ISyE 6669-OAN
The Simplex Method

Andy Sun

In Modules 15 and 16, we study one of the most celebrated algorithms in optimization, the simplex method for solving linear optimization problems. The goal is to understand both the general *algorithmic ideas* and the *inner workings* of the simplex method. We first overview the procedure of finding basic solutions discussed in these modules and see why it works.

We always work with the following standard form linear program:

$$\begin{aligned} \min \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

1 The procedure to find basic solutions and why it works

Procedure for constructing basic solutions for the above standard form LP:

1. Choose m linearly independent columns $\boldsymbol{A}_{B(1)}, \ldots, \boldsymbol{A}_{B(m)}$ from $\boldsymbol{A}$ and form the matrix $\boldsymbol{B} = [\boldsymbol{A}_{B(1)}, \ldots, \boldsymbol{A}_{B(m)}]$. Denote the rest of $\boldsymbol{A}$ as matrix $\boldsymbol{N}$.

2. The basic solution is $\boldsymbol{x} = [\boldsymbol{x}_B, \boldsymbol{x}_N]$, where the basic variables are $\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b}$ and the nonbasic variables are $\boldsymbol{x}_N = \boldsymbol{0}$.

If the basic solution $\boldsymbol{x} = [\boldsymbol{x}_B, \boldsymbol{x}_N]$ thus obtained is also feasible, which means $\boldsymbol{x} \geq \boldsymbol{0}$ and in particular $\boldsymbol{x}_B \geq \boldsymbol{0}$, then $\boldsymbol{x}$ is a BFS, an extreme point. This is an algebraic procedure to find basic solutions, but why it works? Why the solution thus found $\boldsymbol{x} = [\boldsymbol{x}_B, \boldsymbol{x}_N]$ is a basic solution for the LP?

The definition of a basic solution is that there are n linearly independent active constraints at this solution, and all the equality constraints must be satisfied. So why is it true that there are n linearly independent active constraints at the solution found by the above procedure? The reasoning is summarized below.

- First, it is easy to realize that the matrix $\boldsymbol{A}$ of m rows and n columns is most likely a "short" and "wide" matrix, in other words, $m \leq n$. Why? Because, if $m > n$, then there are more linear constraints than the number of dimensions, which results in an overdetermined system. Without loss of generality, we always assume the rows of $\boldsymbol{A}$ are linearly independent, because we can always remove equations that are linear independent. In this case, if $m = n$, the equalities $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ has a unique solution, and the feasible region of the LP has one or no solution. If $m < n$, the equalities define a non-empty affine subspace (i.e. a linear subspace moved away from the origin), which is the more interesting case.

- Second, by definition, all the equality constraints in the LP must be satisfied at the basic solution $\boldsymbol{x}$. This means that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ are all active constraints. So this gives m active constraints. Since $m < n$, we still need another $n - m$ active constraints, which, together with the existing m equality constraints, form n linear independent constraints. Where do these $n - m$ active constraints come from? They can only come from the nonnegativity constraints $\boldsymbol{x} \geq \boldsymbol{0}$. Such a constraint takes the form $x_i = 0$ for some i. We can also write this linear constraint as

$$\underset{\dots,i^{th},\dots}{[0, \dots, 0, 1, 0, \dots, 0]} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = 0.$$

- Third, let's put all n active constraints together:

$$\left[\begin{array}{ccc|cccc} & \boldsymbol{B} & & & \boldsymbol{N} & & \\ \hline 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 1 \end{array}\right] \left[\begin{array}{c} \boldsymbol{x}_B \\ \hline \boldsymbol{x}_N \end{array}\right] = \left[\begin{array}{c} \boldsymbol{b} \\ \hline \boldsymbol{0} \end{array}\right]$$

Table 1: n linearly independent active constraints at $\boldsymbol{x} = [\boldsymbol{x}_B, \boldsymbol{x}_N]$.

The left-hand side matrix denoted as $\tilde{\boldsymbol{A}}$ can be written as a block matrix:

$$\tilde{\boldsymbol{A}} := \begin{bmatrix} \boldsymbol{B} & \boldsymbol{N} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{1}$$

Note that $\boldsymbol{B}$ is an $m \times m$ invertible matrix, $\boldsymbol{N}$ is $m \times (n-m)$, the zero matrix in the left bottom corner is $(n-m) \times m$, and the identity matrix at the right bottom corner is $(n-m) \times (n-m)$.

Proposition 1. *Assume $\boldsymbol{B}$ is invertible and $\tilde{\boldsymbol{A}}$ is defined in (1). Then, $\tilde{\boldsymbol{A}}$ is invertible.*

Proof. To show $\tilde{\boldsymbol{A}}$ is invertible, Let us look at the following equation:

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{N} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_B \\ \boldsymbol{x}_N \end{bmatrix} = \boldsymbol{0},$$

which clearly implies $\boldsymbol{x}_N = \boldsymbol{0}$. Therefore, we have $\boldsymbol{B}\boldsymbol{x}_B = \boldsymbol{0}$. Since $\boldsymbol{B}$ is invertible, the only solution is $\boldsymbol{x}_B = \boldsymbol{0}$. Therefore, $\tilde{\boldsymbol{A}}$ has linearly independent columns, thus is invertible. □

The matrix $\tilde{\boldsymbol{A}}$ is invertible means all the active constraints in $\tilde{\boldsymbol{A}}$ are linearly independent! This shows why $\boldsymbol{x} = [\boldsymbol{x}_B, \boldsymbol{x}_N]$ thus found is a basic solution. The above picture is very useful. Keep it in mind. We will use it again later.

2 The simplex method

2.1 Algorithmic idea: Local search

Local search is an iterative algorithm, i.e. an algorithm that goes through iterations, where each iteration t can be described on a high level as follows:

Algorithm 1 Local search framework.

1: Start from a feasible solution $\boldsymbol{x}_t$.

2: Find a "good" direction $\boldsymbol{d}$ that (a) points inside the feasible region and (b) decreases the objective value.

3: Find a "good" step length θ along $\boldsymbol{d}$ to move to next iterate: $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t + \theta \boldsymbol{d}$.

4: If no good direction or step length can be found, terminate. Otherwise $t \leftarrow t+1$ and go to step 1.

