CSCI 570 Homework 5
Spring 2023

Due Date: Apr. 13, 2023 at 11:59 P.M.

1. There is a precious diamond that is on display in a museum at m disjoint time intervals. There are n security guards who can be deployed to protect the precious diamond. Each guard has a list of intervals for which he or she is available to be deployed. Each guard can be deployed to at most M time slots and has to be deployed to at least L time slots. Design an algorithm that decides if there is a deployment of guards to intervals such that each interval has either one or two guards deployed.

   solution:

   We create a circulation network as follows. For the i-th guard, introduce a vertex gi and for the j-th time interval, introduce a vertex tj. If the i-th guard is available for the jth interval, then introduce an edge from gi to tj of capacity 1. Add a source s and a sink t. To every guard vertex add an edge from s of capacity M and lower bound L. From every interval vertex add an edge to t of capacity 2 and lower bound 1. Add an edge from t to s of infinite capacity.

   Next we reduce this to a problem with no lower bounds. We will get the following demands: d(s) = n L, d(gi) = -L, d(tj) = 1, d(t) = -M.

   Then we reduce it to a flow problem, by creating a super source S and super sink T. We connect S to t and gi. We connect s and tj to T. We also reduce capacities.

   Claim: there exists a valid deployment if and only if the above network has a max flow equal to m + n L.

2. A company makes three products and has 4 available manufacturing plants. The production time (in minutes) per unit produced varies from plant to plant as shown below:

   Similarly, the profit ($) contribution per unit varies from plant to plant as below:

   If, one week, there are 35 working hours available at each manufacturing plant how much of each product should be produced given that we need

		Manufacturing Plant			
		1	2	3	4
Product	1	5	7	4	10
	2	6	12	8	15
	3	13	14	9	17

		Manufacturing Plant			
		1	2	3	4
Product	1	10	8	6	9
	2	18	20	15	17
	3	15	16	13	17

at least 100 units of product 1, 150 units of product 2, and 100 units of product 3. The goal is to maximize the total profit. Formulate this problem as a linear program. You do not have to solve the resulting LP.

solution:

Variables

At first sight, we are trying to decide how much of each product to make. However, on closer inspection, it is clear that we need to decide how much of each product to make at each plant. Hence let x_{ij} = amount of product $i (i = 1, 2, 3)$ made at plant $j (j = 1, 2, 3, 4)$ per week. Although (strictly) all the x_{ij} variables should be integers they are likely to be quite large and so we let them take fractional values and ignore any fractional parts in the numerical solution. Note that the question explicitly asks us to formulate the problem as an LP rather than as an IP.

Constraints

We first formulate each constraint in words and then in a mathematical way. Limit the number of minutes available each week for each workstation

$$5x_{11} + 6x_{21} + 13x_{31} \leq 35$$

$$7x_{12} + 12x_{22} + 14x_{32} \leq 35$$

$$4x_{13} + 8x_{23} + 9x_{33} \leq 35$$

$$10x_{14} + 15x_{24} + 17x_{34} \leq 35$$

A lower limit on the total amount of each product produced

$$x_{11} + x_{12} + x_{13} + x_{14} \geq 100$$

$$x_{21} + x_{22} + x_{23} + x_{24} \geq 150$$

$$x_{31} + x_{32} + x_{33} + x_{34} \geq 100$$

All variables are greater than equal to zero.

