Linear Programming

Formulating and solving large problems

http://academia.utp.ac.pa/Humberto-alvarez

Recalling some concepts

- As said, LP is concerned with the optimization of a linear function while satisfying a set of linear inequalities.
- Assumptions:
  - Proportionality
  - Additivity
  - Divisibility
  - Deterministic

Problem representation

- Standard vs. Canonical Formats
  - In the standard format all constraints are equalities and all the variables are non negative.
  - In the canonical format are the variables are non negative and the constraints are inequalities depending on the objective function:
    - Minimization: $\geq$
    - Maximization: $\leq$
- Matrix Format

Standard vs. Canonical Forms

	MINIMIZATION PROBLEM	MAXIMIZATION PROBLEM
STANDARD FORM	Minimize $\sum_{j=1}^{n} c_j x_j$ subject to $\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1,\dots,m$ $x_j \geq 0, \quad j = 1,\dots,n.$	Maximize $\sum_{j=1}^{n} c_j x_j$ subject to $\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1,\dots,m$ $x_j \geq 0, \quad j = 1,\dots,n.$
CANONICAL FORM	Minimize $\sum_{j=1}^{n} c_j x_j$ subject to $\sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1,\dots,m$ $x_j \geq 0, \quad j = 1,\dots,n.$	Maximize $\sum_{j=1}^{n} c_j x_j$ subject to $\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1,\dots,m$ $x_j \geq 0, \quad j = 1,\dots,n.$

Mokhtar S., B., et al (2004) *Linear Programming and Network Flows*, Wiley-Interscience, U. S. A.,

Matrix Form

Denote de row vector ($c_1, c_2, \ldots, c_n$) as **c** and consider the following column vectors **x** and **b** and the $m \times n$ matrix **A**

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

The problem can be formulated as follows:

Optimize **z**: **cx**

subject to:

$$\mathbf{Ax} = \mathbf{b}$$
$$\mathbf{x} \geq \mathbf{0}$$

Mokhtar S., B., et al (2004) *Linear Programming and Network Flows*, Wiley-Interscience, U. S. A.,

Handling large problems

- Indexing:
  - Indexes or subscrits permit representing collections of similar quantities with a single symbol.
  - Thus, the first step is to choose appropiate indexes for the different dimensions of the problem.
  - To describe large problems it is necessary to assign indexed symbolic names to most input parameters even if they are constant.

- Families of similar constraints distinguished by indexes may be expressed in single-line format.
- A function is linear if it is a constant-weighted sum of decision variables.
- Summations are used because of the linearity.
- Use as many summations as indexes are in the corresponding expression.

Example (From Rardin, R. (1997) Optimization in Operations Research, Prentice Hall: U. S. A.)

A large manufacturer of corn seed operates 20 facilities producing seeds of 25 hybrid corn varieties and distributes them to customers in 30 sales regions. They want to know how to carry out these production and operations at minimum cost.

A variety of parameters have been estimated:

- The cost per bag of producing each hybrid at each facility
- The corn processing capacity of each facility
- The number of bags of corn that must be processed to make a bag of each hybrid.
- The number of bags of each hybrid demanded in each customer region
- The cost per bag of shipping each hybrid from each facility to each customer region

Pertinent information:

- **Dimensions of the problem**

$f \triangleq$ production facility number ($f = 1, \ldots, \ell$)

$h \triangleq$ hybrid variety number ($h = 1, \ldots, m$)

$r \triangleq$ sales region number ($r = 1, \ldots, n$)

- **Parameter of the model**

$p_{f,h} \triangleq$ cost per bag of producing hybrid h at facility f

$u_f \triangleq$ corn processing capacity of facility f in bushels

$a_h \triangleq$ number of bushels of corn that must be processed to obtain a bag of hybrid h

$d_{h,r} \triangleq$ number of bags of hybrid h demanded in sales region r

$s_{f,h,r} \triangleq$ cost per bag of shipping hybrid h from facility f to sales region r

- **Decision variables**

$x_{f,h} \triangleq$ number of bags of hybrid h produced at facility f ($f = 1, \ldots, \ell$; $h = 1, \ldots, m$)

$y_{f,h,r} \triangleq$ number of bags of hybrid h shipped from facility f to sales region r ($f = 1, \ldots, \ell$; $h = 1, \ldots, m$; $r = 1, \ldots, n$)

Objective function

total cost = total production cost + total shipping cost

$$\min \sum_{f=1}^{\ell} \sum_{h=1}^{m} p_{f,h} x_{f,h} + \sum_{f=1}^{\ell} \sum_{h=1}^{m} \sum_{r=1}^{n} s_{f,h,r} y_{f,h,r}$$