Local search is a very general algorithmic framework. Many algorithms fall into this framework. For instance, the famous *gradient descent algorithm*, which minimizes a differentiable function without any constraint. The gradient descent algorithm moves along the negative gradient direction of each iterate. If the algorithm terminates, it will find a point with zero gradient. This point can be a *local minimum*, i.e. there is no other better solution in a neighborhood of the one found by the algorithm. Because at each iteration, the algorithm only looks locally around a point, and it is likely that the algorithm can be trapped in a local minimum and never gets out to the true *global minimum*. Finding a global minimum can be very difficult for general optimization problems.

The only class of optimization problems for which local search can guarantee to find a global minimum is the class of convex programs, i.e. minimization problems with a convex objective function and a convex feasible region, because for this class of problems, a local minimum is a global minimum. Let us state this theorem with a simple proof.

Theorem 1. *A locally minimum of a convex optimization problem is also a global minimum.*

Proof. Let $\boldsymbol{x}^*$ be a local minimum of $\min_{\boldsymbol{x}\in X} f(\boldsymbol{x})$, where $f(\boldsymbol{x})$ is a convex function and X is a convex set. Then by definition, there exists a neighborhood of $\boldsymbol{x}^*$ in X, denoted as $U_\epsilon(\boldsymbol{x}^*) = \{\boldsymbol{x} : \|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \epsilon\}$ for some $\epsilon > 0$, where $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in U_\epsilon(\boldsymbol{x}^*)$. Now suppose there is a different point $\hat{\boldsymbol{x}} \in X$ such that $f(\hat{\boldsymbol{x}}) < f(\boldsymbol{x}^*)$. Since X is a convex set, for any $0 < \lambda < 1$, the point $\boldsymbol{x} = \lambda\hat{\boldsymbol{x}} + (1-\lambda)\boldsymbol{x}^*$ is in X. We can choose λ to be very close to zero so that $\boldsymbol{x}$ is also in the neighborhood $U_\epsilon(\boldsymbol{x}^*)$. Since f is convex, we have $f(\boldsymbol{x}) \leq \lambda f(\hat{\boldsymbol{x}}) + (1-\lambda)f(\boldsymbol{x}^*) < \lambda f(\boldsymbol{x}^*) + (1-\lambda)f(\boldsymbol{x}^*) = f(\boldsymbol{x}^*)$, i.e. $\boldsymbol{x}$ is a point in $U_\epsilon(\boldsymbol{x}^*)$ with a strictly lower objective value than $\boldsymbol{x}^*$, which contradicts with $\boldsymbol{x}^*$ being a local minimum in $U_\epsilon(\boldsymbol{x}^*)$. □

2.2 Simplex method as a local search

Remember LP is a special class of convex programs, therefore, we can expect that a good local search algorithm should be able to solve LP. Of course, how to design a functioning local search algorithm for LP is a highly nontrivial matter. It requires the work of a genius (more on this later). The simplex method as a local search algorithm is outlined below. The aspects that are specific to the simplex methods are put in italic.

Algorithm 2 The simplex method framework.

1: Start from a *basic feasible solution* $\boldsymbol{x}_t$.
2: Find a direction $\boldsymbol{d}$ that (a) points to *an adjacent BFS* and (b) decreases the objective value.
3: Find a step length θ so that *the next iterate,* $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t + \theta \boldsymbol{d}$*, is a BFS.*
4: If no such direction or step length can be found, terminate. O.w. $t \leftarrow t+1$ and go to step 1.

An important observation is that the simplex method goes from one BFS to another BFS. Why should we only care about BFS? This is due to an important property of LP. Namely, *if an LP has a BFS (which is true for any standard form LP) and if the optimal objective value is bounded, then a BFS is an optimal solution.*

2.3 Development of the simplex method

To implement Algorithm 2, we need to clarify the concept of "an adjacent BFS" and also specify algebraically how to carry out steps 2, 3, and 4. From now on in this notes, all the LPs are in standard form.

$$\begin{aligned} \min \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ and the rows of the matrix $\boldsymbol{A}$ are linearly independent.

The simplex method is developed step by step below.

1. *Start from a BFS*: Recall that, algebraically speaking, a basic feasible solution is a feasible solution where all equality constraints $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ are satisfied and out of all the active constraints, n of them are linearly independent. A basic feasible solution $\boldsymbol{x}$ can be partitioned into two parts: $\boldsymbol{x} = [\boldsymbol{x}_B, \boldsymbol{x}_N]$, the basic variable part $\boldsymbol{x}_B$ and the nonbasic variable part $\boldsymbol{x}_N$. The nonbasic variable is always zero, i.e., $\boldsymbol{x}_N = \boldsymbol{0}$. The basic variable $\boldsymbol{x}_B = [x_{B(1)}, \ldots, x_{B(m)}]$ is associated with the basis $\boldsymbol{B} = [\boldsymbol{A}_{B(1)}, \ldots, \boldsymbol{A}_{B(m)}]$. In particular, we have

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x} = [\boldsymbol{B}, \boldsymbol{N}] \begin{pmatrix} \boldsymbol{x}_B \\ \boldsymbol{x}_N \end{pmatrix} = \boldsymbol{B}\boldsymbol{x}_B + \boldsymbol{N}\boldsymbol{x}_N = \boldsymbol{B}\boldsymbol{x}_B \quad \Rightarrow \quad \boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b}.$$

2. *Find a feasible direction*: Starting from a basic feasible solution $\boldsymbol{x}$, the simplex method considers a feasible direction $\boldsymbol{d}$ to move away from the BFS $\boldsymbol{x}$ to $\hat{\boldsymbol{x}} := \boldsymbol{x} + \theta \boldsymbol{d}$. The new point $\boldsymbol{x} + \theta \boldsymbol{d}$ needs to be (a) a feasible point and (b) an adjacent BFS. Let us look at these two requirements.

