Linear Programming

Abhinav Anand

Background

Linear programming involves maximizing or minimizing a linear objective function subject to linear inequality constraints.

Illustration 1:

We consider a trivial example: suppose the objective function is $f(x) = 2x$ and suppose that x is constrained to be a positive number not more than 5. More formally,

$$\max 2x : x \leq 5, x > 0$$

While this is clearly an admissible linear program, it's fairly trivial to solve. Since $2x$ is linear and monotonic in x, its solution will occur at the end point of $x = 5$ where it attains its maximum of 10.

Illustration 2:

For a linear program whose objective function has two variables, consider a classic portfolio analysis problem: bonds generate 5% returns, stocks generate 8% returns. The total budget is \$1000. How much of each asset should be bought?

We can translate this problem into a linear programming problem:

$$\max 0.05b + 0.08s : b + s \leq 1000, b \geq 0, s \geq 0$$

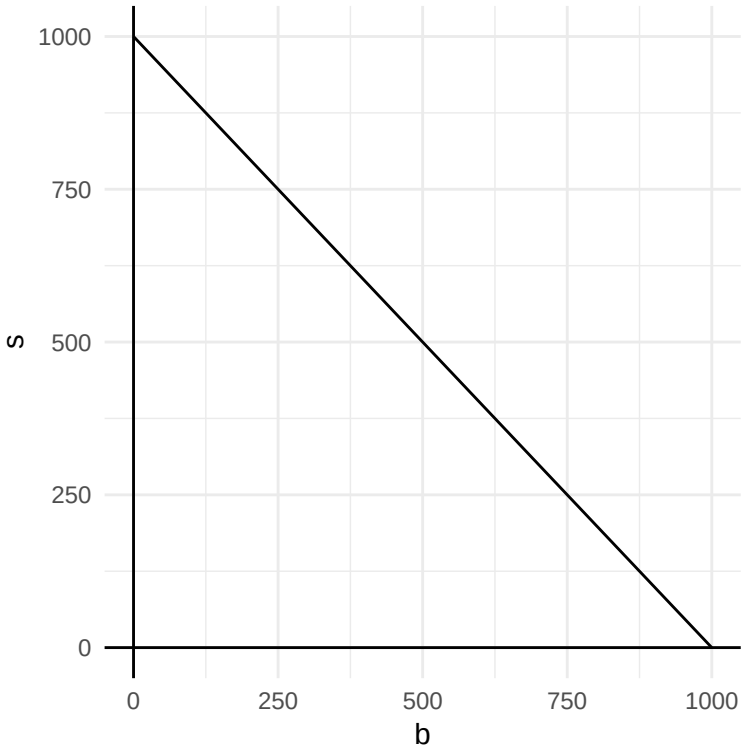
where b, s respectively are the amount (in dollars) invested in bonds and stocks.

The Feasible Set

Any combination of bonds and stocks that satisfies the inequalities: $b, s \geq 0$ and $b + s \leq 1000$ is *feasible*. In the problem above, the feasible set is the following triangular region:

```
b <- 0:1000
s <- 1000 - b

ggplot(data.frame(cbind(b, s)), aes(b, s)) +
  geom_line() +
  geom_vline(xintercept = 0) + #vertical line
  geom_hline(yintercept = 0) + #horizontal line
  theme_minimal()
```



In the problem above and more generally, in any linear programming problem, the feasible set is an intersection of *half-spaces*. Clearly, the more constraints we have, the smaller the feasible set is. The feasible set in general can be of three varieties:

1. It is empty. In this case there is no solution.
2. It is not empty but the objective function is unbounded over it. ($f(x) \in \{\infty, -\infty\}$.)
3. It is not empty *and* the objective function is bounded over it. ($f(x) \in (\infty, -\infty)$.)

Only the last case has practical value.

Finding the Maximum

In principle, to find the maximum, all we need to do is to evaluate the objective function at all feasible points; and then see which point yields the maximum. Clearly, this is not practical since there are a continuum of points in this case.

