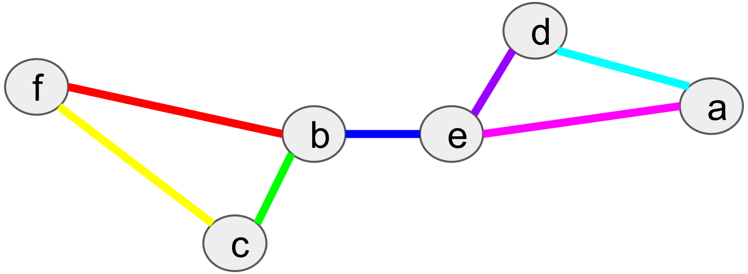
Hypergraph Partitioning via Geometric Embeddings

Sepideh Maleki* Udit Agarwal
Keshav Pingali

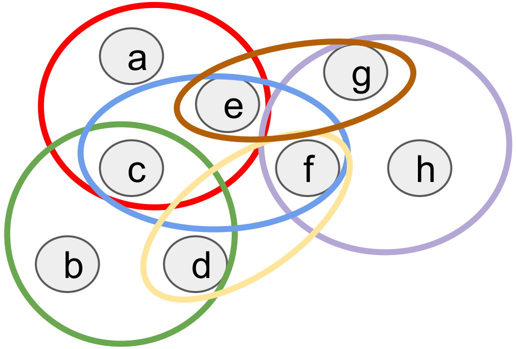
The University of Texas at Austin

What is a hypergraph?

- Graph G = (V, E)
  - V = set of nodes
  - E = set of edges
  - An edge (hyperedge) can connect **two nodes**

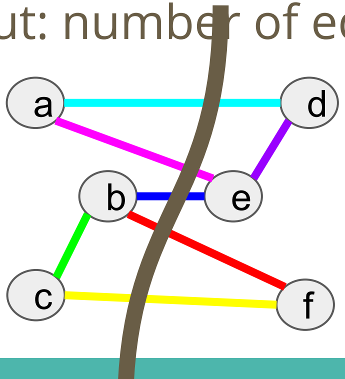


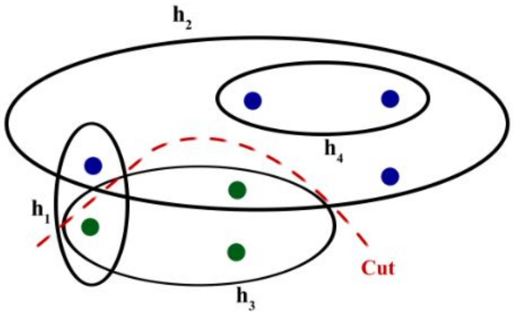
- Hypergraph H = (V, E)
  - V = set of nodes
  - E = set of edges
  - An edge (hyperedge) can connect **any number of nodes**



Preliminaries

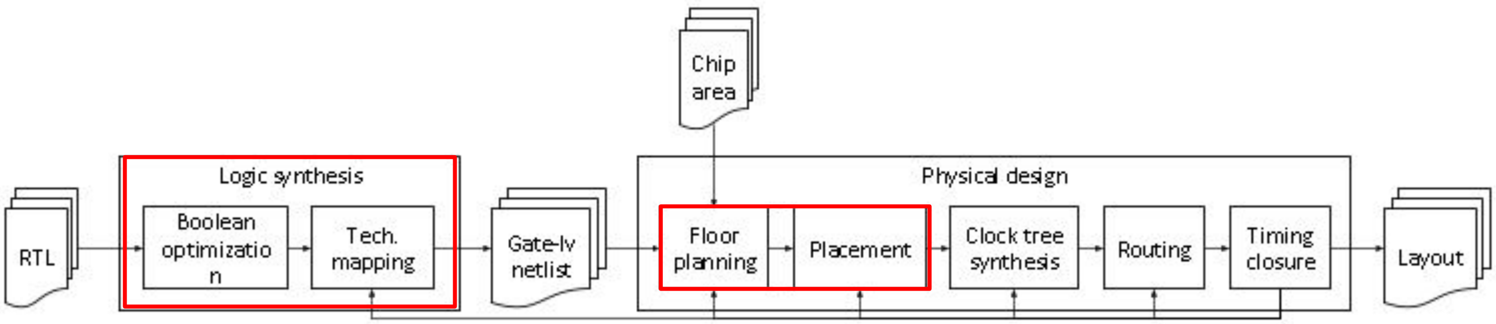
- Hyperedge Partitioning: given a number $K > 1$
    - partition Π of V with blocks of nodes $\Pi = (V1, \ldots, Vk)$: $V1 \cup \cdots \cup Vk = V$ and $Vi \cap Vj = \varnothing \ \forall i \mathrel{!=} j$
- All blocks must follow the balance constraint
    - $\forall i \in \{1, \ldots, k\} : |Vi| \leq L_{max} = (1+\varepsilon)[|V| / k]$ for some imbalance parameter $\varepsilon \in R_{\geq 0}$
- Goal of partitioning: minimize or maximize a particular objective function
- Minimize total cut: number of edges (hyperedges) connecting different partitions





