Endsem

Time : 150 minutes Full Marks :80

Submission Guidelines.

- The exam is open book and open notes but no electronic device can be used
- Keep your final answers brief and precise. Meaningless rambles fetch negative credits.
- Strictly my perception of hardness of the problems : $P2 \prec P4 \preceq P3 \prec P1$

Problem 1. (20 points)

Let $G = (V, E)$ be a directed graph with an edge–cost function $c : E \to \mathbb{Z}$ (costs may be negative). Let $s, t \in V$ be two distinct vertices. Consider the following linear program:

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in E} c_e \, x_e \\ \text{subject to} \quad & \sum_{e \in \delta^+(S)} x_e \;\geq\; 1 \qquad \text{for every subset } S \subseteq V \text{ with } s \in S,\ t \notin S, \\ & x_e \;\geq\; 0 \qquad \text{for all } e \in E. \end{aligned}$$

(a) (5 points) Show that the LP is feasible if and only if an s–t path exists in G.

textbfSolution. Claim: The LP is feasible if and only if an $s-t$ path exists in G.

- **Necessity ($\implies$):** Suppose the LP is feasible. Let S be the set of all vertices reachable from s in G. By definition, $s \in S$. If no $s-t$ path exists, then $t \notin S$. For this specific subset S, there are no edges (u, v) such that $u \in S$ and $v \notin S$ (otherwise v would be reachable, contradicting the definition of S). Thus, $\delta^+(S) = \emptyset$, and the sum $\sum_{e \in \delta^+(S)} x_e = 0$. This violates the constraint $\sum x_e \geq 1$. Therefore, an $s-t$ path must exist.
- **Sufficiency ($\impliedby$):** Suppose an $s-t$ path P exists. Define a solution by setting $x_e = 1$ if $e \in P$ and $x_e = 0$ otherwise. For any subset S where $s \in S$ and $t \notin S$, the path P must cross from S to $V \setminus S$ at least once to reach t. Thus, there is at least one edge $e \in P \cap \delta^+(S)$, ensuring $\sum_{e \in \delta^+(S)} x_e \geq 1$.

Rub : $+2.5$ for each part

(b) (5 points) Write the dual of this LP and write a one-two sentence 'intuition' of the dual constraints.

Solution.

$$\begin{aligned} \text{maximize} \quad & \sum_{S: s\in S, t\notin S} y_S \\ \text{subject to} \quad & \sum_{S: e\in\delta^+(S)} y_S \le c_e \quad \forall e \in E \\ & y_S \ge 0 \quad \forall S \subseteq V : s \in S, t \notin S \end{aligned}$$

Intuition: The dual variables y_S represent "widths" or "distances" assigned to the cuts separating s and t[cite: 20]. The objective is to maximize the total distance between s and t such that for any edge e, the sum of distances of the cuts it crosses does not exceed its cost c_e.

Rub: $+1$ for objective, $+2$ for the edge constraints, $+1$ for non-negativity constraint, $+1$ for intuition

(c) (10 points) Assume now that all costs satisfy $c_e \ge 0$. Let $d(t)$ denote the shortest–path distance from s to t with respect to the costs c_e. Show that the dual LP has a feasible solution whose objective value is exactly $d(t)$.

(Hint : Try to use the shortest-path distances from s to 'construct' a feasible dual. If you did things correctly, your dual variables should belong to subsets of V. The trick is to decide which subsets will have positive dual and how much. Think of Dijkstra's algorithm conceptually if that helps)

Solution. Assume $c_e \ge 0$. Let $d(v)$ be the shortest path distance from s to v. We construct a dual solution with objective value $d(t)$.

