Kernighan–Lin (KL)

KL is the most widely used *local refinement* algorithm for GP. It is an iterative method based on inner and outer iterations.

Original KL: $O(n^3)$ time pr. outer iter.
Fiduccia & Mattheyses: $O(m)$ time pr. outer iter.

The FM algorithm is similar to KL but uses more advanced data structures.

Basic operation: move a vertex from one set to the other and measure the *gain*. For bisection ($V = A \cup B$) the gain when moving node i from A to B is:

$$gain(i) = \sum_{v_j \in B} w_{ij} - \sum_{v_j \in A} w_{ij}$$

Sketch of KL-FM Algorithm

Variant of Fiduccia–Matheyses ('82)

Until No better partition is discovered
    Best Partition := Partition
    Compute all initial gains
    Until Termination criteria reached
        Select vertex to move (greedily)
        Perform move
        Update gains of all neighbors of moved vertex
        If Partition balanced & better than Best Partition **Then**
            Best Partition := Partition
    End Until
    Partition := Best Partition
End Until

Remarks:

- Original KL was based on swapping equal sets of nodes, we move one vertex at a time. We will have temporary imbalance.
- Some gains may be negative, but we still make the move since we may win more later.
  $\Rightarrow$ KL can climb out of local minima.
- The algorithm can be generalized to p partitions.

Overview of Multilevel Approach

KL is good locally, but not globally.
Idea: Do KL refinement on many different *scales* of the graph.
Similar in spirit to multigrid techniques.

(1) Construct sequence of smaller and smaller graphs
which preserve essential properties of original.

(2) Partition smallest graph (*coarse grid solver*).

(3) Move partition through graph sequence (*prolongation*),
periodically refining (*smoothening*).

The local refinement should be powerful on the coarse
levels since there are fewer local minima.

Construction of Smaller Graphs

Goal: Preserve essential properties of original graph.

1. Total weight of edges cut.
2. Total weight of vertices in each set.
3. Retain overall structure $\Longrightarrow$ coarsen uniformly

Two alternatives:

- Select independent set of vertices (coloring)
- Select independent set of edges (matching)

Most algorithms now use matching and edge contraction.

Fundamental Operation: Weighted edge contraction.

Merge two vertices joined by edge.
Add their vertex weights.
Add edge weights of any coincident edges.

- Properties:

  Preserves weighted balance and cut size.
  Preserves planarity, connectivity, etc.

- Note: Propagation of partition is trivial.

Partitioning Smallest Graph

- Must handle weighted graphs.

- Minimal impact on runtime & solution if
  Smallest graph much smaller than original.

- Random or greedy partition is fine if:
  Smallest graph is small enough, and
  Refinement is sufficiently powerful.

- Better to use spectral partitioning?

References

- Bui & Jones ('93) proposed a pure
  multilevel algorithm but **without weights**.

- Hendrickson & Leland ('93) independently proposed a
  multilevel algorithm **with weights**.

Software

- Chaco (Hendrickson & Leland, Sandia Labs)

- METIS (Karypis & Kumar, Univ. of Minnesota)