

USC CSCI 570 Homework5 Date: 2021-04-02

1. Maximizing Profit (15 points)

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 dollar, 8 dollar, and 5 dollar, 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 to only 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:

>

lets denote x_i is the number of change of couches of type i . so the x_i might be a positive or negative number. And our goal is to $\min(-10x_1 - 8x_2 - 5x_3)$, in other words, to maximize $(10x_1 + 8x_2 + 5x_3)$.

and the constrains will be:

$$\begin{aligned} 3x_1 + 2x_2 + x_3 &\leq -600 \\ x_1 + 2x_2 + 2x_3 &\leq 100 \end{aligned}$$

and since the factory were able to produce 500 type I couches, 300 type II and 200 type III. So x_i also need to within these limits:

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

2. Dual Program (15 points)

Consider the following linear program:

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

subject to

$$\begin{aligned} x_1 - x_2 + x_3 &\leq 4 \\ 2x_1 + x_2 + 3x_3 &\leq 6 \\ -x_1 + 2x_3 &= 3 \\ x_1 + x_2 + x_3 &\leq 8 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

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

Solution:

>

1. we first convert the equality constraint $-x_1 + 2x_3 = 3$ to:

$$\begin{aligned} -x_1 + 2x_3 &\leq 3 \\ x_1 - 2x_3 &\leq -3 \end{aligned}$$

so, overall constrains are:

$$\begin{aligned} x_1 - x_2 + x_3 &\leq 4 \\ 2x_1 + x_2 + 3x_3 &\leq 6 \\ -x_1 + 2x_3 &\leq 3 \\ x_1 - 2x_3 &\leq -3 \\ x_1 + x_2 + x_3 &\leq 8 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

2. then we build matrix $x \ c \ A \ h$

2. then we build matrix x, c, z_1, z_2

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$c = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & -1 & 1 \\ 2 & 1 & 3 \\ -1 & 0 & 2 \\ 1 & 0 & -2 \\ 1 & 1 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} 4 \\ 6 \\ 3 \\ -3 \\ 8 \end{bmatrix}$$

3. next, we create matrix y and transpose matrix A, b .

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}$$

$$A^T = \begin{bmatrix} 1 & 2 & -1 & 1 & 1 \\ -1 & 1 & 0 & 0 & 1 \\ 1 & 3 & 2 & -2 & 1 \end{bmatrix}$$

$$b^T = [4 \ 6 \ 3 \ -3 \ 8]$$

4. then we build dual LP.

$$\max(c^T \dot{x}) \Rightarrow \min(b^T \dot{y}) = \min(4y_1 + 6y_2 + 3y_3 - 3y_4 + 8y_5)$$

$$A\dot{x} \leq b \Rightarrow A^T \dot{y} \geq c = \begin{bmatrix} y_1 + 2y_2 - y_3 + y_4 + y_5 \geq 3 \\ -y_1 + y_2 + y_5 \geq 2 \\ y_1 + 3y_2 + 2y_3 - 2y_4 + y_5 \geq 1 \end{bmatrix}$$

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

3. Spectrum Management (15 points)

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_2, \dots, s_n , a list of frequencies f_1, \dots, 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:

>

Even though there has m available frequency channels, we only need n of them and assign each station. So we can build a $n \times n$ matrix X , where $x_{i,j}$ denotes whether the station s_i uses frequency f_j .

$$X = \begin{bmatrix} 0 & 1 & \dots & \dots & 0 \\ 1 & 0 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & 1 \end{bmatrix}$$

also we need to create a vector (or counter) k to track where frequency f_j is used or not.

$k = (1, 1, 0, 0, 1, \dots, 1)$, the length of k is m . In this way, our goal is to minimize the sum of k vector is

$$\min \sum_{j=1}^m k_j$$

and the constraints will be:

1. those adjacent stations should use different frequencies.

- $$x_{ij} + x_{i'j} \leq 1 \quad \forall (i', i) \text{ s.t. } (s_i, s_j) \in E$$
- the value of item in both X and k should be zero or one.
 $x_{ij} \in \{0, 1\}$ and $k_j \in \{0, 1\}$
 - and there is only one frequency for each station.
 $\sum_{j=1}^n x_{i,j} = 1$

4. Short Questions (15 points)

Prove or disprove the following statements.

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

Solution:

>

- False. Lets say in a decision problems, we are asked to determine whether if the input string is in the given language. An empty language is the language containing no strings. People can transform the empty language \emptyset to any $NP - hard$ problem, but the empty language \emptyset is not $NP - hard$.
- True. B is in NP , and A is not harder than B . Combining the certifier for B and the polynomial reduction map, we can construct a certifier for A as well. And also this assertion is a theorem.
- 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 $P = NP$.
- True. Each of possible solution can be checked by a polynomial-time deterministic turing machine, since everyone can use the exponential time to try all of them (possible solution).
- 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 a contradiction.

5. Finding a Satisfying Assignment (20 points)

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:

>

Lets say in an given $3 - SAT$ instance who consists variables $X = \{x_1, \dots, x_n\}$ and also has the clauses set $\{C_1, \dots, C_m\}$.

Next, Lets denote $\varphi(x_1, \dots, x_n) = C_1 \wedge C_2 \wedge \dots \wedge C_k$ is the corresponding Boolean formula.

Also the CNF formula φ_i is obtained from φ by setting x_i to $TRUE$ and $FALSE$, respectively.

And we can name the given algorithm is $S(\varphi)$. It is able to traverse all the variable X , and decide whether a given instance has a satisfying assignment.

More specifically, for the x_1 , using algorithm to assign x_1 value. if $S(\varphi_1) == 1$, then we assign $True$ to the x_1 , or we denote x_1 is $FALSE$. Once x_1 finished assignment, we can continue to verify rest of variables.

After finished all the variables assignment, we get one satisfying assignment, and the time complexity is still polynomial.

6. Multi-Lane Highway (20 points)

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. $\min_{u \in P} d_{uv}$).

Next, we define the aggregate remoteness of P as $r(P) = \sum_{v \in V} d(v, P)$

In the example shown 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.

Solution:

>

1. Claim:

This problem is in *NP*.

A feasible path of remoteness less than k can be verified by traversing the path in polynomial time, including checking whether there are no repeated vertices, the sum of remoteness is k , and all edges of the path P are in E .

2. Claim:

G has a Hamiltonian path if and only if G has a feasible path P whose remoteness at most

$$k = \sum_{v \in V'} d(v, P).$$

Assume there is an instance of the Hamiltonian path problem: a graph with n vertices.

Normally, a Hamiltonian path in G is equivalent to a simple path (no repeated vertices) of length $n - 1$, which n is the number of all vertices. and its remoteness is 0.

But in a graph G' like mentioned in the question, a Hamiltonian may not reach all vertices. for example, a simple path could be $P = ABXCD$, the rest of nodes Y and Z are not the path P . We can use V' to denote those vertices. $V' = n - V_p$. And V_p means a set of vertices in path P .

So we can say a Hamiltonian path in G is equivalent to a simple path (no repeated vertices) of length at most $n - 1$ in G .

In this way, it turns out that there is a polynomial time reductions from the Hamiltonian path problem to this problem. Specifically, if there is a Hamiltonian path P , then that path P is a simple path and its length is at most $n - 1$. The remoteness becomes $r(P) = \sum_{v \in V'} d(v, P)$. On the other hand, if there is a simple path of remoteness at most $\sum_{v \in V'} d(v, P)$, then it is a Hamiltonian path.

Therefore, we can conclude that this problem is *NP-complete*