Assume standard complexity theoretic assumptions for all the questions.

A 2-approximation algorithm for the multiway cut problem operates as follows.

For each distinguished vertex s_i, *it computes a minimum isolating cut* F_i *between* s_i *and the set of remaining distinguished vertices* $\{s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_k\}$. *This is achieved by introducing a sink vertex* t *into the graph and connecting it with infinite cost edges from all distinguished vertices except* s_i. *The algorithm then calculates a minimum* s_i-t *cut. The final solution is obtained by taking the union of all such cuts:* $\bigcup_{i=1}^{k} F_i$.

(a) The algorithm that returns the cheapest $k - 1$ minimum isolating cuts is a $(2 - k)$-approximation algorithm.

(b) The algorithm that returns the cheapest $k - 1$ minimum isolating cuts is a $(2 - \frac{2}{k})$-approximation algorithm.

(c) The algorithm that returns the cheapest $k - 1$ minimum isolating cuts is a $(2 + \frac{2}{e})$-approximation algorithm.

(d) The algorithm that returns the cheapest $k - 1$ minimum isolating cuts is a $(2 - \frac{2}{e})$-approximation algorithm.

b.

Consider the improved version of the algorithm in Question 1 i.e. the algorithm that returns the cheapest $k - 1$ minimum isolating cuts. What is the optimal and the approximate solution for the graph given below where the terminals are s_1, s_2, s_3 and s_4:

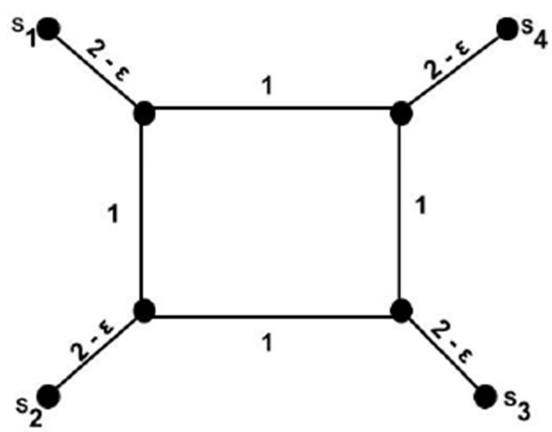

Figure 1: Question 2

(a) 3 and 4(2 - ε) respectively

(b) 7 and 8(2 - ε) respectively

(c) 4 and 3(2 - ε) respectively

(d) 8 and 7(2 - ε) respectively

c.

Recall that another way of looking at the multiway cut problem is finding an optimal partition of V into sets C_i such that $s_i \in C_i$ for all $i \in k$, and such that the cost of $F = \bigcup_{i=1}^{k} \delta(C_i)$ is minimized. Given this perspective, we define two variables: x_u^i and z_e^i, such that

- ▷ $x_u^i = 1$ if vertex u is assigned to set C_i, and 0 otherwise.
- ▷ $z_e^i = 1$ if edge e belongs to the cut-set $\delta(C_i)$, and 0 otherwise.

Which of the following best describes the objective function of the integer program?

(a) Minimize $\frac{1}{2} \sum_{e \in E} c_e \sum_{i=1}^{k} z_e^i$

(b) Maximize $\sum_{e \in E} c_e \sum_{i=1}^{k} z_e^i$

(c) Minimize $\sum_{u \in V} c_v \sum_{i=1}^{k} x_u^i$

(d) Maximize $\sum_{u \in V} c_v \sum_{i=1}^{k} z_u^i$

a.

. Which of the following correctly captures the constraints of the multiway cut problem for the objective function given in Question 3?

(a)
$$\begin{aligned} &\textstyle\sum_{i=1}^{k} x_u^i = 1, \quad \forall u \in V \\ &z_e^i \geqslant x_u^i - x_v^i, \quad \forall e = (u,v) \in E \\ &z_e^i \geqslant x_v^i - x_u^i, \quad \forall e = (u,v) \in E \\ &x_{s_i}^i = 1, \quad i = 1, \dots, k \\ &x_u^i \in \{0,1\}, \quad \forall u \in V,\ i = 1, \dots, k \end{aligned}$$

(b)
$$\begin{aligned} &\textstyle\sum_{i=1}^{k} x_u^i \leqslant 1, \quad \forall u \in V \\ &z_e^i \leqslant x_u^i - x_v^i, \quad \forall e = (u,v) \in E \\ &z_e^i \leqslant x_v^i - x_u^i, \quad \forall e = (u,v) \in E \\ &x_{s_i}^i = 1, \quad i = 1, \dots, k \\ &x_u^i \in \{0,1\}, \quad \forall u \in V,\ i = 1, \dots, k \end{aligned}$$

(c)
$$\begin{aligned} &\textstyle\sum_{i=1}^{k} x_u^i = 1, \quad \forall u \in V \\ &z_e^i \geqslant x_u^i - x_v^i, \quad \forall e = (u,v) \in E \\ &z_e^i \geqslant x_v^i - x_u^i, \quad \forall e = (u,v) \in E \\ &x_{s_i}^i = 1, \quad i = 1, \dots, k \\ &x_u^i \in \{0,1\}, \quad \forall u \in V,\ i = 1, \dots, k \end{aligned}$$

(d)
$$\begin{aligned} &\textstyle\sum_{i=1}^{k} x_u^i = 1, \quad \forall u \in V \\ &z_e^i \leqslant x_u^i - x_v^i, \quad \forall e = (u,v) \in E \\ &z_e^i \leqslant x_v^i - x_u^i, \quad \forall e = (u,v) \in E \\ &x_{s_i}^i = 1, \quad i = 1, \dots, k \\ &x_u^i \in \{0,1\}, \quad \forall u \in V,\ i = 1, \dots, k \end{aligned}$$

c.

