Problem 1

Important note: the max-flow min-cut theorem for two commodities holds only for *undirected* graphs – that is, for any edge $e = (uv)$, the total flow $u \to v$ plus the flow $v \to u$ must be at most the capacity c_e. [1]

The maximum concurrent flow LP can be written as follows. We have a variable f^i_{uv} for each arc (directed edge) $e = (uv)$ and each commodity $i = 1, 2$. We let D^i_e be the demand of each commodity at a particular vertex, i.e. $D^i_v = 1$ for $v = t_i$, $D^i_v = -1$ for $v = s_i$, and $D^i_v = 0$ otherwise.

$$\text{maximize} \quad \lambda$$

subject to

$$\begin{aligned} \forall i = 1,2, \forall v \in V \quad \sum_{u \in V} f^i_{uv} - \sum_{w \in V} f^i_{vw} &\leq D^i_v \lambda \\ \forall e = \{uv\} \quad \sum_{i=1,2} f^i_{uv} + f^i_{vu} &\leq u_e \lambda. \end{aligned}$$

The dual LP (D) looks like

$$\text{minimize} \quad \sum_{e=uv \in E} c_e d_e$$

subject to

$$\begin{aligned} p^1_{t_1} - p^1_{s_1} + p^2_{t_2} - p^2_{s_2} &\geq 1 \\ \forall e = (u,v) \in E : \qquad d_e &\geq \max\{|p^1_u - p^1_v|, |p^2_u - p^2_v|\}. \end{aligned}$$

Note that in the optimal solution the inequalities $d_e \geq \max\{|p^1_u - p^1_v|, |p^2_u - p^2_v|\}$ are tight. Hence d can be interpreted as an L_∞ metric in $\mathbb{R}^2$.

However, d can be interpreted also as an L_1 metric using the transformation $x^1_u = (p^1_u + p^2_u)/2$, $x^2_u = (p^1_u - p^2_u)/2$ for $u \in V$. Indeed, it is not hard to verify that $|x_u - x_v|_1 = |x^1_u - x^1_v| + |x^2_u - x^2_v| = \max(|p^1_u - p^1_v|, |p^2_u - p^2_v|) = |p_u - p_v|_\infty$. It is a handy trick to know that L_1 and L_∞ in two dimensions are essentially the same thing.

Hence, an equivalent formualtion of (D) is:

$$\text{minimize} \quad \sum_{e=uv \in E} c_e d_e$$

subject to

$$\begin{aligned} d(s_1, t_1) + d(s_2, t_2) &\geq 1 \\ d \text{ is an } L_1 \text{ metric in } \mathbb{R}^2. \end{aligned}$$

which is the same as (D2):

[1] For one commodity it does not matter – there is no point in sending flow in both directions. However, for two commodities it's easy to construct examples where the theorem does not hold in directed, or even bidirected graphs.

$$\text{minimize} \quad \frac{\sum_{e=uv\in E} c_e d_e}{d(s_1,t_1)+d(s_2,t_2)}$$

over all L_1 metrics d in $\mathbb{R}^2$.

Compare this to the sparsest cut formulation (SC) for two commodities:

$$\text{minimize} \quad \frac{\sum_{e=uv\in E} c_e d_e}{d(s_1,t_1)+d(s_2,t_2)}$$

over all cut metrics metrics d. (A cut metric d is a metric for which $d(u,v) = 0$ if $|S \cap \{u,v\}| = 0, 2$ and $d(u,v) = 1$ if $|S \cap \{u,v\}| = 1$ for some set $S \in V$). By strong LP duality, there is a solution to (D2) with value equal to the optimal concurrent flow. Using the machinery of Claim 0.1, we can convert any solution to (D2) to a solution to (SC), without increasing its cost. Hence, the sparsest cut for two commodities equals maximum concurrent flow. ■

Problem 2

In Bisection, we are supposed to divide the vertex V set into two parts, $S, \overline{S}$ of size $|V|/2$ each. Hence we know that for any set $U \subseteq V$, at least $|U| - |V|/2$ vertices must be on each side of the cut. Hence we shall require that for any vertex u and any set U, there are at least $|U| - |V|/2$ vertices of U on the side that does not include u.

In our relaxation, we will have variables d_{uv} for each pair u, v of vertices. We would like to have $d_{uv} = 0$ if u and v are on the same side of the bisection, and $d_{uv} = 1$ if they are on opposite sides. Note that with this definition, d is symmetric and satisfies triangle inequality (if d_{uv} and d_{vw} are both 0, then d_{uw} has to be 0 as well).

