**CS602 Applied Algorithms** **Jul 2023**

# Assignment 2

*Total Marks: 40* *Deadline: Nov 10, 14:00*

Note: The assignment needs to be done individually. Any kind of discussion with other students is not allowed. If you feel the need to discuss anything, you can reach out to the instructor. You can directly use any result proved in the class. You can use other sources (book/internet), but you should give a reference and specify for which part it has been used.

**Que 1 [10 marks].** Recall that in the Steiner tree problem we are given a edge weighted graph $G$ with a set of terminal vertices and we have to compute a minimum weight subgraph where all terminal vertices are connected with each other. Consider the following algorithm for minimum weight Steiner tree problem.

Compute shortest path for each pair of terminals. Construct a complete graph $H$ on the terminal vertices where edge weights are the computed shortest path lengths between pairs of terminals. Construct a minimum spanning tree $T$ for the graph $H$. Take the union of the paths in $G$ corresponding to the MST edges. Remove any redundant edges in any order.

Assume that the number of terminals given is 3. Prove that the above algorithm gives a 4/3-approximate Steiner tree.

Hint: consider the optimal Steiner tree on 3 terminals. Try to upper bound this in terms of sum of paths between two pairs of terminals (appropriately chosen), with a multiplicative factor. Upper bound the sum by weight of a tree in the graph $H$.

**Ans 1.** Let the three terminals be $a, b, c$. Let $d_{a,b}$ be the weight of the shortest path between $a$ and $b$ in $G$. Similarly, define $d_{a,c}$ and $d_{b,c}$. The weight of the MST in $H$ is simply the sum of two smallest numbers among these three. Equivalently, it is

$$\begin{aligned} w(\text{MST in } H) = d_{a,b} + d_{b,c} + d_{a,c} - \max\{d_{a,b}, d_{b,c}, d_{a,c}\} &\leq d_{a,b} + d_{b,c} + d_{a,c} - 1/3(d_{a,b} + d_{b,c} + d_{a,c}) \\ &= 2/3(d_{a,b} + d_{b,c} + d_{a,c}). \end{aligned}$$

Rearranging,

$$d_{a,b} + d_{b,c} + d_{a,c} \geq 3/2 \times w(\text{MST in } H). \tag{1}$$

The inequality follows because of maximum of three numbers is at least the average of three numbers.

Let the optimal Steiner tree in $G$ be $T^*$. Let $P_{a,c}$ be the unique path from $a$ to $c$ in $T^*$, and let $P_{a,b}$ and $P_{b,c}$ be defined similarly. Four different types of Steiner trees are shown in Figure 1.

Observe that, in all scenarios,

$$2w(T^*) = w(P_{a,c}) + w(P_{b,c}) + w(P_{a,b}).$$

This is because, in the right hand sum every edge of $T^*$ is counted twice. By the definition of shortest path and using equation (1), we can write

$$\begin{aligned} 2w(T^*) &= w(P_{a,c}) + w(P_{b,c}) + w(P_{a,b}) \\ &\geq d_{a,b} + d_{b,c} + d_{a,c} \\ &\geq 3/2 \times w(\text{MST in } H). \end{aligned}$$

We get that

$$w(\text{MST in } H) \leq 4/3 \times w(T^*).$$

The Steiner tree output by the algorithm is a subset of the union of the paths corresponding to edges in the MST in $H$. Hence, the weight of the the Steiner tree given by the algorithm is at most

$$w(\text{MST in } H) \leq 4/3 \times w(T^*).$$

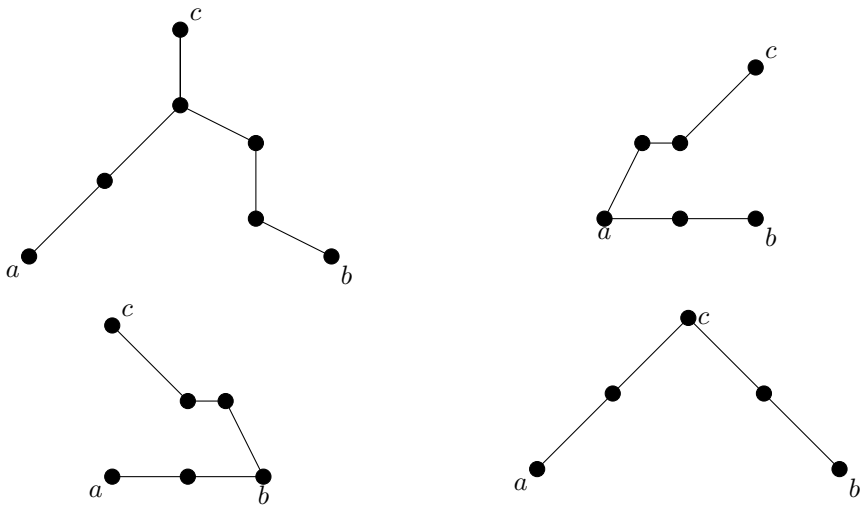


Figure 1: Different types of Steiner trees between three terminals.

