The Simplex algorithm

Sticking to certain *pivoting* rules prevents cycling:

E.g., Bland’s rule: among multiple candidates for entering/leaving the basis always choose the one with the smallest subscript.

This answers the third issue (Termination):

**Theorem.** The simplex method with Bland’s rule terminates after a finite number of steps.

**Proof.** Since the algorithm does not cycle and there are only \( \binom{n+m}{m} \) different dictionaries, the claim follows.

Unfortunately, pathological instances exist (e.g., the Klee-Minty cube), for which the Simplex method needs *exponential* time. However,

- in practice, the method is fast.
- other methods (e.g., Ellipsoid method) run in polynomial time.
We are left with only one issue (Initialization):

How do we find an initial dictionary if

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

has an infeasible origin?

Problems:

- Is there a feasible solution at all? (The problem might be infeasible)
- If so, how to find it?
Solution: Auxiliary problem

\[
\begin{align*}
\text{min} & \quad x_0 & \quad \text{(AUX)} \\
\text{subject to} & \quad \sum_{j=1}^{n} a_{ij}x_j - x_0 & \leq b_i & \quad i = 1, 2, \ldots, m \\
& \quad x_j & \geq 0 & \quad j = 0, 1, \ldots, n
\end{align*}
\]

Now, a feasible solution for (AUX) is easily found:

Set \( x_j = 0 \) for \( j \in \{1, 2, \ldots, n\} \) and make \( x_0 \) sufficiently large.

Furthermore: the original problem has a feasible solution if and only if the optimum value of (AUX) is zero.

Thus, we solve (AUX) first.
The Simplex algorithm

Example.

Maximize \( x_1 - x_2 + x_3 \)
subject to
\[
\begin{align*}
2x_1 - x_2 + 2x_3 & \leq 4 \\
2x_1 - 3x_2 + x_3 & \leq -5 \\
-x_1 + x_2 - 2x_3 & \leq -1 \\
x_1, x_2, x_3 & \geq 0
\end{align*}
\]
leads to

Maximize \(-x_0\) (AUX)
subject to
\[
\begin{align*}
2x_1 - x_2 + 2x_3 - x_0 & \leq 4 \\
2x_1 - 3x_2 + x_3 - x_0 & \leq -5 \\
-x_1 + x_2 - 2x_3 - x_0 & \leq -1 \\
x_0, x_1, x_2, x_3 & \geq 0
\end{align*}
\]

The first dictionary for (AUX) then looks like

\[
\begin{align*}
x_4 &= 4 - 2x_1 + x_2 - 2x_3 + x_0 \\
x_5 &= -5 - 2x_1 + 3x_2 - x_3 + x_0 \\
x_6 &= -1 + x_1 - x_2 + 2x_3 + x_0 \\
w &= -x_0
\end{align*}
\]

which is also infeasible! So where's the advantage?
We can make it feasible by one single pivot, namely by having \( x_0 \) enter the basis and having \( x_5 \) leave it.

This yields the feasible dictionary

\[
\begin{align*}
    x_0 &= 5 + 2x_1 - 3x_2 + x_3 + x_5 \\
    x_4 &= 9 - 2x_2 - x_3 + x_5 \\
    x_6 &= 4 + 3x_1 - 4x_2 + 3x_3 + x_5 \\
    w &= -5 - 2x_1 + 3x_2 - x_3 - x_5
\end{align*}
\]

from which we can read off the first feasible solution for (AUX)

\[
    x = (5, 0, 0, 0, 9, 0, 6) \quad \text{with} \quad w = -5
\]
Two more iterations, namely

\[
\begin{align*}
    x_2 &= 1 + \frac{3}{4} x_1 + \frac{3}{4} x_3 + \frac{1}{4} x_5 - \frac{1}{4} x_6 \\
    x_0 &= 2 - \frac{1}{4} x_1 - \frac{1}{4} x_3 + \frac{1}{4} x_5 + \frac{3}{4} x_6 \\
    x_4 &= 7 - \frac{2}{5} x_1 - \frac{2}{5} x_3 + \frac{1}{5} x_5 + \frac{2}{5} x_6 \\
    w &= -2 + \frac{1}{4} x_1 + \frac{5}{4} x_3 - \frac{1}{4} x_5 - \frac{3}{4} x_6 \\
\end{align*}
\]

and

\[
\begin{align*}
    x_3 &= \frac{8}{5} - \frac{1}{5} x_1 + \frac{1}{5} x_5 + \frac{3}{5} x_6 - \frac{4}{5} x_0 \\
    x_2 &= \frac{11}{5} + \frac{3}{5} x_1 + \frac{2}{5} x_5 - \frac{1}{5} x_6 - \frac{3}{5} x_0 \\
    x_4 &= 3 - x_1 - x_6 + 2 x_0 \\
    w &= -x_0
\end{align*}
\]

solve (AUX) and its optimal value is \( w = 0 \). Therefore, we can read off a first feasible solution

\[(0, \frac{11}{5}, \frac{8}{5}, 3, 0, 0) \ldots\]
... and a first feasible dictionary:

\[
\begin{align*}
    x_3 &= \frac{8}{5} - \frac{1}{5}x_1 + \frac{1}{5}x_5 + \frac{3}{5}x_6 \\
    x_2 &= \frac{11}{5} + \frac{3}{5}x_1 + \frac{4}{5}x_5 + \frac{1}{5}x_6 \\
    x_4 &= 3 - x_1 - x_6 \\
    z &= x_1 - x_2 + x_3 = x_1 - \left( \frac{11}{5} + \frac{3}{5}x_1 + \frac{4}{5}x_5 + \frac{1}{5}x_6 \right) + \left( \frac{8}{5} - \frac{1}{5}x_1 + \frac{1}{5}x_5 + \frac{3}{5}x_6 \right) \\
    &= \frac{3}{5} + \frac{1}{5}x_1 - \frac{1}{5}x_5 + \frac{2}{5}x_6
\end{align*}
\]

Now, we can go on with the regular Simplex method.
The Simplex algorithm (9)

General method (first phase of two-phase Simplex):

We solve

\[
\begin{align*}
\text{max} & \quad -x_0 \quad \text{(AUX)} \\
\text{s.t.} & \quad \sum_{j=1}^{n} a_{ij}x_j - x_0 \leq b_i \quad i = 1, 2, \ldots, m \\
& \quad x_j \geq 0 \quad j = 1, 2, \ldots, n
\end{align*}
\]

by starting with an infeasible dictionary

\[
x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij}x_j + x_0 \quad i = 1, 2, \ldots, m
\]

\[
w = -x_0
\]

We arrive at a feasible dictionary by swapping \( x_0 \) with the "most infeasible" \( x_{n+i} \), more precisely, with \( x_{n+ (\arg \min_{i=1,\ldots,m} b_i)} \).
The Simplex algorithm

One more special rule when solving (AUX):

Whenever \( x_0 \) is a candidate for leaving the basis, we pick it.

Why? Because we obtain a feasible solution with \( x_0 = 0 \) and thus \( w = 0 \) due to the properties of a dictionary.

Do other cases exist? After termination of phase one

- \( x_0 \) may be basic, and the value of \( w \) is zero. But then, in the previous iteration, we had \( w < 0 \) and thus \( x_0 > 0 \) due to \( w = -x_0 \). So, we have not followed the special rule for picking \( x_0 \) whenever possible; thus, this case may not occur.

- \( x_0 \) may be basic, and the value of \( w \) is non-zero. This case proves that the original problem is infeasible.
The Simplex algorithm

We are now ready for the

Fundamental theorem of linear programming. Every LP problem has the following three properties:

1. If it has no optimal solution, then it is either infeasible or unbounded.

2. If it has a feasible solution, then it has a basic feasible solution.

3. If it has an optimal solution, then it has a basic optimal solution.

Proof (constructive). The first phase of the two-phase Simplex algorithm either discovers that the problem is infeasible or computes a basic feasible solution. The second phase then finds a basic optimal solution or discovers that the problem is unbounded.
Duality
Consider

\[
\begin{align*}
\text{max} & \quad 4x_1 + x_2 + 5x_3 + 3x_4 \\
\text{subject to} & \quad x_1 - x_2 - x_3 + 3x_4 \leq 1 \\
& \quad 5x_1 + x_2 + 3x_3 + 8x_4 \leq 55 \\
& \quad -x_1 + 2x_2 + 3x_3 - 5x_4 \leq 3 \\
& \quad x_1, x_2, x_3, x_4 \geq 0
\end{align*}
\]

Let us try to find a quick estimate on the optimal solution value \(z^*\).

Lower bounds? Rather run Simplex. . .

Upper bounds?
Duality: Introductory example

Blackboard calculations lead to the dual problem

\[
\begin{align*}
\text{min} & \quad y_1 + 55y_2 + 3y_3 \\
\text{subject to} & \quad y_1 + 5y_2 - y_3 \geq 4 \\
& \quad - y_1 + y_2 + 2y_3 \geq 1 \\
& \quad - y_1 + 3y_2 + 3y_3 \geq 5 \\
& \quad 3y_1 + 8y_2 - 5y_3 \geq 3 \\
& \quad y_1, y_2, y_3 \geq 0
\end{align*}
\]
In general, the dual of

\[
\text{max} \quad \sum_{j=1}^{n} c_j x_j \quad \text{(primal problem)}
\]

subject to \[
\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad i = 1, 2, \ldots, m
\]
\[
x_j \geq 0 \quad j = 1, 2, \ldots, n
\]
is

\[
\text{min} \quad \sum_{i=1}^{m} b_i y_i \quad \text{(dual problem)}
\]

subject to \[
\sum_{i=1}^{m} a_{ij} y_i \geq c_j \quad j = 1, 2, \ldots, n
\]
\[
y_i \geq 0 \quad i = 1, 2, \ldots, m
\]

Lemma. (Weak duality) \[
\sum_{j=1}^{n} c_j x_j \leq \sum_{i=1}^{m} b_i y_i
\]
Proof. Blackboard.