Combining these two observations, we write the following integer program (IP):

$$\text{minimize} \quad \sum_{e=uv\in E} d_{uv} c_{uv}$$

subject to

$$\forall u \in V, \forall U \subseteq V : \quad \sum_{v\in U} d_{uv} \quad \geq \quad |U| - \frac{|V|}{2}$$

d is a metric, i.e.

$$\begin{aligned} \forall u, v \in V : \quad d_{uv} &= d_{vu} \\ \forall u, v, w \in V : \quad d_{uv} + d_{vw} &\geq d_{uw} \\ \forall u, v \in V : \quad d_{uv} &\in \{0, 1\} \end{aligned}$$

We obtain a spreading LP relaxation (LP) by replacing the last set of constraints by $0 \leq d_{uv} \leq 1$. ■

Integrality gap. As is the case with about 50% of the problems in this assignment, we turn to our good friend, the bounded degree expander for a lower bound example. Let G be the expander, let Δ be its maximum degree, α the expansion, and $n = |G|$ the number of vertices.

We claim that the shortest path metric d' for G, scaled down by a factor $\lambda = (\log_\Delta n/2 - 1)^{-1}$ is a feasible solution to (LP). Technically, let $d_{uv} = \min(1, \lambda d'_{uv})$. It is easy to see that d is a metric. Note that the cost of the solution d is $|E|\lambda = O(n/\log n)$.

To see that d is a feasible solution, pick any vertex $u \in V$ and a set $U \subseteq V$. We claim that (*) there at most $n/2$ nodes within distance strictly less than 1 from u. This clearly establishes feasibility. We show (*) by noting that a vertex v can be within distance less than 1 from u in d only if it lies within distance $1/\lambda$ in the d' metric. The ball of radius $r = 1/\lambda$ around u (in the metric d') contains at most $1 + \Delta + \Delta^2 + \ldots + \Delta^r \le \Delta^{\log_\Delta n/2} \le n/2$ nodes. Hence for any set U, at least $|U| - n/2$ of its vertices will be at distance 1 from u, showing that d is feasible for (LP).

Finally, observe that the cost of any bisection of G is at least $\alpha n = \Omega(n)$, since G is an expander. ∎

Problem 3

(a) As in the previous problem, let α be the expansion, Δ the maximum degree, and $n = |V|$ the number of vertices of a bounded-degree expander G. We claim that any embedding ϕ of the shortest path metric d into L_1 must have distortion $\Omega(\log n)$. We prove the claim by looking at different relaxations for the uniform Sparsest Cut problem for G.

The Sparsest Cut problem can be formulated as follows: (SC1)

$$\text{minimize} \quad \frac{\sum_{e=(uv)\in E} d_{uv} c_{uv}}{\sum_{u,v\in V} d_{uv}} \tag{1}$$

over all cut metrics, i.e. metrics of the form $d_{uv} = |\{u, v\} \cap S| \bmod 2$ for some set S.

The first relaxation, (SC2), will be to minimize 1 over any L_1 metric. Clearly, this is a relaxation, as any cut metric is an L_1 metric. The second relaxation, (SC3), will be to minimize 1 over *any* metric.

Claim 0.1 *Any solution to (SC2) can be converted into a solution to (SC1) of the same or lower cost.*

Proof. Let d be a solution to (SC2). Since d is a L_1 metric, it can be written as a nonnegative combination of cut metrics, i.e. $d_{uv} = \sum_{i=1}^k w_i d^i_{uv}$, where w_i are nonnegative weights and d^i are cut metrics for $i = 1, \ldots, k$. Hence the value of (SC2) can be written as

$$\frac{\sum_{i=1}^k \sum_{e=(uv)\in E} d^i_{uv} c_{uv}}{\sum_{i=1}^k \sum_{u,v\in V} d^i_{uv}} \ge \min_{i=1,\ldots,k} \frac{\sum_{e=(uv)\in E} d^i_{uv} c_{uv}}{\sum_{u,v\in V} d^i_{uv}}.$$

The metric d^i that achieves the minimum on the right hand side is the desired solution to (SC1), as it is a cut metric and its cost is no more than the cost of d. ∎

Claim 0.2 *Let the metrics d and d' be solutions to (SC3), and suppose that for some $\delta \ge 1$ the following holds:*

- *$d'_{uv} \ge d_{uv}$ for any $u, v \in V$, and*
- *$d'_{uv} \le \delta d_{uv}$ for any $u, v \in V$.*

then the cost of d' is at most δ times the cost of d.

Proof. By definition, $\sum_{e=(uv)\in E} d'_{uv} \le \delta \sum_{e=(uv)\in E} d'_{uv}$ and $\sum_{u,v\in V} d'_{uv} \ge \sum_{u,v\in V} d'_{uv}$. Combining these inequalities gives the result. ∎

Claim 0.3 *Any solution to (SC1) on the expander G has cost at least α/n.*

Proof. Let $(S, V - S)$ be the solution. Since G is an expander, there are at least $\alpha \min(|S|, |V - S|)$ edges crossing the cut, which is equal to the numerator of the objective function. The denominator is $|S| \cdot |V - S|$. ∎

Claim 0.4 *The shortest path metric on the expander G is a solution to (SC3) of cost $O(1/(n \log n))$.*

Proof. Let d be the shortest path metric on G. We have $\sum_{e=(uv)\in E} d_{uv} = |E| \leq \Delta n$. On the other hand, for each vertex u, there are $\Omega(n)$ vertices within distance $\Omega(\log n)$, and hence $\sum_{u,v\in V} \geq \Omega(n^2 \log n)$. ■

Finally, we arrive at our desired result.

