Linear Programming III : Simplex Method

Asst. Prof. Dr.-Ing. Sudchai Boonto

Department of Control System and Instrumentation Engineering
King Mongkut's Unniversity of Technology Thonburi
Thailand

Objective

- Linear programming (LP) problems occur in a diverse range of real-life applications in economic analysis and planning, operations research, computer science, medicine, and engineering.
- These prolems, it is known that nay minima occur at the vertices of the feasible region and can be determined through a "brute-force" or exhaustive approach by evaluating the objective function at all the vertices of the feasible region.
- The number of variables involved in practical LP problem is often vary large and an exhaustive approach would entail a considerable amount of computation.
- In 1947, Dantzig developed a method for solving LP problems known as the *simplex method*. He solved this problem because he came to the class late and thought an unsolved problem on a blackboard was homework.
- Named one of the "Top 10 algorithms of the 20th century" by *Computing in Science & Engineering magazine.* Full list at: `https://www.siam.org/pdf/news/637.pdf`
- The simplex method has been the primary method for solving LP problems since its introduction.

Simplex Method for Standard Form

Consider an example of the standard LP problem:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) = x_1 - 2x_2 - x_4 \\ \text{subject to} \quad & 3x_1 + 4x_2 + x_3 \quad = 9 \\ & 2x_1 + x_2 + x_4 \quad = 6 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0 \end{aligned}$$

We have

$$A = \begin{bmatrix} 3 & 4 & 1 & 0 \\ 2 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 9 \\ 6 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad x \in \mathbb{R}^4, \quad p = 2$$

Simplex Method for Standard Form

The p equality constraints are always treated as active constraints denoted by $\tilde{A}x = \tilde{b}$. Assume B is a matrix that consists of p linearly independent column of $\tilde{A}$. The we have

$$\tilde{A}x = \tilde{b} \Longrightarrow \tilde{A}x = \begin{bmatrix} B & \big| & N \end{bmatrix} \begin{bmatrix} x_B \\ \hline x_N \end{bmatrix} = \left[\begin{array}{cc|cc} 3 & 4 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ \hline x_3 \\ x_4 \end{bmatrix} = Bx_B + Nx_N = \tilde{b}$$

- The variables contained in x_B and x_N are called basic and non basic variables, respectively.
- B is nonsingular, we can express the basic variables in terms of the nonbasic variables as

$$x_B = B^{-1}b - B^{-1}Nx_N$$

Simplex Method for Standard Form

- At vertex x_k, there is at least n active constraints. In addition to the p equality constraints, there are at least $n-p$ inequality constraints that become active in x_k.
- Therefore, for the standard-form LP problem a vertex contains at least $n-p$ zero components.

Theorem: Linear independence of columns in matrix $\tilde{A}$
The columns of $\tilde{A}$ corresponding to strictly positive of a vertex x_k are linearly independent.

Proof: Let $\hat{B}$ be formed by the columns of $\tilde{A}$ that correspond to strictly positive components of x_k $(x_k \geq 0)$, and let $\hat{x}_k$ be the collection of the positive components of x_k. If $\hat{B}\hat{w} = 0$ for some nonzero $\hat{w}$, then it follows that

$$\tilde{A}x_k = \hat{B}\hat{x}_k = \hat{B}(\hat{x} + \alpha\hat{w}) = b \text{ for any scalar } \alpha$$

Simplex Method for Standard Form

Since $\hat{x}_k > 0$, there exists a sufficiently small $\alpha_+ > 0$ such that

$$\hat{y}_k = \hat{x}_k + \alpha\hat{w} > 0 \text{ for } -\alpha_+ \leq \alpha \leq \alpha_+.$$

$y \in \mathbb{R}^{n\times 1}$ be such that the components of y_k corresponding to $\hat{x}_k$ are equal to the components of $\hat{y}_k$ and the remaining correspondents of y_k are zero. Note that with $\alpha = 0$, $y_k = x_k$ is a vertex, and when α varies from $-\alpha_+$ to α_+ , vertex x_k would lie between two feasible points on a straight line, which is a contradiction. Hence $\hat{w}$ must be zero and the columns of $\hat{B}$ are linearly independent.