   (a) Maintain feasibility: For $\hat{\boldsymbol{x}}$ to be feasible, we need $\boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{b}$, which means $\boldsymbol{A}(\boldsymbol{x} + \theta \boldsymbol{d}) = \boldsymbol{b}$. Since we already have $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ and $\theta > 0$, we should have $\boldsymbol{A}\boldsymbol{d} = \boldsymbol{0}$. Let us break down $\boldsymbol{A}$ into $[\boldsymbol{B}, \boldsymbol{N}]$ and $\boldsymbol{d}$ correspondingly to $\boldsymbol{d}_B$ and $\boldsymbol{d}_N$. Then, we have

$$\boldsymbol{A}\boldsymbol{d} = [\boldsymbol{B}, \boldsymbol{N}] \begin{pmatrix} \boldsymbol{d}_B \\ \boldsymbol{d}_N \end{pmatrix} = \boldsymbol{B}\boldsymbol{d}_B + \boldsymbol{N}\boldsymbol{d}_N = \boldsymbol{0}. \tag{2}$$

   We need to figure out how to find $\boldsymbol{d}_B$ and $\boldsymbol{d}_N$. For this, we need the definition of an adjacent BFS.

(b) Adjacent BFS: The new iterate $\hat{\boldsymbol{x}}$ is a BFS adjacent to $\boldsymbol{x}$. Mathematically, an adjacent BFS is defined below.

Definition 1 (Adjacent BFS)**.** *Two basic feasible solutions $\hat{\boldsymbol{x}}$ and $\boldsymbol{x}$ of a polyhedron P are called* adjacent*, if they share the same $n-1$ linearly independent active constraints.*

Geometrically, the solutions of $n-1$ linearly independent equations in n variables form a line, i.e. a 1-dimensional linear structure. So, two adjacent BFS are connected by a line segment on this line. This line segment is called an *edge* of the polyhedron. That is, two adjacent BFS' are connected by an edge, which is consistent with our intuition of what "adjacency" means. If you want to get a picture, flip to Section 4, Figure 1, where BFS' $(0, 10)$ and $(6, 16)$ are connected by an edge and therefore are adjacent. They share 1 common active constraints in $\mathbb{R}^2$. The concept of adjacent BFS' is applicable to any polyhedron. Now, we specialize it to a standard form LP.

Proposition 2. *In a standard form LP, two BFS' $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are adjacent if they have the same $n-m-1$ nonbasic variables.*

Proof. First, each BFS has $n-m$ nonbasic variables, which are all zero. Since $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ share $n-1$ active constraints and both have to satisfy all the m equality constraints $\boldsymbol{Ax} = \boldsymbol{b}$, then they can only differ in one nonbasic variable. In other words, they have the same $n-m-1$ nonbasic variables. □

For example, if the BFS $\boldsymbol{x} \in \mathbb{R}^5$ has two basic variables x_1, x_2 and three nonbasic variables x_3, x_4, x_5, then its adjacent BFS $\hat{\boldsymbol{x}}$ should share two same nonbasic variables x_3, x_4 or x_3, x_5 or x_4, x_5, plus one different nonbasic variable x_1 or x_2. In other words, going from $\boldsymbol{x}$ to $\hat{\boldsymbol{x}}$, two nonbasic variables of $\boldsymbol{x}$ stay as nonbasic variables of $\hat{\boldsymbol{x}}$ and only one of the nonbasic variables of $\boldsymbol{x}$ becomes a basic variable of $\hat{\boldsymbol{x}}$.

Proposition 2 gives us a way to specify $\boldsymbol{d}_N \in \mathbb{R}^{n-m}$: *Because $n-m-1$ nonbasic variables of $\boldsymbol{x}$ need to remain nonbasic, i.e. at zero value, $\boldsymbol{d}_N$ must have $n-m-1$ components at zero value; and because one nonbasic variable of $\boldsymbol{x}$ needs to become basic, i.e. to increase from zero value to some positive value, then the corresponding component of $\boldsymbol{d}_N$ has to be a positive number, and without loss of generality, we can fix this component at value* 1 *and use the step length θ to control its size.*

In terms of the Step 2 in the simplex method (Algorithm 2), we can do the following:

- Select a nonbasic variable x_j (remember initially $x_j = 0$), increase x_j to $\theta \geq 0$, while keeping other nonbasic variables at zero.

In other words, the simplex method makes $\boldsymbol{d}_N = [0, \ldots, 0, 1, 0, \ldots, 0]^\top := \boldsymbol{e}_j^\top$ for some $d_j = 1$, where $\boldsymbol{e}_j$ is the j-th unit vector. It seems that only the nonbasic direction is determined by this choice. However, the cool thing about starting from a BFS is that the basic direction part $\boldsymbol{d}_B$ is also uniquely determined simultaneously. To see this, From (2), it follows that

$$\boldsymbol{B}\boldsymbol{d}_B + \boldsymbol{N}\boldsymbol{d}_N = \boldsymbol{B}\boldsymbol{d}_B + \boldsymbol{A}_j = \boldsymbol{0},$$

where $\boldsymbol{A}_j$ is the column in $\boldsymbol{N}$ associated with the 1 component in $\boldsymbol{d}_N$, which is of course a column of the matrix $\boldsymbol{A}$. Since $\boldsymbol{B}$ is invertible, we have

$$\boldsymbol{d}_B = -\boldsymbol{B}^{-1}\boldsymbol{A}_j. \tag{3}$$

Put together, we have $\boldsymbol{d} = [-\boldsymbol{B}^{-1}\boldsymbol{A}_j, \boldsymbol{e}_j]$. We refer to this direction as the **j-th basic direction**.

3. *Measure the cost change*: How much can we change the objective cost by moving along the j-th basic direction? The change of cost can be calculated as:

$$\boldsymbol{c}^\top(\boldsymbol{x} + \theta\boldsymbol{d}) - \boldsymbol{c}^\top\boldsymbol{x} = \theta\boldsymbol{c}^\top\boldsymbol{d} = \theta\left(\boldsymbol{c}_B^\top\boldsymbol{d}_B + \boldsymbol{c}_N^\top\boldsymbol{d}_N\right) = \theta\left(-\boldsymbol{c}_B^\top\boldsymbol{B}^{-1}\boldsymbol{A}_j + c_j\right) := \theta\bar{c}_j.$$

Here we define a very important quantity $\bar{c}_j$:

Definition 2. *Let $\boldsymbol{x}$ be a basic solution, let $\boldsymbol{B}$ be an associated basis matrix, and let $\boldsymbol{c}_B$ be the vector of costs of the basic variables. For each j, we define the **reduced cost** $\bar{c}_j$ of the variable x_j to be $\bar{c}_j = c_j - \boldsymbol{c}_B^\top\boldsymbol{B}^{-1}\boldsymbol{A}_j$.*

Intuitively, $\bar{c}_j$ is the unit change of cost when we move along the j-th basic direction. Clearly, we would select the j-th basic direction only if $\bar{c}_j < 0$, i.e. the cost can be reduced by going this direction (remember we are minimizing the cost). This is the *criterion* for selecting a feasible direction from $n - m$ possible feasible directions.