Complete formulation

$$\min \sum_{f=1}^{\ell}\sum_{h=1}^{m} p_{f,h}x_{f,h} + \sum_{f=1}^{\ell}\sum_{h=1}^{m}\sum_{r=1}^{n} s_{f,h,r}y_{f,h,r} \qquad \text{(total cost)}$$

$$\text{s.t.} \quad \sum_{h=1}^{m} a_h x_{f,h} \le u_f \qquad f = 1,\ldots,\ell \qquad \text{(capacity)}$$

$$\sum_{f=1}^{\ell} y_{f,h,r} = d_{h,r} \qquad h = 1,\ldots,m; \quad r = 1,\ldots,n \qquad \text{(demands)}$$

$$\sum_{r=1}^{n} y_{f,h,r} = x_{f,h} \qquad f = 1,\ldots,\ell; \quad h = 1,\ldots,m \qquad \text{(balance)}$$

$$x_{f,h} \ge 0 \qquad f = 1,\ldots,\ell; \quad h = 1,\ldots,m \qquad \text{(nonnegativity)}$$

$$y_{f,h,r} \ge 0 \qquad f = 1,\ldots,\ell; \quad h = 1,\ldots,m; \quad r = 1,\ldots,n$$

The general LP problem: the mixing problem

A plant produces products mixing several resources in such a way that the mixture meets certain availability or levels of each resources. Supose that n *products* $i=1, 2, \ldots, n$ and m resources $j = 1, 2, \ldots, m$ are to be considered. The unit income of each product is d_i and the availability of the resources is b_j. Each unit of product x_i needs certain amount of a_{ij} resource b_j to be produced. The general formulation is:

$$Maximize\ Z = \sum_{i=1}^{n} d_i x_i$$

$Subject\ to$:

$$\sum_{j=1}^{m} a_{i,j} x_i \leq b_v \forall i = 1,2,..,n$$

$$x_i \geq 0$$

Production scheduling

It is necessary to determine the production rate over a planning period of T units of time such as the known demand is satisfied and the total production and inventory cost is minimized. Let the known demand at time t be $g(t)$, and similarly denote the production rate and inventory at time t be x(t) and y(t) respectively. Suppose that the initial inventory at t=0 be y_0, and the desired inventory at the end of the planning period is y_T. Suppose that the planning period T is divided into n smaller and equal periods of length Δ, such that $T = n\Delta$. Furthermore, the total cost of inventory is proportional to the units in storage in certain period of time , such that the inventory cost can at time i be approximated to $(c_1\Delta)y_i$ for $c_1 > 0$ and known.

Production scheduling…, cont.

Similarly, it is possible to assume that the production cost is proportional to the production rate such that the total production cost at period *i* can be determined by $(c_2\Delta)x_i$. Additionally, no backlogs are allowed and the production rate can not be greater than b_1 and the inventory level hast to be less or equal to b_2 at any time. The problem can be formulated as:

$$Minimize\ Z = \sum_{i=1}^{n}\left((c_1\Delta)y_i + (c_2\Delta)x_i\right)$$

$Subject\ to$:

$$y_i = y_{i-1} + (x_i - g_i)\Delta, \forall\ i = 1, 2, \dots, n$$

$$y_n = y_T$$

$$0 \le x_i \le b_1, \forall 1 = 1, 2, \dots, n$$

$$0 \le y_i \le b_2, \forall 1 = 1, 2, \dots, n$$

Cutting Stock problem

This problem is concerned with the production and cutting of pieces of material from standard elements of raw material of width *w* and length *l*. Orders are placed for parts of width *w* but various lengths. In particular, b_i parts of length l_i and width *w* for *i=1, 2, …, m.* The objective is to cut the standard elements in such a way as to satisfy the order and to minimize the waste.

It is possible to cut the elements in may way, each way called a cutting pattern $\boldsymbol{a_j}$, each composed of a column vector of if *i* components. For each pattern, the component a_{ij} is a nonnegative integer denoting the number of parts of length *i* in the jth pattern. The vector $\boldsymbol{a_j}$ is a cutting pattern iif $\sum_i^n a_{ij} l_i \leq l$, and each $a_{ij} \in \Im \geq 0$. The number of cutting patterns is infinite. Let x_j be number of standards rolls cut according to the jth pattern, the problem can be formulated as follows

$$Minimize\ Z = \sum_{j=1}^{n} x_j$$

$$subject\ to:$$

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_j \forall j = 1, 2, \ldots, n$$

$$x_j \geq 0 \in \Im$$

The diet problem

- The goal of the **diet problem** is to select a set of foods that will satisfy a set of daily nutritional requirement at minimum cost.
- The problem is formulated as a **linear program** where the objective is to minimize cost and the constraints are to satisfy the specified nutritional requirements. The diet problem constraints typically regulate the number of calories and the amount of vitamins, minerals, fats, sodium, and cholesterol in the diet.

