Tutorial 4

Approximation and Online Algorithms

Submission Deadline - 20th September 2026

All problems marked (*) need to be submitted by the submission deadline.

---

1. Consider an instance of weighted MAX-k-SAT. Design a randomized algorithm that independently sets every variable to True with probability 1/2. Prove that the expected weight of satisfied clauses is at least

$$\left(1 - 2^{-k}\right) \sum_{j=1}^{m} w_j$$

when every clause has exactly k literals.

2. Derandomize the MAX-k-SAT algorithm from Problem 1 using the method of conditional expectations. In particular, give a polynomial-time procedure for deciding the value of each variable so that the final assignment satisfies at least as much total clause weight as the randomized algorithm's expectation.

3. Consider a MAX-SAT instance containing both positive and negative literals. Instead of setting every variable to True with probability 1/2, set every variable to True independently with probability p. Determine the value of p that maximizes the worst-case probability of satisfying a clause, and prove the resulting approximation guarantee.

4. Given the standard LP relaxation of weighted MAX-SAT with variables y_i and z_j, propose a randomized rounding algorithm that sets variable x_i to True with probability y_i. Prove an approximation guarantee by relating the probability that clause C_j is satisfied to its LP variable z_j.

5. For a weighted MAX-SAT instance, consider the two randomized algorithms: (i) unbiased random assignment and (ii) LP randomized rounding. Run both algorithms and return the assignment satisfying the larger total clause weight. Prove that this "better of two" algorithm achieves a 3/4-approximation.

6. In the LP randomized rounding algorithm for MAX-SAT, replace the rounding probability y_i by a function $f(y_i)$. Design a suitable non-linear function $f : [0,1] \to [0,1]$ and prove that the resulting algorithm achieves a 3/4-approximation.

7. Consider the Prize-Collecting Steiner Tree problem with edge costs c_e and penalties π_v. Given an optimal solution to its standard LP relaxation, design the randomized rounding algorithm based on a uniformly chosen threshold $\alpha \in [\gamma, 1]$. Analyze the expected edge cost and penalty cost, and choose an appropriate value of γ to obtain the best approximation guarantee.

8. Consider the metric Uncapacitated Facility Location problem with facility-opening variables y_i and assignment variables x_{ij} from the standard LP relaxation. Design the randomized rounding algorithm that opens facilities and assigns clients using the fractional solution. Prove that the resulting algorithm is a randomized 3-approximation.

9. Consider the problem of scheduling jobs on a single machine with release dates so as to minimize

$$\sum_{j=1}^{n} w_j C_j.$$

Given the time-indexed LP relaxation, design the randomized rounding algorithm using a uniformly chosen $\alpha \in [0,1]$ (an α-point) and prove that the expected weighted completion time is at most twice the LP optimum.

10. Let $X_1, \ldots, X_n$ be independent random variables taking values in $\{0, a_i\}$, where $0 < a_i \leq 1$, and let

$$X = \sum_{i=1}^{n} X_i, \qquad \mu = \mathbb{E}[X].$$

Prove a Chernoff-type upper-tail bound for $\Pr[X \geq (1+\delta)\mu]$ and explain how this bound can be used to control edge congestion in randomized rounding.

11. (*) Consider the minimum-capacity multicommodity flow problem: given an undirected graph and terminal pairs (s_i, t_i), choose one s_i–t_i path for every commodity so as to minimize the maximum number of paths using any edge. Given an optimal fractional multicommodity flow, design a randomized rounding algorithm that independently samples one path for each commodity. Prove, using Chernoff bounds and a union bound, that the maximum edge congestion is

$$O\left(\frac{\log n}{\log \log n}\right)$$

times the fractional optimum with high probability.

12. (*) Let $G = (V, E)$ be a δ-dense 3-colorable graph, i.e.,

$$|E| \geq \delta \binom{|V|}{2}$$

for a constant $\delta > 0$. Design a randomized sampling algorithm that selects a small set of vertices, uses their colors to infer the coloring of the remaining vertices, and produces a proper 3-coloring with high probability. Prove appropriate bounds on the sample size and the probability of failure.