However, if we find $\bar{c}_j \geq 0$ for every index j, it is not benefitial to move along any basic direction, which implies the current BFS $\boldsymbol{x}$ is locally optimal, thus globally optimal. Conversely, if $\boldsymbol{x}$ is optimal and nondegenerate, then $\bar{c}_j \geq 0$ for every j. This in fact gives the optimality conditions for linear optimization:

Theorem 2 (Optimality Conditions)**.** *Consider a basic feasible solution $\boldsymbol{x}$ associated with a basis matrix $\boldsymbol{B}$, and let $\bar{\boldsymbol{x}}$ be the corresponding vector of reduced costs.*

(a) If $\bar{\mathbf{c}} \geq \mathbf{0}$, then $\boldsymbol{x}$ is optimal.

(b) If $\boldsymbol{x}$ is optimal and nondegenerate, then $\bar{\mathbf{c}} \geq \mathbf{0}$.

4. *Find the step length*: Once a basic direction is selected, i.e. $\bar{c}_j < 0$ for some $j \in N$, we want to go along this direction as far as possible. In this way, the cost can be reduced by the maximum amount. We want to move until some basic variable $x_{B(l)}$ becomes zero at $\boldsymbol{x} + \theta^*\boldsymbol{d}$, where

$$\theta^* := \max\{\theta \geq 0 \mid \boldsymbol{x} + \theta\boldsymbol{d} \in P\}.$$

Let us summarize for now. We start from a BFS, select a basic feasible direction that reduces the cost, and we go along this direction as far as we can while still remaining feasible. Now the question is, Does this lead us to a position that we can repeat the procedure?

5. *Arrive at an adjacent BFS*: Indeed, the above procedure leads us to a new BFS. From there, we can start the procedure again. Let us see what the new BFS is. Remember after moving by $\theta^*\boldsymbol{d}$, we have $x_{B(l)} = 0$, i.e. $x_{B(l)}$ becomes a nonbasic variable. At the same time, $x_j = \theta > 0$

becomes a basic variable. We say that x_j enters the basis and $x_{B(l)}$ leaves the basis. If we look at the initial basis matrix $\boldsymbol{B}$ and the new basis matrix $\overline{\boldsymbol{B}}$, they only differ in one column:

$$\begin{aligned}\boldsymbol{B} &= \left[\boldsymbol{A}_{B(1)}, \ldots, \boldsymbol{A}_{B(l-1)}, \boldsymbol{A}_{B(l)}, \boldsymbol{A}_{B(l+1)}, \ldots, \boldsymbol{A}_{B(m)}\right] \in \mathbb{R}^{m\times m} \\ \overline{\boldsymbol{B}} &= \left[\boldsymbol{A}_{B(1)}, \ldots, \boldsymbol{A}_{B(l-1)}, \boldsymbol{A}_{j}, \boldsymbol{A}_{B(l+1)}, \ldots, \boldsymbol{A}_{B(m)}\right] \in \mathbb{R}^{m\times m}\end{aligned}$$

Theorem 3.

(a) $\overline{\boldsymbol{B}}$ *is a basis matrix, that is, columns* $\boldsymbol{A}_{B(i)}, i \neq l$ *and* $\boldsymbol{A}_j$ *are linearly independent.*

(b) The vector $\boldsymbol{y} = \boldsymbol{x} + \theta^* \boldsymbol{d}$ *is a BFS associated with the basis matrix* $\overline{\boldsymbol{B}}$*.*

Putting everything together, we complete an iteration of the simplex method.

1. We start the simplex method with a basic feasible solution $\boldsymbol{x}$ and the associated basis consisting of the basis matrix $\boldsymbol{B} = [\boldsymbol{A}_{B(1)}, \ldots, \boldsymbol{A}_{B(m)}]$, where $\boldsymbol{A}_{B(i)}$'s are the basic columns of $\boldsymbol{A}$.
2. Compute the reduced cost $\bar{c}_j = c_j - \boldsymbol{c}_B^\top \boldsymbol{B}^{-1}\boldsymbol{A}_j$ for each nonbasic variable x_j. There are two possibilities:
   (2.1) If all the reduced costs are nonnegative, the current BFS is optimal, and the algorithm terminates;
   (2.2) Otherwise, select some j with $\bar{c}_j < 0$, so x_j *enters the basis.*
3. Compute $\boldsymbol{d}_B = -\boldsymbol{B}^{-1}\boldsymbol{A}_j$. There are two possibilities:
   (3.1) If $\boldsymbol{d}_B \geq \boldsymbol{0}$, we know the optimal cost is $-\infty$, and the algorithm terminates.
   (3.2) Otherwise, if some entry of $\boldsymbol{d}_B$ is negative, continue to the next step.
4. If some entry of $\boldsymbol{d}_B$ is negative, compute the stepsize θ^* by the following *min-ratio test*:

$$\theta^* = \min_{\{i=1,\ldots,m \mid d_{B(i)}<0\}} \frac{x_{B(i)}}{-d_{B(i)}}.$$

   Suppose the index $B(l)$ achieves the minimum: $x_{B(l)}$ *exits the basis.*
5. Form the new basis matrix $\overline{\boldsymbol{B}}$ by replacing $\boldsymbol{A}_{B(l)}$ column with $\boldsymbol{A}_j$. The new BFS $\boldsymbol{y}$ has basic variable part $y_{B(i)} = x_{B(i)} + \theta^* d_{B(i)}$ and $y_j = \theta^*$. Start a new iteration.

2.4 Correctness of the simplex method

The correctness of the above algorithm is stated in the following theorem. In essense, under a favorable condition, the above simplex method always terminates in a finite number of steps, with either an optimal solution, or a direction along which the optimal cost goes to $-\infty$.