Which is INCORRECT about the randomized rounding algorithm for the multiway cut problem?

(a) The probability that an edge $e = (u, v)$ belongs to the cut-set is at most $\frac{3}{2}\|x_u - x_v\|_1$,.

(b) For any index l and any two vertices $u, v \in V$, we have $|x_u^l - x_v^l| \geqslant \frac{1}{2}\|x_u - x_v\|_1$

(c) The algorithm is a $\frac{3}{2}$ factor approximation algorithm.

(d) The algorithm picks a random permutation π of $\{1, \ldots, k\}$ in $O(k)$.

b.

Consider the Linear Programming formulation for the Multicut problem.

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in E} c_e x_e && (1) \\ \text{subject to} \quad & \sum_{e \in P} x_e \geqslant 1, \quad \forall P \in \mathcal{P}_i,\ 1 \leqslant i \leqslant k, && (2) \\ & x_e \in \{0, 1\}, \quad \forall e \in E. && (3) \end{aligned}$$

It seems the inequality in (2), $\sum_{e \in P} x_e \geqslant 1$ has an exponential number of constraints. To tackle this which of the following CANNOT be applied?

(a) A polynomial time separation oracle to solve the relaxed LP

(b) An alternative equivalent polynomially-sized linear program

(c) The separation oracle works as follows: Given a solution x, we consider the graph G in which the length of each edge e is x_e. For each i, $1 \leqslant i \leqslant k$, we compute the length of the shortest path between s_i and t_i. If for some i, the length of the shortest path P is less than 1, we return it as a violated constraint, since we have $\sum_{e \in P} x_e < 1$ for $P \in \mathcal{P}_i$. If for each i, the length of the shortest path between s_i and t_i is at least 1, then the length of every path $P \in \mathcal{P}_i$ is at least 1, and the solution is feasible.

(d) The separation oracle works as follows: Given a solution x, we construct a network flow problem on the graph G in which the capacity of each edge e is set to x_e. For each vertex i, we check whether the maximum flow from i to the root r is at least 1. If not, then the minimum cut S separating i from r gives a violated constraint such that $\sum_{e \in P} x_e < 1$ for $P \in \mathcal{P}_i$. If the flow is at least 1, then by the max-flow/min-cut theorem the solution is feasible.

d.

Given a feasible solution to the linear program given in Question 6, for any s_i one can find in polynomial time a radius $r < \frac{1}{2}$ such that

(a) $c(\delta(B_x(s_i), r)) \leqslant (\frac{1}{2}\ln(k+1))V_x(s_i, r)$

(b) $c(\delta(B_x(s_i), r)) \geqslant (2\ln(k+1))V_x(s_i, r)$

(c) $c(\delta(B_x(s_i), r)) \geqslant (4\ln(k+1))V_x(s_i, r)$

(d) $c(\delta(B_x(s_i), r)) \leqslant (2\ln(k+1))V_x(s_i, r)$

d.

Consider the algorithm given below for the multicut problem.

Algorithm 1 Algorithm for the multicut problem.

```
1: Let x be an optimal solution to the LP
2: F ← ∅
3: for i ← 1 to k do
4:     if s_i and t_i are connected in (V, E − F) then
5:         Choose radius r < 1/2 around s_i
6:         F ← F ∪ δ(B_x(s_i, r))
7:         Remove B_x(s_i, r) and incident edges from graph
8:     end if
9: end for
10: return F
```

Which of the following is true?

(a) For any value of r, such that $V(r)$ is differentiable, we have $\frac{dV}{dr} = 0$.

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

(c) Line 5 of the given algorithm should be: *Choose radius* $r < \frac{1}{2}$ *around* s_i *such that* $c(\delta(B_x(s_i, r))) \geqslant (2\ln(k+1)) \cdot V_x(s_i, r)$.

(d) Line 6 should be replaced by $F \leftarrow F \cap \delta(B_x(s_i, r))$

b.

Which of the following is true about the balls $B_x(s_i, r)$ and volumes $V_x(s_i, r)$?

(a) The balls $B_x(s_i, r)$ and volumes $V_x(s_i, r)$ are taken with respect to the entire initial graph, not considering the edges and vertices removed in previous iterations.

(b) In each iteration, the balls $B_x(s_i, r)$ and volumes $V_x(s_i, r)$ are computed based on a fixed radius r that does not vary across different iterations.

(c) The volumes $V_x(s_i, r)$ are calculated based on the edges and vertices remaining in the current graph, but the balls $B_x(s_i, r)$ are taken with respect to the original graph, including all removed edges and vertices.

(d) In any iteration, the balls $B_x(s_i, r)$ and volumes $V_x(s_i, r)$ are taken with respect to the edges and vertices remaining in the current graph.

d.

The expected value of $\frac{c(r)}{V(r)}$ for r chosen uniformly from $[0, \frac{1}{2})$ is

(a) $\mathbb{E}\left[\frac{c(r)}{V(r)}\right] = 2\sum_{j=0}^{l-1} \int_{r_j}^{r^-_{j+1}} \frac{1}{V(r)} \frac{dV}{dr}\, dr$

(b) $\mathbb{E}\left[\frac{c(r)}{V(r)}\right] < 2\sum_{j=0}^{l-1} [\ln V(r)]_{r_j}^{r^-_{j+1}}$

(c) $\mathbb{E}\left[\frac{c(r)}{V(r)}\right] > 2\sum_{j=0}^{l-1} \left[\ln V(r^-_{j+1}) - \ln V(r_j)\right]$

(d) $\mathbb{E}\left[\frac{c(r)}{V(r)}\right] \geqslant 2\sum_{j=0}^{l-1} [\ln V(r_{j+1}) - \ln V(r_j)]$

a.