1. **Subsets:** Let $0 = \rho_0 < \rho_1 < \cdots < \rho_k < d(t)$ be the distinct shortest path distances from s to vertices in V.
2. **Construction:** Define $S_i = \{v \in V \mid d(v) \le \rho_i\}$. Note $s \in S_i$ and $t \notin S_i$[cite: 23]. Set:

   $$y_{S_i} = \rho_{i+1} - \rho_i \quad (\text{where } \rho_{k+1} = d(t))$$

   Set $y_S = 0$ for all other S.
3. **Objective Value:** $\sum y_S = (\rho_1 - \rho_0) + (\rho_2 - \rho_1) + \cdots + (d(t) - \rho_k) = d(t)$.
4. **Feasibility:** For an edge $e = (u, v)$, the sum $\sum_{S: e\in\delta^+(S)} y_S$ telescopes to $d(v) - d(u)$ if $d(v) > d(u)$, or 0 otherwise. By the triangle inequality $d(v) \le d(u) + c_e$, we have $d(v) - d(u) \le c_e$, satisfying the constraint.

Remark. The above is a more general way of setting the dual (as given by Gemini Pro). However, you may also choose the sets by 'running' Dijkstra and using the laminar cuts created in the process. It's essentially the same thing.

Rub : $+5$ for correctly specifying the duals, $+1$ for arguing objective function is $d(t)$. $+4$ for arguing feasibilily (a common mistake is only arguing for edges along the shortest path. this will only fetch half the score).

Problem 2. (**20 points**) Consider the standard LP relaxation for the weighted vertex cover problem:

$$\begin{aligned} \min \quad & \sum_{v \in V} w_v x_v \\ \text{s.t.} \quad & x_u + x_v \ \geq \ 1 \qquad \text{for every edge } (u,v) \in E, \\ & x_v \ \geq \ 0 \qquad \text{for all } v \in V. \end{aligned} \tag{VC-LP}$$

(a) (10 points) Prove that every extreme point optimal solution of (VC-LP) is *half-integral*; that is, each variable satisfies

$$x_v \in \{0, \tfrac{1}{2}, 1\} \qquad \text{for all } v \in V.$$

Solution. Suppose x is an optimal extreme point solution. Assume, for contradiction, that x is not half-integral. Define the following sets of vertices based on their LP values:

- $V_0 = \{v \in V \mid 0 < x_v < 1/2\}$
- $V_1 = \{v \in V \mid 1/2 < x_v < 1\}$

If x is not half-integral, then $V_0 \cup V_1$ must be non-empty[cite: 32, 33]. We can define a small epsilon $\epsilon > 0$ and two new vectors y and z such that:

- $y_v = x_v + \epsilon$ if $v \in V_0$, $y_v = x_v - \epsilon$ if $v \in V_1$, and $y_v = x_v$ otherwise.
- $z_v = x_v - \epsilon$ if $v \in V_0$, $z_v = x_v + \epsilon$ if $v \in V_1$, and $z_v = x_v$ otherwise.

For a sufficiently small ϵ:

(a) **Feasibility:** For any edge (u,v), the constraint $x_u + x_v \geq 1$ remains satisfied[cite: 27, 28]. If one endpoint is in $\{0, 1, 1/2\}$, the small change in the other ($\pm\epsilon$) does not violate the sum. If both endpoints are in $(0,1) \setminus \{1/2\}$, the total change in $x_u + x_v$ is either 0, 2ϵ, or -2ϵ. However, since $x_u + x_v > 1$ was strictly true for these non-half-integral values (unless they sum exactly to 1, in which case the adjustments $+\epsilon$ and $-\epsilon$ balance out), feasibility is maintained.

(b) **Linearity:** Since $x = \frac{1}{2}y + \frac{1}{2}z$, if x is an extreme point, it cannot be represented as a convex combination of two other distinct feasible solutions y and z.

The existence of such y and z contradicts x being an extreme point unless the set of non-half-integral variables is empty. Thus, $x_v \in \{0, \frac{1}{2}, 1\}$ for all $v \in V$

Rub : +2 for even a correct idea. +2x2 for defining the two solution correctly. +2 for showing they are both feasible. +2 for arguing that $x^\star$ is a convex combination of the two.

(b) (10 points) Now assume the graph G is 4-colorable and a proper 4-coloring $V = C_1 \cup C_2 \cup C_3 \cup C_4$ is given as part of the input. In other words, the graph can be partitioned in to 4 disjoint sets $C_i, i \in 1,2,3,4$ such that *every* edge in the graph has its two endpoints in different partitions. Using part (a), design a polynomial-time algorithm that returns a vertex cover of weight at most

$$\tfrac{3}{2} \cdot \text{OPT},$$

where OPT is the optimal value of (VC-LP). Prove the approximation guarantee.