Theorem 4 (Correctness of the Simplex Method)**.** *If every basic feasible solution of a standard form linear program is nondegenerate, then the simplex method always terminates in a finite number of steps, either*

1. *finds an optimal solution* $\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b}$ *and* $\boldsymbol{x}_N = \boldsymbol{0}$*, with the associated optimal basis matrix* $\boldsymbol{B}$*;*

2. *or finds a direction* $\boldsymbol{d} = (\boldsymbol{d}_B, \boldsymbol{d}_N)$ *such that* $\boldsymbol{A}\boldsymbol{d} = \boldsymbol{0}$*,* $\boldsymbol{d}_B \geq \boldsymbol{0}$*,* $\boldsymbol{d}_N = (0, \ldots, 0, 1, 0, \ldots, 0)^T$ *with* j*-th element being* 1*,* $\boldsymbol{c}^T\boldsymbol{d} < 0$*, and the optimal cost is* $-\infty$*.*

So the favorable condition is that every basic feasible solution is *nondegenerate*. We have talked about degeneracy before. The definition is the following:

Definition 3. *1. A basic feasible solution of a standard form LP is said to be* ***nondegenerate****, if every basic variable is positive, i.e.* $x_{B(i)} > 0$ *for all* $i = 1, \ldots, m$*.*

2. A basic feasible solution of a standard form LP is said to be ***degenerate****, if some basic variable* $x_{B(i)}$ *is zero.*

The key insight in Theorem 3 is that, when every BFS is non-degenerate (i.e. $\boldsymbol{x}_B > \boldsymbol{0}$), the algorithm moves by a positive amount $\theta^* > 0$ along a direction $\boldsymbol{d}$, and no BFS can be visited twice (why?). Since any polyhedron can only have a finite number of BFS, the algorithm must terminate after a finite number of iterations.

2.5 The simplex method for degenerate problems

As we carry out the simplex method, there can be multiple choices of entering variables and exiting variables. How to choose which variables to enter and exit is important for a degenerate problem. If not careful, the simplex method may run into cycles and never terminates. A simple rule that prevents cycling is the following one:

Bland's rule:

1. Among all the eligible choices of nonbasic variables to enter the basis, select the one with the smallest subscript.

2. Among all the eligible choices of basic variables to exit the basis, select the one with the smallest subscript.

For example, if in an iteration of the simplex method, we have basic variables $\boldsymbol{x}_B = (x_5 = 0, x_6 = 0, x_7 = 1)$ and the reduced costs of the nonbasic variables x_1, x_2, x_3, x_4 are $(-1/2, 20, -3/4, 6)$, respectively. Then, both x_1 and x_3 are candidates to enter the basis. According to the Bland's rule, choose x_1 to enter, which has smaller subscript than x_3. Suppose the feasible direction is $\boldsymbol{d}_B = -\boldsymbol{B}^{-1}\boldsymbol{A}_1 = \begin{bmatrix} -1/4 \\ -1/2 \\ 0 \end{bmatrix}$. The min-ratio test has $\theta^* = \min\{0/(1/4), 0/(1/2)\} = 0$, so both x_5 and x_6 can exit. Bland rule picks x_5 since it has a smaller subscript.

3 The Phase I/Phase II Simplex Method

So far we have assumed that we already have a basic feasible solution to start the simplex method. However, in reality, finding a basic feasible solution is not an easy task — in fact, it turns out to be as hard as solving a linear program. Do not confuse finding a basic feasible solution with finding a basic solution. To find a basic solution, we only need to find a basis matrix $\boldsymbol{B}$. However, the resulting basic solution may not be feasible (because $\boldsymbol{B}^{-1}\boldsymbol{b}$ may not be nonnegative), therefore, may not be a basic feasible solution.

In the following, we will develop a two-phase method, in the first phase (Phase-I), it finds an initial BFS for a standard form LP, or detects the LP is infeasible; in the second phase (Phase-II), it uses the initial BFS to start the simplex method.

1. For any standard form LP, we can always make the right-hand side vector $\boldsymbol{b}$ a nonnegative vector, i.e. $\boldsymbol{b} \geq \boldsymbol{0}$ by multiplying -1 to both sides of the i-th equality constraint if $b_i < 0$. For example, if we have $3x_1 - x_2 = -2$ in the constraints, we can always write it as $-3x_1 + x_2 = 2$. So we want to solve the standard form LP:

$$\begin{aligned} \min \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{aligned}$$

   where $\boldsymbol{b} \geq \boldsymbol{0}$.

2. Now construct the following auxiliary problem, which we call the Phase-I LP:

$$\begin{aligned} \text{(Phase-I LP)} \quad z^* = \min \quad & y_1 + y_2 + \cdots + y_m \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{I}\boldsymbol{y} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{y} \geq \boldsymbol{0}. \end{aligned}$$

   Here we introduce auxilliary variable $\boldsymbol{y} = (y_1, \ldots, y_m)$, similar to the case where we introduce slack variables (but here the purpose is not to transform a LP into standard form, but to find an initial BFS or detect infeasibility).

   For this problem, we can easily find an initial basic feasible solution, namely $(\boldsymbol{x} = \boldsymbol{0}, \boldsymbol{y} = \boldsymbol{b})$. Since the identity matrix $\boldsymbol{I}$ is invertible, this is a basic solution, and since $\boldsymbol{b} \geq \boldsymbol{0}$, this solution is feasible. Therefore, it is indeed a BFS for the Phase-I LP. So we can solve the Phase-I LP using the simplex method starting with the BFS $(\boldsymbol{x} = \boldsymbol{0}, \boldsymbol{y} = \boldsymbol{b})$.

   After we solve the Phase-I problem, the optimal cost z^* has two possibilities: $z^* = 0$ or $z^* > 0$.

   (a) If $z^* = 0$, then the optimal solution $y_1^*, \ldots, y_m^*$ must be all zero, because each of $y_i^* \geq 0$ and their sum is zero. In this case, the x-variable part $\boldsymbol{x}^* = (x_1^*, \ldots, x_n^*)$ is a feasible solution to the original problem, i.e. $\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{b}$ and $\boldsymbol{x}^* \geq \boldsymbol{0}$. Furthermore, either $\boldsymbol{x}^*$ already contains all the basic variables, or after some simple operations, we can obtain a basic feasible solution to the original problem.

   (b) If $z^* > 0$, then some $y_i^* > 0$. This in fact shows the original problem is infeasible, as proved in the following theorem.

