Decomposition of Planar, Straight-sided Shapes using Reinforcement Learning

Cristina Garcia Cardona, Navamita Ray, Ben DiPrete, Rao Garimella

Los Alamos National Laboratory

1. Summary

We demonstrate a novel approach to block-decomposition of planar straight-sided shapes using a reinforcement learning framework. This work is a proof-of-principle study in the path towards the goal of automatically decomposing 3D CAD models such that they can be meshed with good quality hexahedra.

2. Introduction

- All-hex mesh generation long standing problem - not fully automated.
- 3+ decades of research - plastering, sweeping, whisker weaving, and more.
- Good quality hex meshing still requires at least some human intervention.
- Meshing can take up to 50% of the time in design-analysis cycle

- Can we leverage new AI/ML techniques to shift the needle towards automation?

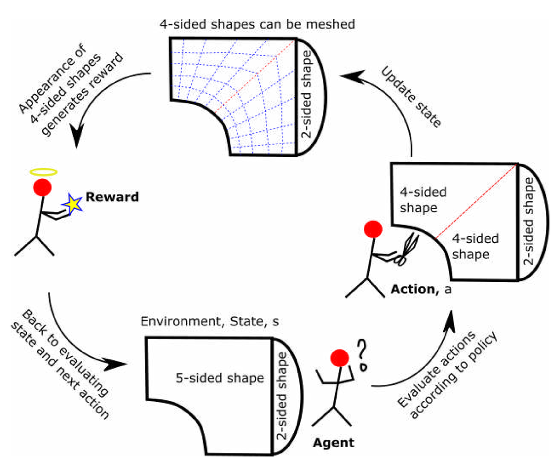
- We present proof-of-principle of using AI for model decomposition.
- Restricted to planar, axis-aligned, straight-sided models.

3. Reinforcement Learning

- Mimics human learning, video game play

- **Training**:
  Agent **explores** environment taking actions, getting rewards
  Agent adjusts policy (state $\rightarrow$ action) to maximize cumulative reward

- **Deployment**:
  Agent **exploits** learned policy to execute task

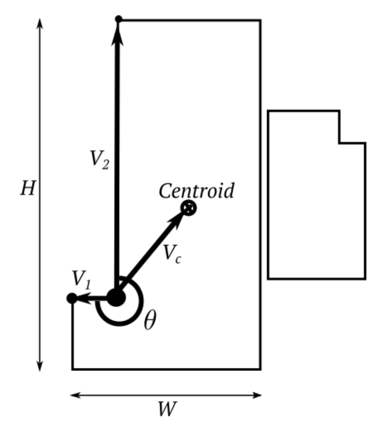


4. RL Applied to Block Decomposition of CAD Models

- Agent acts on CAD model (environment)
- Picks a vertex at which to make a modification
- Picks a direction along which to make a cut (X or Y)
- Gets reward based on the quality of subdomains
- Adjusts policy according to rewards
- Moves to next subdomain of CAD model

5. Environment State or Observations

- Environment (CAD model) is not fixed; evolves as cuts are made
- Number of global observations change - not suited for NNs
- Use a fixed set of local observations [Pan, et.al. 2022]
  - Vectors to adjacent vertices
  - Vector to centroid of shape
  - Interior angle at vertex
  - Aspect ratio of shape

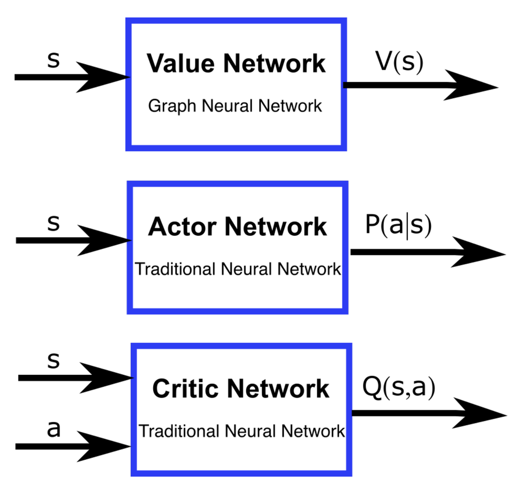


6. Soft-Actor-Critic Reinforcement Learning [Haarnoja, et.al. 2018]

- **Value Neural Network**: Computes the expected value of each state
- **Actor Neural Network**: Generates probabilities for actions to take in a state
- **Critic Neural Network**: Generates feedback on quality of action in a state