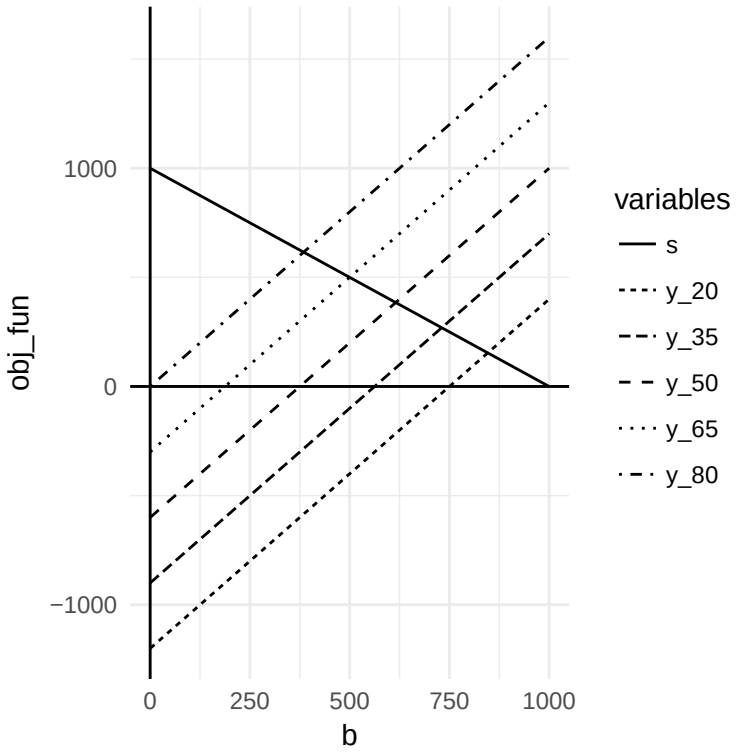
The key idea is to consider a sequence of contour lines for the objective function. Here we consider the family of lines $0.05b + 0.08s = \{20, 35, 50, \ldots\}$ etc. Then we consider which of these intersect with our feasible set. The maximum of the objective function will be attained at a *corner point*. (Can you see why? Hint: See illustration 1.)

```
y_20 <- (20-0.08*s)/0.05 #contour line: .05b+.08s=20
y_35 <- (35-0.08*s)/0.05 #contour line: .05b+.08s=35
y_50 <- (50-0.08*s)/0.05
y_65 <- (65-0.08*s)/0.05
y_80 <- (80-0.08*s)/0.05

data_plot_w <- cbind(b, s, y_20,
                     y_35, y_50,
                     y_65, y_80) %>%
  dplyr::as_tibble() #wide format

data_plot_l <- tidyr::gather(data_plot_w,
                             s:y_80,
                             key = "variables",
                             value = "obj_fun") #long format
```

```
ggplot(data = data_plot_l, aes(b, obj_fun)) +
  geom_line(aes(linetype = variables)) +
  geom_vline(xintercept = 0) +
  geom_hline(yintercept = 0) +
  theme_minimal()
```



This plot suggests that the optimal cannot occur at a strictly interior point in the feasible set and that it must occur at some corner point.[1] This is simple to see since the contour lines increase steadily upwards. This yields a tempting tentative solution which computes the objective function at all (finitely many) corners and just compares all values to find the optimal. However, for general problems, there could be several million corner points and this approach does not scale. Hence we prefer to reach the maximum in a more systematic way.

[1] In general the solution could occur along some edge as well.

The Simplex Idea

Devised by George Dantzig, the simplex method relies on a simple insight: for a canonical cost minimization program, look for an optimal solution by starting from a corner and visiting some accessible corner with lower cost until we reach a corner for which there is no accessible corner with cost any lower.

Corners

In Illustration 1, we have to maximize $2x : x \leq 5, x > 0$. (What (if any) are the corner(s) here?) In one-dimension ($x \in \mathbb{R}$), corners are merely end points. The problem above has a corner at $x = 5$. (Is $x = 0$ also a corner?)

In two dimensions, corners are intersection points of two lines. This is clearly the case in Illustration 2 which possesses three corners: (0,0), (1000,0) and (0,1000).

In three dimensions corners are intersection points of three planes—for example, ceiling corners are intersections of the three walls (planes).

Similarly in n dimensions corners are intersections of n hyperplanes. In general for a linear program there are $\binom{n+m}{n}$ possible corners. Hence brute-force evaluation of the objective function over all corners is rendered impractical.

