First the general problem:
Input: A weighted graph $G = (V, E)$ with non-negative integer edge capacities c_e for each edge $e \in E$. Distinguished vertices $s_i, t_i \in V$ for $i = 1, \ldots, k$.
Output: Subset $E' \subseteq E$ such that removal of the edges in E' from G disconnects each s_i from the corresponding t_i, and $\sum_{e \in E'} c_e$ is minimized.

The problem is a generalization of the $s - t$ mincut problem, which can be obtained by selecting $k = 1$ in the above. It turns out to be NP-complete, even for the special case in which G is a tree.

In this lecture we give a 2 approximation for this special case of trees.

1 IP Formulation and LP relaxation

We have a 0-1 integer variable x_e for each edge. We will have $x_e = 1$ denote $e \in E'$, and $x_e = 0$ otherwise. Since we are considering trees, there is a unique path P_i from each s_i to the corresponding t_i. We require that at least one edge on this path be selected in E'. Thus:

$$\forall i : \quad \sum_{e \in P_i} x_e \geq 1$$

Our objective is:

$$\text{minimize} \quad z = \sum_e c_e x_e$$

The integrality condition is:

$$\forall i : x_i \in \{0, 1\}$$

For the LP relaxation, instead of the condition above, we have:

$$\forall i : x_i \in [0, 1]$$

Let x^*, x' denote the optimal solutions to the IP, LP relaxation respectively, and z^*, z' the respective optimal objective function values.

2 Algorithm

The first step is to solve the LP relaxation and find x', z'. While we do not know x^*, z^*, we know that z' is a lower bound on z^*, i.e. $z' \leq z^*$.

Next we will somehow round x' to a 0-1 solution x, with objective value z.

2.1 Randomized rounding

An interesting way to round x' is to use randomization. Suppose we set $x_e = 1$ with probability x'_e, then we get[1] the expected value of the objective $E[z] = \sum_e E[x_e] = \sum_e x'_e = z'$. This is good; we are essentially paying no penalty for the rounding! Further, by linearity of expectation[2], we have, $E[\sum_{e \in P_i} x_e] = \sum_{e \in P_i} E[x_e] = \sum_{e \in P_i} x'_e \geq 1$. Thus, at least in expectation, the constraints will also be satisfied! This is of course not enough – we want to be sure that *all* the constraints are satisfied, or at least satisfied with high probability.

2.2 A variation

The algorithm is as follows:

1. Root the tree arbitrarily.
2. Consider x'_e to be the length of each edge e.
3. Pick a random number α uniformly at random from $[0, 0.5]$.
4. Consider each root-leaf path, and on this path, place marks at distances $\alpha + k/2$ for all possible integers k.
5. For edges e which receive a mark, set $x_e = 1$, otherwise set $x_e = 0$.

Clearly this runs in polynomial time.

The right visualization of this algorithm is: think of each edge e as a string of length x'_e. The tree hangs from the root. We are going to draw horizontal lines $1/2$ unit apart over the hanging tree, the only randomness is in the vertical offset α that we choose for the lines.

Notice that whether or not two edges get marked is not independent anymore (as opposed to our discussion in Section 2.1, but as you will see the lack of independence works in our favour.

Lemma 1 $\forall i : \quad \sum_{e \in P_i} x_e \geq 1.$

Proof: The path P_i maybe viewed as two segments: a path $s_i - m_i$ and a path $m_i - t_i$, where m_i is the least common ancestor of s_i, t_i in the tree. Now, we know that $\sum_{e \in P_i} x'_i \geq 1$, thus the length of one of these two paths is at least $1/2$. Without loss of generality, say $s_i - m_i$. But now note that marks are placed a distance $1/2$ apart. Thus this path must get at least 1 mark, independent of the choice of α. Thus some edge e on the path $s_i - m_i$, and hence on the path $s_i - t_i$ must have $x_e = 1$. ■

[1] By definition $E[x] = \sum_i i \cdot Pr[x = i]$, if x only takes values 0,1, we get $E[x] = Pr[x = 1]$.

[2] $E[a_1 x_1 + \cdots + a_m x_m] = a_1 E[x_1] + \cdots + a_m E[x_m]$

Lemma 2 $E[z] \leq 2z'$

Proof: We first prove that $Pr[x_e = 1] \leq 2x'_e$. If e has length x'_e at least $1/2$, then we know by the above argument that it must receive a mark, thus $Pr[x_e = 1] = 1 \leq 2x'_e$. Suppose that $x'_e < 1/2$. Let the closer end point of e be at distance p from the root, and the further endpoint at distance q from the root. For any real r, define $r \bmod 1/2$ to be r' where $0 \leq r' < 1/2$ and $r = r' + k/2$ for some integer k. Suppose $p' = p \bmod 1/2$, $q' = q \bmod 1/2$, and suppose $p' < q'$. Then clearly the e will be marked if $\alpha \in [p', q']$. But since α is chosen uniformly from $[0, 1/2]$ the probability that it lies in $[p', q']$ is $q' - p'/(1/2) = 2(q' - p') = 2x'_e$. A similar argument works for the case $q' \leq p'$.

Now $E[z] = E[\sum_e c_e x_e] = \sum_e c_e E[x_e] = \sum_e c_e Pr[x_e = 1] \leq \sum_e c_e(2x'_e) = 2z'$. ■

The algorithm as we have given will always give a feasible solution; the objective value will depend upon the random numbers chosen, and its average will be at most $2z'$. Thus on the average we will get a good objective value. This is itself very good, but we can do better.

2.3 A deterministic algorithm

The key to what follows is in the following puzzle. Suppose you are going on a drive which takes you through several states. Each state has a different price for petrol, and let us say that throughout the drive your car gives the same mileage. You want to minimize the amount you spend on the petrol, but you also want to keep things simple while on the road: basically you want to fill petrol when the tank is almost empty (let us say that the road is full of petrol pumps and you can fill anytime), and when you fill, you fill the tank fully. You do not know how much petrol you had when you started – say it is some random quantity between empty tank and full. What is your expected spending? What is the minimum spending?

We know from the analysis of the last section that averaged over all choices of α, the value of z is at most $2z'$. Clearly, there must exist a value of α for which z is guaranteed to be at most $2z'$. It turns out that we can find that value of α for which z is minimum (and hence no larger than $2z'$ in polynomial time).

Consider the interval $I = [0, 1/2]$. Let p_e, q_e denote the distance of the closer and farther endpoints of e from the root. Let $p'_e = p_e \bmod 1/2$ and $q'_e = q_e \bmod 1/2$. Now mark p'_e, q'_e on I for all e. Thus I receives $2m$ marks where m is the number of edges, and thus I is divided into $2m + 1$ subintervals.

The key observation is: all choices of α within any single subinterval give the same value for z. Thus we only need to consider $2m + 1$ different choices and pick the one which gives the smallest z value.

Exercise

Let G denote the star graph on 4 vertices, i.e. $V = \{1, 2, 3, 4\}$, $E = \{(1, 2), (1, 3), (1, 4)\}$. Let $s_1, s_2, s_3 = 2, 3, 4$ respectively, and $t_1, t_2, t_3 = 3, 4, 2$ respectively.

Solve this multicut problem using the algorithm above. How many different values of *alpha* do you need to consider?