The Simplex algorithm: Introductory example

The following introduction to the Simplex algorithm is from the book *Linear Programming* by V. Chvátal.

Example:

$$\begin{array}{rl} \max & 5x_1 + 4x_2 + 3x_3 \\ \text{subject to} & 2x_1 + 3x_2 + x_3 \leq 5 \\ & 4x_1 + x_2 + 2x_3 \leq 11 \\ & 3x_1 + 4x_2 + 2x_3 \leq 8 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

The Simplex algorithm: Introductory example (2)

Introduce *slack variables* and obtain *standard form*[1]:

$$\begin{aligned} x_4 &= 5 - 2x_1 - 3x_2 - x_3 \\ x_5 &= 11 - 4x_1 - x_2 - 2x_3 \\ x_6 &= 8 - 3x_1 - 4x_2 - 2x_3 \\ z &= 5x_1 + 4x_2 + 3x_3 \end{aligned}$$

$$\max z \quad \text{subject to} \quad x_1, \ldots, x_6 \geq 0$$

Slack variables: x_4, x_5, x_6

Decision variables: x_1, x_2, x_3

The Simplex algorithm: Introductory example (3)

Idea of the Simplex algorithm: Start with *feasible solution* $x_1, \ldots, x_6$ and try to find another feasible solution $\overline{x}_1, \ldots, \overline{x}_6$ with

$$5\overline{x}_1 + 4\overline{x}_2 + 3\overline{x}_3 > 5x_1 + 4x_2 + 3x_3 \ .$$

Iterate until optimal solution is found.

Problem 1: Find first feasible solution. In our example it is easy:

$$x = (0, 0, 0, 5, 11, 8)$$

will do and yields $z = 0$.

The Simplex algorithm: Introductory example (4)

Problem 2: Finding a better feasible solution.

Looking at z, we have to increase x_1, x_2, or x_3.

[1] Note: Chvátal calls the form on the previous slide standard form.

Blackboard calculations tell us: $x_1 \leq \frac{5}{2}, x_1 \leq \frac{11}{4}, x_1 \leq \frac{8}{3}$, so

$$\bar{x} = (\tfrac{5}{2}, 0, 0, 0, 1, \tfrac{1}{2}) \quad \text{yielding} \quad z = \tfrac{25}{2} \ .$$

Finding an even better solution? Not so easy. We rewrite our system of equations such that, again, nonzero variables appear at the left-hand side and zero variables on the right-hand side (thus, x_1 and x_4 swap their positions):

$$x_1 = \frac{5}{2} - \frac{1}{2}x_4 - \frac{3}{2}x_2 - \frac{1}{2}x_3$$

The Simplex algorithm: Introductory example (5)

New system:

$$\begin{aligned} x_1 &= \frac{5}{2} - \frac{1}{2}x_4 - \frac{3}{2}x_2 - \frac{1}{2}x_3 \\ x_5 &= 1 + 5x_2 + 2x_4 \\ x_6 &= \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\ z &= \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4 \end{aligned}$$

Next step: increase x_3.

Blackboard calculations: $x_3 = 1$, yielding $(2, 0, 1, 0, 1, 0)$ and $z = 13$.

The Simplex algorithm: Introductory example (6)

New system:

$$\begin{aligned} x_3 &= 1 + x_2 + 3x_4 - 2x_6 \\ x_1 &= 2 - 2x_2 - 2x_4 + x_6 \\ x_5 &= 1 + 5x_2 + 2x_4 \\ z &= 13 - 3x_2 - x_4 - x_6 \end{aligned}$$

We are done!

Why? We have found a solution with $z = 13$ and every feasible solution must satisfy $z = 13 - 3x_2 - x_4 - x_6$ and $x_2, x_4, x_6 \geq 0$.

The Simplex algorithm

Given a problem

$$\begin{aligned} \max \quad & \sum_{j=1}^{n} c_j x_j & & \\ \text{subject to} \quad & \sum_{j=1}^{n} a_{ij} x_j \leq b_i & & i = 1, 2, \ldots, m \\ & x_j \geq 0 & & j = 1, 2, \ldots, n \end{aligned}$$

we introduce slack variables and arrive at our first *dictionary*

$$x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij} x_j \qquad\qquad i = 1, 2, \ldots, m$$
$$z = \sum_{j=1}^{n} c_j x_j \ ,$$

which characterizes each feasible solution as $n+m$ non-negative numbers $x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m}$, where $x_{n+1}, \ldots, x_{n+m}$ depend on $x_1, \ldots, x_n$.