   Theorem 5. *The original LP is feasible if and only if $z^* = 0$.*

   Proof. If the original LP is feasible, there exists some $\boldsymbol{x}^*$ such that $\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{b}$ and $\boldsymbol{x}^* \geq \boldsymbol{0}$. Then, $(\boldsymbol{x} = \boldsymbol{x}^*, \boldsymbol{y} = \boldsymbol{0})$ is also feasible for the Phase-I problem. The cost associated with this feasible solution is 0. But since we know $z^* \geq 0$, this feasible solution is actually optimal. That is, we have $z^* = 0$.

For the other direction, if $z^* = 0$, let the optimal solution of the Phase-I problem be $(\boldsymbol{x}^*, \boldsymbol{y}^*)$. Then $y_1^* = \cdots = y_m^* = 0$, therefore, $\boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{I}\boldsymbol{y}^* = \boldsymbol{b}$ which implies $\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{b}$. We also have $\boldsymbol{x}^* \geq \boldsymbol{0}$. So, the original problem is feasible. □

To summarize, we have the following **Phase-I/Phase-II simplex method** to solve any standard form LP:

1. Phase I:

   Solve the Phase-I LP. Denote the optimal cost as z^*. We will have two possibilities:

   (a) If $z^* > 0$, the original LP is infeasible. The algorithm terminates.

   (b) If $z^* = 0$, a feasible solution to the original LP is found, from which we can obtain a BFS for the original LP.

2. Phase II:

   Solve the original LP by the simplex method, starting with the BFS found in Phase I.

[See next page for examples]

4 Examples

4.1 Simplex Method in Detail

Consider the following linear program:

$$\begin{aligned} \max \quad & 2x_1 + 3x_2 \\ \text{s.t.} \quad & -x_1 + x_2 \leq 10 \\ & 3x_1 + 2x_2 \leq 60 \\ & 2x_1 + 3x_2 \leq 6 \\ & x_1, x_2 \geq 0. \end{aligned}$$

1. First let us draw the feasible region of this LP in $\mathbb{R}^2$ in Figure 1.

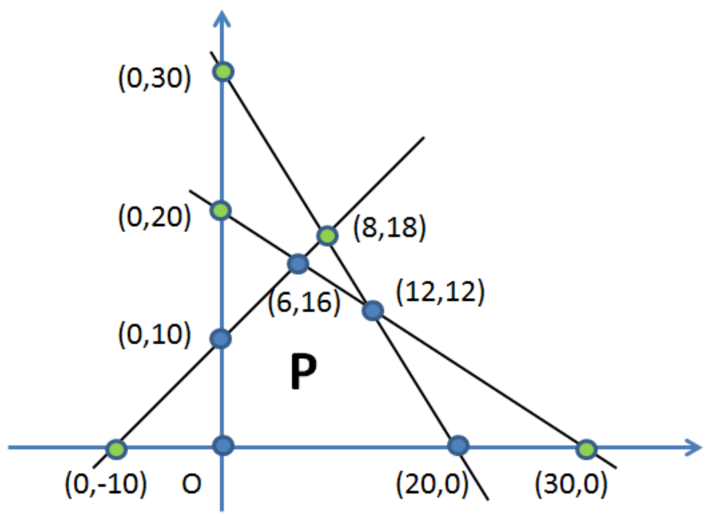


Figure 1: A simplex example.

The blue dots are basic feasible solutions. The green dots are basic solutions but not feasible. So in total, there are 10 basic solutions.

2. Transform to a standard form LP: The simplex method works on standard form LPs, so let us first transform the above LP into the standard form.

$$\begin{array}{lrrrrrcl} \min & -2x_1 & -3x_2 & & & & & \\ \text{s.t.} & -x_1 & +x_2 & +x_3 & & & = & 10 \\ & 3x_1 & +2x_2 & & +x_4 & & = & 60 \\ & 2x_1 & +3x_2 & & & +x_5 & = & 60 \\ & \multicolumn{5}{c}{x_1, x_2, x_3, x_4, x_5 \geq 0.} & & \end{array}$$

Remember a standard form LP has the following form:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T \boldsymbol{x} && \text{[Minimization]} \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} && \text{[Only equality constraints]} \\ & \boldsymbol{x} \geq \boldsymbol{0} && \text{[All variables nonnegative]} \end{aligned}$$

To facilitate the simplex method, it helps to write out explicitly the $\boldsymbol{c}, \boldsymbol{A}, \boldsymbol{b}$:

$$\boldsymbol{c} = \begin{bmatrix} -2 \\ -3 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{A} = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 \\ 3 & 2 & 0 & 1 & 0 \\ 2 & 3 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} 10 \\ 60 \\ 60 \end{bmatrix}$$

3. Start the simplex method:

   Iteration 1:

   (a) **Choose a starting BFS:** Let us select the basis $\boldsymbol{B} = [\boldsymbol{A}_3, \boldsymbol{A}_4, \boldsymbol{A}_5] = \left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right]$. The corresponding basic solution is

   $$\boldsymbol{x}_B = \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix} = \boldsymbol{B}^{-1}\boldsymbol{b} = \begin{bmatrix} 10 \\ 60 \\ 60 \end{bmatrix}, \boldsymbol{x}_N = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

   and the cost coefficients associated with basic and nonbasic variables:

   $$\boldsymbol{c}_B = \begin{bmatrix} c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{c}_N = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} -2 \\ -3 \end{bmatrix}.$$

   Since $\boldsymbol{x}_B \geq \boldsymbol{0}$ (and of course $\boldsymbol{x}_N \geq \boldsymbol{0}$), the current basic solution is a basic feasible solution. So we are ready to start the simplex method.

   (b) **Compute reduced costs for nonbasic variables:**

   $$\bar{c}_1 = c_1 - \boldsymbol{c}_B^T \boldsymbol{B}^{-1} \boldsymbol{A}_1 = -2$$
   $$\bar{c}_2 = c_2 - \boldsymbol{c}_B^T \boldsymbol{B}^{-1} \boldsymbol{A}_2 = -3$$

   Both $\bar{c}_1$ and $\bar{c}_2$ are negative. Therefore, the current BFS is not optimal, and both x_1 and x_2 are candidates to enter the basis, i.e. to increase to a positive value. Let us take x_2 to enter the basis, and keep x_1 zero.

   (c) **Compute feasible direction $\boldsymbol{d} = \left[\begin{smallmatrix} \boldsymbol{d}_B \\ \boldsymbol{d}_N \end{smallmatrix}\right]$:** Since we decide to increase x_2 and keep x_1 at zero, the nonbasic variable part of the feasible direction $\boldsymbol{d}_N$ is $\boldsymbol{d}_N = \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right]$, and the basic variable part of the feasible direction $\boldsymbol{d}_B$ is

   $$\boldsymbol{d}_B = -\boldsymbol{B}^{-1}\boldsymbol{A}_2 = \begin{bmatrix} -1 \\ -2 \\ -3 \end{bmatrix}.$$

   Since some components of $\boldsymbol{d}_B$ is negative, we do not have an unbounded optimal solution, and we need to decide how far to go along this direction while still remaining feasible.

(d) **Min-ratio test:** By going along the above calculated direction, we are going from the initial BFS $\boldsymbol{x}$ to a new point $\boldsymbol{x}+\theta\boldsymbol{d}$. Let us write it out componentwise:

$$\boldsymbol{x}+\theta\boldsymbol{d} = \begin{bmatrix} \boldsymbol{x}_B+\theta\boldsymbol{d}_B \\ \boldsymbol{x}_N+\theta\boldsymbol{d}_N \end{bmatrix} = \begin{bmatrix} x_{B(1)}+\theta d_{B(1)} \\ x_{B(2)}+\theta d_{B(2)} \\ x_{B(3)}+\theta d_{B(3)} \\ x_1+\theta\cdot 0 \\ x_2+\theta\cdot 1 \end{bmatrix} = \begin{bmatrix} x_3+\theta d_3 \\ x_4+\theta d_4 \\ x_5+\theta d_5 \\ x_1 \\ x_2+\theta \end{bmatrix} = \begin{bmatrix} 10+\theta\cdot(-1) \\ 60+\theta\cdot(-2) \\ 60+\theta\cdot(-3) \\ 0 \\ \theta \end{bmatrix} = \begin{bmatrix} 10-\theta \\ 60-2\theta \\ 60-3\theta \\ 0 \\ \theta \end{bmatrix}$$

To decide the largest θ so that $\boldsymbol{x}+\theta\boldsymbol{d} \geq \boldsymbol{0}$, we need to do the min-ratio test:

$$\theta^* = \min_{\{i=1,\dots,m|d_{B(i)}<0\}} \frac{x_{B(i)}}{-d_{B(i)}} = \min\{\frac{10}{1}, \frac{60}{2}, \frac{60}{3}\} = 10.$$

So $x_{B(1)} = x_3$ exits the basis.

(e) **The new basis:** The new basis $\bar{\boldsymbol{B}} = [\boldsymbol{A}_2, \boldsymbol{A}_4, \boldsymbol{A}_5]$, which differs from the original basis only in one column: $\boldsymbol{A}_3$ is replaced by $\boldsymbol{A}_2$ i.e. x_3 exits the basis and x_2 enters the basis. The new basic variables and nonbasic variables are

$$\boldsymbol{x}_{\bar{B}} = \begin{bmatrix} x_2 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 10 \\ 40 \\ 30 \end{bmatrix}, \ \boldsymbol{x}_{\bar{N}} = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

We are ready for a new iteration of the simplex method.

Iteration 2:

(a) Let us write the new basis and its inverse:

$$\boldsymbol{B} = [\boldsymbol{A}_2, \boldsymbol{A}_4, \boldsymbol{A}_5] = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix}, \ \boldsymbol{B}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix}.$$

The cost coefficients for basic and nonbasic variables:

$$\boldsymbol{c}_B = \begin{bmatrix} c_2 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix}, \ \boldsymbol{c}_N = \begin{bmatrix} c_1 \\ c_3 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \end{bmatrix}.$$

(b) **Compute reduced costs:**

$$\bar{c}_1 = c_1 - \boldsymbol{c}_B^T\boldsymbol{B}^{-1}\boldsymbol{A}_1 = -2 - \begin{bmatrix} -3 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 3 \\ 2 \end{bmatrix} = -2 - \begin{bmatrix} -3 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 3 \\ 2 \end{bmatrix} = -5$$

$$\bar{c}_3 = c_3 - \boldsymbol{c}_B^T\boldsymbol{B}^{-1}\boldsymbol{A}_3 = 0 - \begin{bmatrix} -3 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = 0 - \begin{bmatrix} -3 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = 3.$$

$\bar{c}_1 < 0$, so the current BFS is not optimal, and x_1 enters the basis.

(c) **Feasible direction:**

$$\boldsymbol{d}_N = \begin{bmatrix} d_1 \\ d_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \ \boldsymbol{d}_B = \begin{bmatrix} d_2 \\ d_4 \\ d_5 \end{bmatrix} = -\boldsymbol{B}^{-1}\boldsymbol{A}_1 = -\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ -5 \\ -5 \end{bmatrix}.$$

Since some components of $\boldsymbol{d}_B$ are negative, the optimal solution is not unbounded.

(d) **Min-ratio test:** Going along the direction calculated above, we move from the current BFS to a new point

$$\boldsymbol{x}_B + \theta \boldsymbol{d}_B = \begin{bmatrix} 10 \\ 40 \\ 30 \end{bmatrix} + \theta \begin{bmatrix} 1 \\ -5 \\ -5 \end{bmatrix}.$$

Do the min-ratio test to decide how far to move to keep $\boldsymbol{x}_B + \theta \boldsymbol{d}_B \geq \boldsymbol{0}$:

$$\theta^* = \min_{\{i=1,\ldots,m | d_{B(i)}<0\}} \frac{x_{B(i)}}{-d_{B(i)}} = \min\{\frac{40}{-(-5)}, \frac{30}{-(-5)}\} = 6.$$

$x_{B(3)} = x_5$ becomes zero, so x_5 exits the basis.

(e) **The new basis:** $\bar{\boldsymbol{B}} = [\boldsymbol{A}_2, \boldsymbol{A}_4, \boldsymbol{A}_1]$, since x_1 enters the basis and x_5 exits the basis. The new non-basis matrix $\overline{\boldsymbol{N}} = [\boldsymbol{A}_5, \boldsymbol{A}_3]$. The new BFS is

$$\boldsymbol{x}_B = \begin{bmatrix} x_2 \\ x_4 \\ x_1 \end{bmatrix} = \begin{bmatrix} 16 \\ 10 \\ 6 \end{bmatrix}, \ \boldsymbol{x}_N = \begin{bmatrix} x_5 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

To decide if this BFS is optimal, we need to start another iteration.