Simplex Method for Standard Form

- Using above theorem, we can use the columns of $\hat{B}$ as a set of core basis vectors to construct a nonsingular square matrix B. If $\hat{B}$ already contains p columns, we assume that $B = \hat{B}$, otherwise, we augment $\hat{B}$ with additional columns of A to obtain a square nonsingular B.
- Let the index set associated with B at x_k be denoted as $\mathcal{I}_\beta = \{\beta_1, \beta_2, \ldots, \beta_p\}$. With matrix B so formed, matrix N can be constructed with those $n - p$ columns of $\tilde{A}$ that are not in B. Let $\mathcal{I}_N = \{v_1, v_2, \ldots, v_{n-p}\}$ be the index set for the columns of N and let I_N be the $(n - p) \times n$ matrix composed of rows $v_1, v_2, \ldots, v_{n-p}$ of the $n \times n$ identity matrix.
- It is clear that at vertex x_k the active constrain matrix A_{a_k} contains the working-set matrix

$$\hat{A}_{a_k} = \begin{bmatrix} \tilde{A} \\ I_N \end{bmatrix}$$

as an $n \times n$ submatrix.

Simplex Method for Standard Form

- It can be shown that matrix $\hat{A}_{a_k}$ is nonsingular. If $\hat{A}_{a_k}x = 0$ for some x, then we have

$$Bx_B + Nx_N = 0 \text{ and } x_N = 0 \quad \Longrightarrow \quad x_B = -B^{-1}Nx_N = 0$$
$$x = \begin{bmatrix} x_B & x_N \end{bmatrix}^T = 0.$$

Therefore, $\hat{A}_{a_k}$ is nonsingular. In summary, at a vertex x_k a working set of active constraints for the application of the simplex method can be obtained with three simple steps as follows:

1. Select the columns in matrix A that correspond to the strictly positive components of x_k to form matrix $\hat{B}$.
2. If the number of columns in $\hat{B}$ is equal to p, take $B = \hat{B}$; otherwise, $\hat{B}$ is augmented with additional columns of A to form a square nonsingular matrix B.
3. Determine the index set $\mathcal{I}_n$ and form matrix I_N.

Simplex Method for Standard Form

Example

Identify working sets of active constraints at vertex $x = [3\ 0\ 0\ 0]^T$ for the LP problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) = x_1 - 2x_2 - x4 \\ \text{subject to} \quad & 3x_1 + 4x_2 + x_3 \quad = 9 \\ & 2x_1 + x_2 + x_4 \quad = 6 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0 \end{aligned}$$

Solution Using $r = Ax - b$, we can verify that the point $x = [3\ 0\ 0\ 0]^T$ is a degenerate vertex at which there are five active constraints. (count the zero element in r). Since x_1 is the only strictly positive component, $\hat{B}$ contains only the first column of A, i.e., $B = \begin{bmatrix} 3 & 2 \end{bmatrix}^T$. Matrix $\hat{B}$ can be augmented, by using the second column of A to generate a nonsingular $\hat{B} = B$ as

$$B = \begin{bmatrix} 3 & 4 \\ 2 & 1 \end{bmatrix}$$

Simplex Method for Standard Form

Example

This leads to

$$\mathcal{I}_N = \{3, 4\} \quad \text{and} \quad \hat{A}_a = \begin{bmatrix} 3 & 4 & 1 & 0 \\ 2 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The vertex x is degenerate, matrix $\hat{A}_a$ is not unique. There are two possibilities for augmenting $\hat{B}$. Using the third column of A for the augmentation, we have

$$B = \begin{bmatrix} 3 & 1 \\ 2 & 0 \end{bmatrix}, \ \mathcal{I}_N = \{2,\ 4\}, \ \hat{A}_a = \begin{bmatrix} 3 & 4 & 1 & 0 \\ 2 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Simplex Method for Standard Form

Example

Alternatively, augmenting $\hat{B}$ with the fourth column of A yields

$$B = \begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix}, \; \mathcal{I}_N = \{2, 3\}, \text{ and } \hat{A}_a = \begin{bmatrix} 3 & 4 & 1 & 0 \\ 2 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

It can be easily verified that all three $\hat{A}_a$'s are nonsingular.

