Homework 12

CSCI570 Spring 2025

Due: May 4, 2025

1. (**15 points**)

   A small coffee shop sells three types of coffee-based drinks. The first drink uses 10g coffee, 12g sugar, and 8 fl oz of milk. The second uses 16g coffee, 6g sugar, 3 fl oz milk. The third type uses 8g coffee, 15g sugar, 10 fl oz milk. The profit made on the three types of drinks is \$2, \$2.8, and \$1.5, respectively. The cafe makes 200 drinks of the first type, 120 of the second type, and 180 of the third type every day (with no ingredients leftover and there is enough demand that they're completely sold). On a certain day, there is a shortage of coffee by 600 g, and sugar by 400 g, but 2 gallons extra milk is available (1 gallon = 128 fl oz). How should the production of the three types of drinks be adjusted (increased or decreased as necessary) **to minimize the decrease in profit**? Formulate this problem as a linear programming problem (no need to solve the LP).

   (a) Define your variables. (Describe what they represent in English)

   (b) What is the objective function?

   (c) What are the constraints in your LP?

   Solution:

   (a) Suppose the number of drinks of each type that are produced **in addition** to the usual be x_1, x_2, x_3.

   (b) maximize: $2x_1 + 2.8x_2 + 1.5x_3$

   (c) subject to:
   $$10x_1 + 16x_2 + 8x_3 \leq -600,$$
   $$12x_1 + 6x_2 + 15x_3 \leq -400,$$
   $$8x_1 + 3x_2 + 10x_3 \leq 256,$$
   $$x_1 \geq -200,$$
   $$x_2 \geq -120,$$
   $$x_3 \geq -180.$$

   Explanation: The net additional profit is $2x_1 + 2.8x_2 + 1.5x_3$ which should be maximized (equivalent to minimizing decrease in profit). This objective is subject to the following constraints. The additional consumption of coffee is $10x_1 + 16x_2 + 8x_3$ g which can be at most 600 (owing to a shortage of 600), similarly, the additional consumption of sugar is $12x_1 + 6x_2 + 15x_3$ which can be at most -400 (owing to a shortage of 400), while the additional milk consumption is $8x_1 + 3x_2 + 10x_3$ which can be at most 256. The total number of drinks of each type is $200 + x_1, 120 + x_2, 180 + x_3$ resp., all of which should be non-negative. The resultant LP formulation is as follows:

   Alternatively, define x_1, x_2, x_3 as the decrements in drink quantities produced to get the following LP:

minimize:	$2x_1 + 2.8x_2 + 1.5x_3$
subject to:	$10x_1 + 16x_2 + 8x_3 \geq 600,$
	$12x_1 + 6x_2 + 15x_3 \geq 400,$
	$8x_1 + 3x_2 + 10x_3 \geq -256,$
	$x_1 \leq 200,$
	$x_2 \leq 120,$
	$x_3 \leq 180.$

Alternatively, define x_1, x_2, x_3 as the new drink quantities produced. This requires computing the default ingredient supplies as 5360g, 5820g and 3760 fl oz resp., and in turn, the new supplies as 4760g, 5420g and 4016. Then, we get the following LP:

maximize:	$2x_1 + 2.8x_2 + 1.5x_3$
subject to:	$10x_1 + 16x_2 + 8x_3 \leq 4760,$
	$12x_1 + 6x_2 + 15x_3 \leq 5420,$
	$8x_1 + 3x_2 + 10x_3 \leq 4016,$
	$x_1 \geq 0,$
	$x_2 \geq 0,$
	$x_3 \geq 0$

Rubrics (15 points):

- The scoring break-up is the same in all 3 cases: 3 for defining the variables and constraints, 3 for objective, 2+2+2+1+1+1 for constraints as shown.
- In each of the cases, the objective max $f(x)$ can be switched to min $-f(x)$ and vice versa.
- Any objective expression $f(x)$ can also be expressed as $f(x) + c$ for some constant c (e.g., the usual day's profit).

2. (**12 points**) A clique in a graph $G = (V, E)$ is a subset of vertices $C \subseteq V$ such that each pair of vertices in C is adjacent, i.e., $\forall u, v \in C, (u, v) \in E$. We want to find the largest clique (i.e., a clique with the maximum number of vertices) in a given graph. Write an integer linear program that will find the largest clique.

   (a) Define your variables. (Describe what they represent in English)

   (b) What is the objective function?

   (c) What are the constraints in your LP?