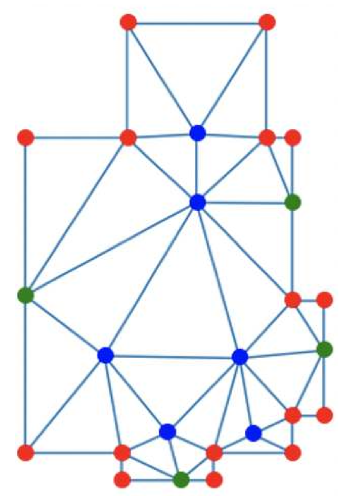
- Actor, Critic networks implented using PyTorch; Value network implemented using PyTorch Geometric
- Off-policy formulation for exploration and Entropy Maximization for dialing back exploration
- Implementation details in paper [DiPrete, Garimella, Cardona, Ray, 2022]

- SAC framework can handle continuous state and actions (e.g. cuts at arbitrary angles)



7. Value Network

- Value network based on triangulation of the domain
- Uses Spline CNN on a Graph Neural Network [Fey, et.al. 2017]
- Input indicates if vertex is on model vertex, edge or interior (1-hot encoding)
- Computes expected value at each vertex based on rewards
- Actor network uses value to choose a vertex to take next action



8. Reward Function

$$10\frac{N_q}{N} - 5(1-\delta_{1N}) + 3*\left[\left(\frac{N}{\sum_i R_i^2}\right)^2 - \left(\frac{\sum_i (A_i-\bar{A})^2}{\sum_i A}\right)\right]$$

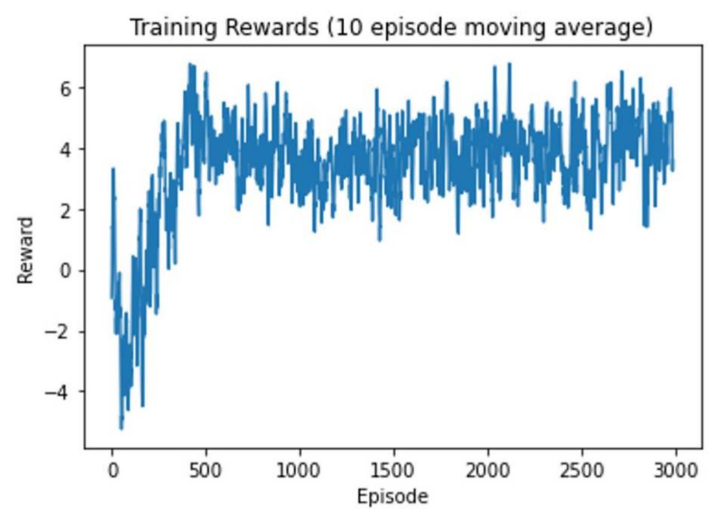
- Reward is maximized if a cut generates all squares with equal area

- N is number of new subdomains, N_q is number of new quadrilaterals
- A_i is area of i'th subdomain, $\bar{A}$ is average area of subdomains
- R_i is the aspect ratio of the i'th subdomain
- $(1-\delta_{1N})$ penalizes cuts that don't produce new subdomains

9. Data

- 37 training models, 12 testing models
- Union of randomly scaled+translated rectangles
- Scripted using Python API to OpenCascade

10. Convergence

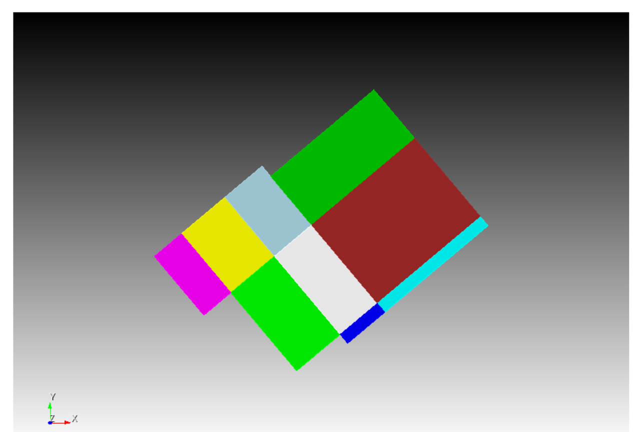


11. Decompositions

Test Shape | Block decomposition returned by the agent | Quad mesh

12. Ongoing Work

- Extending work to rotated planar, straight sided shapes
- Cut at a vertex along connected edge directions
- Extend to 3D polyhedra



13. Future Work

- Generalize to curved planar shapes
- Generalize to cuts at arbitrary angles
- Extend to 3D polyhedral models
- Generalize to curved 3D shapes
- Incorporate partial cuts, booleans and other operations

14. Acknowledgments

This work was performed under the auspices of the National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396, and was supported by the LDRD (Lab Directed Research and Development Program