CSCI 570 - Analysis of Algorithms
Assignment 5
Harshitha Kurra

---

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 (with no wood resources leftover). 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.**

Solution:

	Framing Wood (in feet)	Cabinet Wood (in feet)	Profit	Number of couches produced
Couch 1	1	3	$10	500
Couch 2	2	2	$8	300
Couch 3	2	1	$5	200

Let x_i be the change in the number of couches produced each month. x_i will be positive if the production increases and negative if the production decreases. Therefore, our profit will be written as

$$\Delta p = 10x_1 + 8x_2 + 5x_3$$

Since we are told that the supply of framing wood would be increased by 100, we write it as

$$x_1 + 2x_2 + 2x_3 \leq 100$$

We are told the supply of cabinet wood would be decreased by 600, we write it as:

$$3x_1 + 2x_2 + x_3 \leq -600$$

Since the couches produced can't be less than 0, we can write it as:

$$x_1 \geq -500, x_2 \geq -300, x_3 \geq -200$$

We need to minimize the loss. Which implies we need to maximize the profit. So, out problem will be:

$$max(10x_1 + 8x_2 + 5x_3)$$

$$x_1 + 2x_2 + 2x_3 \leq 100$$

$$3x_1 + 2x_2 + x_3 \leq -600$$

$$x_1 \geq -500, x_2 \geq -300, x_3 \geq -200$$

But this is not in our standard maximum problem. So, we re-write it as:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \text{X} = X^1 + \begin{bmatrix} -500 \\ -300 \\ -200 \end{bmatrix}$$

$$X \geq \begin{bmatrix} -500 \\ -300 \\ -200 \end{bmatrix}$$

Now, the equations become: $z = c^T X$

$$AX \leq b$$

$$X \geq d$$

$$\text{c} = \begin{bmatrix} 10 \\ 8 \\ 5 \end{bmatrix}, \text{b} = \begin{bmatrix} 100 \\ -600 \end{bmatrix}, \text{A} = \begin{bmatrix} 1 & 2 & 2 \\ 3 & 2 & 1 \end{bmatrix}, \text{d} = \begin{bmatrix} -500 \\ -300 \\ -200 \end{bmatrix}$$

$z = c^T X = c^T(X^1 + d) = c^T X - 8400$

$AX \leq b \Rightarrow A(X^1 + d) \leq b \Rightarrow AX^1 \leq \begin{bmatrix} 1600 \\ 1700 \end{bmatrix}$. Let $\begin{bmatrix} 1600 \\ 1700 \end{bmatrix}$ be b^1

$X \geq d$

Now,

$z^1 = c^T X^1$

$AX^1 \leq b^1$

$X^1 \geq 0$

Finally,

$$max(10X_1^1 + 8X_2^1 + 5X_3^1)$$

subject to

$$X_1^1 + 2X_2^1 + 2X_3^1 \leq 1600$$

$$3X_1^1 + 2X_2^1 + 1X_3^1 \leq 1700$$

$$X_1^1, X_2^1, X_3^1 \geq 0$$

---

2. Consider the following linear program:

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

subject to

$$x_1 - x_2 + x_3 \leq 4$$

$$2x_1 + x_2 + 3x_3 \leq 6$$

$$-x_1 + 2x_3 = 3$$

$$x_1 + x_2 + x_3 \leq 8$$

$$x1, x2, x3 \geq 0$$

Write the dual problem. You do not need to demonstrate intermediate steps.

Solution: We are first going to convert the problem into the standard form.

$-x_1 + 2x_3 = 3$ should be written as $-x_1 + 2x_3 \leq 3$ and $x_1 - 2x_3 \leq -3$

So, the question becomes the following;

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

subject to

$$x_1 - x_2 + x_3 \le 4$$

$$2x_1 + x_2 + 3x_3 \le 6$$

$$-x_1 + 2x_3 \le 3$$

$$x_1 - 2x_3 \le -3$$

$$x_1 + x_2 + x_3 \le 8$$

$$x1, x2, x3 \ge 0$$

Here,

A = $\begin{bmatrix} 1 & -1 & 1 \\ 2 & 1 & 3 \\ -1 & 0 & 2 \\ 1 & 0 & -2 \\ 1 & 1 & 1 \end{bmatrix}$, X = $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$, B = $\begin{bmatrix} 4 \\ 6 \\ 3 \\ -3 \\ 8 \end{bmatrix}$, C = $\begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$

Now, we write the Dual Problem:

Objective: $min(B^T y)$

Subject to

$A^T Y \ge C$

$Y \ge 0$

$$A^T = \begin{bmatrix} 1 & 2 & -1 & 1 & 1 \\ -1 & 1 & 0 & 0 & 1 \\ 1 & 3 & 2 & -2 & 1 \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}, B^T = [4 \quad 6 \quad 3 \quad -3 \quad 8], C = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$$

Therefore, the dual function will be:

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

subject to

$$y_1 + 2y_2 - y_3 + y_4 + y_5 \geq 3$$

$$-y_1 + y_2 + y_5 \geq 2$$

$$y_1 + 3y_2 + 2y_3 - 2y_4 + y_5 \geq 1$$

$$y_1, y_2, y_3, y_4, y_5 \geq 0$$

---

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 $s_1,\ldots,s_n$, a list of frequencies $f_1,\ldots,f_m$, 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:

- Variables:

Let x_{ik} be the k^{th} frequency being assigned to the i^{th} station.

x_{ik} = 1 if the frequency is assigned to that station

= 0 if the frequency is not assigned to that station

Let AOS_k denote the k^{th} frequency being assigned to at least one station

AOS_k = 1 if f_k has been assigned to at least one station

= 0 if f_k has not been assigned to at least one station

- Objective Function

We are supposed to use as less frequencies as possible. That is, the min function.

