CO250 Notes

Chapter 1 keywords

Formulation, Linear Program, Integer Program, Shortest Path, Non linear Program

D Formulate a real world problem to a linear program

	Machine		Labor		Price sold
Product	I	II	skilled	unskilled	
1	11	4	8	7	300
2	7	6	5	8	260
3	6	5	5	7	220
4	5	4	6	4	180
available	700	500	600	650	

hours:

Take a look at this problem: a manufacturer wants to make the most amount of profit from the above resources. How many unit of product 1,2,3,4 should it produce to achieve maximum profit? We can formulate this in by assigning variables to each product, and the labor time used.

let x_1, x_2, x_3, x_4 represents the amount of product 1,2,3,4 we decided to produce. Let y_1, y_2 represents the amount of work time is used for skilled/unskilled labor, respectively. Then we have these constraints:

$$11x_1 + 7x_2 + 6x_3 + 5x_4 \leq 700$$

$$4x_1 + 6x_2 + 5x_3 + 4x_4 \leq 500$$

$$8x_1 + 5x_2 + 5x_3 + 6x_4 \leq y_1$$

$$7x_1 + 8x_2 + 7x_3 + 4x_4 \leq y_2$$

$$y_1 \leq 600$$

$y_2 \leq 650$

$\forall x, y \geq 0$ cannot produce negative amounts

A solution is a set of $x_1, x_2, x_3, x_4, y_1, y_2$.

A feasible solution is a set of x, y such that all constraint from above are true.

Definition 1.1.

A formulation of a program is correct if:

every single possible arrangement correspond to exactly one feasible solution, and every feasible solution correspond to exactly one arrangement.

In this case, arrangement means the amount of production, and labor hour used.

Definition 1.2:

An affine function, $f: \mathbb{R}^n \Rightarrow \mathbb{R}$, is

$$f(x) = \vec{a}^T \vec{x} + c \quad , \text{ where } a \in \mathbb{R}^n, c \in \mathbb{R}$$

if $c = 0$, f is linear.

Definition 1.3:

A linear constraint is one of the following:

$$f(x) \leq c \ , \ f(x) \geq c \ , \ f(x) = c \quad \text{for linear } f, \ c \in \mathbb{R}$$

Definition 1.4:

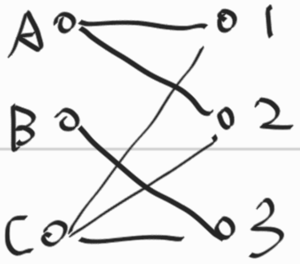
A linear program maximize/minimize an affine function, subjected to linear constraints.

Definition 1.5:

An integer program is a linear program that require at least one variable to take integer values.

▷ Assigment Problem:

How can we encode the fact that:



A can do job 1, 2,

B can do job 3,

C can do job 1,2,3?

A graph $G = (\underset{\uparrow}{V}, \underset{\nwarrow}{E})$ can represent it.

vertice edge

(ABC123) (A1, A2, B3, C1, C2, C3)

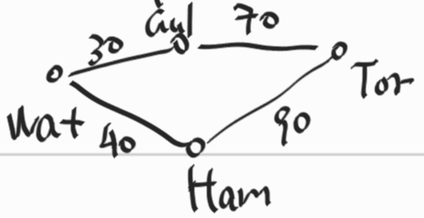
i can do job j iff $ij \in E$.

Bipartite: a partition of vertice so that each edge has one end in partition A, another one in partition B.

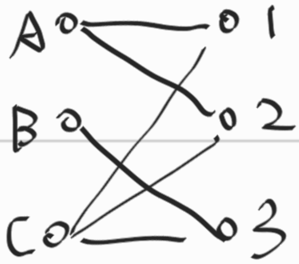
Matching: a set of edges where no two edges share a vertex

Perfect matching: every vertice is connected to an edge in the matching.

