CSCI 570 – Homework 5
Due: April 16th, 2021

Name: Rohan Mahendra Chaudhari
USC ID: 6675-5653-85
Email ID: rmchaudh@usc.edu

1. A furniture company produces three types of couches. The first type uses 1 foot of framing wood and 3 feet of cabinet wood. The second type uses 2 feet of framing wood and 2 feet of cabinet wood. The third type uses 2 feet of framing wood and 1 foot of cabinet wood. The profit of the three types of couches is \$10, \$8, and \$5, respectively. The factory produces 500 couches each month of the first type, 300 of the second type, and 200 of the third type. However, this month there is a shortage of cabinet wood by 600 feet, but the supply of framing wood is increased by 100 feet. How should the production of the three types of couches be adjusted to minimize the decrease in profit? Formulate this problem as a linear programming problem.

   <u>Solution</u>:

   Let x_1, x_2, x_3 be the change in the number of couches of type 1, 2 and 3 produced each month respectively – positive for increase in production and negative for decrease.

   The change in profit is then

   $10x_1 + 8x_2 + 5x_3$

   Since supply of framing wood will be increased by 100 feet, the change in the amount of framing wood used will be

   $x_1 + 2x_2 + 2x_3 <= 100$

   whereas due to the shortage of cabinet wood by 600 feet, the change in amount of cabinet wood will be

   $3x_1 + 2x_2 + x_3 <= -600$

   Finally, since the number of each type of couch produced cannot be less than zero, we have

   $x_1 >= -500$, $x_2 >= -300$, $x_3 >= -200$

Minimizing the loss means maximizing change in profit. Thus, the LP is

Maximize $10x_1 + 8x_2 + 5x_3$
Subject to $x_1 + 2x_2 + 2x_3 <= 100$
$3x_1 + 2x_2 + x_3 <= -600$
$x_1 >= -500$, $x_2 >= -300$, $x_3 >= -200$

OR

Minimize $-10x_1 – 8x_2 – 5x_3$
Subject to $x_1 + 2x_2 + 2x_3 <= 100$
$3x_1 + 2x_2 + x_3 <= -600$
$x_1 >= -500$, $x_2 >= -300$, $x_3 >= -200$

This problem is not in the form of a standard maximum problem, since we do not have the positivity condition.

2. Consider the following linear program:

$\max(3x_1 + 2x_2 + x_3)$

subject to
$x_1 – x_2 + x_3 <= 4$
$2x_1 +x_2 +3x_3 <=6$
$-x_1 + 2x_3 = 3$
$x_1 + x_2 + x_3 <= 8$
$x_1,x_2,x_3 \geq 0$
Write the dual problem. You do not need to demonstrate intermediate steps.

Solution:

$x_1 – x_2 + x_3 <= 4$
$2x_1 + x_2 + 3x_3 <= 6$
$-x_1 + 2x_3 = 3$
$x_1 + x_2 + x_3 <= 8$
$x_1,x_2,x_3 >= 0$

Rewrite the equality constraint $-x_1 + 2x_3 = 3$ as follows: $-x_1 + 2x_3 <= 3$ and $x_1 – 2x_3 <= -3$

Multiple each equation by a new variable $y_k >= 0$
$y_1(x_1 – x_2 + x_3) + y_2(2x_1 + x_2 + 3x_3) + y_3(x_1 + x_2 + x_3) + y_4(-x_1 + 2x_3) + y_5(x_1 – 2x_3) <= 4y_1 + 6y_2 + 8y_3 + 3y_4 – 3y_5$

Collect terms with respect to x_k
$x_1(y_1 + 2y_2 + y_3 – y_4 + y_5) + x_2(-y_1 + y_2 + y_3) + x_3(y_1 + 3y_2 + y_3 + 2y_4 – 2y_5) <= 4y_1 + 6y_2 + 8y_3 + 3y_4 – 3y_5$

Choose y_k in a way that $A^Ty >= c$
Note that $b^T = (4,6,8,3,-3)$ and $c^T = (3,2,1)$.