Simplex Method for Standard Form

Algorithm

We could change steps 2 and 3 of the previous simplex algorithm to reduce the computational complexity.

- At a vertex x_k, the nonsingularity of the working-set matrix $\hat{A}_{a_k}$ given by $\hat{A}_{a_k} = \begin{bmatrix} \tilde{A} \\ I_N \end{bmatrix}$ implies that there exist $\lambda_k \in \mathbb{R}^{p\times 1}$ and $\hat{\mu}_k \in \mathbb{R}^{(n-p)\times 1}$ such that

$$c = \hat{A}_{a_k}^T \begin{bmatrix} -\lambda_k \\ \hat{\mu}_k \end{bmatrix} = -\tilde{A}^T \lambda_k + I_N^T \hat{\mu}_k$$

If $\mu_k \in \mathbb{R}^{n\times 1}$ is the vector with zero basic variables and the components of $\hat{\mu}_k$ as its nonbasic variables, then the above equation can be expressed as

$$c = -\tilde{A}^T \lambda_k + \mu_k$$

The vertex x_k is a minimizer if and only if $\hat{\mu}_k \geq 0$.

Simplex Method for Standard Form

Algorithm

- If we use a permutation matrix P to rearrange the components of c in accordance with the partition of x_k into basic and nonbasic variables then

$$Pc = \begin{bmatrix} c_B \\ c_N \end{bmatrix} = -P\tilde{A}^T\lambda_k + PI_N^T\hat{\mu}_k = -\begin{bmatrix} B^T \\ N^T \end{bmatrix} \lambda_k + \begin{bmatrix} 0 \\ \hat{\mu}_k \end{bmatrix}$$

It follows that

$$B^T\lambda_k = -c_B \text{ and } \hat{\mu}_k = c_N + N^T\lambda_k$$

Since B is nonsingular, λ_k and $\hat{\mu}_k$ can be computed. The size of the matrix is $p \times p$, which is much smaller than $n \times n$ of the simplex method for the non-standard form.

Simplex Method for Standard Form

Algorithm

- If some entry in $\hat{\mu}_k$ is negative, then x_k is not a minimizer and a search direction d_k needs to be determined. Note the Lagrange multipliers $\hat{\mu}_k$ are not related to the equality constraints in $\tilde{A}x = \tilde{b}$ but are related to those bound constraints $x \geq 0$ that are active and are associated with the nonbasic variables.
- If the search direction d_k is partitioned according to the basic and nonbasic variables, x_B and x_N, into $d_k^{(B)}$ and $d_k^{(N)}$, respectively, and if $(\hat{\mu}_k)_l < 0$, then assigning

  $d_k^{(N)} = e_l$ where e_l is the lth column of the $(n-p) \times (n-p)$ identity matrix.

  d_k makes the v_lth constraint inactive without affecting other bound constraints that are associated with the nonbasic variables.
- In order to assure the feasibility of d_k, it is also required that $\tilde{A}d_k = 0$. This requirement can be described as

$$\tilde{A}d_k = Bd_k^{(B)} + Nd_k^{(N)} = Bd_k^{(B)} + Ne_l = 0$$

Simplex Method for Standard Form

Algorithm

- $d_k^{(B)}$ can determined by solving the system of equations

$$Bd_k^{(B)} = -a_{v_l} \text{ where } a_{v_l} = Ne_l$$

  Altogether we can determine the search direction d_k. It follows that

$$c^T d_k = -\lambda_k^T A d_k + \hat{\mu}_k^T I_N d_k = \hat{\mu}_k^T d_k^{(N)} = \hat{\mu}_k^T e_l = (\hat{\mu}_k)_l < 0$$

  Therefore, d_k is a feasible descent direction.

