Chapter 8: Cuts and Metrics

(cp. Williamson & Shmoys, Chapter 8)

The Multiway Cut Problem

Given: Graph $G = (V, E)$; costs $c_e \geq 0$, $e \in E$; k terminals $s_1, \ldots, s_k \in V$.

Task: Find $F \subseteq E$ of minimum cost $c(F)$ disconnecting all s_i, $i = 1, \ldots, k$.

Simple Algorithm

1. for $i = 1, \ldots, k$ find min-cost $F_i \subseteq E$ isolating s_i from all s_j, $j \neq i$;
2. return $F := \bigcup_{i=1}^{k} F_i$;

Theorem 8.1.
The algorithm is a 2-approximation algorithm for multiway cuts.

Proof:. . . □

Corollary 8.2.
Taking only the cheapest $k - 1$ cuts F_i yields a $(2 - \frac{2}{k})$-approximation algorithm.

Proof:. . . □

IP Formulation of Multiway Cut Problem

In an optimal solution F, subgraph $(V, E \setminus F)$ has exactly k connected components C_i, with $s_i \in C_i$, $i = 1, \ldots, k$.

For $u \in V$ and $e \in E$ introduce decision variables

$$x_u^i = \begin{cases} 1 & \text{if } u \in C_i, \\ 0 & \text{otherwise,} \end{cases} \qquad z_e^i = \begin{cases} 1 & \text{if } e \in \delta(C_i), \\ 0 & \text{otherwise.} \end{cases}$$

IP formulation:

$$\begin{aligned} \min \quad & \frac{1}{2} \sum_{e \in E} c_e \sum_{i=1}^{k} z_e^i \\ \text{s.t.} \quad & \sum_{i=1}^{k} x_u^i = 1 && \text{for all } u \in V, \\ & z_e^i \geq |x_u^i - x_v^i| && \text{for all } e = uv \in E, \\ & x_{s_i}^i = 1 && \text{for all } i = 1, \ldots, k, \\ & x_u^i \in \{0, 1\} && \text{for all } u \in V,\ i = 1, \ldots, k. \end{aligned}$$

LP Relaxation

In an optimal IP solution $z_e^i = |x_u^i - x_v^i|$ and the objective function value is

$$\frac{1}{2} \sum_{e = uv \in E} c_e \cdot \|x_u - x_v\|_1$$

with $x_u \in \Delta_k := \left\{ x \in \mathbb{R}^k_{\geq 0} \mid \sum_{i=1}^{k} x^i = 1 \right\}$ (k-simplex).

LP relaxation:

$$\begin{aligned} \min \quad & \frac{1}{2} \sum_{e = uv \in E} c_e \cdot \|x_u - x_v\|_1 \\ \text{s.t.} \quad & x_{s_i} = e_i && \text{for all } i = 1, \ldots, k, \\ & x_u \in \Delta_k && \text{for all } u \in V. \end{aligned}$$

LP Rounding Algorithm

For $i = 1, \dots, k$ and $0 \le r < 1$ let

$$B(e_i, r) := \left\{ u \in V \,\middle|\, \tfrac{1}{2}\|e_i - x_u\|_1 \le r \right\} \ .$$

LP Rounding Algorithm

1. compute optimal solution to LP relaxation;
2. set $C_i := \emptyset$ for all $i = 1, \dots, k$; set $X := \emptyset$; (nodes assigned already)
3. draw $r \in (0, 1)$ uniformly at random;
4. pick random permutation π of $\{1, \dots, k\}$;
5. for $i = 1, \dots, k-1$ set $C_{\pi(i)} := B(e_{\pi(i)}, r) \setminus X$ and $X := X \cup C_{\pi(i)}$;
6. set $C_{\pi(k)} := V \setminus X$ and return $F := \bigcup_{i=1}^{k} \delta(C_i)$;

Theorem 8.3.
The algorithm is a randomized 3/2-approximation algorithm.

Proof:. . . □

The Multicut Problem

Given: Graph $G = (V, E)$ with edge costs $c_e \ge 0$, $e \in E$; k source-sink pairs $(s_1, t_1), \dots, (s_k, t_k) \in V \times V$.

Task: Find $F \subseteq E$ of minimum cost $c(F)$ whose removal disconnects s_i from t_i, $i = 1, \dots, k$.

