Simplex Algorithm

In the worst-case this algorithm for LP runs in exponential in the number of variables and constraint's (m+n), but in practice it runs often quite fast.

Consider the following LP in standard form:

Maximize \[ 3x_1 + x_2 + 2x_3 \]

S.t. \[ x_1 + x_2 + 3x_3 \leq 30 \]
\[ 2x_1 + 2x_2 + 5x_3 \leq 24 \]
\[ 4x_1 + x_2 + 2x_3 \leq 36 \]
\[ x_1, x_2, x_3 \geq 0 \]

Corresponding slack form:

\[ z = 3x_1 + x_2 + 2x_3 \]
\[ x_4 = 30 - x_1 - x_2 - 3x_3 \]
\[ x_5 = 24 - 2x_1 - 2x_2 - 5x_3 \]
\[ x_6 = 36 - 4x_1 - x_2 - 2x_3 \]

A solution is feasible if \( x_i \geq 0, \forall i = 1, \ldots, 6 \).
Basic solution: Set all non-basic variables on the RHS to zero.

Basic sol: \((\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_6) = (0, 0, 0, 30, 24, 36)\)
has objective value
\[ Z = (3.0) + (1.0) + (2.0) = 0 \]

- Simplex algorithm re-writes constraints and objective function so that basic and non-basic variables are exchanged.

- By above exchange, LP solution does not change.

- A feasible basic solution is almost always maintained by the algorithm.

- The goal is to rewrite the LP so that new basic solution has better objective value.
- Select a non-basic variable $x_e$ whose coefficient $c_e$ in the objective function is positive.

- Increase the value of $x_e$ as much as possible to increase the objective function value.

- In our example $x_1$ has coefficient $+3$ in the objective function.

- We cannot increase $x_1$ arbitrarily since $x_4, x_5, x_6$ decrease with increasing $x_1$ and they have to remain positive.

- The constraint

\[ x_6 = 36 - 4x_1 - x_2 - 2x_3 \]

is the tightest which allows $x_1$ to increase up to 9.

- Switch the roles of $x_6$ and $x_1$

\[ x_1 = 9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4} \]
Rewrite other constraints with $x_6$ on the right:

\[
x_4 = 30 - x_1 - x_2 - 3x_3 \\
= 30 - (9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4}) - x_2 - 3x_3 \\
= 21 - \frac{3x_2}{4} - \frac{5x_3}{2} + \frac{x_6}{4}
\]

\[x_5 = \ldots\ldots\ldots\]

Similarly, eliminate $x_1$ from the objective function and bring in $x_6$. New LP in re-written form:

\[
Z = 27 + \frac{x_2}{4} + \frac{x_3}{2} - \frac{3x_6}{4}
\]

\[
x_1 = 9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4}
\]

\[
x_4 = 21 - \frac{3x_2}{4} - \frac{5x_3}{2} + \frac{x_6}{4}
\]

\[
x_5 = 6 - \frac{3x_2}{2} - 4x_3 + \frac{x_6}{2}
\]

This entire operation is called **pivot**.

Pivot chooses a nonbasic variable $x_e$ called **entering variable** and make it basic, replacing a basic variable called **leaving variable** denoted $x_e$. 
Continue Pivoting:

- Choose \( x_3 \): The third constraint
  \[ x_5 = 6 - \frac{3x_2}{2} - 4x_3 + \frac{x_6}{2} \]
  is the tightest.

So, \( x_e = x_3 \), \( x_l = x_5 \), LP re-written:

\[
Z = \frac{111}{4} + \frac{x_2}{16} - \frac{x_5}{8} - \frac{11x_6}{16}
\]
\[
x_1 = \frac{33}{4} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{5x_6}{16}
\]
\[
x_3 = \frac{3}{2} - \frac{3x_2}{8} - \frac{x_5}{4} + \frac{x_6}{8}
\]
\[
x_4 = \frac{69}{4} + \frac{3x_2}{16} + \frac{5x_5}{8} - \frac{x_6}{16}
\]

Basic soln: \( (\frac{33}{4}, 0, \frac{3}{2}, \frac{69}{4}, 0, 0) \)
\[ Z = \frac{111}{4} \]

- Choose \( x_2 \) (this is the only way to increase objective value)

Three constraints have 132, 4, \( \infty \) as upper bounds.
So, \( x_e = x_2 \), \( x_l = x_3 \)
\[ z = 28 - \frac{x_2}{6} - \frac{x_5}{6} - \frac{2x_6}{3} \]
\[ x_1 = 8 + \frac{x_2}{6} + \frac{x_5}{6} - \frac{x_6}{3} \]
\[ x_2 = 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3} \]
\[ x_9 = 18 - \frac{x_3}{2} + \frac{x_5}{2} \]

All coefficients now in the objective function are negative. At this point we have achieved optimal solution.

So, the solution \((8,4,0,18,0,0)\) which gives objective value 28 is the optimal solution.

Pivot: takes \((N,B,A,b,c,u)\) in slack form, index \(l\) for leaving variable, index \(e\) for entering variable.