Objective

Maximize

$$10x_{11}+8x_{12}+6x_{13}+9x_{14}+18x_{21}+20x_{22}+15x_{23}+17x_{24}+15x_{31}+16x_{32}+13x_{33}+17x_{34}$$

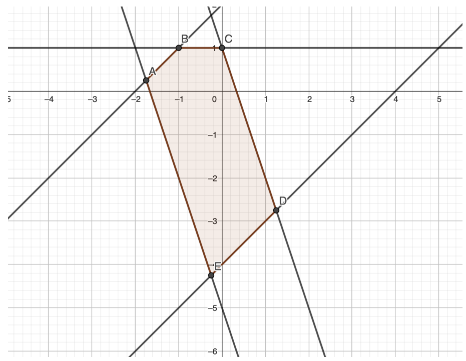

Figure 1: The feasible region

3. Solve the following linear program using the fundamental theorem. Specifically, find all vertices of the feasible region, calculate the values of the objective function at those points, and conclude the optimal solution. (*Hint: plot the feasible region in 2D*)

$$\max(-x_1 + 4x_2)$$

subject to

$$\begin{aligned} 3x_1 + x_2 &\leq 1 \\ 3x_1 + x_2 &\geq -5 \\ x_1 - x_2 &\leq 4 \\ x_1 - x_2 &\geq -2 \\ x_2 &\leq 1 \end{aligned}$$

solution:
We plot the feasible region in Figure 1. There are 5 vertices $(-1, 1)$, $(0, 1)$, $(-1.75, 0.25)$, $(1.25, -2.75)$, $(-0.25, -4.25)$ with values 5, 4, 2.75, -12.25, -16.75. Therefore, we conclude the solution is 5.

4. There are m basic nutritional ingredients, and Andy has to receive at least b_i units of the i-th nutrient per day to satisfy the basic minimum nutritional requirements. There are n available foods. The j-th food sells at a price c_j per unit. Additionally, each unit of the j-th food contains a_{ij} units of the i-th nutrient. Help Andy find the lowest cost per day to

satisfy the requirement. Formulate this problem as a linear programming problem in the standard form.

solution:

Denote the number of units of food j by x_j. The total cost is $\sum_{j=1}^{n} c_j x_j$ and the problem is to minimize the cost, which is equivalent to

$$\max_x \sum_{j=1}^{n} (-c_j) x_j.$$

As for the constraints, the units are non-negative, so we have $x_j \geq 0,\ \forall j$. Additionally, x_j units of food j provide $a_{ij} x_j$ units of the i-th nutrient. In total, all n foods provide $\sum_{j=1}^{n} a_{ij} x_j$ units of food j, which has to be at least b_i to satisfy the requirement. Therefore, we have

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i,$$

which is equivalent to $\sum_{j=1}^{n} (-a_{ij}) x_j \leq -b_i$. In summery, the constraints are

$$\sum_{j=1}^{n} (-a_{ij}) x_j \leq -b_i,\ \forall i = 1, 2, \ldots, m$$
$$x_j \geq 0,\ \forall j = 1, 2, \ldots, n.$$

5. Given an undirected graph $G = (V, E)$, a vertex cover is a subset of V so that every edge in E has at least one endpoint in the vertex cover. The problem of finding a minimum vertex cover is to find a vertex cover of the smallest possible size. Formulate this problem as an integer linear programming problem.

   solution:

   For every vertex $v \in V$, introduce a variable x_v and consider the following integer linear program:

$$\min \sum_{v \in V} x_v$$

subject to

$$x_u + x_v \geq 1, \quad \forall (u,v) \in E$$
$$x_v \in \{0,1\}, \quad \forall v \in V$$

In the following, we prove that G has a vertex cover of size k if only if the above linear program with the objective being k. Suppose the G has a vertex cover A of size k. Set $x_v = 1$ if $v \in A$ and $x_v = 0$ if $v \notin A$. Since A is a vertex cover, for every edge $(u,v) \in E$, either $u \in A$ or $v \in A$, which ensures that $x_u + x_v \geq 1$. Hence all the constraints in the integer linear program are satisfied, which shows that x obtained from A is a feasible solution. Moreover, the objective is $\sum_{v \in V} x_v = |A| = k$.