   Solution:
   Let x_u be an integer variable that gets a value 1 if u is included in the clique, and 0 otherwise. The ILP formulation is:

$$
\begin{array}{lll}
\text{maximize} & \displaystyle\sum_{u \in V} x_u & \\
\text{subject to} & x_u + x_v \leq 1 & \forall (u, v) \notin E \\
& x_u \in \{0, 1\}, & \forall u \in V
\end{array}
$$

   Alternatively, you can associate the integer variables with the exclusion of variables instead of inclusion, i.e., each x_u is 1 if u is excluded, and 0 if included. Then the ILP is:

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{u \in V} x_u & \\
\text{subject to} & x_u + x_v \geq 1 & \forall (u, v) \notin E \\
& x_u \in \{0, 1\}, & \forall u \in V
\end{array}
$$

   Rubrics (12 points):

   - 3 pts: Correctly defining the variables
   - 3 pt: Correct objective
   - 6 pts: Correct constraints (4+2)
   - Note that having extra variables and constraints in addition to what is required often makes the solution incorrect.

3. (**16 points**) In today's fast paced world, online grocery shopping is becoming increasingly common as customers can enjoy the comfortable option of home-delivered groceries. Suppose there are n stores in your area to choose from. Suppose you want to buy m items which are all <u>divisible</u> goods - i.e., you could buy them in quantities expressed in real numbers. Suppose store j sells item i for a price p_{ij} per unit. Your consumption needs you want to buy at least a_i units of item i in total, but due to the limitations of storage and preservation, you want to buy at most b_i units of it. To afford the transportation costs of delivering, the stores do not accept orders that are too small - store j requires the total order cost to be at least S_j (all $S_j' s$ are positive). Finally, you want to place the order with any particular store as a single transaction on your credit card which has a per-transaction cost limit of W.

   Grocery stores commonly sell some items for cheap to attract customers, and some items expensive to reap high profits. As a smart customer, you are planning to strategically spread your orders across various stores to minimize your expenditure - i.e. you want to decide how much of each item to order from each of the stores. Also, you are planning to place an order <u>with all n stores</u> not to leave any store unexplored. Formulate this problem as a Linear Program (LP).

   (a) Define your variables. (Describe what they represent in English)

   (b) What is the objective function?

   (c) What are the constraints in your LP?

   Solution:

   (a) X_{ij} denotes the quantity of item i ordered from store j.

   (b) $\min \sum_i \sum_j X_{ij} p_{ij}$

   (c) $a_i \leq \sum_j X_{ij} \leq b_i \qquad \forall i \in \{1, ..., m\}$
   $S_j \leq \sum_i X_{ij} p_{ij} \leq W \qquad \forall j \in \{1, ..., n\}$
   $0 \leq X_{ij} \qquad \forall i \in \{1, ..., m\} \ \forall j \in \{1, ..., n\}$

   Rubrics (16 points):

   - 3 pts: Correctly defining the variables
   - 3 pt: Correct objective
   - 10 pts: Correct constraints (4+4+2)

4. (**17 points**) Given a graph G and a partition (A, B), let $E(A, B)$ denote the set of edges with one endpoint in A and one endpoint in B. The Max Equal Cut problem is defined as follows
Instance Graph $G(V, E), V = 1, 2, ..., 2n$.
Question Find a partition of V into two n-vertex sets A and B, maximizing the size of $E(A, B)$.

Provide a $\frac{1}{2}$-approximation algorithm for solving the Max Equal Cut problem.
(Hint: Consider an algorithm that iteratively builds the partition (A, B) in a *greedy* manner while ensuring A, B have equal size at any point.)

Solution:
Start with empty sets A, B, and perform n iterations:
In iteration i, pick vertices $2i - 1$ and $2i$, and place one of them in A and the other in B, according to which choice maximizes $|E(A, B)|$ at that point.
Explanation: In a particular iteration, when we have cut (A, B) and we want to add u and v, suppose u has N_{A_u}, N_{B_u} neighbours in A, B respectively and suppose v has N_{A_v}, N_{B_v} neighbours in A, B respectively. Then, adding u to A and v to B adds $N_{B_u} + N_{A_v}$ edges to the cut, whereas doing the other way round adds $N_{B_v} + N_{A_u}$ edges to the cut. Since the sum of these two options is nothing but the total number of edges being added to this partial subgraph, the bigger of the two must be at least half the total number of edges being added to this partial subgraph (a subtle point to consider is whether u and v have an edge between them and it can be seen that the argument holds in either case). Since this is true for each iteration, at the end, when all the nodes (and edges) are added, our algorithm is bound to add at least half of the total $|E|$ edges. Naturally since the max equal cut capacity $OPT \leq |E|$, our solution is a $\frac{1}{2}$-approximation.

