Linear Programming

Abhinav Anand

Background

Linear programming involves maximizing or minimizing a linear objective functions 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:

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

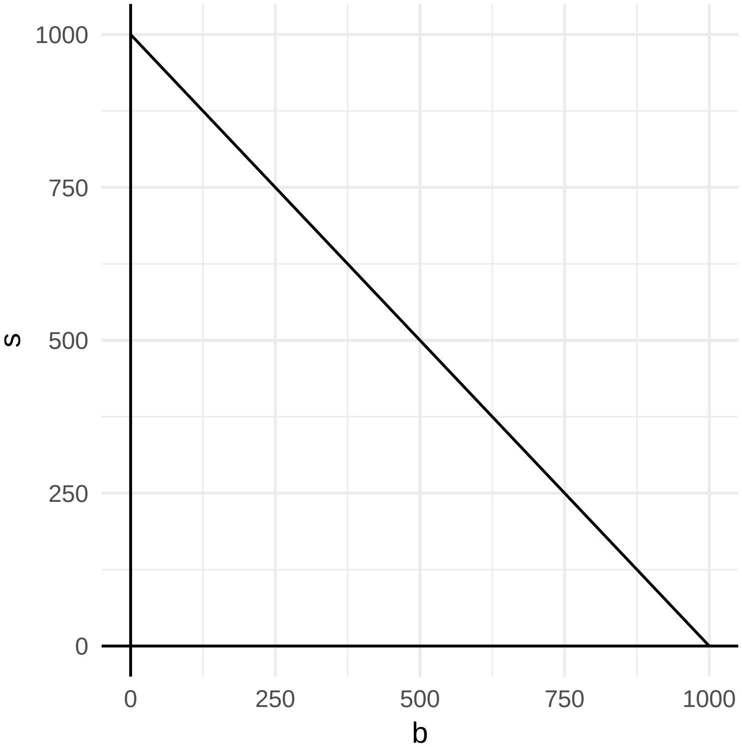
where b, s respectively are the amount (in dollars) invested in bonds and stocks.

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:

```r
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 Minimum

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

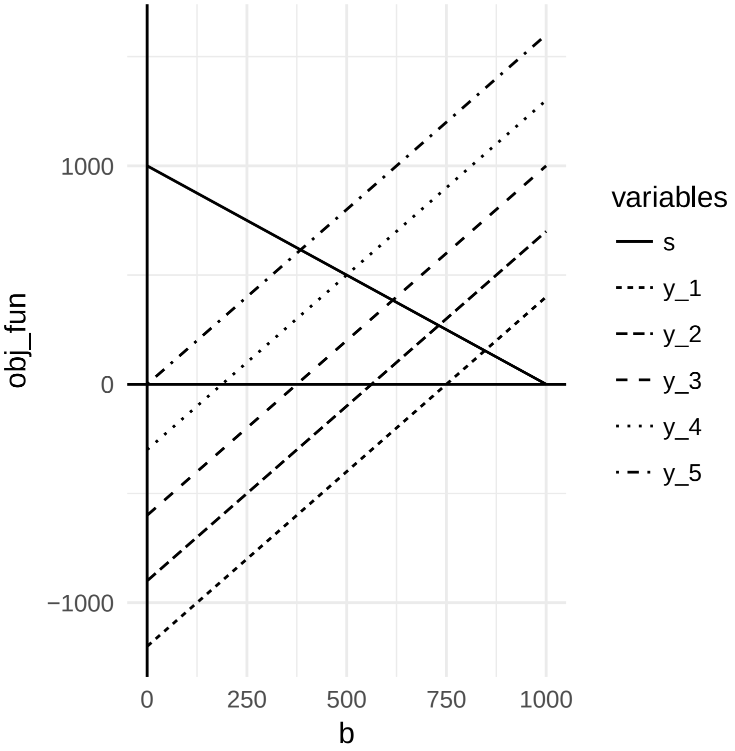
The key idea is to consider a sequence of contour lines for the objective function. Here we consider the family of lines $0.05b + 0.08s = \{20, 35, 50, \ldots\}$ etc. Then we consider which of these intersect with our feasible set. The maximum of the objective function will be attained at a *corner point*.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

```r
y_1 <- (20-0.08*s)/0.05 #contour line: .05b+.08s=20
y_2 <- (35-0.08*s)/0.05
y_3 <- (50-0.08*s)/0.05
y_4 <- (65-0.08*s)/0.05
y_5 <- (80-0.08*s)/0.05

data_plot_w <- cbind(b, s, y_1,
                     y_2, y_3,
                     y_4, y_5) %>%
  dplyr::as_tibble() #wide format

data_plot_l <- tidyr::gather(data_plot_w,
                             s:y_5,
                             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 move steadily upwards until they intersect the feasible set. This yields a tempting tentative solution which compute 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 minimum in a more systematic way.

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

The Simplex Method

Devised by George Dantzig, the simplex method relies on a simple insight: 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.