**Que 2 [10 marks].** Consider the minimum weight Steiner forest problem with terminal pairs $(s_1, t_1), (s_2, t_2)$. That is, we want a Steiner forest that connects $s_1$ with $t_1$ and $s_2$ with $t_2$. Recall the primal dual algorithm for the Steiner forest problem discussed in the class. Show that the algorithm gives a solution with approximation factor 3/2.

Hint: look closely into the analysis done for the approximation factor

**Ans 2.** Recall the analysis where we compared the change in the primal objective and the change in the dual objective in any iteration. Let the final Steiner forest output by the algorithm is $F$. Let $C_1, C_2, \ldots, C_q$ be the sets of vertices whose corresponding dual variables $y_{C_1}, y_{C_2}, \ldots, y_{C_q}$ were increased in a particular iteration. Suppose all dual variables were increased by a quantity $\epsilon$ in this iteration. Hence, the total change in the dual objective $(\sum_C y_C)$ in this iteration is $q\epsilon$.

The final primal cost is

$$\sum_{e \in F} w_e = \sum_{e \in F} \sum_{C : e \in \delta(C)} y_C.$$

This is because only tight edges are selected in the Steiner forest. Let us see how much change was caused in the primal cost in the particular iteration. Recall a few facts from the algorithm. Whenever we increase the dual variable $y_{C_i}$, the set $C_i$ is a connected component in the subgraph constructed so far. And hence, no edges inside $C_i$ are added in any subsequent iterations. Since at the end we have a forest, we can say that there is a unique path in $F$ that connects $C_i$ to $C_j$. Suppose $C_i$ and $C_j$ are connected via a path in $F$, which does not pass through any other component $C_p$. The edges on this path will see a total increase of $2\epsilon$ – $\epsilon$ on the edge which leaves $C_i$ and $\epsilon$ on the edge which enters $C_j$.

How many such paths between pairs of components are there? The answer is $q - 1$. To see this, imagine each component $C_i$ as a vertex, then $F$ induces a forest on these. And there are at most $q - 1$ edges in a forest with $q$ vertices. Hence, the total increase in the primal objective in the particular iteration is at most $2(q - 1)\epsilon$.

Since this holds for every iteration, we can upper bound the ratio of the primal objective and the dual objective by

$$\max_q 2(q-1)/q,$$

where the possible values of $q$ are the number of dual variables increased in any iteration.

We are given that the number of terminals is 4. Recall that each component $C_i$ must separate some terminal pair and moreover, the components $C_i$'s are disjoint. Hence, $q \leq 4$. Thus, the approximation ratio we get is

$$\max\{2/2, 2 \times 2/3, 2 \times 3/4\} = 3/2.$$

**Que 3 [8+2 marks].** Consider a variant of the vertex cover problem. Here, we have weights on both edges and vertices. We want to maximize the weight of edges covered and minimize the weight of selected vertices. You can think of this as getting a reward for covering an edge and also getting a reward for not selecting a vertex.

Formally, let $G(V, E)$ be a graph. Let $P_1, P_2, \ldots, P_n$ be the weights of the vertices. Let $\{Q_{i,j} : (i, j) \in E\}$ be the weights on the edges. The goal is select a subset $S \subseteq V$ of vertices so as to maximize the below expression.

$$\sum_{i \notin S} P_i + \sum_{\substack{(i,j) \in E: \\ i \in S \\ \text{or } j \in S}} Q_{i,j}.$$

We write the following integer program.

$$\begin{aligned} \max \sum_{i=1}^{n} (1 - y_i) P_i + \sum_{(i,j) \in E} z_{i,j} Q_{i,j} \text{ subject to.} \\ y_i \in \{0, 1\} \text{ for } 1 \leq i \leq n \\ z_{i,j} \in \{0, 1\} \text{ for } (i, j) \in E \\ z_{i,j} \leq y_i + y_j \text{ for } (i, j) \in E \end{aligned}$$

First relax the integer constraint and write a linear program. We find an optimal solution for that LP using some LP solver. Let the optimal solution be $(y^*, z^*)$. The optimal solution can be fractional. We convert it to a Boolean solution with the following rounding scheme: For each $1 \leq i \leq n$, select vertex $i$ with probability $(1 - \lambda) + \lambda y_i^*$.