The dual problem is:

$\min(4y_1 + 6y_2 + 8y_3 +3y_4 – 3y_5)$

subject to
$y_1 + 2y_2 + y_3 – y_4 + y_5 >= 3$
$-y_1 + y_2 + y_3 >= 2$
$2y_1 + 3y_2 + y_3 + 2y_4 – 2y_5 >= 1$
$y_1,y_2,y_3,y_4,y_5 >= 0$

Or equivalently, letting $w = y_4 – y_5$ unconstrained. The dual problem is:
$\text{Min}(4y_1 + 6y_2 + 8y_3 + 3w)$

subject to
$y_1 + 2y_2 + y_3 – w >= 3$
$-y_1 + y_2 + y_3 >= 2$
$y_1 + 3y_2 + y_3 + 2w >= 1$
$y_1,y_2,y_3 >= 0$

3. Spectrum management is the process of regulating the use of radio frequencies to promote efficient use and gain a net social benefit. Given a set of broadcast emitting stations $s_1,...,s_n$, a list of frequencies $f_1,...,f_m$, where $m \geq n$, and the set of adjacent stations $\{(s_i,s_j)\}$ for some $i,j \in [n]$. The goal is to assign a frequency to each station so that adjacent stations use different frequencies and the number of used frequencies is minimized. Formulate this problem as an integer linear programming problem.

Solution:

We make use of binary variables a_{ij} where $1 <= i,j <= n$ to denote whether station s_i uses frequency f_j. There would be n^2 a_{ij} binary variables. Additionally, we introduce another binary variable x_j for each frequency f_j, denoting whether frequency f_j is used or not. From the question it can be noted that since there are n stations, we would require n frequencies for the assignment. The ILP is:

$$\text{Min} \sum_{k=1}^{n} x_k$$

Subject to

For all(i,i') which denote adjacent stations and for all(j) which denote frequencies such that the edge between 2 stations belongs to E and i,j belongs to [n]

$a_{i1}+a_{i2}+\ldots.+a_{in} = 1$ – (1)
$a_{ij} + a_{i'j} <= 1$ - (2)
$a_{ij} <= x_j$ - (3)
x_j, a_{ij} belongs to {0,1}

The first inequality (1) denotes that each station must have exactly one frequency. The second inequality (2) denotes that adjacent stations must have different frequencies. The third inequality (3) means we can only use those frequencies which are available to use/active. Thus the above inequality is formulated as an Integer Linear programming problem.

4. Prove or disprove the following statements.

Solution:

a. False. Let A be the empty language. It is trivially reducible to, say 3SAT: transform any input into the formula $a \wedge \neg a$. But only the empty language is reducible to the empty language, thus A is not NP-hard. One can transform the empty language ∅ to any NP-hard problem, but the empty language is not NP-hard.
b. True. One can construct a certifier for A by composing the certifier for B and the polynomial reduction map.

c. True. Since 2−SAT is in P and 3−SAT is in NP and the reduction shows that 2−SAT is in NP, we have P = NP.
d. True. Since one can use exponential time to try all possible certificates, each of them can be checked by a polynomial time deterministic Turing machine. This class of decision problems is called EXPTIME.
e. False. Let A be a problem in P and B be a problem that requires exponential time to solve. If $B \leq_p A$, then there exists a polynomial time algorithm for B, which leads to an contradiction.

5. 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.

