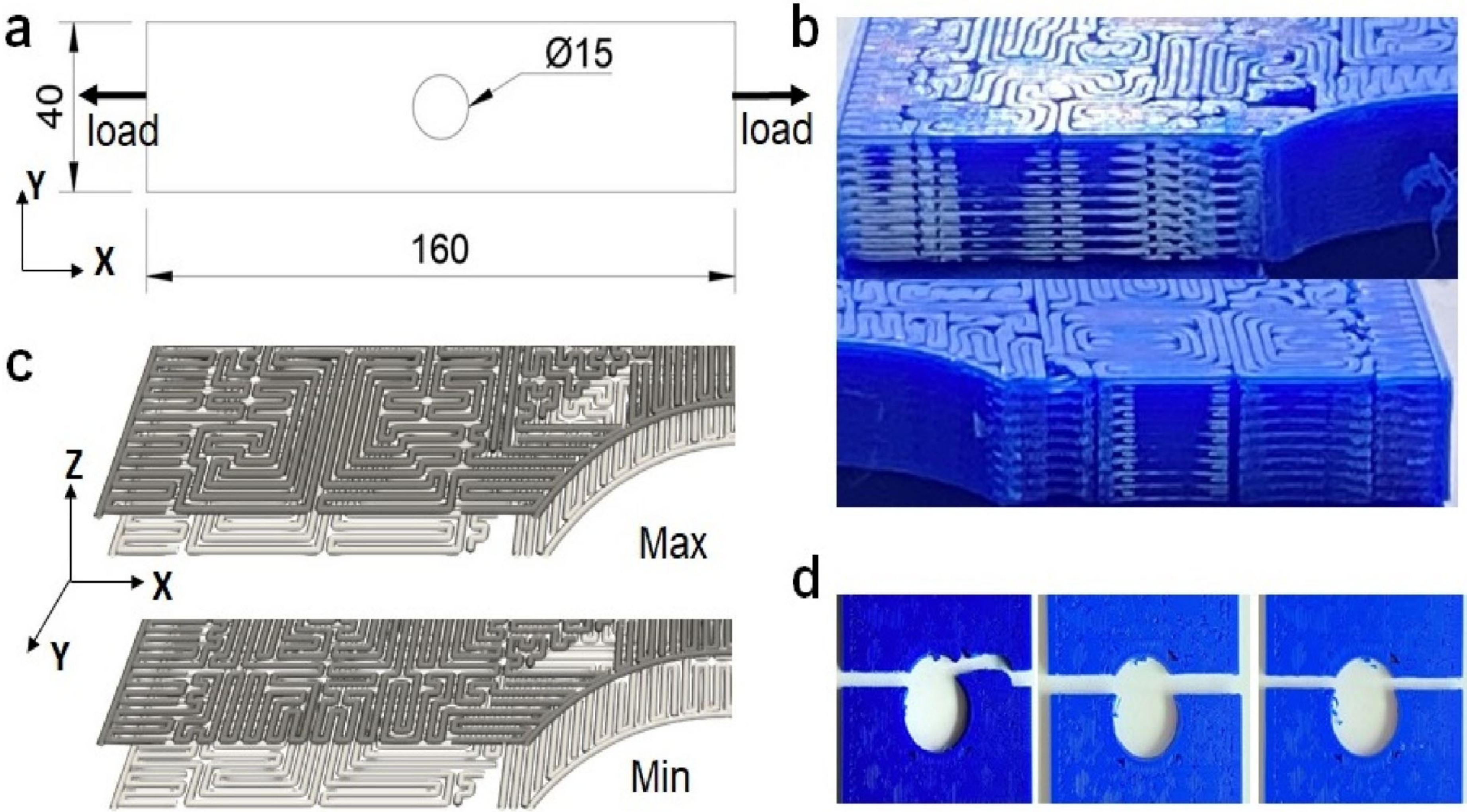


Bunny: layer 117

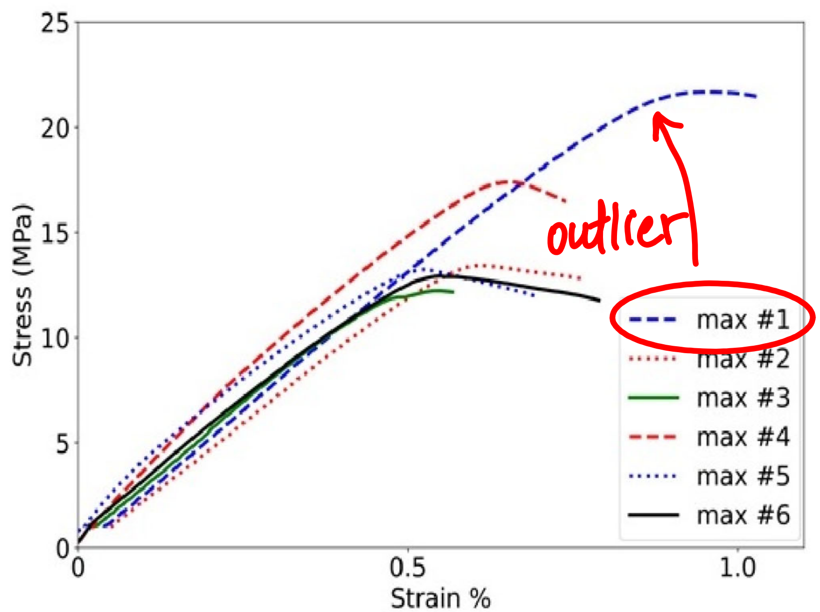


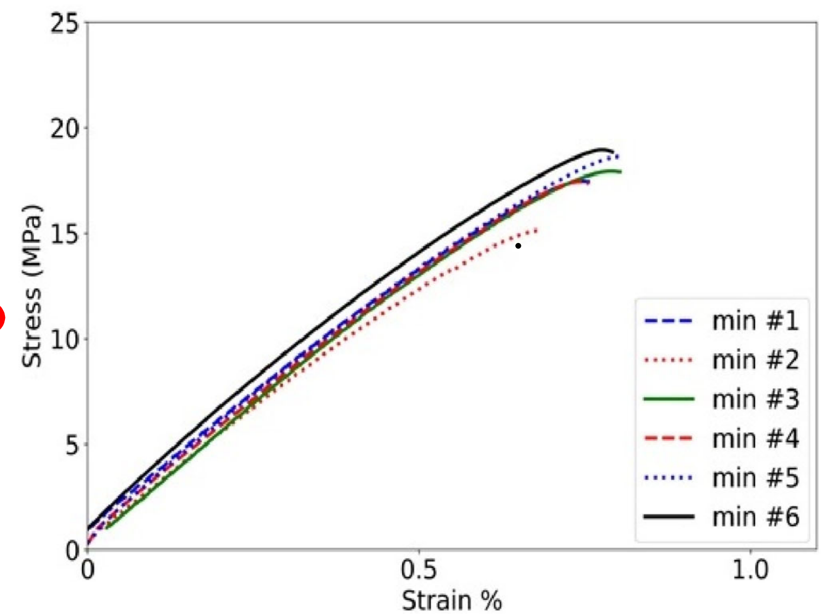
min edge overlap

Mechanical Testing

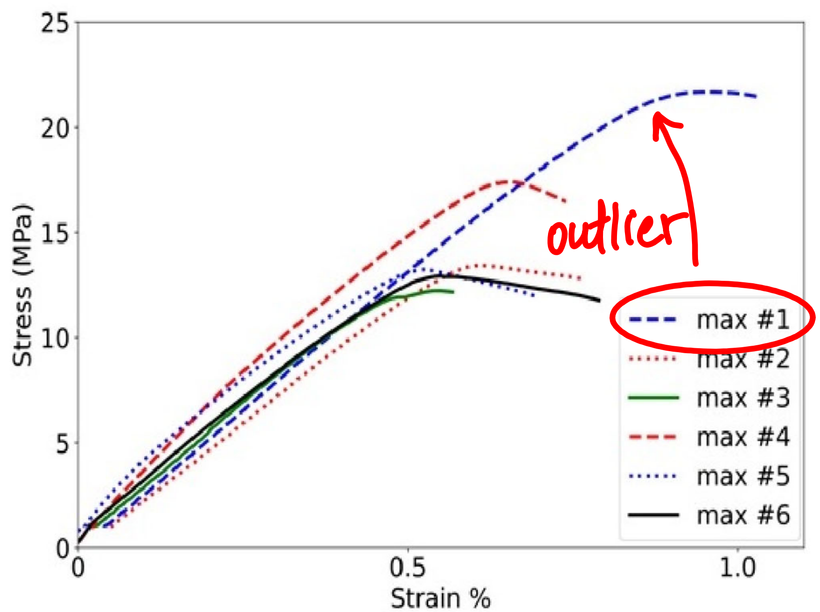


MECHANICAL TESTING

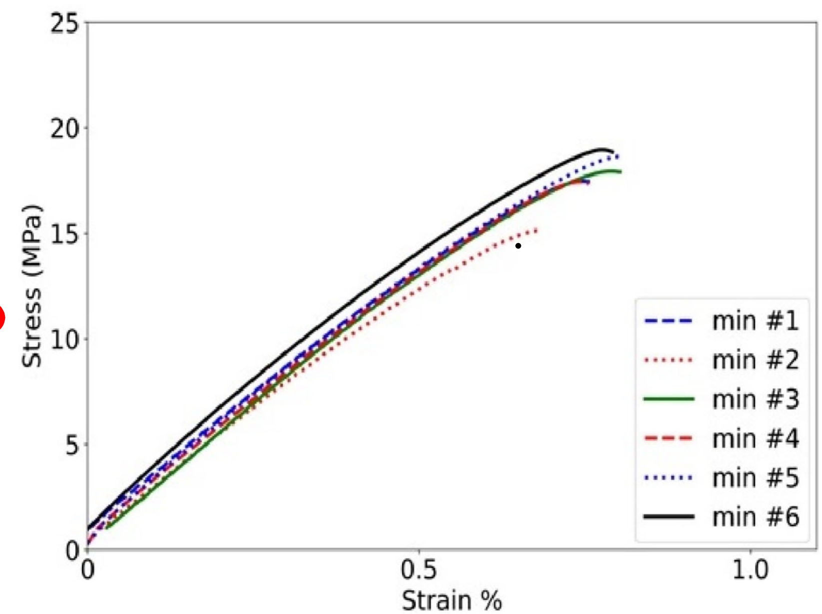




MECHANICAL TESTING

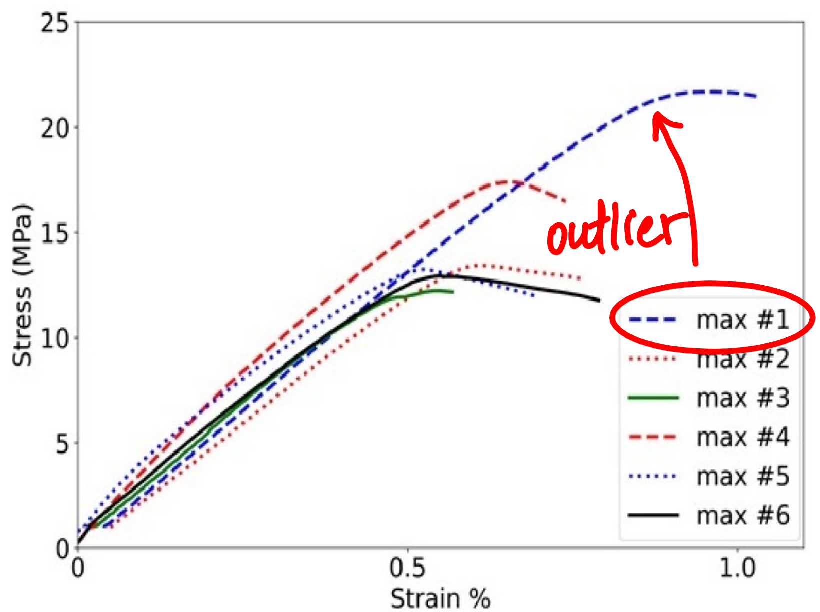


max overlap
avg. strength = 13.84 MPa
(w/o outlier)

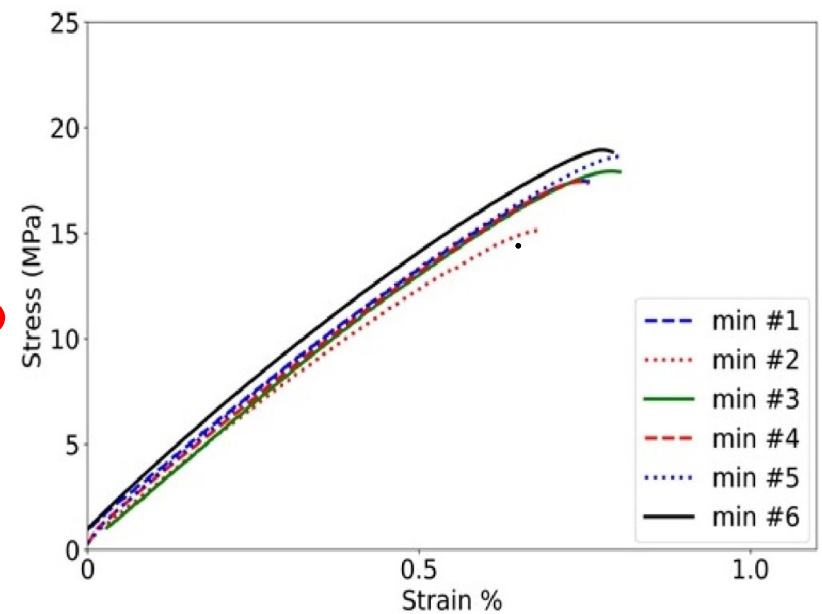


min overlap
avg. strength = 17.60 MPa

MECHANICAL TESTING



max overlap
avg. strength = 13.84 MPa
(w/o outlier)



min overlap
avg. strength = 17.60 MPa

* min edge overlap had better strength

POINTS TO NOTE

* Edge weights could capture various properties
  - can we optimize internal microstructure?

POINTS TO NOTE

* Edge weights could capture various properties
  – can we optimize internal microstructure?
* Could use approx. algos for subdomain path planning (instead of MIP)

POINTS TO NOTE

* Edge weights could capture various properties – can we optimize internal microstructure?
* Could use approx. algos for subdomain path planning (instead of MIP)
* Could combine with other approaches within subdomains (e.g., Fermat spiral, alignment maximization, ...)

Points to Note

* Edge weights could capture various properties – can we optimize internal microstructure?
* Could use approx. algos for subdomain path planning (instead of MIP)
* Could combine with other approaches within subdomains (e.g., Fermat spiral, alignment maximization, ...)

Thank You!