Show that the approximation factor (i.e., rounded solution cost divided by LP optimal) is at least $\min\{\lambda, 1 - \lambda^2/4\}$.

What should be the choice of $\lambda$ that gives the best approximation factor. What is the approximation factor for this choice.

**Ans 3.** Relaxing the integer constraints.

$$\begin{aligned} \max \sum_{i=1}^{n} (1 - y_i) P_i + \sum_{(i,j) \in E} z_{i,j} Q_{i,j} \text{ subject to.} \\ 0 \leq y_i \leq 1 \text{ for } 1 \leq i \leq n \\ 0 \leq z_{i,j} \leq 1 \text{ for } (i, j) \in E \\ z_{i,j} \leq y_i + y_j \text{ for } (i, j) \in E \end{aligned}$$

Let the optimal solution be $(y^*, z^*)$. The LP optimal value will be

$$\text{LP-OPT} = \sum_{i=1}^{n} (1 - y_i^*) P_i + \sum_{(i,j) \in E} z_{i,j}^* Q_{i,j}.$$

Let us denote the two terms above as LP-OPT-V and LP-OPT-E. We want to compare this with the solution obtained after rounding. The solution computed is randomized, and thus, we will consider the expected value. The expected value will be (using linearity of expectation)

$$\sum_{i=1}^{n} \Pr[\text{vertex } i \text{ is not selected}] \cdot P_i + \sum_{(i,j) \in E} \Pr[\text{edge } (i, j) \text{ is covered}] \cdot Q_{i,j}.$$

Let us denote the above two terms as ALG-V and ALG-E. According to the rounding scheme, probability that vertex $i$ is not selected is $\lambda(1-y_i^*)$. Hence,

$$\text{ALG-V} = \sum_{i=1}^{n} \lambda(1-y_i^*) \cdot P_i = \lambda \times \text{LP-OPT-V}. \tag{2}$$

Moving on to the second part, probability that the edge $(i,j)$ is covered is equal to the probability that at least one of the two vertices $i$ and $j$ is selected. This probability is

$$1 - \lambda(1-y_i^*) \times \lambda(1-y_j^*).$$

Thus,

$$\text{ALG-E} = \sum_{(i,j)\in E} \left(1 - \lambda(1-y_i^*) \times \lambda(1-y_j^*)\right) \cdot Q_{i,j}$$

To compare ALG-E with LP-OPT-E, we need to compare $z_{i,j}^*$ with $1-\lambda^2(1-y_i^*)\times(1-y_j^*)$. Using AM-GM inequality,

$$1 - \lambda^2(1-y_i^*) \times (1-y_j^*) \geq 1 - \lambda^2(2-y_i^*-y_j^*)^2/4.$$

From feasibility of $(y^*, z^*)$, we know that $z_{i,j}^* \leq y_i^* + y_j^*$. Hence,

$$\begin{aligned} 1 - \lambda^2(1-y_i^*) \times (1-y_j^*) &\geq 1 - \lambda^2(2-z_{i,j}^*)^2/4 \\ &= 1 - \lambda^2 - \lambda^2(z_{i,j}^*)^2/4 + \lambda^2 z_{i,j}^* \\ &\geq (1-\lambda^2)z_{i,j}^* - \lambda^2 z_{i,j}^*/4 + \lambda^2 z_{i,j}^* \\ &= z_{i,j}^*(1-\lambda^2/4) \end{aligned}$$

From this inequality, we get that

$$\text{ALG-E} \geq (1-\lambda^2/4)\text{LP-OPT-E} \tag{3}$$

Combining (2) and (3),

$$\begin{aligned} \text{ALG-V} + \text{ALG-E} &\geq \lambda \times \text{LP-OPT-V} + (1-\lambda^2/4)\text{LP-OPT-E} \\ &\geq \min\{\lambda, 1-\lambda^2/4\} \times (\text{LP-OPT-V} + \text{LP-OPT-E}). \end{aligned}$$

Thus, the approximation ratio is at least $\min\{\lambda, 1-\lambda^2/4\}$.

$1-\lambda^2/4$ is a decreasing function of $\lambda$. To achieve the best possible approximation ratio, we can put

$$\lambda = 1 - \lambda^2/4.$$

Solving this we get $\lambda = -2+2\sqrt{2}$. And the approximation factor will be the same $-2+2\sqrt{2}$.