Usually, an edge in a graph will be associated with some value, called the cost of an edge. Think of the following example:



Imagine if someone wants to go to Tor from Wat. They can take Wat-Cul-Tor, which has a cost of 100, or Wat-Ham-Tor, which has a cost of 130. Since cost represents distance in this case, normally we would want to minimize our distance. We would like to develop an algorithm, that given any graph, we can find the minimum cost of it. (much later in Chapter 3)



Back to this example. How do we model this graph using an IP? Let c_e denote the cost of edge e. let x_e be a binary variable, such that if $x_e=1$, e is used in our matching, and not used if $x_e=0$.

We have

$$\max \; c_{A1}x_{A1} + c_{A2}x_{A2} + c_{B3}x_{B3} + c_{C1}x_{C1} + c_{C2}x_{C2} + c_{C3}x_{C3}$$

$$\begin{aligned} \text{s.t.} \quad & x_{A1}+x_{A2}=1 & \quad & x_{A1}+x_{C1}=1 \\ & x_{B3}=1 & & x_{A2}+x_{C2}=1 \\ & x_{C1}+x_{C2}+x_{C3}=1 & & x_{B3}+x_{C3}=1 \end{aligned}$$

$$0 \leq x \leq 1, \quad \text{int } x.$$

Generic:

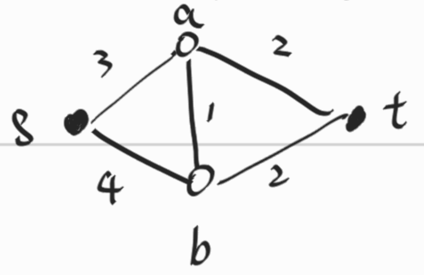
$$\min \; \Sigma c_e x_e$$

$$\text{s.t. } \Sigma x_e : e \in \delta(v) = 1$$ (requirement of a matching) (or, for every vertex, we must have one and only one edge in the matching adjacent to it)

$$0 \leq x_e \leq 1, \quad \text{int } x_e$$

Shortest path problem:

Consider the following graph:



(G)

We wish to find the shortest path, starting from s to t.

Let $G=(V,E)$, $U \subseteq V$, let $\delta(U)$ be the cut induced by U (see Math 239 for cut). For example,

if $u=\{s\}$ then $\delta(u)=\{sa, sb\}$

$u=\{s,a\}$ $\delta(u)=\{sb, ab, at\}$

$u=\{s,b\}$ $\delta(u)=\{sa, ab, bt\}$

$u=V\setminus\{t\}$ $\delta(u)=\{at, bt\}$

or $\{s,a,b\}$

An s-t path is a set of distinct vertices that starts with s and ends with t. For example s-sa-a-at-t (or s-a-t, sa-at).

Theorem 1.1:

An st path intersects every st cut.

and its converse is also true.

Let S be a set of edges that contains at least one edge from every st cut.

∃ an st path which is contained in S.

By the converse, we know in order to construct an s-t path it suffices to pick one edge from each st cut. With this in mind, we can formulate (G) as:

let $x = [x_{sa} \; x_{sb} \; x_{ab} \; x_{at} \; x_{bt}]^T$. Note that if $x_{ij} = 1$, the edge is "selected" and $x_{ij} = 0$ if the edge is not selected.

$$\min [3 \; 4 \; 1 \; 2 \; 2] x$$

$$\text{s.t} \quad \begin{array}{c|ccccc} & sa & sb & ab & at & bt \\ \hline s & 1 & 1 & 0 & 0 & 0 \\ S,a & 0 & 1 & 1 & 1 & 0 \\ s,b & 1 & 0 & 1 & 0 & 1 \\ V \backslash t & 0 & 0 & 0 & 1 & 1 \end{array} \; x \geq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\text{int } x \geq 0$$

Note that $x \leq 1$ is not used here. Since every edge has positive cost, it will not decrease the OB if $x > 1$

Non linear Program:

NLP are in the form of:

for $f: \mathbb{R}^n \to \mathbb{R}$, $g_i: \mathbb{R}^n \to \mathbb{R}$, $i \in \{1 \cdots m\}$ a NLP is in the form of

$$\begin{aligned} \min \; & f(x) \\ \text{s.t. } & g_1(x) \leq 0 \\ & \vdots \\ & g_m(x) \leq 0 \end{aligned}$$