(Hint: Start thinking about the toy example K_4 - the complete graph on 4 vertices and then generalize)

Solution. Algorithm:

(a) Solve (VC-LP) to obtain an extreme point optimal solution x^*.

(b) Identify the set of fractional vertices $V_{1/2} = \{v \in V \mid x^*_v = 1/2\}$.

(c) Partition $V_{1/2}$ into four sets C'_1, C'_2, C'_3, C'_4 based on the given 4-coloring.

(d) Identify the color class with the **maximum** weight: $C'_{max} = \text{argmax}_{C'_i} w(C'_i)$. Note that $w(C'_{max}) \geq \frac{1}{4} w(V_{1/2})$.

(e) Return the vertex cover $S = \{v \mid x^*_v = 1\} \cup (V_{1/2} \setminus C'_{max})$.

Analysis: *Feasibility:* Any edge (u, v) with $x^*_u = x^*_v = 1/2$ must have endpoints in different color classes. Since we discard only one class (C'_{max}), at least one endpoint remains in S. Here we are crucially using the property that every $x^\star_v = 1/2$. Edges with an endpoint $x^*_v = 1$ are covered automatically.

Approximation Ratio: The weight of our solution is:

$$w(S) = \sum_{x^*_v = 1} w_v + w(V_{1/2}) - w(C'_{max}) \leq \sum_{x^*_v = 1} w_v + \frac{3}{4} w(V_{1/2})$$

Since the LP objective is $OPT_{LP} = \sum_{x^*_v=1} w_v + \frac{1}{2} w(V_{1/2})$, we observe:

$$w(S) \leq \frac{3}{2} \left(\sum_{x^*_v=1} w_v + \frac{1}{2} w(V_{1/2}) \right) = \frac{3}{2} OPT_{LP} \leq \frac{3}{2} OPT$$

Rub : $+5$ for the algorithm description and $+5$ for the analysis. This part is a bit subjective.

Problem 3. (**15 points**) Consider the simple random walk on the complete graph K_n. The walk starts at an arbitrary vertex (does not matter which one). At each step, the walk moves to a vertex chosen independently and uniformly at random from the n vertices (yes ! it may stay at the same vertex too). Let T denote the *cover time* of the walk, i.e., the number of steps until every vertex has been visited at least once. Show that $\mathbb{E}[T] = O(n \log n)$.

(Hints : Let T_k be the number of steps needed to go from having visited $k-1$ distinct vertices to having visited k distinct vertices. Can you express T in terms of T_k's ? What is $\mathbb{E}[T_k]$)?

Solution.

1. **Defining Intermediate Steps**
   Following the hint, let T_k be the number of steps needed to go from having visited $k-1$ distinct vertices to having visited k distinct vertices. The total cover time T can be expressed as the sum of these independent intervals:

$$T = \sum_{k=1}^{n} T_k$$

Rub : $+3$

2. **Calculating the Expectation of T_k**
   When $k-1$ distinct vertices have already been visited, there are $n-(k-1)$ vertices remaining that have not been visited. At each step, the probability p_k of moving to a new (unvisited) vertex is:
   $$p_k = \frac{n-(k-1)}{n}$$
   Since each step is independent and the walk can stay at the same vertex, T_k follows a *geometric distribution* with success probability p_k. The expected value of a geometric random variable is $1/p$:
   $$\mathbb{E}[T_k] = \frac{1}{p_k} = \frac{n}{n-k+1}$$
   Rub : +7 for probability, +3 for expectation

3. **Calculating the Total Expected Cover Time $\mathbb{E}[T]$**
   By the linearity of expectation:
   $$\mathbb{E}[T] = \mathbb{E}\left[\sum_{k=1}^{n} T_k\right] = \sum_{k=1}^{n} \mathbb{E}[T_k]$$
   Substituting the values:
   $$\mathbb{E}[T] = \sum_{k=1}^{n} \frac{n}{n-k+1} = n\left(\frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{1}\right)$$
   The term in the parentheses is the n-th harmonic number, denoted as H_n. It is a well-known result that $H_n = \sum_{i=1}^{n} \frac{1}{i} = \Theta(\log n)$.
   Thus:
   $$\mathbb{E}[T] = n \cdot H_n = O(n \log n)$$
   Rub : +2 for finishing calculation

Problem 4. (**25 points**) There are n jobs and m *unrelated* machines - this means that Job i has processing time $p_{ij} \in [0,1]$ on machine j. We want to assign each job to exactly one machine so as to minimize the maximum load over all machines.

