

CS4810 / CS7800: Advanced Algorithms
Homework 4: Due Friday, October 28, 2022

Fall 2022

Mahsa Derakhshan and Jonathan Ullman

Collaboration and Honesty Policy Reminder. Collaboration in the form of discussion is allowed and encouraged. However, all forms of cheating are not allowed and will be penalized harshly. These include, but are not limited to, copying parts of an assignment from a classmate, finding answers to problems on the internet or from anyone not enrolled in the class, and plagiarizing from research papers or old posted solutions. A rule of thumb is that you should be able to walk away from discussing a homework problem with no notes and write your solution on your own.

- You must write up all solutions by yourself, and may not share any written solutions, even if you collaborate with others to solve the problem.
- You must identify all your collaborators. If you did not collaborate, write “no collaborators” or something to that effect. You may have a maximum of two collaborators per assignment, and collaboration is transitive (if you list a collaborator they must list you).
- Asking and answering questions in class forums (lectures, office hours, Piazza) is allowed and encouraged, and you do not need to list these interactions as collaborators.
- Seeking out alternative sources (e.g. classmates, textbooks, the internet) for general concepts you need (e.g. greedy algorithms, probability) is allowed and encouraged.

Assigned Problems

Problem 1 In this problem we will generalize the maximum (bipartite) matching problem to a weighted version. In the *maximum-weight (bipartite) matching problem* (MWM) you are given a bipartite graph $G = (L \cup R, E)$ with edge weights $\{w(e)\}$, and the goal is to find a matching $M \subseteq E$ maximizing the total weight $\sum_{e \in M} w(e)$.

Using a reduction to the minimum-cost perfect (bipartite) matching problem (MCPM), design an efficient algorithm for solving MWM. Justify the correctness of your reduction and analyze its running time, including the time required to solve MCPM. You will receive full credit for any efficient reduction.

Problem 2 As a top agent in the National Cyberwarfare Unit, you have been assigned the task of disrupting communication between enemy *operators* and their embedded *spies*. You have identified the communication network being used by the enemy, and have the network specified as a directed graph G with n nodes and m edges. You are also given a set of operator nodes O and spy nodes S and these sets are disjoint.

The goal is to determine the minimum number of edges to remove in order to make it impossible for any operator to contact any spy. That is, after removing these edges, no node $v \in S$ should be reachable via a directed path from any node $u \in O$. An example is illustrated in Figure 1.

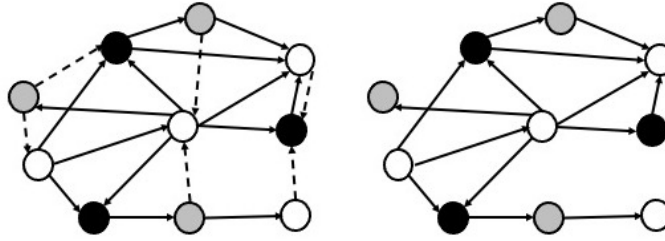


Figure 1: Communication network G shown on the left. The gray nodes are operators and the black nodes are spies. Removing the dashed edges makes it so no operator can reach any spy.

Using a reduction to one of the problems we've studied in class, design an efficient algorithm that determines the smallest number of edges to remove.

Problem 3 Each of the following questions describes a computational problem and asks you to write the problem as an equivalent linear program.

3.1 Multiple-goods maximum flow. In real flow networks we often need to share the network to route more than one type of good. We are given a directed, capacitated graph $G = (V, E, \{c(e)\})$. There are k goods and for each good i we must find a flow f_i to route the good from source s_i to sink t_i . We have three types of constraints to satisfy:

- (Non-negativity) For every edge, the flow of every good is non-negative.
- (Aggregate Capacity) For every edge, the sum of the flow of every good is at most the capacity.
- (Flow conservation) For every good, the flow is conserved except at that good's source or sink.

Our objective is to maximize the sum of the flow of all k goods. Show how to write this optimization problem as a linear program.

3.2 Soft linear classification. This is a problem that arises in machine learning. We are given m labeled pairs (x_i, y_i) where $x_i \in \mathbb{R}^n$ is a vector and $y_i \in \{-1, +1\}$ is a binary label. Ideally we would like to find a *linear classifier* $w \in \mathbb{R}^n$ such that $\text{sign}(w \cdot x_i) = y_i$ for every pair. However, such a classifier often doesn't exist. A common goal is to find a classifier that minimizes a *soft* version of this objective. Specifically, for every i we can compute the penalty

$$\max\{0, -y_i(w \cdot x_i)\}$$

which is 0 whenever $\text{sign}(w \cdot x_i) = y_i$ and is negative otherwise with magnitude proportional to the magnitude of the sign error. Our overall objective function becomes

$$\sum_{i=1}^n \max\{0, -y_i(w \cdot x_i)\}$$

Show how to write this *non-linear* objective function as a linear program.

Problem 4 Recall the Bake-Sale Problem from HW2. You need to staff an event from time 0 until time T . Each of n parents gave you a window of time $[s_i, f_i)$ that they are available. (*Note that we changed the problem slightly so that the parents' intervals do not include the time f_i . Parents are busy and will skip out a few seconds early if they can.*) Your goal is to find the smallest set of parents $S \subseteq \{1, \dots, n\}$ such that $\bigcup_{i \in S} [s_i, f_i) \supseteq [0, T]$. You may assume that all values $s_1, f_1, \dots, s_n, f_n$ are distinct and that using all n parents would successfully staff the event. In this question we'll take a fresh look at this problem using linear programming.

- 4.1 Show that there exists some finite set of points $F \subseteq [0, T]$ such that any set of parents that can staff the table at every time in F can also staff the table for all of $[0, T]$.
- 4.2 Formulate the Bake-Sale Problem as a linear program with decision variables x_1, \dots, x_n where x_i indicates whether or not parent i is used in the solution.
- 4.3 Write the dual of your LP.
- 4.4 Write an interpretation of the dual LP.
- 4.5 Prove that both the LP and its dual have *integral* optimal solutions.¹

¹**Hint:** You may want to use the fact that the greedy algorithm from HW2 solves this problem optimally.