Graph and Hypergraph Partitioning for Parallel Computing

Edmond Chow
School of Computational Science and Engineering
Georgia Institute of Technology

June 29, 2016

Graph and hypergraph partitioning

References:

- Graph Partitioning for High Performance Scientific Simulations, Kirk Schloegel, George Karypis, and Vipin Kumar, 2000
- Hypergraph-Partitioning-Based Decomposition for Parallel Sparse-Matrix Vector Multiplication, Umit V. Catalyurek and Cevdet Aykanat, 1999

Some figures below are from the above references.

Distributed sparse matrix-vector multiplication

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1p} \\ A_{21} & A_{22} & \cdots & A_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} & A_{p2} & \cdots & A_{pp} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$$

Processor i stores $A_{i\star}$, x_i and computes y_i
(block row-wise partitioning).

Distributed sparse matrix-vector multiplication

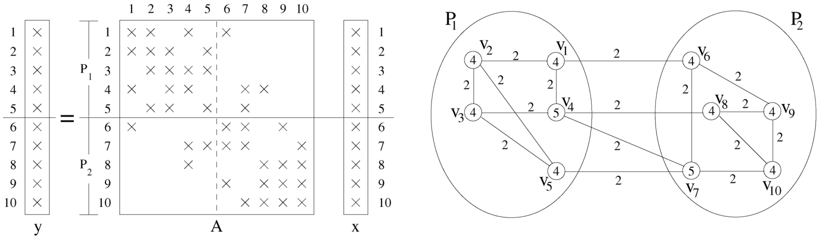
$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1p} \\ A_{21} & A_{22} & \cdots & A_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} & A_{p2} & \cdots & A_{pp} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$$

On processor i:

1. Send components of x_i needed by other processors
2. Compute $y_{local} = A_{ii} x_i$
3. Receive components of x_j from other processors j needed by processor i
4. Compute $y_{ext} = A_{i,ext} x_{ext}$
5. Form $y_i = y_{local} + y_{ext}$

If processor i contains nonzeros in column k, then processor i needs the kth scalar component of x.

Graph model (for $p = 2$)



The graph model shows which nonzeros (edges) induce communication in SpMV.

Graph partitioning problem

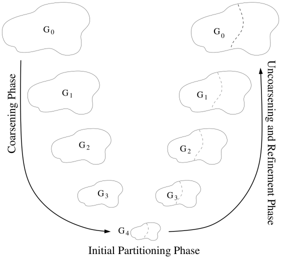
Given $G = (V, E)$, and number of parts p,
find the partitioning of the vertices $P_1, P_2, \ldots, P_p$, where P_i contains a subset of the vertices in V,
such that the P_i are disjoint and complete
and

$$|P_i| \leq \left(\frac{1}{k}\sum |P_j|\right)(1+\epsilon)$$

for all i (number of vertices in each partition is approximately equal) and the number of **cut edges** is minimized. A cut edge straddles two partitions.

This problem is NP-hard.

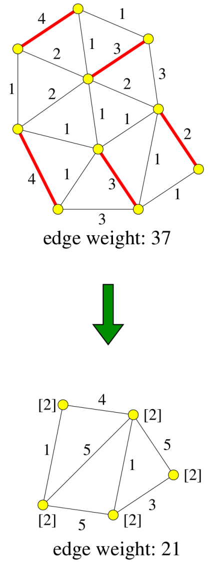
Multilevel graph partitioning



1. Coarsen the graph by collapsing heavy edges. Vertex weights and edge weights are updated. Vertex and edge weights are initially 1.
2. Repeat until the graph is small enough, then partition the coarsest level graph by some method.
3. Uncoarsen. After each uncoarsening step, refine the partition (using, e.g., Kerninghan-Lin).

Software: METIS, Chaco, SCOTCH.