- To determine the step size α_k, we note that a point $x_k + \alpha d_k$ with any α satisfies the constraints $\tilde{A}x = \tilde{b}$, i.e.

$$\tilde{A}(x_k + \alpha d_k) = \tilde{A}x_k + \alpha \tilde{A} d_k = b$$

  The only constraints that are sensitive to step size α_k are those that are associated with the basic variables and are decreasing along direction d_k.

Simplex Method for Standard Form

Algorithm

- When limited to the basic variables, d_k becomes $d_k^{(B)}$. Since the normals of the constraints in $x \geq 0$ are simply coordinate vectors, a bound constraint associated with a basic variable is decreasing along d_k if the associated component in $d_k^{(B)}$ is negative.
- The special structure of the inequality constraints in $x \geq 0$ implies that the residual vector, when limited to basic variables in x_B, is x_B itself.
- The above analysis lead to a simple step that can be used to determine the index set

$$
\begin{aligned}
\mathcal{I}_k &= \{i : (d_k^{(B)})_i < 0\} \text{ and, if } \mathcal{I} \text{ is not empty} \\
\alpha_k &= \min_{i \in \mathcal{I}_k} \left[\frac{(x_k^{(B)})_i}{(-d_k^{(B)})_i} \right]
\end{aligned}
$$

where $x_k^{(B)}$ denotes the vector for the basic variables of x_k.

Simplex Method for Standard Form

Algorithm

- If i^* is the index in $\mathcal{I}_k$ that achieves α_k, then the i^*th component of $x_k^{(B)} + \alpha_k d_k^{(B)}$ is zero. This zero component is then interchanged with the lth component of $x_k^{(N)}$, which is now not zero but α_k.
- The vector $x_k^{(B)} + \alpha d_k^{(B)}$ after this updating becomes $x_{k+1}^{(B)}$ and $x_{k+1}^{(N)}$ remains a zero vector. Matrices B and N as well as the associated index sets $\mathcal{I}_B$ and $\mathcal{I}_N$ also need to be updated accordingly.

Simplex Method for Standard Form

Algorithm

Simplex algorithm for the standard-form LP problem

1. Input vertex x_0 set $k = 0$, and form $B, N, x_0^{(B)}$, $\mathcal{I}_B = \{\beta_1^{(0)}, \beta_2^{(0)}, \ldots, \beta_p^{(0)}$, and $\mathcal{I}_N = \{v_1^{(0)}, v_2^{(0)}, \ldots, v_{n-p}^{(0)}\}$.
2. Partition vector c into c_B and c_N. Solve $B^T\lambda_k = -c_B$ for λ_k and compute $\hat{\mu}_k$ using

   $$\hat{\mu}_k = c_N + N^T\lambda_k$$

   If $\hat{\mu}_k \geq 0$, stop (x_k is a verter minimizer); otherwise, select the index l that corresponds to the most negative component in $\hat{\mu}_k$.
3. Solve $Bd_k^{(B)} = -a_{v_l}$ for $d_k^{(B)}$ where a_{v_l} is the $v_l^{(k)}$th column of A.
4. Form index set $\mathcal{I}_k$ in $\mathcal{I}_k = \{i : (d_k^{(B)})_i < 0\}$. If $\mathcal{I}_k$ is empty then stop (the objective function tends to $-\infty$ in the feasible region); otherwise, compute α_k using $\alpha_k = \min_{i \in \mathcal{I}_k} \left[\frac{(x_k^{(B)})_i}{(-d_k^{(B)})_i} \right]$

Simplex Method for Standard Form

Algorithm

4. (cont.) and record the index i^* with $\alpha_k = \frac{(x_k^{(B)})_i^*}{(-d_k^{(B)})_i^*}$
5. Compute $x_{k+1}^{(B)} = x_k^{(B)} + \alpha_k d_k^{(B)}$ and replace its i^*th zero component by α_k. Set $x_{k+1}^{(N)} = 0$. Update B and N by interchanging the lth column of N with the i^*th column of B.
6. Update $\mathcal{I}_B$ and $\mathcal{I}_N$ by interchanging index $v_l^{(k)}$ of $\mathcal{I}_N$ with index $\beta_{i^*}^{(B)}$ of $\mathcal{I}_B$. Use the $x_{k+1}^{(B)}$ and $x_{k+1}^{(N)}$ obtained in Step 5 in conjunction with $\mathcal{I}_B$ and $\mathcal{I}_N$ to form x_{k+1}. Set $k = k + 1$ and repeat form Step 2.