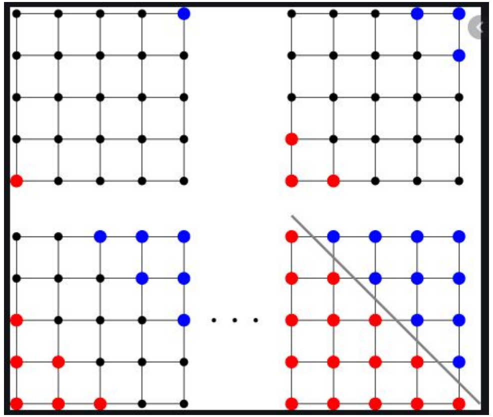
Application

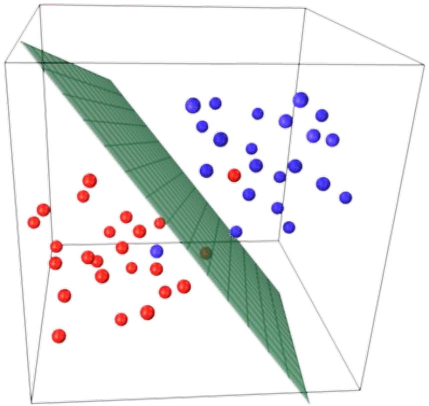
- EDA flow
  - Logic synthesis
    - Physical design
  - Floorplacement

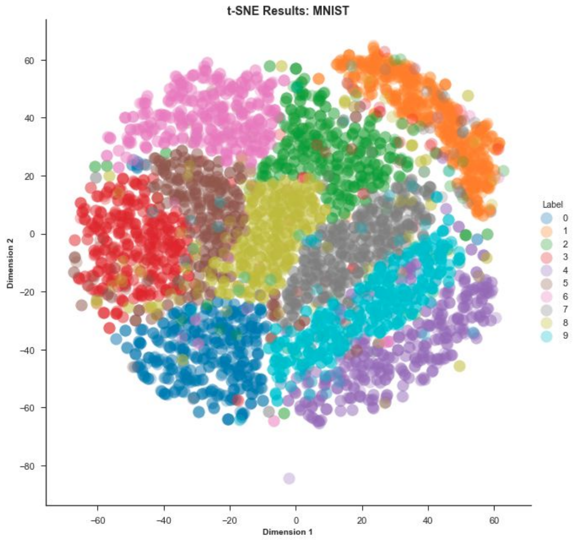


Existing approaches to partitioning

- Techniques
    - Topology based methods: use node connectivity (Metis, GMetis, BiPart)
    - Geometry based methods: nodes are points in R^d (Miller et al 93)