The diet problem

$$
\begin{aligned}
x[\text{food}] &= \text{amount of } \textit{food} \text{ to eat} \\
c[\text{food}] &= \text{cost of 1 serving of } \textit{food} \\
A[\text{food}] &= \text{amount of Vitamin A in 1 serving of } \textit{food} \\
Cal[\text{food}] &= \text{amount of calories in 1 serving of } \textit{food} \\
MinF[\text{food}] &= \text{minimum number of servings for } \textit{food} \\
MaxF[\text{food}] &= \text{maximum number of servings for } \textit{food} \\
MinN[\text{nutrient}] &= \text{minimum amount of } \textit{nutrient} \text{ required} \\
MaxN[\text{nutrient}] &= \text{maximum amount of } \textit{nutrient} \text{ required}
\end{aligned}
$$

$$
\begin{array}{llcl}
\text{Minimize} & & cost[C]*x[C]+cost[M]*x[M]+cost[W]*x[W] & \\
\text{Subject to} & MinN[VA] \le & A[C]*x[C]+A[M]*x[M]+A[W]*x[W] & \le MaxN[VA] \\
 & MinN[Cal] \le & Cal[C]*x[C]+Cal[M]*x[M]+Cal[W]*x[W] & \le MaxN[Cal] \\
 & MinF[C] \le & x[C] & \le MaxF[C] \\
 & MinF[M] \le & x[M] & \le MaxF[M] \\
 & MinF[W] \le & x[W] & \le MaxF[W]
\end{array}
$$

The knapsack problem

- Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.
- The most common problem being solved is the **0-1 knapsack problem**, which restricts the number x_i of each kind of item to zero or one. Given a set of n items numbered from 1 up to n, each with a weight w_i and a value v_i, along with a maximum weight capacity W,. Here x_i represents the number of instances of item i to include in the knapsack. Informally, the problem is to maximize the sum of the values of the items in the knapsack so that the sum of the weights is less than or equal to the knapsack's capacity.

$$\text{maximize} \sum_{i=1}^{n} v_i x_i$$

$$\text{subject to} \sum_{i=1}^{n} w_i x_i \leq W \text{ and } x_i \in \{0,1\}$$

The knapsack problem

- The **bounded knapsack problem** (**BKP**) removes the restriction that there is only one of each item, but restricts the number of copies of each kind of item to a maximum non-negative integer value *c*:

$$\text{maximize} \sum_{i=1}^{n} v_i x_i$$

$$\text{subject to} \sum_{i=1}^{n} w_i x_i \leq W \text{ and } 0 \leq x_i \leq c$$

- The **unbounded knapsack problem** (**UKP**) places no upper bound on the number of copies of each kind of item and can be formulated as above except for that the only restriction on x_i is that it is a non-negative integer.

$$\text{maximize} \sum_{i=1}^{n} v_i x_i$$

$$\text{subject to} \sum_{i=1}^{n} w_i x_i \leq W \text{ and } x_i \geq 0$$

The bin packing problem

- In the **bin packing problem**, objects of different volumes must be packed into a finite number of bins or containers each of volume *V* in a way that minimizes the number of bins used.
- There are many variations of this problem, such as 2D packing, linear packing, packing by weight, packing by cost, and so on. They have many applications, such as filling up containers, loading trucks with weight capacity constraints, etc.
- The problem of maximizing the value of items that can fit in the bin is known as the knapsack problem

Given a set of bins $S_1, S_2 \ldots$ with the same size V and a list of n items with sizes $a_1, \ldots, a_n$ to pack, find an integer number of bins B and a B-partition $S_1 \cup \cdots \cup S_B$ of the set $\{1, \ldots, n\}$ such that $\sum_{i \in S_k} a_i \leq V$ for all $k = 1, \ldots, B$.

$$\text{minimize } B = \sum_{i=1} y_i$$

$$\text{subject to } B \geq 1,$$

$$\sum_{j=1}^{n} a_j x_{ij} \leq V y_i, \forall i \in \{1, \ldots, n\}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad \forall j \in \{1, \ldots, n\}$$

$$y_i \in \{0, 1\}, \qquad \forall i \in \{1, \ldots, n\}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall i \in \{1, \ldots, n\} \forall j \in \{1, \ldots, n\}$$

where $y_i = 1$ if bin i is used and $x_{ij} = 1$ if item j is put into bin i.