**Que 4 [5+5 marks].** Suppose we are given $n$ objects, their pairwise dissimilarities $\{d_{i,j} : 1 \leq i < j \leq n\}$ and also their pairwise similarities $\{s_{i,j} : 1 \leq i < j \leq n\}$. Similarities and dissimilarities are positive real numbers, and not necessarily related to each other. We want to partition the objects into **two** clusters so as to maximize the total dissimilarity of pairs lying in different clusters and the total similarity of pairs lying in same clusters. That is, maximize the following objective function

$$\sum_{\substack{i,j \\ \text{which are in} \\ \text{different clusters}}} d_{i,j} + \sum_{\substack{i,j \\ \text{which are in} \\ \text{same clusters}}} s_{i,j}.$$

We can write the following integer program for the problem.

$$\max \sum_{i,j} d_{i,j}(1-z_i z_j)/2 + \sum_{i,j} s_{i,j}(1+z_i z_j)/2 \text{ subject to}$$
$$z_i \in \{-1, 1\} \text{ for } 1 \leq i \leq n$$

Relax the program to a vector program. Write the corresponding semidefinite program. Suppose we round the SDP optimal solution by the same scheme as in Max-cut problem. Show that we get the same approximation factor (roughly 0.878).

**Ans 4.** The vector program.

$$\max \sum_{i,j} d_{i,j}(1 - z_i^T z_j)/2 + \sum_{i,j} s_{i,j}(1 + z_i^T z_j)/2 \text{ subject to}$$
$$z_i \in \mathbb{R}^n \text{ for } 1 \le i \le n$$
$$z_i^T z_i = 1 \text{ for } 1 \le i \le n$$

This can be solved using the following SDP. We consider variables $x_{i,j}$ for $z_i^T z_j$.

$$\max \sum_{i,j} d_{i,j}(1 - x_{i,j})/2 + \sum_{i,j} s_{i,j}(1 + x_{i,j})/2 \text{ subject to}$$
$$\begin{pmatrix} 1 & x_{1,2} & \cdots & x_{1,n} \\ x_{1,2} & 1 & \cdots & x_{2,n} \\ & \vdots & & \\ x_{1,n} & x_{2,n} & \cdots & 1 \end{pmatrix} \succeq 0.$$

SDP optimal: Suppose the the optimal solution for the SDP is $X^*$ and the corresponding solution for the vector program is $z_1^*, z_2^*, \ldots, z_n^* \in \mathbb{R}^n$. Let $\theta_{i,j}$ be the angle between $z_i^*$ and $z_j^*$. That is,

$$\cos\theta_{i,j} = (z_i^*)^T z_j^*.$$

Then the optimal value for the vector program/SDP is

$$\sum_{i,j} d_{i,j}(1 - \cos\theta_{i,j})/2 + \sum_{i,j} s_{i,j}(1 + \cos\theta_{i,j})/2.$$

Rounding: We pick a random hyperplane $H$ passing through origin (with orthogonal vector $h \in \mathbb{R}^n$) Put object $i$ in the left cluster if $h^T z_i^* < 0$ and put object $i$ in the right cluster if $h^T z_i^* > 0$. What is the probability that two objects $i$ and $j$ fall into the same cluster? It is same as the probability that the $z_i^*$ and $z_j^*$ are on the same side of the hyperplane $H$. As discussed in the class, this probability is equal to

$$1 - \theta_{i,j}/\pi.$$

Hence, the expected value of the rounded solution will be

$$\sum_{i,j} d_{i,j} \Pr[i \text{ and } j \text{ fall into different clusters}] + \sum_{i,j} s_{i,j} \Pr[i \text{ and } j \text{ fall into same cluster}]$$
$$= \sum_{i,j} d_{i,j} \times \theta_{i,j}/\pi + \sum_{i,j} s_{i,j} \times (1 - \theta_{i,j}/\pi)$$

Using calculus, we can show that

$$\min_{0 \le \theta \le \pi} \frac{\theta/\pi}{(1 - \cos\theta)/2} \approx 0.878$$

Similarly,

$$\min_{0 \le \theta \le \pi} \frac{1 - \theta/\pi}{(1 + \cos\theta)/2} \approx 0.878$$

One can just obtain this by replacing $\theta$ with $\pi - \theta$ in the above inequality. Hence,

$$\theta_{i,j}/\pi \ge 0.878 \times (1 - \cos\theta)/2$$

$$1 - \theta_{i,j}/\pi \ge 0.878 \times (1 + \cos\theta)/2$$

Hence, the expected value of the rounded solution will be at least 0.878 times the SDP optimal value. Thus, the approximation factor is 0.878.