

## 18 Cuts & Metrics

### Shortest Path

$$\begin{array}{ll} \min & \sum_e c(e)x_e \\ \text{s.t.} & \forall S \in \mathcal{S} \quad \sum_{e \in \delta(S)} x_e \geq 1 \\ & \forall e \in E \quad x_e \in \{0, 1\} \end{array}$$

$\mathcal{S}$  is the set of subsets that separate  $s$  from  $t$ .

#### The Dual:

$$\begin{array}{ll} \max & \sum_S y_S \\ \text{s.t.} & \forall e \in E \quad \sum_{S: e \in \delta(S)} y_S \leq c(e) \\ & \forall S \in \mathcal{S} \quad y_S \geq 0 \end{array}$$

The **Separation Problem** for the Shortest Path LP is the Minimum Cut Problem.

## 18 Cuts & Metrics

### Minimum Cut

$$\begin{array}{ll} \min & \sum_e c(e)x_e \\ \text{s.t.} & \forall P \in \mathcal{P} \quad \sum_{e \in P} x_e \geq 1 \\ & \forall e \in E \quad x_e \in \{0, 1\} \end{array}$$

$\mathcal{P}$  is the set of path that connect  $s$  and  $t$ .

#### The Dual:

$$\begin{array}{ll} \max & \sum_P y_P \\ \text{s.t.} & \forall e \in E \quad \sum_{P: e \in P} y_P \leq c(e) \\ & \forall P \in \mathcal{P} \quad y_P \geq 0 \end{array}$$

The **Separation Problem** for the Minimum Cut LP is the Shortest Path Problem.

## 18 Cuts & Metrics

### Observations:

Suppose that  $\ell_e$ -values are solution to Minimum Cut LP.

- ▶ We can view  $\ell_e$  as defining the **length** of an edge.
- ▶ Define  $d(u, v) = \min_{\text{path } P \text{ btw. } u \text{ and } v} \sum_{e \in P} \ell_e$  as the **Shortest Path Metric** induced by  $\ell_e$ .
- ▶ We have  $d(u, v) = \ell_e$  for every edge  $e = (u, v)$ , as otw. we could reduce  $\ell_e$  without affecting the distance between  $s$  and  $t$ .

#### Remark for bean-counters:

$d$  is not a metric on  $V$  but a semimetric as two nodes  $u$  and  $v$  could have distance zero.

### How do we round the LP?

- ▶ Let  $B(s, r)$  be the ball of radius  $r$  around  $s$  (w.r.t. metric  $d$ ).  
Formally:

$$B = \{v \in V \mid d(s, v) \leq r\}$$

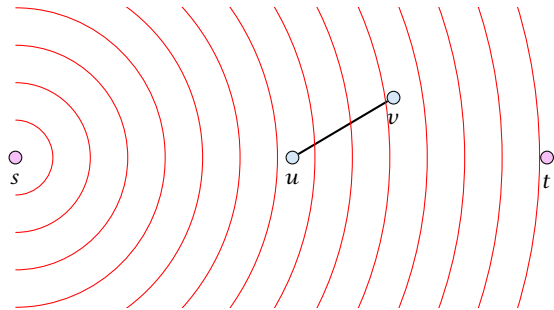
- ▶ For  $0 \leq r < 1$ ,  $B(s, r)$  is an  $s$ - $t$ -cut.

Which value of  $r$  should we choose? **choose randomly!!!**

Formally:

choose  $r$  u.a.r. (**uniformly at random**) from interval  $[0, 1)$

What is the probability that an edge  $(u, v)$  is in the cut?



► assume wlog.  $d(s, u) \leq d(s, v)$

$$\Pr[e \text{ is cut}] = \Pr[r \in [d(s, u), d(s, v))] \leq \frac{d(s, v) - d(s, u)}{1 - 0} \leq \ell_e$$

What is the expected size of a cut?

$$\begin{aligned} E[\text{size of cut}] &= E\left[\sum_e c(e) \Pr[e \text{ is cut}]\right] \\ &\leq \sum_e c(e) \ell_e \end{aligned}$$

On the other hand:

$$\sum_e c(e) \ell_e \leq \text{size of mincut}$$

as the  $\ell_e$  are the solution to the Mincut LP relaxation.

Hence, our rounding gives an optimal solution.

