1 Outline of the (Revised) Simplex Algorithm.

A standard form LP is:

$$\min \mathbf{c}^T\mathbf{x} \tag{1}$$
$$\text{st. } \mathbf{A}\mathbf{x} = \mathbf{b}, \tag{2}$$
$$\mathbf{x} \geq 0. \tag{3}$$

Matrix $\mathbf{A}$ has n columns and m rows, and $n > m$.

Usually the problem is originally defined with the use of both equality and inequality constraints, as well as special inequality constraints called *bounds*, i.e., instead of (2) we have:

$$\mathbf{A}^E\mathbf{x} = \mathbf{b}^E, \tag{4}$$
$$\mathbf{A}^L\mathbf{X} \leq \mathbf{b}^L, \tag{5}$$
$$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}. \tag{6}$$

Note that (5)–(6) can be transformed into constraint of type (2) by extending the vector of variables $\mathbf{x}$ by *slack* variables and adding them to the lefthand side of (5). Also the inequalities of type "greater than" can be replaced by that of type "less than" by multiplying both sides by -1. Thus we assume that the problem is transformed into the standard form (1)–(3) before we run Simplex.

Terminology

A *basis* B is an ordered set of m linearly independent columns. By $\mathbf{A}_B$ we denote the matrix consisting of columns in B, and by $\mathbf{c}_B$ we denote the vector of coefficients restricted to the elements of $\mathbf{c}$ that correspond to the indices of columns in B. We write $\mathbf{A}_N$ to denote the matrix consisting of non-basic columns, and, similarly, $\mathbf{c}_N$ to denote the vector of non-basic coefficients.

A *bfs* (basic feasible solution) is a vector $\mathbf{x} = (\mathbf{x}_B, \mathbf{x}_N)$, where $\mathbf{x_N}$ is a zero vector and $\mathbf{x}_B$ is a solution of the system $\mathbf{A}_B\mathbf{x}_B = \mathbf{b}$ that satisfies $\mathbf{x}_B \geq 0$. Such a vector corresponds to a vertex of polytope defined by the set of LP constraints.

A bfs is *degenerate* if $\mathbf{x}_B$ contains zero. This means that the vertex corresponding to this bfs also can be obtained by taking another bfs (i.e., the same vertex is obtainable as an intersection of different choices of polytope facets).

Phase I

The purpose of the Phase I is to determine an initial *bfs*. There are two cases:

1. Initially, all constraints were inequalities of the form (5), with $\mathbf{b}^L \geq 0$. Then we have introduced m slack variables when transforming them into equality constraints. Such slack variables also define a basis, which clearly is feasible. We proceed directly to Phase II from this point.

2. No initial basis is known, but we can add m artificial variables that would form a basis.

Assuming that the second case occurs, we add to each constraint an artificial variable x_i^a, $i = 1, \ldots, m$, and solve an auxiliary LP with an objective function:

$$\sum_{i=1}^{m} x_i^a,$$

and the set of constraints:

$$\mathbf{A}\mathbf{x} + \mathbf{I}\mathbf{x}^a = \mathbf{b}, \quad \mathbf{x}, \mathbf{x}^a \geq 0.$$

The simplex method itself can be used for solving this LP with initial bfs $\mathbf{x}_B^a = \mathbf{b}$.

If the auxiliary LP has optimal value greater than zero, then the original LP (1)–(3) is infeasible, and we stop at this point. If the original LP is feasible, then the auxiliary LP has optimal value 0. The basis corresponding to the solution of auxiliary LP should now consist entirely of non-artificial variables, and can be used as an initial basis in Phase II. If otherwise some artificial variable remains in the basis corresponding to the optimal solution of value 0, then this variable can be moved out from the basis, and replaced by a variable corresponding to any nonzero element in the row corresponding to the artificial variable [TODO: this is not implemented yet.]. If there are no nonzero elements in that row, then the original matrix $\mathbf{A}$ does not have a full rank, and the redundant row must be removed.

Phase II

In this phase we solve the original LP (1)–(3), given an initial bfs, obtained in Phase I.

The Algorithm

1. Given a basis B, construct m-by-m matrix $\mathbf{A}_B$.
2. Compute $\mathbf{x} = \mathbf{A}_B^{-1}\mathbf{b}$.
3. Compute $\mathbf{y} = (\mathbf{A}_B^T)^{-1}\mathbf{c}_B$.
4. Compute $\mathbf{s} = \mathbf{c}_N - \mathbf{A}_N^T\mathbf{y}$.
5. If $\mathbf{s} \geq 0$ then STOP. Return optimal value $(\mathbf{x}_B, \mathbf{x}_N = \mathbf{0})$.
6. Select entering index j, such that $\mathbf{s}_j < 0$ and j is the smallest.
7. Compute $\mathbf{d} = \mathbf{A}_B^{-1}\mathbf{A}_j$.
8. If $\mathbf{d} \leq 0$ then STOP. Problem is unbounded.
9. Select leaving index i, to be the smallest one in the set $\min\{\frac{x_i}{d_i},\ d_i > 0\}$.
10. Let $B \leftarrow B \setminus \{i\} \cup \{j\}$. Go to step 1.

The rules for selecting entering and leaving indices are called *Bland's rules*.