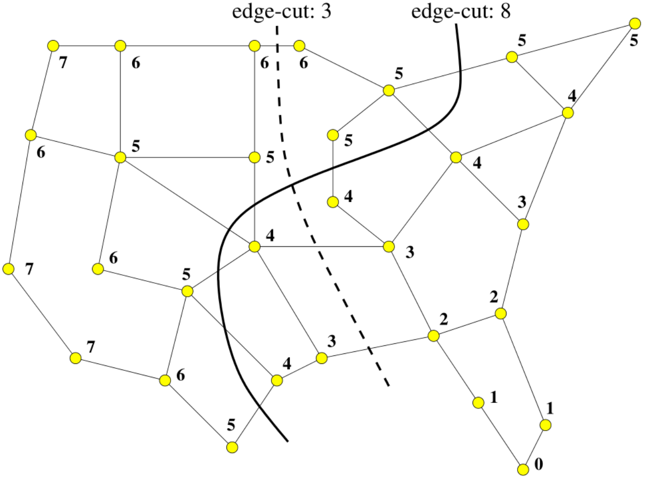
Graph coarsening



- Edges are collapsed by using a heavy-edge matching.
- A matching is a set of edges such that no two edges share a vertex.

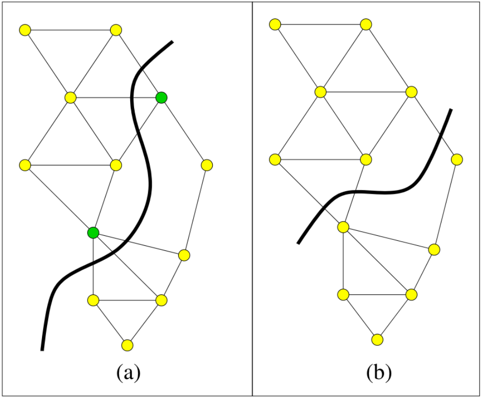
Levelized nested dissection

Possibility for partitioning the coarsest level graph.



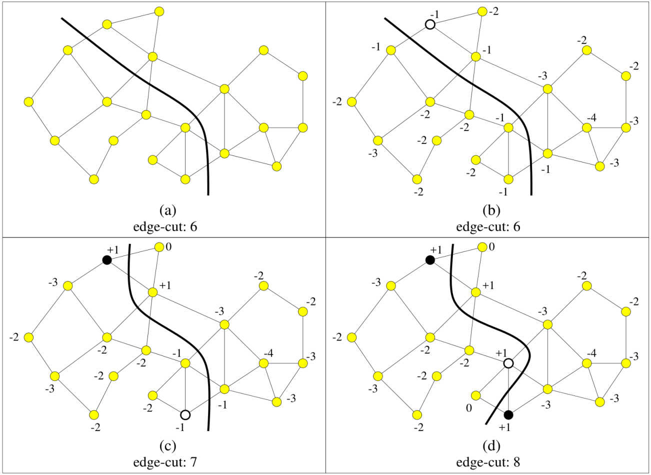
Kernighan-Lin

Find pairs of nodes on opposites sides of the partition that can change partitions so that the edge-cut is improved. Repeat until all nodes have been moved and then restore the best configuration. Repeat above procedure until the best configuration found is unchanged.



Fiduccia-Mattheyses

Compute gain for each node. Change partition of node with largest gain. Update gains. Gains are stored in priority queue and a node that has moved will not move again until all others have moved.