Simplex Method for Standard Form

Example

Solve the standard-form LP problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) = 2x_1 + 9x_2 + 3x_3 \\ \text{subject to} \quad & -2x_1 + 2x_2 + x_3 - x_4 \quad = 1 \\ & x_1 + 4x_2 - x_3 - x_5 \quad = 1 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{aligned}$$

Solution: We have

$$A = \begin{bmatrix} -2 & 2 & 1 & -1 & 0 \\ 1 & 4 & -1 & 0 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \text{ and } c = \begin{bmatrix} 2 & 9 & 3 & 0 & 0 \end{bmatrix}^T$$

To identify a vertex, we set $x_1 = x_3 = x_4 = 0$ and solve the system

$$\begin{bmatrix} 2 & 0 \\ 4 & -1 \end{bmatrix} \begin{bmatrix} x_2 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \text{ for } x_2 \text{ and } x_5.$$

Simplex Method for Standard Form

Example

We have $x_2 = 1/2$ and $x_5 = 1$; hence $x_0 = \begin{bmatrix} 0 & \frac{1}{2} & 0 & 0 & 1 \end{bmatrix}^T$ is a vertex. Associated with x_0 are $\mathcal{I}_B = \{2, 5\}$, $\mathcal{I}_N = \{1, 3, 4\}$

$$B = \begin{bmatrix} 2 & 0 \\ 4 & -1 \end{bmatrix}, \quad N = \begin{bmatrix} -2 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix}, \text{ and } x_0^{(B)} = \begin{bmatrix} \frac{1}{2} & 1 \end{bmatrix}^T$$

Partitioning c into

$$c_B = \begin{bmatrix} 9 & 0 \end{bmatrix}^T \text{ and } c_N = \begin{bmatrix} 2 & 3 & 0 \end{bmatrix}^T$$

and solving $B^T \lambda_0 = -c_B$ for λ_0, we obtain $\lambda_0 = \begin{bmatrix} -\frac{9}{2} & 0 \end{bmatrix}^T$. Hence

$$\hat{\mu}_0 = c_N + N^T \lambda_0 = \begin{bmatrix} 2 \\ 3 \\ 0 \end{bmatrix} + \begin{bmatrix} -2 & 1 \\ 1 & -1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} -\frac{9}{2} \\ 0 \end{bmatrix} = \begin{bmatrix} 11 \\ -\frac{2}{3} \\ \frac{9}{2} \end{bmatrix}$$

Simplex Method for Standard Form

Example

Since $(\hat{\mu}_0)_2 < 0$, x_0 is not a minimizer, and $l = 2$. Next, we solve $Bd_0^{(B)} = -a_{v_2}$ for $d_0^{(B)}$ with $v_2^{(0)} = 3$ and $a_3 = \begin{bmatrix} 1 & -1 \end{bmatrix}^T$, which yields

$$d_0^{(B)} = \begin{bmatrix} -\frac{1}{2} \\ -3 \end{bmatrix} \text{ and } \mathcal{I}_0 = \{1, 2\}$$

Hence

$$\alpha_0 = \min\left(1, \frac{1}{3}\right) = \frac{1}{3} \text{ and } i^* = 2$$

To find $x_1^{(B)}$, we compute

$$x_0^{(B)} + \alpha_0 d_0^{(B)} = \begin{bmatrix} \frac{1}{3} \\ 0 \end{bmatrix}$$