The Simplex algorithm (2)

An iteration of the Simplex algorithm consists of replacing a feasible solution

$$x_1, \ldots, x_{n+m} \quad \text{by} \quad \overline{x}_1, \ldots, \overline{x}_{n+m} \ ,$$

with

$$\sum_{j=1}^{n} c_j \overline{x}_j > \sum_{j=1}^{n} c_j x_j \ .$$ [2]

We do so by choosing a (non-basic) variable x_N from the right-hand side with positive objective function coefficient and increasing its value maximally, thereby setting a left-hand side (basic) variable x_B to zero.

We compute a new dictionary where x_N is left (basic) and x_B is right (non-basic).

The Simplex algorithm (3)

Definition: (dictionary)

- The equations in a dictionary express m of the variables $x_1, \ldots, x_{m+n}$ and the objective function z in terms of the remaining n variables.
- Every solution of a dictionary must be also a solution of the first dictionary.

A dictionary is *feasible* if setting the non-basic variables to zero and evaluating the basic variables yields a feasible solution.

The Simplex algorithm moves from feasible dictionary to feasible dictionary.

The Simplex algorithm (4)

Are we done? No!

- How do we find the first feasible dictionary? (Initialization)
- Can we always find an entering and leaving variable to construct the next dictionary? (Iteration)
- Does the process terminate? (Termination)

The Simplex algorithm (5)

[2] As we will later see, the inequality is not always strict.

We start with the second issue (Iteration):

Consider the last row of a dictionary

$$z = z^* + \sum_{j \in N} \bar{c}_j x_j \quad ,$$

where N contains the indices of non-basic variables. If $\bar{c}_j \leq 0$ for all $j \in N$, we are done.

Otherwise, choose any variable with positive coefficient for *entering* the basis, e. g., the one with the largest coefficient $\bar{c}_j$.

Now the *leaving* variable is that basic variable whose non-negativity imposes the most stringent upper bound on the increase of the entering variable.

Problems:

1. no candidate
2. multiple candidates

The Simplex algorithm (6)

Problem 1 (no candidate). Consider

$$\begin{aligned} x_2 &= 5 + 2x_3 - x_4 - 3x_1 \\ x_5 &= 7 - 3x_4 - 4x_1 \\ z &= 5 + x_3 - x_4 - x_1 \end{aligned}$$

No restrictions on x_3. Setting $x_3 = t$ yields

$$x = (0, 5 + 2t, t, 0, 7) \quad \text{with} \quad z = 5 + t \quad ,$$

that is, the problem is *unbounded*. Of course, this holds in general.

Problem 2 (multiple candidates). In this case, we may choose *any* candidate. This leads, however, to *degenerate* solutions.

The Simplex algorithm (7)

A basic solution is *degenerate* if one or more basic variables are at zero.

This might have an annoying side effect: non-increasing iterations.

Example:

$$\begin{aligned} x_4 &= 1 - 2x_3 \\ x_5 &= 3 - 2x_1 + 4x_2 - 6x_3 \\ x_6 &= 2 + x_1 - 3x_2 - 4x_3 \\ z &= 2x_1 - x_2 + 8x_3 \end{aligned}$$

If, as in this example, only a few iterations are degenerate, this is not a big problem. Otherwise, we may *cycle*, which is a serious problem!

The Simplex algorithm (8)

Cycling.

Can the Simplex go through an endless sequence of iterations without ever finding an optimal solution?

Yes:

$$
\begin{aligned}
x_5 &= -\frac{1}{2}x_1 + \frac{11}{2}x_2 + \frac{5}{2}x_3 - 9x_4 \\
x_6 &= -\frac{1}{2}x_1 + \frac{3}{2}x_2 + \frac{1}{2}x_3 - x_4 \\
x_7 &= 1 - x_1 \\
z &= 10x_1 - 57x_2 - 9x_3 - 24x_4
\end{aligned}
$$

Now, choosing the entering variable according to the “largest coefficient” rule and the leaving variable according to the “lexicographic” rule, we will see the same dictionary again after six iterations!