Rubrics (16 points):

- 8 pts: Algorithm
- 9 pts: Proving the approximation bound

Ungraded problems

For all of the LP problems, follow the structure:

(a) Define your variables. (Describe what they represent in English)

(b) What is the objective function?

(c) What are the constraints in your LP?

5. Write down the problem of finding a Min-s-t-Cut of a directed network with source s and sink t as an Integer Linear Program and explain your program.

The variable x_u indicates if the vertex u is on the side of s in the cut. That is, $x_u = 1$ if and only if u is on the side of s. Likewise, the variable $x_{(u,v)}$ indicates if the edge (u, v) crosses the cut.

$$\begin{aligned} \text{minimize} \sum_{(u,v)\in E} c(u,v).x_{(u,v)} \\ \text{subject to} : x_v - x_u + x_{(u,v)} \geq 0 \quad \forall (u,v) \in E \\ x_u \in \{0,1\} \quad \forall u \in V : u \neq s, u \neq t \\ x_{(u,v)} \in \{0,1\} \quad \forall (u,v) \in E \\ x_s = 1 \\ x_t = 0 \end{aligned}$$

Setting $x_s = 1$ and $x_t = 0$ ensures that s and t are separated. The first constraint ensures that if u is on the side of s and v is on the side of t, then the edge (u, v) must be included in the cut. For completeness, one should argue that with the above correspondence (that is, $x_{(u,v)}$ indicating if an edge crosses the cut), every min-s-t-cut corresponds to a feasible solution and vice versa.

6. The edge-coloring problem is to color the edges of a graph with the fewest number of colors in such a way any two edges that share a vertex have different colors. Suppose you are given the algorithm ALG that colors a graph with at most $m/n + d$ colors when the graph has m edges, n vertices, and a maximum vertex-degree of d (We do not need to know how the algorithm works). Prove that this algorithm is a $\frac{3}{2}$-approximation algorithm for the edge-coloring problem. You may assume that the input graph is not empty (i.e., has at least one edge).

   Solution:
   The maximum degree of the graph is d. This implies that there exists at least one vertex (say v) which is connected to d other vertices. All the d edges connected to v would need to have different colors to satisfy the edge coloring. Therefore, the lower bound on the optimal solution is $OPT \geq d$.

   Our algorithm ALG returns a coloring of at most $m/n+d$ colors. Since max-degree is d, the number of edges m is at most $nd/2$, and thus $m/n \leq d/2$ Hence, the upper bound of the algorithm is $ALG \leq 3d/2$. The approximation ratio ρ would be upper bound on $\frac{ALG}{OPT}$ which is the upper bound of ALG divided by the lower bound of the optimal algorithm OPT which is:

$$\rho = \frac{3d/2}{d} = 3/2$$

7. 720 students have pre-enrolled for the "Analysis of Algorithms" class in Fall. Each student must attend one of the 16 discussion sections, and each discussion section i has capacity for D_i students. The **aggregate** happiness level of students assigned to a discussion section i is proportionate to $\alpha_i(D_i - S_i)$, where α_i is a known parameter reflecting how well the air-conditioning system works for the room used for section i (the higher the better), and S_i is the actual number of students assigned to that section. We want to find out how many students to assign to each section in order to maximize total student happiness. Express the problem as an integer linear program problem.

   Solution:
   Our variables will be the S_i. Our objective function is:

$$\begin{aligned} \text{maximize} \sum_{i=1}^{16} \alpha_i(D_i - S_i) \\ \text{subject to: } D_i - S_i \geq 0 \text{ for } 0 < i \leq 16 \\ S_i \geq 0 \text{ for } 0 < i \leq 16 \\ \sum_{i=1}^{16} S_i = 720 \end{aligned}$$

8. Mary is a primary school teacher. She decides to distribute candy to the students in her class to congratulate them on their exam grades. There are n students sitting in a line. The j^{th} student in the line has grades g_j. Mary has m different type of candy, and N_i pieces available for each candy type i. A single piece of candy i costs c_i.

   Mary wants to distribute the candy satisfying the following criteria:

   (a) Each student receives at least 3 pieces of candy, and each should be of different kind.

   (b) For any students s and s' that are sitting next to each other in the line:

      i. If they have equal grades, they should get equal pieces of candy, if not, the one with higher grades should get more pieces.

      ii. They should not both get the candy of any same kind.

   Formulate a linear program to model this problem with an objective to minimize the overall money spent by Mary.

   Solution:

   Let X_{ij} denote the number of pieces of candy i given to student j (X_{ij} can also be declared binary since it is not more than 1 as per condition a)).