Conversely, assume that there exists a solution x' (that is, each x'_v is assigned a 0/1 integer such that all the constraints are satisfied) to the integer linear program such that $\sum_{v \in V} x'_v = k$. If $x'_v = 1$, we add v to set B. For every edge $(u,v) \in E$, since $x'_u + x'_v \geq 1$, either $x'_u \geq 1$ or $x'_v \geq 1$. Thus for every edge $(u,v) \in E$, either $u \in B$ or $v \in B$, which implies that B is a vertex cover of G. Moreover, the size of B is $\sum_{v \in V} x'_v = k$

Thus G has a vertex cover of size k if and only if the corresponding integer linear program has a solution where the objective is k.

6. Write down the dual program of the following linear program. There is no need to provide intermediate steps.

$$\max(x_1 - 3x_2 + 4x_3 - x_4)$$

subject to

$$\begin{aligned} x_1 - x_2 - 3x_3 &\leq -1 \\ x_2 + 3x_3 &\leq 5 \\ x_3 &\leq 1 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}$$

solution:

$$\min(-y_1 + 5y_2 + y_3)$$

subject to

$$\begin{aligned} y_1 &\geq 1 \\ -y_1 + y_2 &\geq -3 \\ -3y_1 + 3y_2 + y_3 &\geq 4 \\ y_1, y_2, y_3 &\geq 0 \end{aligned}$$

We also have a trivial constraint $0 \geq -1$.

7. Determine whether the following linear programs are feasible bounded, feasible unbounded, or infeasible.

(a)

$$\max(x_1 + x_2)$$

subject to

$$\begin{aligned} x_1 + 2x_2 &\leq 3 \\ 3x_1 - x_2 &\leq 2 \\ -4x_1 - x_2 &\leq 2 \end{aligned}$$

(b)

$$\max(x_1 + x_2)$$

subject to

$$x_1 + 2x_2 \geq 3$$
$$3x_1 - x_2 \geq 2$$
$$-4x_1 - x_2 \geq 2$$

(c)

$$\max(x_1 + x_2)$$

subject to

$$x_1 + 2x_2 \geq 3$$
$$3x_1 - x_2 \leq 2$$
$$-4x_1 - x_2 \leq 2$$

solution:

(a) The LP is feasible and bounded as shown in Figure 2

(b) The LP is infeasible as the first two inequalities imply $4x_1 + x_2 \geq 5$ but the last one implies $4x_1 + x_2 \leq -2$

(c) The LP is feasible but not bounded as shown in Figure 3. For example, $x_1 = 1$ and every $x_2 \geq 1$ are feasible but the objective $x_1 + x_2$ is unbounded in this case.

8. Show that vertex cover remains NP-Complete even if the instances are restricted to graphs with only even degree vertices.

   solution: We claim that vertex cover with only even degree vertices remains NP-Complete by showing it is polynomial time reducible from the original vertex cover problem.

   Let $(G = (V, E), k)$ be an input instance of Vertex Cover where k is the size of the vertex cover. Because each edge in E contributes a count of 1 to the degree of each of the vertices with which it connects, the sum of the degrees of the vertices is exactly $2|E|$, an even number. Hence, there is an even number of vertices in G that have odd degrees.

   Let U be the subset of vertices with odd degrees in G and $|U|$ is even.

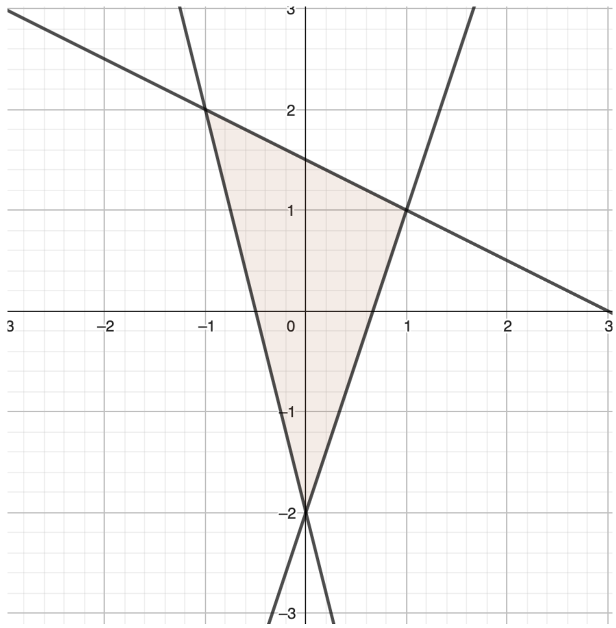

Figure 2: 6(a)

