Example 1

Linear Programming Motivation

A Farmer has 200 acres of land and Produces Corn and wheat.

It costs 100 per acre to Plant Corn
200 " " " " wheat

He Profits 50 Per acre from Corn and
100 per " " wheat.

It takes 10 man days Per acre for Corn
20 " " " " " wheat.

He has a budget of 20k and 1500 mandays.

Objective: Maximize the Profit from the Corn & wheat

Variables : $X \rightarrow$ Total area of Corn

$y \rightarrow$ " " " Wheat

Objective function

Maximize : $Z = 50X + 100y$

Constraint: $100X + 200y \leq 20{,}000$ (budget)

$X + y \leq 200$ (Total area)

$10X + 20y \leq 1500$ (man days)

$X \geq 0$ & $y \geq 0$

Answer: 7500 by Planting wheat of 75 acres

or " " Corn of 150 "

Example 2: **Diet Problem**

There are two types of grains G_1 and G_2 and three types of nutrients Starch, Proteins, Vitamins.

Nutrient Content & Cost Per kg of food

	Starch	Protein	Vitamin	Cost Per kg
G_1	5	4	2	60
G_2	7	2	1	35

The requirement Per day of Starch, Protein & Vitamins is 8, 15 and 3 respectively.

The Problem is to find how much of each food to consume Per day so as to get required amount Per day of each nutrient at minimal Cost.

Variables:

x_1 : # of units of grain G_1 to be Consumed per day

x_2 : # of units of grain G_2 to be " "

Objective function:

In this case, the objective is to minimize the total cost per day, which is $z = 60x_1 + 35x_2$

Constraints:

$$5x_1 + 7x_2 \geq 8$$

$$4x_1 + 2x_2 \geq 15$$

$$2x_1 + x_2 \geq 3$$

$$x_1 \geq 0, \; x_2 \geq 0$$ → Non-negativity Constraint

The diet Problem can be formulated by the following linear programming.

$$\text{Minimize} \quad z = 60x_1 + 35x_2$$

Subject to :

$$5x_1 + 7x_2 \geq 8$$

$$4x_1 + 2x_2 \geq 15$$

$$2x_1 + x_2 \geq 3$$

$$x_1 \geq 0, \; x_2 \geq 0$$

Def: A Solution $x = (x_1, x_2)$ is said to be feasible with respect to the above LP if it satisfies all the above equations.

A feasible solution is Optimal if its objective function value is equal to the smallest value z can take over all feasible solutions.

Representation of Linear Programs

Recall :

The Problem of solving simultaneous linear equations can be represented using matrix vector notation.

Given a matrix A and a vector b, we want to solve the equation $Ax = b$

[Gaussian elimination is a well-known algo for this Problem]

In LP, we have inequalities in place of equations.

Example, find x that satisfies $Ax \geq b$

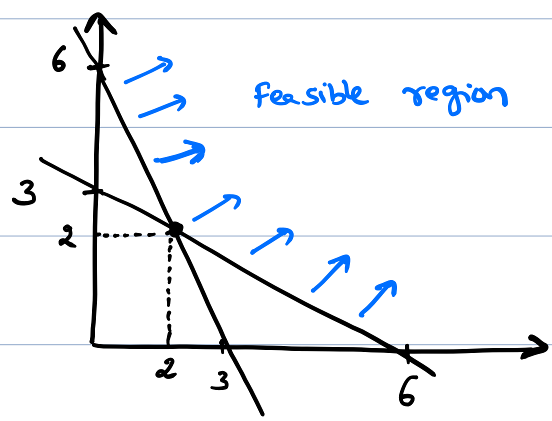
(Coordinate wise comparison)

This system of inequalities define a region in the Plane.

E.g; let $x = (x_1, x_2)$

$$x_1 + 2x_2 \geq 6$$

$$2x_1 + x_2 \geq 6$$



Given a region $Ax \geqslant b$, LP seeks to minimize a linear combination of coordinates of x, over all x belong to the region.

Such a linear combination can be written as $c^T x$, where $c^T x$ denote the inner Product of two vectors.

Standard form of LP

$$\text{Min } c^T x$$

Such that $\quad Ax \geqslant b$

$\qquad\qquad\quad x \geqslant 0$

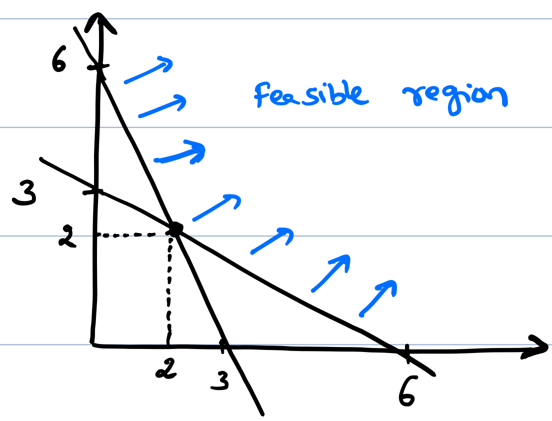
where A is $m \times n$ matrix, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and $x \in \mathbb{R}^n$

E.g; let $x = (x_1, x_2)$

$$x_1 + 2x_2 \geq 6$$

$$2x_1 + x_2 \geq 6$$

Suppose we want to minimize $1.5x_1 + x_2$ over the region defined by the inequalities.