Input: \((N,B,A,b,c,u,b,e)\)

Output: \((\hat{N}, \hat{B}, \hat{A}, \hat{b}, \hat{c}, \hat{u})\) new slack form.
Pivot algorithm

Pivot \((N, B, A, b, c, v, l, e)\)

* Compute coefficients of the equation of \(xe\)

\[ \hat{b}_e := b_e / a_{le} \]

for each \(j \in N - \{e\}\)

\[ \hat{a}_{ej} := a_{ij} / a_{le} \]

\[ \hat{a}_{el} := 1 / a_{le} \]

* Compute coefficients of other constraints

for \(i \in B - \{l\}\)

do \( \hat{b}_i := b_i - a_{ie} \hat{b}_e \)

for each \(j \in N - \{e\}\)

do \( \hat{a}_{ij} := a_{ij} - a_{ie} \hat{a}_{ej} \)

\[ \hat{a}_{ib} := -a_{ie} \hat{a}_{el} \]

* Compute the objective function.

\[ \hat{v} := v + c_e \hat{b}_e \]

for each \(j \in N - \{e\}\)

do \( \hat{c}_j := c_j - c_e \hat{a}_{ej} \)

\[ \hat{c}_l := -c_e \hat{a}_{el} \]

* Compute basic and nonbasic variables

\[ \hat{N} := N - \{e\} U \{l\} \]

\[ \hat{B} := B - \{l\} U \{e\} \]

return \((\hat{N}, \hat{B}, \hat{A}, \hat{b}, \hat{c}, \hat{v})\).
Simplex Algorithm

Issues:
- How do we determine if LP is feasible?
- What do we do if LP is feasible but initial basic solution is not?
- How do we determine LP is unbounded?
- How to choose entering and leaving variable?

Initialize \((A, b, c)\): takes an LP in standard form.

\[- A: \{a_{ij}\} \text{ mxn matrix} \]
\[- b: (b_i) : m \text{-vecnr} \]
\[- c: (c_j): n \text{-vecnr} \]

If LP is infeasible, it returns saying LP is infeasible. Otherwise, it returns a slack form where initial basic solution is feasible.

Simplex \((A, b, c)\): takes LP in standard form returns \(n\)-vector \(\mathbf{x} = (x_j)\), an optimal solution.
Simplex \((A, b, c)\)
\((N, B, A, b, c, u) := \text{Initialize} (A, b, c)\)

while \(\exists j \in N\) has \(c_j > 0\)
do choose \(e \in N\) for which \(c_e > 0\)
for \(i \in B\)
do if \(a_{ie} > 0\)
then \(\Delta_i := b_i / a_{ie}\)
else \(\Delta_i := \infty\)

choose \(e \in B\) that minimizes \(\Delta_i\)

If \(\Delta_e = \infty\)
then return "unbounded"
else
\((N, B, A, b, c, u) := \text{Pivot} (N, B, A, b, c, u, l, e)\)

for \(i := 1\) to \(n\)
do if \(i \in B\)
then \(\bar{x}_i := b_i\)
else \(\bar{x}_i := 0\)

return \((\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)\).
Lemma 1. Simplex returns a feasible solution or an "unbounded" solution.

Proof. Loop invariant for the outer while loop:
At the beginning of the loop:
1. the slack form is equivalent to the original slack form returned by Initialize
2. for \( i \in B, \ b_i \geq 0 \)
3. the basic solution is feasible.

Show the three invariants at
(a) Initialization, (b) in the middle, (c) at termination

(See the proof in the book for all three cases).
Termination

Now we show that Simplex can always be made to terminate. (Why can it cycle?)

\[ Z = x_1 + x_2 + x_3 \]
\[ x_4 = 8 - x_1 - x_2 \]
\[ x_5 = x_2 - x_3 \]
\[ x_6 = x_3 \]
\[ x_7 = x_6 \]
\[ Z = 8 + x_3 - x_4 \]
\[ x_1 = 8 - x_2 - x_4 \]
\[ x_5 = x_2 - x_3 \]

\( \{ \) Objective value didn't change

Fortunately, if we pivot with \( x_2 \) entering and \( x_1 \) leaving, objective value increases. But, it can happen that objective value remains same with successive pivoting. Then, Simplex algorithm "cycles" through identical slack forms.

How can we detect "cycles"?

**Lemma 2** Let \( I \) be a set of indices. For \( i \in I \), \( \alpha_i, \beta_i \) are reals, \( x_i \) real variable, \( Y \) a real no.

If \( \Sigma \alpha_i x_i = Y + \Sigma \beta_i x_i \) then \( \alpha_i = \beta_i \) and \( Y = 0 \).
Lemma 3. Let \((A, b, c)\) be an LP in standard form. Given a set \(B\) of basic variables, the slack form is uniquely determined.

Proof. 
\[
z = v + \sum_{j \in N} c_j x_j, \quad x_i = b_i - \sum_{j \in N} a_{ij} x_j \quad i \in B
\]
\[
z' = v' + \sum_{j \in N} c'_j x_j, \quad x_i = b'_i - \sum_{j \in N} a'_{ij} x_j \quad i \in B
\]
be two slack forms for some \(B\).

\[
0 = (b_i - b'_i) - \sum (a_{ij} - a'_{ij}) x_j, \quad i \in B
\]
\[
\sum_{j \in N} a_{ij} x_j = (b_i - b'_i) + \sum_{j \in N} a'_{ij} x_j, \quad i \in B.
\]

Apply Lemma 2, to claim

\[
b_i = b'_i, \quad a_{ij} = a'_{ij}.
\]

Also, show \(c = c', \quad v = v'\).

Lemma 4. If Simplex fails to terminate in at most \(\binom{n+m}{m}\) iterations, then it cycles.

Proof. There are at most \(\binom{n+m}{m}\) different \(B\) since \(|B| = m\) and total variables is \(n+m\). Thus, there are at most \(\binom{n+m}{m}\) unique slack forms. Conclusion follows.
Cycling can be avoided by choosing entering and leaving variables carefully. Break ties by choosing the variable with the smallest index: Bland's rule.

**Lemma 5.** If ties are always broken with Bland's rule, Simplex terminates.

We will show that when Simplex returns a feasible solution, it is always optimal. This is shown by duality.