(a) (10 points) Write a natural linear programming relaxation for this problem.

Solution. **NOTE:** I am writing the solutions assuming $p_i j \in [0,1]$ since there are versions of Chernoff which work for that case too but has not been mentioned in the class. You get credit for the $(1,\infty)$ case too.

We define x_{ij} as the fractional assignment of job i to machine j, and L as the makespan.

$$\begin{aligned} \text{minimize} \quad & L \\ \text{subject to} \quad & \sum_{j=1}^{m} x_{ij} = 1 \quad \forall i \in \{1, \ldots, n\} \\ & \sum_{i=1}^{n} p_{ij} x_{ij} \leq L \quad \forall j \in \{1, \ldots, m\} \\ & x_{ij} = 0 \quad \text{if } p_{ij} > L \\ & 0 \leq x_{ij} \leq 1 \quad \forall i, j \end{aligned}$$

Rub: $+2$ for objective, $+2$x3 for each of the important constraints (ignore the $x_i j = 0$ one for the purpose of exam)

(b) (15 points) Design a randomized rounding scheme that is $O\left(\frac{\log m}{\log \log m}\right)$-approximation in expectation. In particular, I would expect the following from you :

(i) A clear and concise description of the entire algorithm

Solution.

Algorithm: 1. Solve the LP to find x^*_{ij} and L^*. 2. For each job i, assign it to machine j with probability x^*_{ij} independently.

Rub : $+5$ for the correct algorithm. Points deducted for missing 'independently'.

(ii) The approximation analysis. You can use results mentioned in class or you have read in the notes that have been provided (any results used from sources other than these two needs to be proved).

Solution. Let X_{ij} be the indicator for job i being on machine j. The load on machine j is $Y_j = \sum_{i=1}^{n} p_{ij} X_{ij}$. By linearity of expectation:

$$\mathbb{E}[Y_j] = \sum_{i=1}^{n} p_{ij} Pr[X_{ij} = 1] = \sum_{i=1}^{n} p_{ij} x^*_{ij} \leq L^*$$

Rub : $+3$ for setting up RV's.

Since $p_{ij} \in [0, 1]$ and assignments are independent, we apply the Chernoff bound. Let $1 + \delta = \frac{c \log m}{\log \log m}$.

$$Pr[Y_j > (1 + \delta)L^*] < \left(\frac{e^{\delta}}{(1 + \delta)^{(1+\delta)}}\right)^{L^*} \leq \frac{1}{m^2}$$

By the Union Bound over m machines, the probability that any machine exceeds $(1+\delta)L^*$ is $\leq 1/m$.

Rub : $+2$ for applying Chernoff.

Expectation. We split the expectation:

$$E[Y_{max}] = \int_0^T P(Y_{max} > t)dt + \int_T^n P(Y_{max} > t)dt$$

$$E[Y_{max}] \leq T \cdot 1 + (n - T) \cdot \frac{1}{m}$$

$$E[Y_{max}] \leq \alpha \frac{\ln m}{\ln \ln m} L^* + \frac{n}{m}$$

Since $L^* \geq 1$ (assuming at least one job exists) and n/m is typically small relative to the log factor, the expectation is dominated by the first term:

$$E[\text{Makespan}] = O\left(\frac{\log m}{\log \log m}\right) L^*$$

Rub : $+5$ for finishing expectation calculation.

(b) Dual Formulation and Intuition

The dual of the primal LP is constructed by assigning a variable y_S to each cut constraint[cite: 20].

Dual Linear Program: