

Outline of the (Revised) Simplex Algorithm.

A standard form LP is:

$$\min \mathbf{c}^T \mathbf{x} \quad (1)$$

$$\text{st. } \mathbf{Ax} = \mathbf{b}, \quad (2)$$

$$\mathbf{x} \geq 0. \quad (3)$$

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:

$$\mathbf{A}^E \mathbf{x} = \mathbf{b}^E, \quad (4)$$

$$\mathbf{A}^L \mathbf{x} \leq \mathbf{b}^L, \quad (5)$$

$$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}. \quad (6)$$

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 , $i = 1, \dots, m$, 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 tableaux row corresponding to the artificial variable. 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*.

Simplex method for problems with bounded variables

Let us assume that variables in the problem are bounded, i.e.,

$$\forall_{j=1,\dots,n} \quad L_j \leq x_j \leq U_j.$$

By substituting $x'_j = x_j - L_j$ we can, without the loss of generality, consider the following formulation:

$$\min \mathbf{c}^T \mathbf{x} \tag{7}$$

$$\text{st. } \mathbf{Ax} = \mathbf{b}, \tag{8}$$

$$0 \leq \mathbf{x} \leq \mathbf{u}. \tag{9}$$

where $\mathbf{u} = [U_1 - L_1, U_2 - L_2, \dots, U_n - L_n]^T$, and we use \mathbf{x} instead of \mathbf{x}' for brevity (note that we also omit the constant term from the objective function, as the optimal solution remains the same).

Instead of adding slack variables to singleton upper-bound constraints and appending them to the matrix \mathbf{A} , we consider this special type of constraints separately. We use the notion of *extended basic feasible solution*. In such a solution, all basic variables assume values

$$0 < x_j < U_j,$$

while all nonbasic variables assume

$$x_j = 0 \quad \text{or} \quad x_j = U_j.$$

Theorem. If every nonbasic variable at its lower bound has a nonnegative objective coefficient, and every nonbasic variable at its upper bound has a nonpositive objective coefficient, then the extended basic feasible solution minimizes the objective function over the feasible region. \square

If the objective coefficient \hat{c}_j of nonbasic variable $x_j = 0$ is negative, we may increase x_j . If the objective coefficient \hat{c}_j of nonbasic variable $x_j = U_j$ is positive, we may decrease x_j . In either case, the value of solution is improving.

To simplify the computations, whenever a nonbasic variable assumes $x_j = U_j$, we apply the *upper-bound substitution*:

$$x_j = U_j - x'_j.$$

This substitution affects the objective function coefficient c_j and a_{ij} and b_i of each constraint that involves x_j . After applying it, all nonbasic variables have $x_j = 0$ or $x'_j = 0$. Thus we may use the same condition for selecting entering variable x_s as in the original simplex method:

Select entering index j , such that $\mathbf{s}_j < 0$ and j is the smallest.

To determine the leaving variable we need to take 3 cases into consideration: when increasing entering variable x_s from 0 either x_s reaches its upper bound U_s , some basic variable x_k decreases to 0, or some basic variable x_k increases to its upper bound U_k . Thus we compute the following:

$$t_1 = \min_i \left\{ \frac{x_i}{d_i} : d_i > 0 \right\},$$

$$t_2 = \min_i \left\{ \frac{U_k - x_i}{-d_i} : d_i < 0 \right\},$$

$$\theta = \min \{U_s, t_1, t_2\}.$$

- i) If $\theta = \infty$ then problem is unbounded.
- ii) If $\theta = U_s$ then we apply the upper-bounding substitution to entering variable x_s (basis remains unchanged).
- iii) If $\theta = t_1$ then we perform usual simplex pivot to introduce x_s into basis.
- iv) If $\theta = t_2$ then we first apply the upper-bounding substitution to basic variable x_k (corresponding to the minimum in t_2), and then perform simplex pivot to introduce x_s into basis.

When optimal solution is found (no increase in variable can lead to improvement of objective function) we must reverse all upper-bounding substitutions to retrieve the actual solution.