Note: Your TA may not get to all the problems. This is totally fine, the discussion worksheets are not designed to be finished in an hour. The discussion worksheet is also a resource you can use to practice, reinforce, and build upon concepts discussed in lecture, readings, and the homework.

1 Linear Programming Basics

Plot the feasible region and identify the optimal solution for the following linear program.

$$\text{maximize } 5x + 3y$$
$$s.t. \quad 5x - 2y \geq 0, \quad x + y \leq 7, \quad x \leq 5, \quad x \geq 0, \quad y \geq 0$$

Suppose we want to maximize

$$\min\{5x, 3y\}$$

instead, subject to the same constraints. Describe how we can modify the LP to solve this problem by changing the objective, adding one new variable, and adding two new constraints.

2 Standard Form LP

Recall that any Linear Program can be reduced to a more constrained *standard form* where all variables are nonnegative, the constraints are given by equations and the objective is that of minimizing a cost function.

More formally, our variables are x_i. Our objective is $\min c^\top x = \sum_i c_i x_i$ for some constants c_i. The jth constraint is $\sum_i a_{ij} x_i = b_j$ for some constants a_{ij}, b_j. Finally, we also have the constraints $x_i \geq 0$.

An example standard form LP:

$$\text{minimize } 5x_1 + 3x_2$$
$$s.t. \quad x_1 + x_2 - x_3 = 1, \quad -x_1 + 2x_2 + x_4 = 0, \quad x_1, x_2, x_3, x_4 \geq 0$$

For each of the subparts, what system of variables, constraints, and objectives would be equivalent to the following:

(a) Max Objective: $\max \sum_i c_i x_i$

(b) Min Max Objective: $\min \max(y_1, y_2)$

(c) Upper Bound on Variable: $x_1 \leq b_1$

(d) Lower Bound on Variable: $x_2 \geq b_2$

(e) Bounded Variable: $b_2 \leq x_3 \leq b_1$

(f) Inequality Constraint: $x_1 + x_2 + x_3 \leq b_3$

(g) Unbounded Variable: $x_4 \in R$

3 An LP for Minimum Spanning Tree

Consider the minimum spanning tree problem, where we are given an undirected graph G with edge weights $w_{u,v}$ for every pair of vertices u, v.

An *integer* linear program that solves the minimum spanning tree problem is as follows:

$$\begin{aligned}
\text{Minimize} \quad & \sum_{(u,v)\in E} w_{u,v} x_{u,v} \\
\text{subject to} \quad & \sum_{\{u,v\}\in E: u\in S, v\in V\setminus S} x_{u,v} \geq 1 \quad \text{for all } S \subseteq V \text{ with } 0 < |S| < |V| \\
& x_{u,v} \in \{0,1\}, \quad \forall (u,v) \in E
\end{aligned}$$

(a) Show how to obtain a minimum spanning tree T of G from an optimal solution of the ILP, and prove that T is indeed an MST. Why do we need the constraint $x_{u,v} \in \{0,1\}$?

(b) How many constraints does the program have?

(c) Suppose that we *replaced* the binary constraint on each of the decision variables $x_{u,v}$ with the pair of constraints:

$$0 \leq x_{u,v} \leq 1, \quad \forall (u,v) \in E$$

How does this affect the optimal value of the program? Give an example of a graph where the optimal value of the relaxed linear program differs from the optimal value of the integer linear program.

4 Vertex Cover Rounding

In the vertex cover problem, we are given a graph G, and our goal is to find the smallest set of vertices S such that every edge has at least one endpoint in S.

(a) Let's write an integer linear program (ILP) for the vertex cover problem. There will be one variable x_v for every vertex, and we will set $x_v = 1$ if v is in our solution and $x_v = 0$ if v is not inour solution.

To finish writing the ILP, what is the objective function? What are the constraints?

(b) If we replace the requirement $x_v \in \{0, 1\}$ with the relaxed requirement $x_v \in [0, 1]$, we get a normal linear program (LP). LPs can be solved efficiently but ILPs cannot.

However, this efficiency does not come for free: Give an example of a graph where the optimal solution to this LP has objective function smaller than the size of the minimum vertex cover. (This suggests that we can't solve the vertex cover problem by just solving an LP for it.)

(c) Suppose someone solves this LP (*not* the ILP) and hands you the solution. Given *only* the solution and not G, how can we compute a vertex cover whose size is at most twice the fractional solution's objective function? e.g. if you're handed a solution which gets an objective function of 7, you should output a vertex cover of size at most 14.

(Hint: Include vertices whose x_v are large enough).