Linear Programming: Simplex

Consider the canonical cost minimization linear program for two variables, say x_1, x_2:

$$\min\{c_1x_1 + c_2x_2\} :$$

$$a_{11}x_1 + a_{12}x_2 \geq b_1$$

$$a_{21}x_1 + a_{22}x_2 \geq b_2$$

$$x_1, x_2 \geq 0$$

Or in vector notation:

$$\min c^\top x :$$

$$Ax \geq b, x \geq 0$$

where $c = (c_1, c_2), x = (x_1, x_2), b = (b_1, b_2)$ and the matrix A is the 2×2 matrix of a_{ij}.

In general, $c = (c_1, c_2, \ldots, c_n), x = (x_1, x_2, \ldots, x_n), b = (b_1, b_2, \ldots, b_m)$; and A is $m \times n$. However the general form remains the same:

$$\min c^\top x :$$

$$Ax \geq b, x \geq 0$$

Locating a Starting Corner

In the general formulation of the canonical cost minimization linear program, where $x \in \mathbb{R}^n$, at a typical corner there are n edges that connect it to an adjacent corner. The simplex idea is to compute the values of the objective function at the accessible corners and choose the next one that has cost lower than that of the current corner. Then at the next corner, repeat the process and choose another less costly corner. Eventually there will be a special corner all of whose adjacent corners are more expensive. This is the optimal as reached by the simplex method.

Corners are simply the intersection points of inequalities. In general there are n inequalities from the $x \geq 0$ constraints and m inequalities from $Ax \geq b$ leading to a total of $n + m$ inequalities.

Slack Variables

In Illustration 2, the inequality could be re-written after the introduction of an auxiliary "slack" variable w:

$$b + s \leq 1000, b, s \geq 0 \iff b + s + w = 1000, b, s, w \geq 0$$

This transforms the inequalities $Ax \geq 0$ to $A\tilde{x} = 0$ where $\tilde{x} = (x, w)$; while retaining the other inequalities in the form $(x, w) = \tilde{x} \geq 0$. As may be seen, the slack variables $w = Ax - b$ turn "loose" inequalities into "tight" inequalities. The new costs become $\tilde{c} = (c, 0)$.

The new linear program now becomes:

$$\max(c, 0)^\top (x, w) : Ax - Iw = b, (x, w) \geq 0$$

or,

$$\max \tilde{c}^\top \tilde{x} : [A - I] \begin{bmatrix} x \\ w \end{bmatrix} = b, \begin{bmatrix} x \\ w \end{bmatrix} \geq 0$$

or, simply without loss of generality,

$$\max c^\top x : Ax = b, x \geq 0$$

This is a general canonical form of a linear program. The equality constraints are $Ax = b$ and the inequality constraints are $x \geq 0$. There are still $n + m$ inequalities but the slack variables have moved them from $Ax \geq b$ to $x \geq 0$. The augmented vector now is $x = (x_1, \ldots, x_n, w_1, \ldots, w_m)$.

Starting Corner

Translating the idea of a corner into algebra, we say that a corner of the feasible set occur when exactly n out of the $n + m$ components of x equal 0. These are exactly the *basic feasible* solutions of $Ax = b$.[2] Basic variables are computed by solving $Ax = b$ and free free components of x are set to 0.

Once the starting corner is found, we move along the edge to an adjacent corner with lower cost.

[2] 'Basic', if n components equal 0; and 'feasible' if $x \geq 0$.

Duality

Every linear programming problem (called the *primal* problem) can be converted into a *dual* problem, the solution of which provides an upper bound for the solution of the primal.

Primal : $\min\{c^\top x\} : Ax \leq b, x \geq 0$

The dual problem contains the same fundamental building blocks A, b, c but reverses everything: in the primal, c captures the costs while b contains the constraints. In the dual problem, b enters the objective function and c forms the constraints. Also, while the primal is a minimization program, its dual is a maximization program.

Dual: $\max\{b^\top y\} : A^\top y \geq c, y \geq 0$

There are two fundamental ideas in duality:

1. The dual of the dual (the so-called *bidual*) is the primal. (Can you see how?)
2. Every feasible solution for a primal gives a bound on the optimal value for the dual.

Duality Theorem

When primal and dual are both feasible with optimal vectors x^*, y^* respectively, the minimum of the primal $c^\top x*$ is equal to the maximum of the dual $b^\top y$.