   (To easily understand the solution, note that summing the variables X_{ij} over i gives us the total no. of pieces student j has, and similarly, summing them over j gives us the total no. of pieces of candy i used.)

$$
\begin{aligned}
\min \quad & \sum_i \sum_j c_i X_{ij} && && (1)\\
\text{s.t.} \quad & \sum_i X_{ij} \geq 3 && \forall j \in \{1, \dots, n\} && (2)\\
& X_{ij} \leq 1 && \forall j \in \{1, \dots, n\}\ \forall i \in \{1, \dots, m\} && (3)\\
& \sum_i X_{ij} = \sum_i X_{i(j+1)} && \forall j \in \{1, \dots, n-1\}, g_j = g_{(j+1)} && (4)\\
& \sum_i X_{ij} \geq \sum_i X_{i(j+1)} + 1 && \forall j \in \{1, \dots, n-1\}, g_j > g_{(j+1)} && (5)\\
& \sum_i X_{ij} \leq \sum_i X_{i(j+1)} - 1 && \forall j \in \{1, \dots, n-1\}, g_j < g_{(j+1)} && (6)\\
& X_{ij} + X_{i(j+1)} \leq 1 && \forall j \in \{1, \dots, n-1\}\ \forall i \in \{1, \dots, m\} && (7)\\
& \sum_j X_{ij} \leq N_i && \forall i \in \{1, \dots, m\} && (8)\\
& X_{ij} \geq 0 && \forall j \in \{1, \dots, n\}\ \forall i \in \{1, \dots, m\} && (9)\\
& X_{ij} \in \mathbb{Z} && \forall j \in \{1, \dots, n\}\ \forall i \in \{1, \dots, m\} && (10)
\end{aligned}
$$

   - Objective (1) Minimizes the total cost of all candy pieces distributed.
   - Constraints (2) & (3) reflect the two parts of condition a) respectively.
   - Constraints (4) - (6) capture condition b.i) by considering all the 3 cases for the grades of adjacent students i and i+1.

- (7) captures condition b.ii).
- (8) captures the condition on availability of each candy.
- (9) and (10) are necessary to reflect that 'no. of pieces' needs to be a non-negative integer quantity.

Some additional comments:

(3), (9), (10) could be combined to simply say $X_{ij} \in \{0, 1\}$ as mentioned earlier. **Unless the latter is clearly and correctly used**, all the three constraints are individually necessary.

The question did not specifically use the term 'integer linear program', so no penalty for missing (10).

(7) guarantees (3), so, **IF (7) is completely correct**, (3) can be omitted without a penalty.

9. A set of n space stations need your help in building a radar system to track spaceships traveling between them. The i^{th} space station is located in 3D space at coordinates (x_i, y_i, z_i). The space stations never move. Each space station i will have a radar with power r_i, where r_i is to be determined. You want to figure how powerful to make each space station's radar transmitter, so that whenever any spaceship travels in a straight line from one station to another, it will always be in radar range of either the first space station (its origin) or the second space station (its destination). A radar with power r is capable of tracking space ships anywhere in the sphere with radius r centered at itself. Thus, a space ship is within radar range through its strip from space station i to space station j if every point along the line from (x_i, y_i, z_i) to (x_j, y_j, z_j) falls within either the sphere of radius r_i centered at (x_i, y_i, z_i) or the sphere of radius r_j centered at (x_j, y_j, z_j). The cost of each radar transmitter is proportional to its power, and you want to minimize the total cost of all of the radar transmitters. You are given all of the $(x_1, y_1, z_1), ..., (x_n, y_n, z_n)$ values, and your job is to choose values for $r_1, ..., r_n$. Express this problem as a linear program.

(a) Describe your variables for the linear program (3 pts).
Solution:
r_i=the power of the i^{th} radar transmitter., i=1,2,...n (3 pts)

(b) Write out the objective function (5 pts).
Solution:
Minimize $r_1 + r_2 + ... + r_n$ or $\sum_i^n r_i$
Defining the objective function without mentioning r_i: -3 pts

(c) Describe the set of constraints for LP. You need to specify the number of constraints needed and describe what each constraint represents (8 pts).
Solution:
$r_i + r_j \geq \sqrt{((x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2)}$. Or, $r_i + r_j \geq d_{i,j}$ for each pair of stations i and j, where $d_{i,j}$ is the distance from station i to station j (6 pts).
We need $\sum_{i=1}^{n-1} i = (n^2 - n)/2$ constraints of inequality (The number of constraints is due to the number of unique paths between each pair of space stations) (2pts).