$$\min\left(\sum_{k} AOS_k\right)$$

- Constraints:

  - If x_i and x_j are adjacent nodes, they should not be assigned same frequency.

$$x_{ik} + x_{jk} \leq 1 \text{ where } i, j \in S, f \in F$$

  - We need to use active frequencies only

$$x_{ik} \leq a_k \text{ where } i \in S, f \in F$$

  - We are supposed to assign only 1 frequency to a station

Let n be the total number of frequencies. $\sum_{k=1}^{n} x_{ik} = 1$ where $i \in S, f \in F$

  - $\forall i, j, x_{ij} \in \{0,1\}$
  - $\forall j, AOS_j \in \{0,1\}$

---

4. Prove or disprove the following statements.

1. **If A$\leq_p$B and B is in NP-hard, then A is in NP-hard.**
2. **If A$\leq_p$ B and B is in NP, then A is in NP.**
3. **If 3-SAT $\leq_p$ 2-SAT, then P = NP.**
4. **Any NP problem can be solved in time O($2^{poly(n)}$), where n is the input size and poly(n) is a polynomial.**
5. **If a problem A $\leq_p$ B, then it follows that B $\leq_p$ A.**

Solution:

1. False.

The given information is not enough to say that A is in NP-Hard. For Example, we can transform any empty language problem to NP-Hard but we know that it is not NP-Hard.

2. True.

Given B is in NP so we can verify the solution of B in polynomial time. It is also given that A is reduced to B in polynomial time. So, we can compose a reduction function F with the polynomial verifier of B to prove that solution of A can be verified in polynomial time too.
Implies, A is in NP

3. True.

From the given, we know that 3-SAT belongs to NP and also P. We know that 2-SAT is in P. And it is also said that 3-SAT can be reduced to 2-SAT in polynomial time. By this, we can say that 2-SAT is also NP. Therefore, P = NP.

4. True.

We can use exponential time to solve all the possible certifications of the problem. They can be verified in polynomial time by the deterministic Turing machine. (DTM)

5. False.

Let us assume A belongs to P and B belongs to NP. Polynomial reduction from B to A is not possible as A is not as hard as B. If $B \leq_p A$ then that means there exists a polynomial time solution for B. This is not possible. Therefore, $B \leq_p A$ does not follow.

---

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: Yes, the polynomial time algorithm exists. Let us represent 3-SAT as follows:

(x v y v z) ^ (~x v y v z) ^ …..

Algorithm:

I. Pick the first literal of the first clause

II. Replace every instance of it to true and the negation of it to false.

III. Run the satisfying algorithm with this value(specified in II.) of the literal.

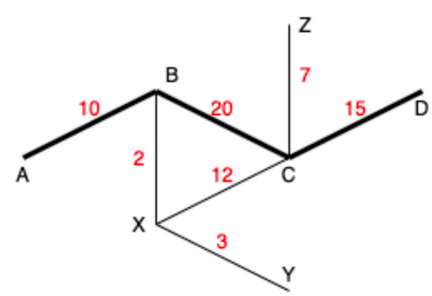
IV. If the problem is satisfied then we continue to next literal and next clauses if necessary. Else, that means we assigned a wrong value to the variable. We make changes to the value and repeat the same process.

The above algorithm that is defined can be run in polynomial time to find out if a satisfying assignment exists for the given 3-SAT instance.

---

6. The government 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 weight of an edge denotes 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.,$min_{u \in P} d_{uv}$). Next, we define the aggregate remoteness of *P* as *r(P)* = $\sum_{v \in V} d(v, P)$.

In the example shown in the figure 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*. Theremoteness 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 the Hamiltonian Path problem.

<u>Solution:</u> Inorder to prove a problem is NP-Complete, we need to do the following:

1. We need to show that it is NP
2. Pick some Y such that Y is NP-Complete and it can be reduced to X. $Y \leq_p X$

We can verify the solution in polynomial time using the following steps:

<u>Step 1:</u> Show the problem is NP.

- Run Dijkstras on all the nodes $n \in P$ as a source vertex. Find the shortest path from the nodes in P to all the nodes that are not in P. Since here we are assuming we have n nodes, we run Dijkstra n times.
- Calculate $r^1(P) = \sum_{v \in V} d(v, P)$ using the values obtained by running Dijkstras.
- Compare $r(P)$ and $r^1(P)$
- If both are same then the solution is correct else, it's wrong.

TimeComplexity of Dijkstras is $O(V + ElogE)$, so the time complexity of our verification algorithm will be $O(V^2(V + ElogE))$

Therefore, our problem is NP.

<u>Step 2:</u> Pick some Y such that Y is NP-Complete and it can be reduced to X. $Y \leq_p X$

we are making a claim that given a graph, hamiltonian path exists if and only if the remoteness of the path is 0.

<u>Proof:</u>

a. If hamiltonian path exists then the remoteness of the path is zero.

As we know hamiltonian path passes through all the nodes of the graph and it should visit each node only once. That means all the nodes lie on the path. So the distance from each node from the path is zero. So, the total value of the remoteness is zero.

$r(P) = 0$

b. If the remoteness of the path is zero then the hamiltonian path exists.

Remoteness value of the graph is zero implies all the nodes lie on the path like a highway. The highway is built in such a way that roads pass through all the cities and

they are visited only once. This means the path on which the highway is built is the hamiltonian path. There exists a hamiltonian path in that graph.
Therefore, G has hamiltonian path.

Hence Proved.