Then the solution is the point $x = (2, 2)$

the value of $c^T x = 1.5 \times 2 + 2 = 5$.

Integer Programming (IP)

General form

$$\text{Min } c_1 x_1 + \cdots + c_n x_n \rightarrow \text{objective function}$$

such that

$$Ax \geq b \rightarrow \text{constraints}$$

$$\forall i : x_i \in \{0,1\} \rightarrow \text{integer variables}$$

where $A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \cdots & & \cdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}_{m \times n}$ $\quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$

if $0 \leq x_i \leq 1$ in the above IP then it is called linear programming.

Integer Linear Programming (ILP)

IP in which the objective function and the constraints are linear.

Note:- ILP is NP_hard

Decision Version of LP (minimization)

Given a matrix A, vectors b and c and a bound k, does there exists x such that $Ax \geq b$ and $c^T x \leq k$.

Remarks ① the decision version of LP is in NP.

② LP Problem can be solved in Polynomial time.

[You can learn more about all this in courses on LP]

In this course we use the algorithm of LP as a black box.

Big Question:

How can LP help us when we want to solve combinatorial problems?

As our first example, we develop a 2-approximation algorithm for weighted vertex cover problem.

[We have already seen a 2-approximation algo for un-weighted vertex cover]

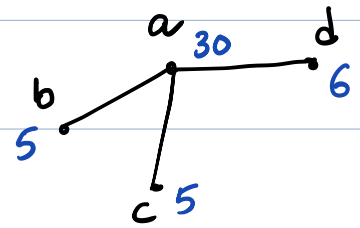
Vertex cover as an integer Program

Recap: A Vertex Cover in a graph $G = (V, E)$ is a set $X \subseteq V(G)$ such that each edge has atleast one end in X.

In a weighted vertex cover problem, each vertex $u \in V(G)$ has a weight $w_u \geq 0$, with the weight of a set X of vertices denoted $w(X) = \sum_{u \in X} w_u$.

Aim is to find a vertex cover X for which $w(X)$ is minimum.

Example :

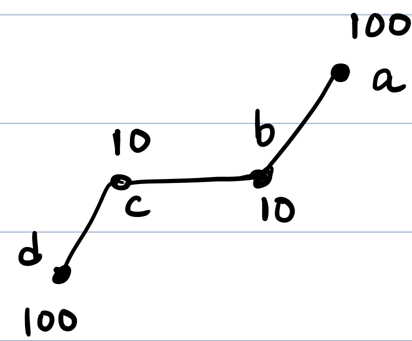


Graph G.

Minimum weight vertex cover

$X = \{b, c, d\}$ of weight 16.

Clearly the greedy 2-approximation algorithm we studied in the previous class doesn't work on the following example.



OPT soln $= \{b, c\}$ of weight $= 20$

Greedy soln $= \{a, b, c, d\}$ of weight $= 220$.

ILP Formulation for Vertex Cover

Given $G = (V, E)$

we will have a variable x_u for each vertex $u \in V(G)$.

$x_u = 0$ will indicate that vertex u is not in the vertex cover.

$x_u = 1$ " " " " " is in the vertex cover.

For each edge $uv \in E(G)$, vertex cover must have at least one end.

we write the inequality $x_u + x_v \geq 1$

Also our objective is to find X such that $\omega(X)$ is minimum.

Combining all, we can formulate the Vertex Cover Problem as follows.

[VCIP] $\min \sum_{u \in V(G)} w_u x_u$

s.t. $x_u + x_v \geq 1$ for all $uv \in E(G)$

$x_u \in \{0,1\}$ for all $u \in V(G)$

Remark: Vertexcover $\leq_p$ Integer Linear Programming

A linear Programming relaxation

We modify VCIP by dropping the requirement that each $x_i \in \{0,1\}$. Instead we have $0 \leq x_i \leq 1$

[VCLP] $$\min \sum_{u \in V(G)} w_u x_u^*$$

s.t. $x_u^* + x_v^* \geq 1$ for all $uv \in E(G)$

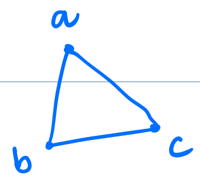
$0 \leq x_u^* \leq 1$ for all $u \in V(G)$

- VCLP can be solved in polynomial time.

Observation 1: Let X denote the vertex cover of minimum weight. Then $w_{LP} \leq w(X)$ → OPT

Pf: follows from the fact that minimum (VCLP) is no larger than that of (VCIP).