1. **Iteration 3:**

(a) Let us write the new basis and its inverse:

$$\boldsymbol{B} = [\boldsymbol{A}_2, \boldsymbol{A}_4, \boldsymbol{A}_1] = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 1 & 3 \\ 3 & 0 & 2 \end{bmatrix}, \ \boldsymbol{B}^{-1} = \begin{bmatrix} 0.4 & 0 & 0.2 \\ 1 & 1 & -1 \\ -0.6 & 0 & 0.2 \end{bmatrix}.$$

The cost coefficients for basic and nonbasic variables:

$$\boldsymbol{c}_B = \begin{bmatrix} c_2 \\ c_4 \\ c_1 \end{bmatrix} = \begin{bmatrix} -3 \\ 0 \\ -2 \end{bmatrix}, \ \boldsymbol{c}_N = \begin{bmatrix} c_5 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

(b) **Compute reduced costs:**

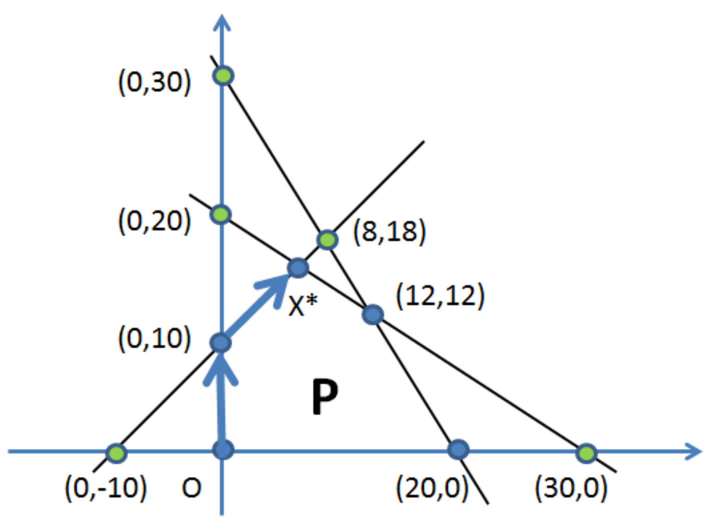
$$\bar{c}_5 = c_5 - \boldsymbol{c}_B^T \boldsymbol{B}^{-1} \boldsymbol{A}_5 = 0 - \begin{bmatrix} -3 & 0 & -2 \end{bmatrix} \begin{bmatrix} 0.4 & 0 & 0.2 \\ 1 & 1 & -1 \\ -0.6 & 0 & 0.2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0 - \begin{bmatrix} 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 1$$

$$\bar{c}_3 = c_3 - \boldsymbol{c}_B^T \boldsymbol{B}^{-1} \boldsymbol{A}_3 = 0 - \begin{bmatrix} -3 & 0 & -2 \end{bmatrix} \begin{bmatrix} 0.4 & 0 & 0.2 \\ 1 & 1 & -1 \\ -0.6 & 0 & 0.2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = 0 - \begin{bmatrix} 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = 0.$$

Since all the reduced costs are nonnegative, the current BFS is optimal. We are done! The final optimal solution is:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 6 \\ 16 \\ 0 \\ 10 \\ 0 \end{bmatrix}.$$

Now let us trace the trajectory of the above simplex iterations on the graph. We started at the initial BFS $(x_1, x_2, x_3, x_4, x_5) = (0, 0, 10, 60, 60)$, which corresponds to the origin $(x_1, x_2) = (0, 0)$ on the 2-D graph. After the first iteration, we moved to a new BFS $(x_1, x_2, x_3, x_4, x_5) = (0, 10, 0, 40, 30)$, which is the extreme point $(x_1, x_2) = (0, 10)$ on the x_2 axis. We decided this is not an optimal solution, so we did one more iteration of simplex. This time, we reached the BFS $(x_1, x_2, x_3, x_4, x_5) = (6, 16, 0, 10, 0)$, which corresponds to $(x_1, x_2) = (6, 16)$ on the graph. It is optimal. This trajectory is shown on the following graph. We can see, geometrically, the simplex method is traversing from one extreme point to another adjacent extreme point, while reducing the objective cost, until it reaches the optimal extreme point.



4.2 Two-Phase Simplex Method

Here is an example to form a Phase-I problem. The original LP is given as:

$$\begin{array}{llllll} \min & x_1 & +3x_2 & +2x_3 & & \\ \text{s.t.} & x_1 & +2x_2 & +x_3 & & = 3 \\ & -x_1 & +2x_2 & & -6x_4 & = 2 \\ & \multicolumn{5}{c}{x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0.} \end{array}$$

The Phase-I problem is formulated as:

$$
\begin{array}{llllllll}
\min & y_1 & +y_2 & & & & & \\
\text{s.t.} & x_1 & +2x_2 & +x_3 & & +y_1 & & =3 \\
 & -x_1 & +2x_2 & & -6x_4 & & +y_2 & =2 \\
 & \multicolumn{7}{l}{x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, y_1 \geq 0, y_2 \geq 0.}
\end{array}
$$

We can choose (y_1, y_2) to be the basic variables, which gives the basic feasible solution $(x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 0, y_1 = 3, y_2 = 2)$. Then, we can start the simplex method to solve the Phase-I problem. We also know, the Phase-I problem always has a finite optimal cost, because $\boldsymbol{y} \geq \mathbf{0}$ which implies $y_1 + \cdots + y_m \geq 0$, i.e. the optimal cost z^* can not be negative.

Solving the Phase-I problem with the simplex method, we get an optimal solution $\boldsymbol{x}^* = (0, 3/2, 0, 1/6)$ and $\boldsymbol{y}^* = (0, 0)$. The $\boldsymbol{x}$ variable part is a BFS for the original LP, with basic variables $(x_2 = 3/2, x_4 = 1/6)$, and nonbasic variables $(x_1 = 0, x_3 = 0)$. You can also go back to the original LP and verify this: The matrix $\boldsymbol{B} = [\boldsymbol{A}_2, \boldsymbol{A}_4]$ is indeed invertible, therefore a basis matrix. The corresponding basic variable is $\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b} = [3/2, 1/6]$, which is positive, therefore it is a BFS of the original LP. Now, we can start solving the original LP with this BFS.