Due date: September 18, 2024

Problem 1: Given an undirected graph $G = (V, E)$ where V is the set of vertices and E is the set of edges, a *vertex cover* is a subset $C \subseteq V$ such that for every edge $(u, v) \in E$, at least one of the endpoints u or v is in C. The vertex-cover problem asks for computing the smallest vertex cover, i.e., the vertex cover of minimum size.

(i) Write an integer LP (ILP) formulation of the vertex-cover problem and its LP relaxation.

(ii) Provide a specific example of a graph where solving the LP relaxation yields a solution that is smaller than the optimal integer solution.

(iii) Write the dual of the relaxed LP formulation.

Problem 2: Let (P) be a feasible and bounded linear program $\max c^\top x$ s.t. $Ax \leq b, x \geq 0$, for vectors $c \in \mathbb{R}^n, b \in \mathbb{R}^m$ and matrix $A \in \mathbb{R}^{m\times n}$. Moreover, let (D) be its dual linear program. Suppose x^* and y^* are optimal choices of variables for (P) and (D), respectively. Prove the complementary slackness conditions

$$y_i^* \left(\sum_{j=1}^{n} A_{ij} x_j^* - b_i \right) = 0 \ \ \forall i, \text{ and} \tag{1}$$

$$x_j^* \left(\sum_{i=1}^{m} A_{ij} y_i^* - c_j \right) = 0 \ \ \forall j. \tag{2}$$

Problem 3: The maximum (s, t)-flow problem consists of a directed graph $G = (V, E)$, two special vertices s and t, and a function assigning a non-negative capacity $c(e)$ to each edge $e \in E$. The goal is to to compute a flow function $f \in \mathbb{R}^E_{\geq 0}$ that satisfies the usual capacity and conservation constraints. The maximum-flow problem can be encoded as a linear program as follows:

$$\text{maximize} \quad \sum_{w:\, s\to w \in E} f(s \to w)$$

subject to:

$$\sum_{w:\, v\to w\in E} f(v \to w) - \sum_{u:\, u\to v\in E} f(u \to v) = 0 \quad \text{for every vertex } v \in V,\, v \neq s, t$$

$$f(u \to v) \leq c(u \to v) \quad \text{for every edge } u \to v \in E$$

$$f(u \to v) \geq 0 \quad \text{for every edge } u \to v \in E$$

(i) Write the dual for the above LP.

(ii) What does the dual compute? Justify your answer.

Problem 4: In a simple game, there are n states, and the goal is to go from state 1 to state n. At each state, the player can choose one of k actions. Choosing action a at state s moves the player to a random state distributed according to $D_{a,s}$. The game ends immediately once the player reaches state n. Assume that there is a path of finite length from every state s to state n.

A *strategy* is a function that maps every state s to a probability distribution over the actions. Let $D_{a,i}(j)$ denote the probability of transitioning to state j when taking action a on state i. The expected number of steps to reach state n using the optimal strategy (i.e., the strategy that minimizes the expected number of steps to reach state n) can be determined by solving the following LP:

$$\begin{aligned} \text{maximize} \quad & x_1 \\ \text{subject to} \quad & x_i \le 1 + \sum_{j=1}^{n-1} D_{a,i}(j) x_j \qquad \forall a \in [k], \quad i \in [n-1] \end{aligned}$$

(i) Write the dual of the above LP.

(ii) Let μ_i denote the expected number of steps to reach state n from state i when following the optimal strategy. Let $x_i := x_i^*$ in an optimal solution of the primal LP. Using strong duality, argue that $x_1^* = \mu_1$.

Problem 5: Given a complete bipartite graph $G = (X \cup Y, X \times Y)$, where $|X| = |Y| = n \in \mathbb{N}$, with edge costs $w \colon X \times Y \to \mathbb{N} \cap [1, \Delta]$ for some arbitrary $\Delta \in \mathbb{N}$. The bottleneck matching problem asks for a perfect matching M minimizing the largest edge cost in M. Provide an algorithm solving the bottleneck matching problem in $O(n^3 \log \Delta)$ time and prove its correctness.