Cut Problems Reference

A graph $G = (V, E)$ is assumed undirected. $n = |V|$ and $m = |E|$. Edges may be weighted with capacities $c(e)$ for $e \in E$. We assume vertices are numbered and so speak of $i \in V$ for $i \in \{1, \ldots n\}$.

Given a set of vertices $S \subset V$, $\delta(S) = \{(i, j) \in E : i \in S, j \notin S\}$, that is, the edges that cross from S into $V \setminus S$.

A *path* is a seqence of edges connected two vertices. $\mathcal{P}$ will denote all paths in a graph, and $\mathcal{P}_{ij}$ all paths between vertices i and j. A *flow* on a graph is an assignment to the paths of G that respects the capacities, that is, if $f(p)$ is the flow assigned to path p, $\sum_{p \ni e} f(p) \leq c(e)$ for all $e \in E$. The *total flow* of a graph is $\sum_{\mathcal{P}} f(p)$.

Minimum Cut Linear Arrangement: Given a permutation σ on $\{1 \ldots n\}$, define $S_i = \{(k, j) \in E : \sigma(k) \leq i < \sigma(j)\}$. The LA problem is to find a σ minimizing $\max_i \sum_{e \in S_i} c(e)$. A ρ-approximation for Graph Bisection or Balanced Cut (see below) gives a $O(\rho \log n)$-approzimation to LA.

Graph Bisection: Find a partition of V into V_1 and V_2 with $|V_1| = \lfloor n/2 \rfloor$ to minimize $\sum_{e \in \delta(V_1)} c(e)$.

Balanced Cut: Find a partition of V into V_1 and V_2 with that $|V_i| < 2n/3$ for $i = 1, 2$ that minimizes $\sum_{e \in \delta(V_1)} c(e)$. Leighton and Rao found a $O(\log n)$ approximation algorithm.

Sparsest Cut: Find a subset of vertices $S \subset V$ minimizing the *sparsity ratio*

$$\rho(S) = \frac{\sum_{e \in \delta(S)} c(e)}{|S| \cdot |\overline{S}|}.$$

There is an alternative definition when k pairs of vertices (s_i, t_i) are distinguished as terminals, with each pair assigned a demand $d(i)$. In this case, let $I(S) = \{i : |S \cap \{s_i, t_i\}| = 1\}$ be the set of terminal pairs that are split by the cut. Then

$$\rho(S) = \frac{\sum_{e \in \delta(S)} c(e)}{\sum_{i \in I(S)} d(i)}.$$

The above cut problems are all NP-hard, and most (?) do not have approximations better than $\log n$ or $\sqrt{\log n}$. Almost nothing is known about hardness of approximation and this is a huge open question.

Minimum Multicut: Given k pairs of terminal vertices (s_i, t_i), a *multicut* is a subset of edges $F \subset E$ that disconnect the terminals, that is in $\overline{G} = (V, E \setminus F)$, no terminal pair is in the same connected component. The Minimum Multicut problem is to find a multicut F that minimizes $\sum_{e \in F} c(e)$.

Maximum Multicommodity Flow: Given a graph G, k pairs of terminal vertices (s_i, t_i), maximize the total flow between terminal vertices. This may be solved by linear programming. The dual LP is a fractional Minimum Multicut instance.

Maximum Concurrent Flow: Each terminal pair is associated with a demand $d(i)$, $i = 1, \ldots, k$. The MCF problem is to find the maximum λ so that there exists a flow f with $\sum_{p \in \mathcal{P}_i} f(p) = \lambda d(i)$ where $\mathcal{P}_i$ denotes all paths between terminal pair i. This may be solved by linear programming.

Linear Programming

A *linear program* (LP) is a problem of the following form. Given a matrix A, a *constraint vector* b, and an *objective vector* c, find a vector x that

$$\text{minimizes} \quad \langle c, x \rangle \tag{1}$$

subject to

$$Ax \geq b, \text{ and} \tag{2}$$

$$x \geq 0. \tag{3}$$

Here the inequalities are interpreted componentwise. A *feasible solution* is an x that satisfies (2–3) but may not be a maximizer. An *integer program* (IP) replaces (3) with the condition $x \in \{0, 1\}$ or $x \in \mathbb{Z}$. The direction of the inequality $Ax \leq b$ and minimizing versus maximizing the objective function are sometimes switched. The *dual* of the above *primal* linear program is to find a vector y that

$$\text{maximizes} \quad \langle b, y \rangle \tag{4}$$

subject to

$$A^t y \leq c, \text{ and} \tag{5}$$

$$y \geq 0. \tag{6}$$

The famous *min-max* theorem states that if x and y are feasible solutions to the primal and dual LPs, respectively, then

1. $\langle c, x \rangle \geq \langle b, y \rangle$ (*weak duality*), and
2. Equality is achieved if and only if x and y are optimal (*strong duality*).

The LP relaxation for MINIMUM MULTICUT and the dual MAXIMUM MULTICOMMODITY FLOW problem are given below. $G = (V, E)$ is an undirected graph with constraints $c(e)$ on each edge e. k pairs $\{(s_i, t_i)\}_{i=1}^k$ are given as terminals. $\mathcal{P}_i$ denotes the set of paths between s_i and t_i and $\mathcal{P}_i(e) = \{P \in \mathcal{P}_i : e \in P\}$.

MINIMUM MULTICUT (Primal)

minimize $\quad \sum_{e \in E} c(e)x(e)$

subject to

$$\sum_{e \in \mathcal{P}} x(e) \geq 1, \quad \text{for each } P \in \mathcal{P}_i,\ i = 1, \ldots, k$$
$$x(e) \geq 0 \quad \text{for each } e \in E$$

MAXIMUM MULTICOMMODITY FLOW (Dual)

maximize $\quad \sum_{i=1}^k \sum_{P \in \mathcal{P}_i} f(P)$

subject to

$$\sum_{i=1}^k \sum_{e \in \mathcal{P}_i(e)} f(P) \leq c(e), \quad \text{for each } e \in E$$
$$f(e) \geq 0 \quad \text{for each } P \in \mathcal{P}_i,\ i = 1, \ldots, k$$