Cuts and Centers

Olli Pottonen
olli.pottonen@tkk.fi

February 15, 2008

Cuts: introduction

- Connected, undirected graph (V, E) with weight function w on edges
- Cut: edges between V' and $V \setminus V'$
- max—flow—min—cut theorem
- *Multiway cut*: given a set $\{s_1, \ldots, s_k\} \subseteq V$ of terminals, disconnect them from each other by removing a set of edges with minimum weight
- *Minimum k-cut*: Divide G into k connected components by removing a set of edges with minimum weight

Cuts: complexity

- Multiway cut is NP-hard for any $k \geq 3$
- Minimum k-cut solvable in $O(n^{k^2})$, NP-hard for arbitrary k
- Both approximable to factor $2 - 2/k$

Approximating multiway cut

- Algorithm:

  1. For $i = 1, \ldots, k$, compute a minimum weight isolating cut C_i for s_i
  2. Discard the heaviest cut and output union of the rest

- Let $A = \cup_i A_i$ be the optimum cut such that A_i isolates c_i. $\sum_{i=1}^{k} w(A_i) = 2w(A)$, since any $e \in A$ belongs to two cuts A_i, A_j. For any i, $w(C_i) \leq w(A_i)$. Discarding heaviest of C_i decreases the weight by factor $1 - 1/k$. Hence: approximating factor $2 - 2/k$.

Gomory-Hu trees

- Approximating k-cuts is more difficult

- Consider edge-weighted graph $G = (V, E, w)$. Tree $T = (V, E', w')$ is *Gomory-Hu tree*, if
  - for any $u, v \in V$, weights of minimum u—v cuts in G and T are equal
  - Any $e \in E'$ divides T into two components: $S, \bar{S}$. For any e, weight of the cut $(S, \bar{S})$ in G is equal to weight of e in T

- Constructing Gomory-Hu trees is an interesting problem, see exercises of Section 4.3 in Vazirani's book

Approximating minimum k-cut

- Algorithm:
  1. Compute Gomory-Hu tree T for G
  2. Output the cuts of G associated with $k-1$ lightest edges of T

- Approximating factor $2-2/k$ as shown here:

- Let $A = \cup_i A_i$ be the optimum k-cut, which divides V into $V_1, \ldots, V_k$. As before, $\sum_{i=1}^{k} w(A_i) = 2w(A)$. Let A_k be the heaviest cut. If, for $i = 1, \ldots, k-1$, we find edge of T with weight $\leq w(A_i)$, the result follows.

Proof continued

- Consider graph with V_i as vertices with edges of T connecting them. Discard edges until a tree remains. Let V_k be the root of the tree (recall that A_k was the heaviest cut). Let e_i be the vertex connecting V_i to its parent. Every e_i corresponds to a cut of G with weight $\leq w(A_i)$.

k-Center

- *Metric k-center*: Let $G = (V, E)$ be a complete undirected graph with metric edge costs, and k be a positive integer. For $v \in V, S \subseteq V$, let $\mathrm{connect}(v, S)$ be the cheapest edge $\{v, s\}$ for any $s \in S$. Find S with $|S| = k$ so as to minimize $\max_{v \in V} \mathrm{connect}(v, S)$

k-Center: inapproximability

- Assuming $P \neq NP$, no polynomial algorithm approximates metric k-center with factor < 2, and no polynomial algorithm approximates non-metric k-center with factor $< \alpha(k)$ for any computable α.

- Reduce dominating set to k-center: given G, set weight of each edge to 1, and add edges with weight 2 or $\alpha(k)$ until the graph is complete. If $\mathrm{dom}(G) \leq k$, the new graph has k-certer of cost 1, and otherwise it has optimum k-center of cost 2 or $\alpha(k)$.

- Factor 2 is achievable

Approximating k-center

- Given graph H, its *square* H^2 is such that $\{u, v\}$ is edge in H^2 fs a path of length at most 2 connects u and v in H

- Triangle inequality: $\max_{e \in E(H^2)} w(e) \leq 2 \max_{e \in E(H)} w(e)$

- Let G_i be a graph with i cheapest edges of G

- Task is to find minimum i such that $\mathrm{dom}(G_i) \leq k$. Let $OPT = \mathrm{cost}(e_i)$.

Approximating k-center

- Theorem: if I is independent set in H^2, $|I| \leq \mathrm{dom}(H)$.

- Algorithm:

  1. Construct $G_1^2, G_2^2, \ldots, G_m^2$
  2. For each G_i, construct maximal independent set I
  3. Return M_j with smallest j such that $|M_j| \leq k$

- Lemma: $\mathrm{cost}(e_j) \leq OPT$.

- Theorem: algorithm has approximating factor 2

Weighted w-center

(Vazirani calls these weighted k-centers)

- k-center: center consists of at most k nodes
- Weighted W-center: center has weight at most W
- Small modification of algorithm necessary: after constructing the maximal independent set M_j, replace each vertex with lightest neighbour
- Approximating factor 3