Theorem 0.5 *The distortion of any embedding of the shortest path metric d of the expander G into L_1 has to be $\Omega(\log n)$.*

Proof. By Claim 0.4, the cost of d is $O(1/(n \log n))$. Suppose we could embed d into an L_1 metric d' with distortion δ better than $\Omega(\log n)$. By Claim 0.2, d' is a solution to (SC2) of cost at most $O(\delta/(n \log n))$, and by Claim 0.1, we can obtain a solution to (SC1) of no higher cost. But this contradicts Claim 0.3, that every solution to (SC1) costs $\Omega(1/n)$. ■

(b) We observe that a distribution over tree metrics can be embedded into a L_1 metric with no distortion. If we could embed expanders into a distribution over tree metrics with some distortion δ better than $\Theta(\log n)$, we could embed them into L_1 with the same distortion. This would contradict the result from (a).

First, observe that any tree metrics d defined by a weighted tree T can be expressed as a positive linear combination of $n-1$ cut metrics. For each edge $e = (uv)$ of the tree, we throw in the cut $(V_e, \overline{V}_e)$ with weight $d_e = d(u, v)$ (V_e and $\overline{V}_e$ are the two components obtained by removing e from T). Clearly, for any pair of vertices u and v, there is exactly one cut $(V_e, \overline{V}_e)$ separating u and v for each edge e on the unique (hence shortest) u-v path P in T, and the weight of each cut is the same as the weight of the corresponding edge.

A distribution over metrics is essentially the same as a convex combination of these metrics. Since a tree metric can be written as a nonnegative combination of cut metrics, it follows that a distribution over tree metrics can also be written as a nonnegative combination of cut metrics. A nonnegative combination of cut metrics can be expressed as an L_1 metric in the obvious way: for each cut $(S, \overline{S})$ of weight w_S we'll have a separate coordinate x^S, and for each vertex $v \in V$ we define $x^S_v = 0$ if $v \in S$ and $x^S_v = 1$ if $v \in \overline{S}$. ■

Problem 4

Here is the recursive algorithm $\mathcal{A}$:

1. if $|V(G)| = 1$, return the trivial ordering $\{v_1\}$.
2. Find a 1/3-balanced cut with cost within a $O(\log n)$ factor of the minimum bisection cost of G. Let $S, \overline{S}$ be the cut found.
3. Recursively call $\mathcal{A}$ to find the orderings $u_1, u_2, \ldots, u_{n_1}$ and $v_1, v_2, \ldots, v_{n_2}$ for the induced subgraphs $G(S)$ and $G(\overline{S})$. Return the concatenation of the two orderings, $u_1, \ldots, u_{n_1}, v_1, \ldots, v_{n_2}$.

Theorem 0.6 *The above algorithm $\mathcal{A}$ is a $O(\log^2 n)$ approximation algorithm for* Min-cut linear arrangement.

For a set of vertices $S \subseteq V$, let $B(S)$ denote the cost of the minimum bisection of the induced subgraph $G(S)$ (for $|S|$ odd, bisection means splitting the graph into two parts of size $\lceil |S|/2 \rceil$ and $\lfloor |S|/2 \rfloor$). Clearly, one of the cuts of the linear arrangement is a bisection of $G(S)$, hence for any S we have that $B(S)$ is a lower bound on the cost of any linear arrangement. Let $B = \max_{S\subseteq V} B(S)$.

To prove Theorem 0.6, it clearly suffices to prove

Theorem 0.7 *The algorithm $\mathcal{A}$ outputs a linear arrangement of cost at most $O(B \log^2 n)$.*

Proof. Let us look at the structure of recursive calls of $\mathcal{A}$. First, since we always find a 1/3-balanced cut, the size of each component shrinks by a constant factor at each level, and hence the depth of the recursion is $O(\log n)$.

The second observation is that the subproblems from the recursive calls are nested. If $G_1, G_2, \ldots, G_m$ are all graphs on which $\mathcal{A}$ has been recursively invoked, we have that for every i, j, either $G_i \subseteq G_j$, $G_j \subseteq G_i$, or $G_i \cap G_j = \emptyset$. Since the depth of recursion is $O(\log n)$, it follows that there are only $O(\log n)$ subgraphs G_j that contain any particular vertex v_k.

To finish the proof, let $v_1, v_2, \ldots, v_n$ be the ordering produced by the algorithm. Consider any cut $(\{v_1, v_2, \ldots, v_k\}, \{v_{k+1}, dots, v_n\})$. Clearly, each edge crossing this cut must have been cut in one of the recursive calls. Note that any such edge could have been cut only by a recursive call on a graph G_i, such that both v_k and v_{k+1} belong to G_i. There can be at most $O(\log n)$ such graphs. The total capacity of edges cut in any recursive step is $O(B \log n)$, hence the total capacity of edges cut by recursive calls on graphs G_i that contain v_k and v_{k+1} is $O(B \log^2 n)$. ■