### Minimum Multicut:

Given a graph  $G = (V, E)$ , together with source-target pairs  $s_i, t_i$ ,  $i = 1, \dots, k$ , and a capacity function  $c : E \rightarrow \mathbb{R}^+$  on the edges. Find a subset  $F \subseteq E$  of the edges such that all  $s_i-t_i$  pairs lie in different components in  $G = (V, E \setminus F)$ .

$$\begin{array}{ll} \min & \sum_e c(e) \ell_e \\ \text{s.t.} & \forall P \in \mathcal{P}_i \text{ for some } i \quad \sum_{e \in P} \ell_e \geq 1 \\ & \forall e \in E \quad \ell_e \in \{0, 1\} \end{array}$$

Here  $\mathcal{P}_i$  contains all path  $P$  between  $s_i$  and  $t_i$ .

Re-using the analysis for the single-commodity case is difficult.

$$\Pr[e \text{ is cut}] \leq ?$$

- If for some  $R$  the balls  $B(s_i, R)$  are disjoint between different sources, we get a  $1/R$  approximation.
- However, this cannot be guaranteed.

- ▶ Assume for simplicity that all edge-length  $\ell_e$  are multiples of  $\delta \ll 1$ .
- ▶ Replace the graph  $G$  by a graph  $G'$ , where an edge of length  $\ell_e$  is replaced by  $\ell_e/\delta$  edges of length  $\delta$ .
- ▶ Let  $B(s_i, z)$  be the ball in  $G'$  that contains nodes  $v$  with distance  $d(s_i, v) \leq z\delta$ .

#### Algorithm 1 RegionGrowing( $s_i, p$ )

```

1:  $z \leftarrow 0$ 
2: repeat
3:   flip a coin ( $\Pr[\text{heads}] = p$ )
4:    $z \leftarrow z + 1$ 
5: until heads
6: return  $B(s_i, z)$ 

```

#### Algorithm 1 Multicut( $G'$ )

```

1: while  $\exists s_i-t_i$  pair in  $G'$  do
2:    $C \leftarrow \text{RegionGrowing}(s_i, p)$ 
3:    $G' = G' \setminus C$  // cuts edges leaving C
4: return  $B(s_i, z)$ 

```

- ▶ probability of cutting an edge is only  $p$
- ▶ a source either does not reach an edge during Region Growing; then it is not cut
- ▶ if it reaches the edge then it either **cuts** the edge or **protects** the edge from being cut by other sources
- ▶ if we choose  $p = \delta$  the probability of cutting an edge is only its LP-value; our expected cost are at most **OPT**.

#### Problem:

We may not cut all source-target pairs.

A component that we remove may contain an  $s_i-t_i$  pair.

If we ensure that we cut before reaching radius  $1/2$  we are in good shape.

- ▶ choose  $p = 6 \ln k \cdot \delta$
- ▶ we make  $\frac{1}{2\delta}$  trials before reaching radius  $1/2$ .
- ▶ we say a Region Growing is not successful if it does not terminate before reaching radius  $1/2$ .

$$\Pr[\text{not successful}] \leq (1-p)^{\frac{1}{2\delta}} = \left( (1-p)^{1/p} \right)^{\frac{p}{2\delta}} \leq e^{-\frac{p}{2\delta}} \leq \frac{1}{k^3}$$

- ▶ Hence,

$$\Pr[\exists i \text{ that is not successful}] \leq \frac{1}{k^2}$$

## What is expected cost?

$$E[\text{cutsize}] = \Pr[\text{success}] \cdot E[\text{cutsize} \mid \text{success}] \\ + \Pr[\text{no success}] \cdot E[\text{cutsize} \mid \text{no success}]$$

$$E[\text{cutsize} \mid \text{succ.}] = \frac{E[\text{cutsize}] - \Pr[\text{no succ.}] \cdot E[\text{cutsize} \mid \text{no succ.}]}{\Pr[\text{success}]} \\ \leq \frac{E[\text{cutsize}]}{\Pr[\text{success}]} \leq \frac{1}{1 - \frac{1}{k^2}} 6 \ln k \cdot \text{OPT} \leq 8 \ln k \cdot \text{OPT}$$

Note: **success** means all source-target pairs separated

We assume  $k \geq 2$ .

If we are not successful we simply perform a trivial  $k$ -approximation.

This only increases the expected cost by at most  $\frac{1}{k^2} \cdot k \text{OPT} \leq \text{OPT}/k$ .

Hence, our final cost is  $\mathcal{O}(\ln k) \cdot \text{OPT}$  in expectation.