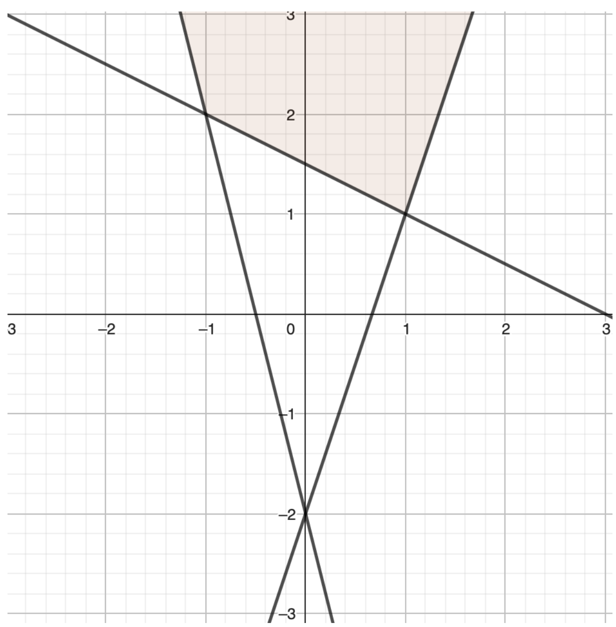

Figure 3: 6(c)

Construct a new instance ($G_0 = (V_0, E_0)$, $k + 2$) of Vertex Cover, where $V_0 = V \cup \{x, y, z\}$ and $E_0 = E \cup \{(x, y), (y, z), (z, x)\} \cup \{(x, v)|v \in U\}$. That is, we make a triangle with three new vertices, and then connect one of them (say x) to all the vertices in U. The degree of every vertex in V_0 is even. Since a vertex cover for a triangle is of (minimum) size 2, it is clear that G_0 has a vertex cover of size $k+2$ if and only if G has a vertex cover of size k.

Hence, vertex cover with only even degree vertices is NP-Complete.

9. Assume that you are given a polynomial time algorithm that given a 3-SAT instance decides in polynomial time if it has a satisfying assignment. Describe a polynomial time algorithm that finds a satisfying assignment (if it exists) to a given 3-SAT instance.

solutions.
The language for the formula satisfiability problem is 3-SAT= $\{\phi_i : \phi$ is a satisfiable boolean formula$\}$. Let ϕ be a boolean formula and A be the polynomial-time algorithm to decide the 3-SAT. ϕ has n variables, denoted as $\{x_1, x_2, ...x_n\}$. Denote $\phi(x_i = 0)$ and $\phi(x_i = 1)$ as the formulas when a variable $x_i, i \in \{1, 2, 3\}$ is assigned 0 or 1, respectively. The following procedure can find a satisfying assignment:

1 Apply A on ϕ to decide whether ϕ is satisfiable. If NO, return "no satisfying assignment can be found"; else continue.

2 Initialize $i = 1$.

3 Let $x_i = 0$, then $\phi(x_i = 0)$ is a boolean formula with $n - i$ variables. Apply A to check the satisfiability of $\phi(x_i = 0)$. If it returns YES, record $x_i = 0$. Else record $x_i = 1$. Since ϕ is satisfiable, there must be an assignment of x_i satisfies ϕ;

4 Replace x_i in ϕ by the recorded value from step (3), then increase i by 1, repeat step (3) until $i = n$

5 Return the recorded assignments for all variables $\{x_1, x_2, ...x_n\}$.

10. **Online Questions.** Please go to DEN (https://courses.uscden.net/) and take the online portion of your assignment.