Linear Programming

Abhinav Anand

Background

Linear programming involves minimizing linear cost functions subject to linear inequality constraints.[1]

Illustration:

To illustrate, 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

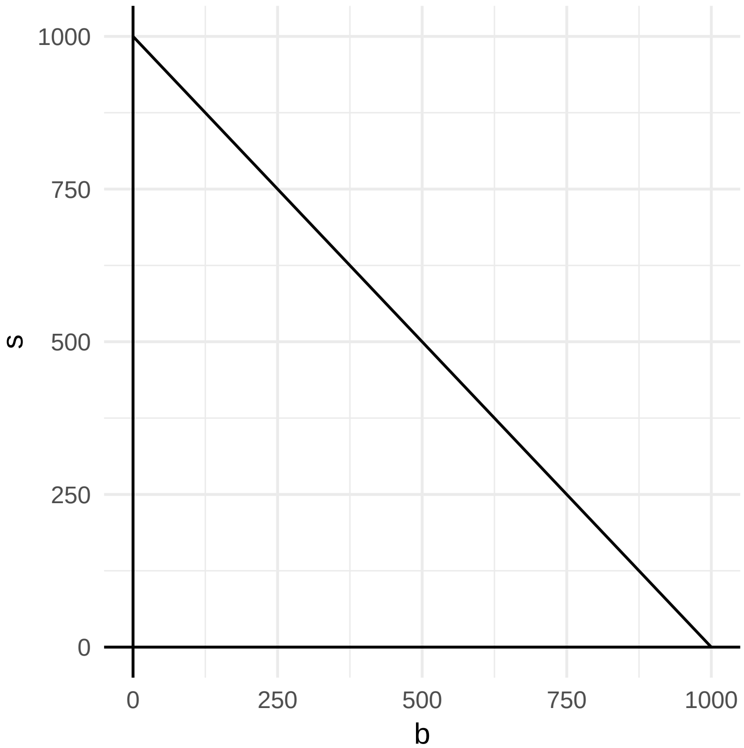
Any combination of bonds and stocks that satisfies the inequalities: $b, s \geq 0$ and $b + s \leq 1000$ is *feasible*. In the problem above, the feasible set is the following triangular region:

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

ggplot(data.frame(cbind(b, s)), aes(b, s)) +
  geom_line() +
```

[1] This is without loss of generality since a maximization program is the negative of the corresponding minimization program.

```r
  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 cost function is unbounded over it. The minimum cost is $-\infty$.
3. It is not empty *and* the cost function is bounded over it.

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. Hence we prefer to reach the minimum in a more systematic way.