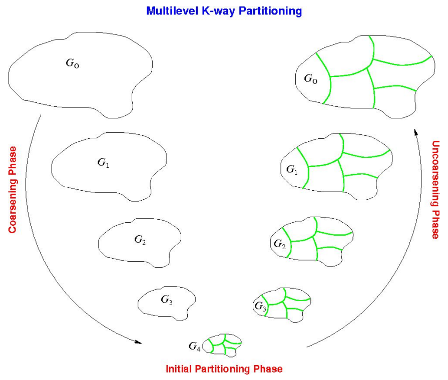




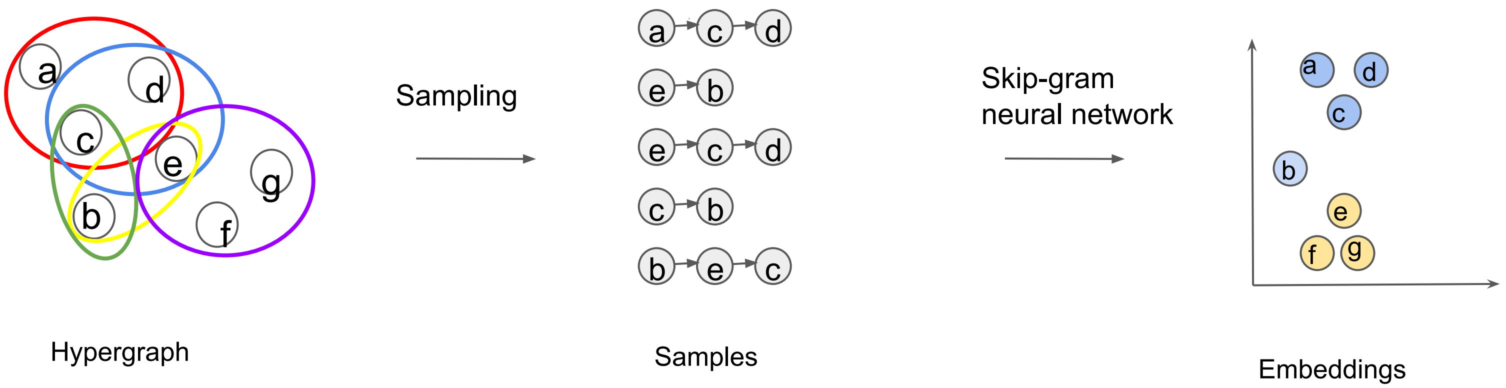


Contributions of paper

- Use machine learning to learn geometry from topology of graph
    - Vector-space models: embedding
- Use the geometry to enhance BiPart
    - Hierarchical hypergraph partitioner
- Provides a general approach to exploiting geometric information in hierarchical partitioners



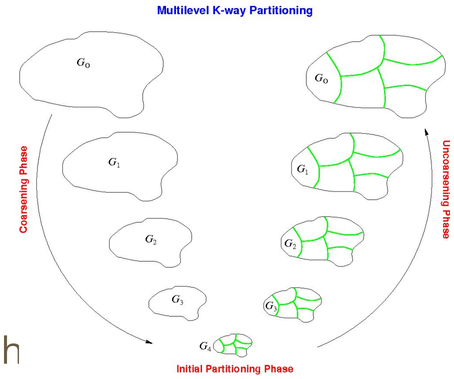
Embedding process for hypergraphs



- Hypergraph embedding
  - Transforms nodes, edges, and features into vector space
  - *Similar nodes* (neighbor nodes) are closer in space

Using embeddings in hypergraph partitioning

- Coarsening phase
  - Find node matchings
- Initial partitioning
  - Use geometry-based partitioning such as spectral partitioning
- Refinement phase
  - Refine partitions by swapping l_{min} nodes between partitions, with nodes that are farther away from the centroid of the current partition



Results

Hypergraph	Nodes	Hyperedges	Edges
Xyce	1,945,099	1,945,099	9,455,545
Webbase	1,000,005	1,000,005	3,105,536
Xenon	157,464	157,464	2,312,497
Stanford	281,903	281,903	2,312,497
Ibm3	23,136	27,401	93,573

	Edge cut	
Hypergraph	Topology based HP	Embedding based HP
Xyce	1,164	2,558
Webbase	1,060	8,81
Xenon	8,157	3,672
Stanford	1,746	302
Ibm3	2,172	1,787

Summary

- Contributions:
  - Given topology, infer geometry using machine learning
  - Use geometry to enhance topological-based hypergraph partitioning scheme
  - If geometry of circuit is available, use that instead of calculating embedding

- Initial implementation: https://github.com/Breakinbad/Galois-1/tree/master/lonestar/experimental/embedding

Thank You!