IP formulation: Let $\mathcal{P}_i$ denote the set of all s_i-t_i-paths

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e \cdot x_e \\ \text{s.t.} \quad & \sum_{e \in P} x_e \ge 1 && \text{for all } P \in \mathcal{P}_i,\ i = 1, \dots, k, \\ & x_e \in \{0, 1\} && \text{for all } e \in E. \end{aligned}$$

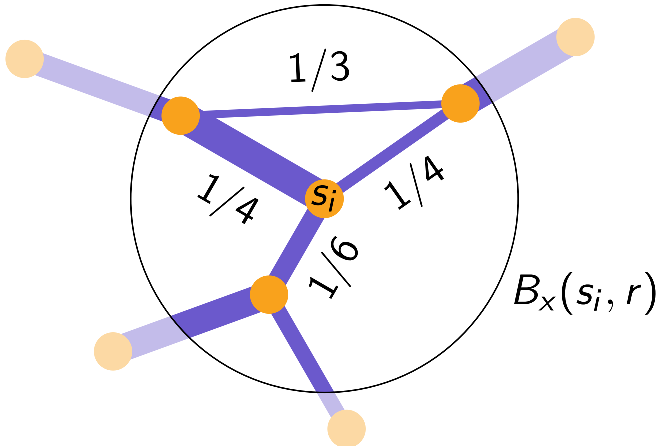
LP relaxation: replace $x_e \in \{0, 1\}$ with $x_e \ge 0$ for all $e \in E$.

- Separation problem consists of k shortest path problems.
- Thus, LP relaxation can be solved in polynomial time.

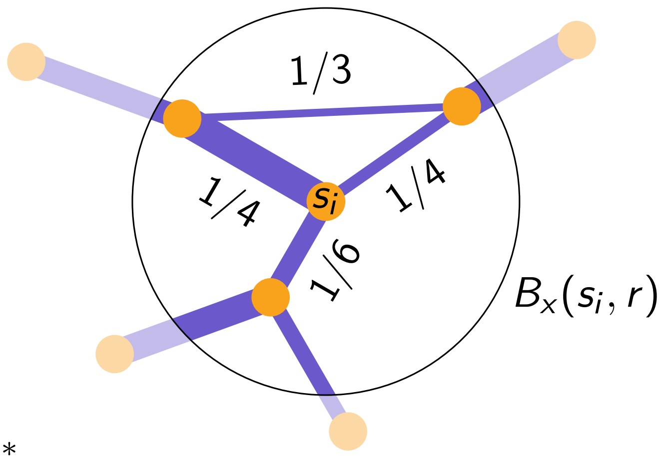
Interpretation of LP Solution

Let x be an optimal LP solution.

- Interpret x_e as the length of edge $e \in E$.
- Let $d_x(u, v) :=$ length of shortest u-v-path w.r.t. edge lengths x_e.
- Notice that $d_x(s_i, t_i) \geq 1$ for all $i = 1, \ldots, k$.
- Let $B_x(s_i, r) := \{v \in V \mid d_x(s_i, v) \leq r\}$ (ball of radius r around s_i).
- Interpret $c_e \cdot x_e$ as the volume of edge $e \in E$.
- LP yields minimum total volume V^* with $d_x(s_i, t_i) \geq 1$, $i = 1, \ldots, k$.



Cut Capacity vs. Volume



$$V_x(s_i, r) := \frac{V^*}{k} + \sum_{\substack{uv \in E: \\ u,v \in B_x(s_i,r)}} c_{uv} \cdot x_{uv} + \sum_{\substack{uv \in E: \\ u \in B_x(s_i,r) \\ v \notin B_x(s_i,r)}} c_{uv} \cdot (r - d_x(s_i, u))$$

Lemma 8.4.

For feasible LP sol. x and s_i, one can efficiently find radius $r < 1/2$ with

$$c(\delta(B_x(s_i, r))) \leq (2 \ln(k + 1)) \cdot V_x(s_i, r) \ .$$

Proof:. . . □

LP Rounding Algorithm

1. compute optimal solution x to LP relaxation; set $F := \emptyset$;
2. for $i = 1, \ldots, k$, if s_i and t_i are connected in $(V, E \setminus F)$, then
3. choose radius r around s_i as in Lemma 8.4;
4. set $F := F \cup \delta\big(B_x(s_i, r)\big)$;
5. set $V := V \setminus B_x(s_i, r)$ and $E := E \setminus \delta\big(B_x(s_i, r)\big)$;

Theorem 8.5.

This is a $\big(4\ln(k+1)\big)$-approximation algorithm for the multicut problem.

Proof:. . . □

Theorem 8.6.

Assuming the *Unique Games Conjecture*, there is no constant-factor approximation algorithm for the multicut problem, unless $P = NP$. □