Example:



$w_a = w_b = w_c = 1$

$X = \{a, c\}$, $w(X) = 2$

VCLP soln: $x_a = x_b = x_c = \frac{1}{2}$,

$w_{LP} = \frac{3}{2}$

Q How can solving VCLP help us to find an approximation algorithm for Vertex Cover Problem.

Rounding the LP solution

Given a fractional solution $\{x_u^*\}_{u \in V(G)}$ of VCLP

define

$(z_u)_{u \in V(G)}$ as

$$z_u = \begin{cases} 1 & \text{if } x_u^* \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

Overview of the Algorithm

① **Solve the LP**

$$\min \sum_u w_u x_u^*$$

S.to $\quad x_u^* + x_v^* \geq 1 \quad \forall\, uv \in E(G)$

$$0 \leq x_u^* \leq 1 \quad \text{for } u \in V(G)$$

② **Round the LP Soln**

$(z_u)_{u \in V(G)}$ defined by

$$z_u = \begin{cases} 1 & \text{if } x_u^* \geq \frac{1}{2} \\ 0 & \text{o.w} \end{cases}$$

③ **Output**

$$X = \{ u \in V(G) : z_u = 1 \}$$

Claim: The set X defined above is a Vertex cover and $w(X) \leq 2\,OPT$

Proof: $\boxed{A}$ Clearly X is a vertex cover because for every edge $uv \in E(G)$, either $x_u^* \geq \frac{1}{2}$ or $x_v^* \geq \frac{1}{2}$

That is atleast one of u or v will be Placed in X.

$uv: \; x_u^* + x_v^* \geq 1$

ie, $x_u^* \geq \frac{1}{2}$ or $x_v^* \geq \frac{1}{2}$

ie, $z_u = 1$ or $z_v = 1$

ie, either u or v in X

$\boxed{B}$ output cost $= \sum_u w_u z_u$

Observe that $z_u \leq 2x_u^*$

from observation ①, we have $\sum_u w_u x_u^* \leq OPT$

$\therefore$ o/p cost

$$= \sum_u w_u z_u \leq 2 \sum w_u x_u^*$$

$$\leq 2\,OPT$$

$\therefore$ It is 2-approximation algorithm.

Alternate explanation for Part-B

Consider the weight $\omega(X)$ of this vertex cover.

The set X only has vertices with $x_u^* \geq \frac{1}{2}$

Thus the linear program "Paid" at least $\frac{1}{2}\omega_u$ for vertex u.

and we "pay" ω_u, at most twice as much.

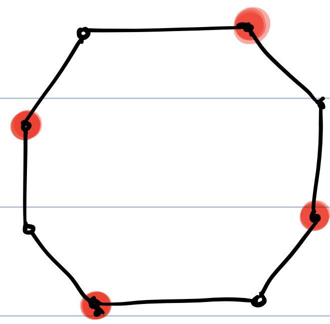
$$\omega_{LP} = \sum_u \omega_u x_u^* \geq \sum_{u \in X} \omega_u x_u^* \geq \frac{1}{2} \sum_{u \in X} \omega_u = \frac{1}{2}\omega(X)$$

$$\omega(X) \leq 2\omega_{LP} \leq 2\omega(X^*)$$ (from observation ①)

hence the algo produces a vertex cover X of

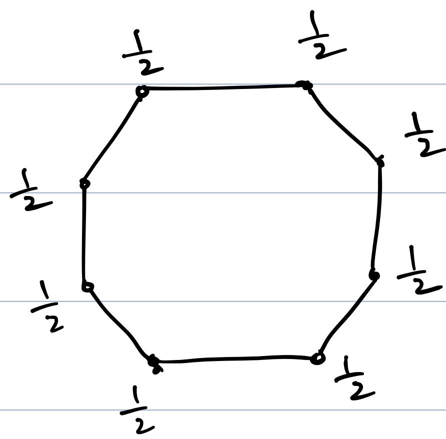
at most twice the minimum possible weight.

The analysis tight



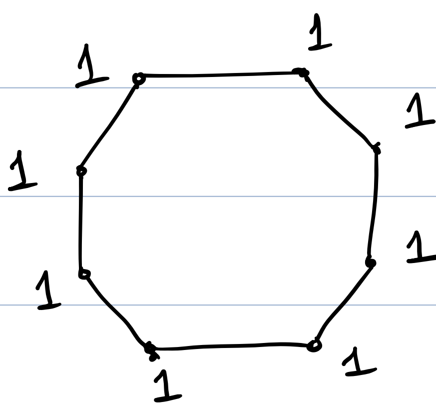
$OPT = 4$

G



LP-soln

After Rounding



Value $= 8$

The algorithm is no better than 2-approximation.

Overview of our Idea

Algorithm

- Find IP
- Solve LP relaxation
- Round solution to integers

Analysis

- Correctness – Satisfies the conditions
- Efficient – Poly time
- Quality – approximation ratio / factor

① O/P can be related to LP value
② O/P " bounded by LP value

Then Combine these two