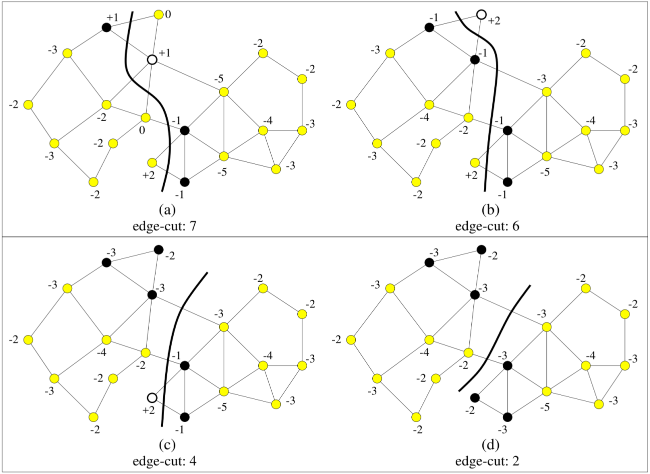
(a) edge-cut: 6

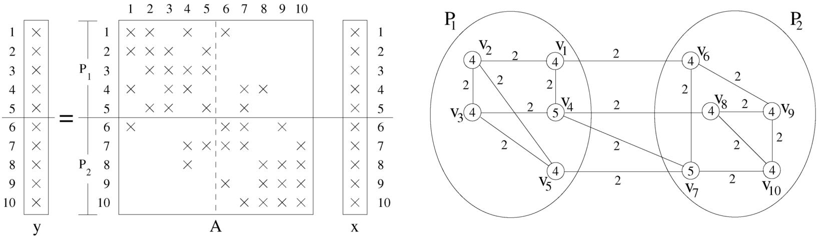
(b) edge-cut: 6

(c) edge-cut: 7

(d) edge-cut: 8

Fiduccia-Mattheyses





The cut size does not measure the communication volume.

This deficiency can be addressed with hypergraph partitioning.

Hypergraph

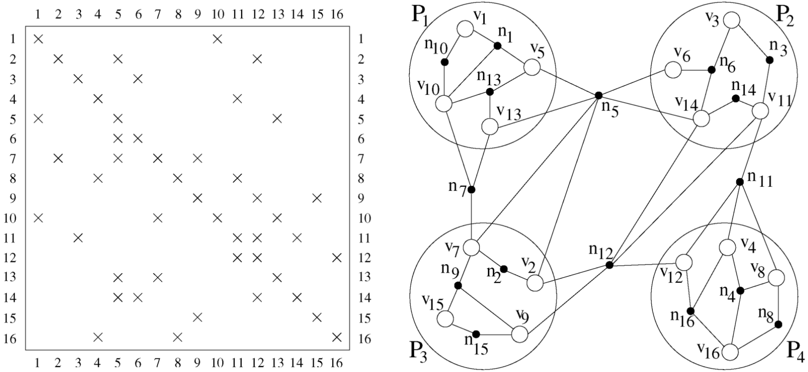
Hypergraph $H = (V, N)$, where V is the set of vertices and N is the set of hyperedges (or nets).

A hyperedge can connect more than 2 vertices.

Hypergraph of a sparse matrix

- Rows are vertices.
- Columns are hyperedges. Nonzeros in column j correspond to the vertices connected by the hyperedge.

Hypergraph


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
P1
P2
P3
P4
v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 v11 v12 v13 v14 v15 v16
n1 n2 n3 n4 n5 n6 n7 n8 n9 n10 n11 n12 n13 n14 n15 n16

Hypergraph partitioning problem

Group the vertices into approximately equal partitions such that

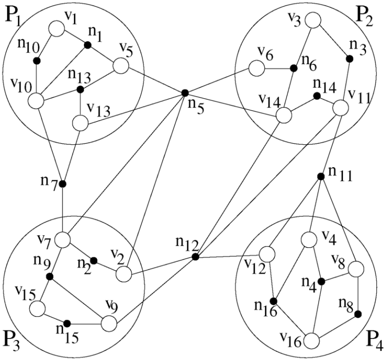
$$\min \sum_{n_i \text{ is cut}} (\lambda_i - 1)$$

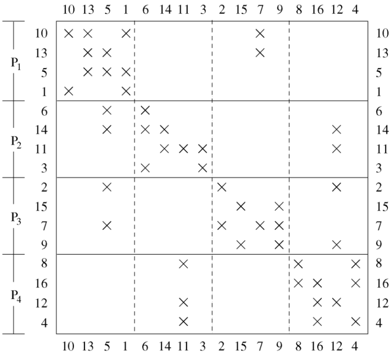
where λ_i is the number of parts connected by hyperedge i.

Each hyperedge belongs in a partition, so the number of neighbors is $\lambda_i - 1$.

NP-hard problem.

Hypergraph partitioning





Hypergraph partitioning

Multilevel algorithm: coarsen hyperedges, refinement with hyperedges.

Software: PaToH, hMETIS.

For FEM meshes, regular graph partitioning is usually good enough. For very unstructured meshes, e.g., from circuit simulation, hypergraph partitioning is needed.

Exercise

Consider this graph consisting of 8 vertices and a partition into two parts (separated by the dotted line):

```
               :
1-----3--:--5-----7
|     |  :  |     |
|     |  :  |     |
2-----4--:--6-----8
               :
```

What is the hypergraph corresponding to this graph? How many hyperedges are cut by the given partitioning?