Approximation Algorithms

Multiway Cut and k-Cut

Kaouri Georgia

NTUA, Spring Semester 2009

Definitions

- A cut on an undirected, connected graph G=(V,E) with weights to edges w: $E \rightarrow R^+$, is defined by a partition of E into two sets, V' and E-V', and consists of all the edges that have one endpoint in each partition.
- Given terminals s,t in G, the cut defined by a partition that separates s from t is called an s-t cut.
- Multiway cut: Given a set of terminals $S=\{s_1,s_2,\ldots,s_k\}$, a multiway cut is a set of edges whose removal separates all terminals from each other.
- k-cut: A set of edges whose removal leaves k connected components is a k-cut.
- We are interested in the minimum weight version of these problems.

2-2/k approximation algorithm for minimum weight Multiway cut

- Algorithm:
  - For each $i=1,2,\ldots,k$, compute a minimum weight isolating cut for s_i, say C_i.
  - Output the union of these cuts, discarding the heaviest.

- The above algorithm is 2-2/k approximative.
- Let A be an optimal multiway cut in G. A is composed by k subsets, say $A_1, A_2, \ldots, A_k$, where A_i is the cut that separates the component containing s_i from the rest of the graph.
- Each edge of A has its two endpoints in two A_is. Therefore $\Sigma_{i=1}^{k} w(A_i) = 2w(A)$.
- A_i is an isolating cut for s_i, but C_i is the minimum, so $w(C_i) \leq w(A_i)$
- We discard the heaviest of the cuts C_i, therefore $w(C) \leq (1-1/k)\Sigma_{i=1}^{k} w(C_i)$. Hence:

$$w(C) \leq (1-1/k)\ \Sigma_{i=1}^{k} w(C_i) \leq (1-1/k)\ \Sigma_{i=1}^{k} w(A_i) = 2(1-1/k)\ w(A)$$

Tight example

- Consider a graph with 2k vertices, k of which form a k-cycle with edge weight equal to 1 and k terminals, each one connected to one of the vertices of the cycle with edges of weight $2-\varepsilon$ for a small $\varepsilon>0$.

- The algorithm computes a solution of weight $(k-1)(2-\varepsilon)$, while the optimal multiway cut has weight k.

Construction of a Gomory-Hu tree

- Consider a tree T with one node, the set S_0=V
- Select a set S_i, $|S_i|\geq 2$ and select 2 vertices u,v of S_i.
- Compute a minimum u-v cut in G', where G' is the graph obtained by G and collapsing each subtree of S_i into a single supernode. We compute the minimum cut between u and v in the new graph and obtain a partition Vn containing u and V_2 containing v.
- Graph T is modified by breaking S_i into two sets $S_{i1}=S_i \cap V_1$ and $S_{i2}=S_i \cap V_2$. We add an edge between them with cost the cost just calculated.
- We connect a subtree of T to S_{i1} if its supernode was in the same partition as u in the minimum cut, otherwise we connect it to S_{i2}.

Properties of a Gomory-Hu tree

- For each pair of vertices u,v in V, the weight of a minimum u-v cut in G is the same as that in T
- For each edge e in T, w'(e) is the weight of the cut associated with e in G.

- Lemma: Let S be the union of cuts in G associated with l edges of T. Then, the removal of S from G leaves the graph with at least l+1 components.

- Let $V_1, V_2, \ldots, V_{l+1}$ be the connected components which are left in T after removing l edges. For any u in V_i and v in V_j, we must have removed some edge in T that disconnects u and v. A cut is associated with this edge in G, which must disconnect u and v in G as well. Thus removing l edges in T, results in at least l+1 connected components in G.

(2-2/k) approximation algorithm

- Compute a Gomory-Hu tree T in G
- Output the union C of the k-1 lightest cuts of the n-1 cuts associated with edges of T in G
  - If more than k components are created, throw back some edges until there are k components.

- Let A be an optimal k-cut in G. As before, let the cuts $V_1, V_2, \ldots, V_{k+1}$ be the k components formed by removing A from G and A_i the cut separating V_i from the rest of the graph and $\Sigma_{i=1}^{k} w(A_i) = 2w(A)$.
- Assume A_k is the heaviest cut.
- Modify T by shrinking vertices corresponding to each V_i into a supernode and remove edges until the graph becomes a tree T'.
- Root T' at the supernode of V_k. Consider the edge (u_j, v_j) connecting a supernode V_i with its parent. These edges belonged in T, therefore $w'(u_i, v_i) \leq w(A_i)$. Thus

$$\Sigma_{i=1}^{k-1} w'(u_i, v_i) \leq \Sigma_{i=1}^{k-1} w(A_i) \leq 2(1-1/k)\, w(A)$$

Tight Example

- Similar as in multiway cut.

- 2k vertices k of which form a cycle with edge costs equal to 1 and k distant nodes (they are no longer called terminals) connected with one node of the cycle with an edge with cost $2-\varepsilon$.

- Using the Gomory-Hu algorithm a solution with cost $(k-1)(2-\varepsilon)$, whereas the optimal algorithm picks all edges with cost 1, i.e. te cost is k.

Thank you!