Solution:

Given an instance of 3-SAT consisting of variables $X = \{x_1 , . . . , x_n \}$ and clauses $C_1,\ldots,C_k$.
Let $b(x_1,\ldots,x_n) = C_1 \wedge \cdots \wedge C_k$ be the corresponding Boolean formula.
Define b_i as the CNF formula obtained from b by setting x_i to TRUE and FALSE, respectively.
Let Algo(b) be the given algorithm that decides whether a given instance has a satisfying assignment.
Suppose Algo(b) returns TRUE, then b is satisfiable, otherwise Algo(b) returns FALSE.
Note that b_i is satisfiable if and only if b has a satisfying assignment where x_i = TRUE. Then the following algorithm constructs satisfying assignment for b if possible, or returns that no such assignment exists.

```
If Algo(b) == FALSE:
    return b is not satisfiable
for i in range(1,n):
    If Algo(bi) == TRUE:
            b ← bi
            xi ← True
    else:
            b ← bi
            xi ← False
Return x = (x1,…,xn)
```

6. The government wants to build a multi-lane highway across the country. The plan is to choose a route and rebuild the roads along this route. We model this problem with a simple weighted undirected graph with the nodes denoting the cities and the edges capturing the existing road network. The weights of the edges denote the length of the road connecting the corresponding two cities.

   Let d_{uv} denote the shortest path distance between nodes u and v.

   Let d(v,P) denote the shortest path distance from a node v to the closest node on a path P (i.e. min d_{uv}).

   Next, we define the aggregate remoteness of P as r(P) = Σ d(v, P).

   In the example shown below, path P = ABCD is the highway. Then, d(A,P) = d(B,P) = d(C,P) = d(D,P) = 0, while d(X,P) = d_{XB} = 2, d(Y,P) = d_{YB} = 5, and, d(Z,P) = d_{ZC} = 7. The remoteness of path ABCD is 0 + 0 + 0 + 0 + 2 + 5 + 7 = 14.

   The government wants a highway with the minimum aggregate remoteness, so that all the cities are somewhat close to the highway. Formally, we state the problem as follows, "Given a graph G, and a number k, does there exist a path P in G having remoteness r(P) at most k"? Show that this problem is NP-complete by reduction from a Hamiltonian Path.

   <u>Solution</u>:

   1. To show that this problem is in NP.

      Certificate: Suppose we have a highway path with min aggregate remoteness r(P).

      Certifier: A simple highway path with min remoteness r(P) of at most k value can be verified by traversing the highway path in linear time and calculating the min remoteness of a highway path P by adding d(v,P) which denotes the min distance between a vertex 'v' and a node on the highway path P. In addition to this, we also check if the total min remoteness of the path P is <= k, if all the edges of the path are in E, if each adjacent vertex of the path are connected by an edge and if the cities are connected to any one of the nodes on the highway path P. If these conditions do not meet, highway path P is not the path we are looking for.

2. To show that the problem is in NP Hard

   Claim: G has an Hamiltonian path if and only if G’ has a simple path with the minimum aggregate remoteness along that path being at-most k.

   =>) By construction.

   Given for all graph G with a Hamiltonian path. Let HP(G) denote a particular Hamiltonian path in G. We then consider this Hamiltonian path and construct a specific G’ with the Hamiltonian path as the main highway path by minimizing ‘k’ to the value 0 as ‘k’ can hold any value.

   Given G has a HP, we must find the minimum aggregate remoteness which is at most k. In order to find this we must follow the Hamiltonian path and calculate the min remoteness along that path. Since the value of k considered is 0 for a specific case of G’, we can state that the problem in G’ becomes a Hamiltonian path problem, since the remoteness between any 2 nodes of a highway path/Hamiltonian path would be 0 in G’, then the minimum aggregate remoteness would also be 0 which would satisfy the condition that the remoteness is at most k.

   <=) Given G’ has a path with minimum aggregate remoteness value of at most k.

   Given G’ with a path with min remoteness of at most k. We need to find a Hamiltonian path in G. We know for a fact that the path in G’ with min remoteness of at most k value is a Hamiltonian path when ‘k’ is 0 as considered for our specific case of G’. Thus, since it is a Hamiltonian path we can be sure that every vertex of the graph is visited exactly once. Therefore, since we get a Hamiltonian path problem by reduction and we know that Hamiltonian path problem is NP-Complete, thus, we can conclude that this problem is NP-Complete.