Simplex Method for Standard Form

Example

Replace i^*th component by α_0, i.e.,

$$x_1^{(B)} = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \end{bmatrix} \text{ with } x_1^{(N)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Update B and N as

$$B = \begin{bmatrix} 2 & 1 \\ 4 & -1 \end{bmatrix} \text{ and } N = \begin{bmatrix} -2 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix}$$

and update $\mathcal{I}_B$ and $\mathcal{I}_N$ as $\mathcal{I}_B = \{2, 3\}$ and $\mathcal{I}_N = \{1, 5, 4\}$. The vertex obtained is $x_1 = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 \end{bmatrix}^T$ to compute the first iteration.
The second iteration starts with the partitioning of c into

$$c_B = \begin{bmatrix} 9 \\ 3 \end{bmatrix} \text{ and } c_N = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$$

Simplex Method for Standard Form

Example

Solving $B^T\lambda_1 = -c_B$ for λ_1, we obtain $\lambda_1 = \begin{bmatrix} -\frac{7}{2} & -\frac{1}{2} \end{bmatrix}^T$ which leads to

$$\hat{\mu}_1 = c_N + N^T\lambda_1 = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -2 & 1 \\ 0 & -1 \\ -1 & 0 \end{bmatrix}^T \begin{bmatrix} -\frac{7}{2} \\ -\frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{17}{2} \\ \frac{1}{2} \\ \frac{7}{2} \end{bmatrix}$$

Since $\hat{\mu}_1 > 0$, x_1 is the unique vertex minimizer.

Tabular Form of the Simplex Method

For LP problems of very small size, the simple method can be applied in terms of a tabular form in which the input data such as A, b, and c are used to form a table.

Consider the standard0form LP problem:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

- Assume that at vertex x_k the equality constraints are expressed as

$$x_k^{(B)} + B^{-1} N x_k^{(N)} = B^{-1} b$$

From $c = -A^T \lambda_k + \mu_k$, the objective function is given by

$$c^T x_k = \mu_k^T x_k - \lambda_k^T A x_k = 0^T x_k^{(B)} + \hat{\mu}_k^T x_k^{(N)} - \lambda_k^T b$$

Tabular Form of the Simplex Method

The important data at the kth iteration can be put together in a tabular form as a table.

x_B^T	x_N^T	
I	$B^{-1}N$	$B^{-1}b$
0^T	$\hat{\mu}_k^T$	$\lambda_k^T b$

- If $\hat{\mu} \geq 0$, x_k is a minimizer.
- Otherwise, and appropriate rule can be used to choose a negative component in $\hat{\mu}_k$, say $(\hat{\mu})_l < 0$. The column in $B^{-1}N$ gives $-d_k^{(B)}$. This column will be referred to as the pivot column. The variable in x_N^T that corresponds to $(\hat{\mu})_l$ is the variable chosen as a basic variable.
- Since $x_k^{(N)} = 0$, $x_k^{(B)} + B^{-1}Nx_k^{(N)} = B^{-1}b$ implies that $x_k^{(B)} = B^{-1}b$. Therefore, the far-right p-dimensional vector gives $x_k^{(B)}$.
- Since $x_k^{(N)} = 0$, $c^T x_k = 0^T x_k^{(B)} + \hat{\mu}_k^T x_k^{(N)} - \lambda_k^T b$ implies that the number in the lower-right corner of the table is equal to $-f(x_k)$.

Tabular Form of the Simplex Method

The important data at the kth iteration can be put together in a tabular form as a table.

Basic variables		Nonbasic variables				
x_2	x_5	x_1	x_3	x_4	$B^{-1}b$	
1	0	-1	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	
0	1	-5	3	-2	1	
0	0	11	$-\frac{3}{2}$	$\frac{9}{2}$	$-\frac{9}{2}$	$\leftarrow \lambda_k^T b$

- From the previous example with x_0, since $(\hat{\mu})_2 < 0$, x_0 is not a minimizer. x_3 is the variable in $x_0^{(N)}$ that will become a basic variable, and the vector above $(\hat{\mu})_2$, $\begin{bmatrix} \frac{1}{2} & 3 \end{bmatrix}^T$, is the pivot column $-d_0^{(B)}$.
- From $\mathcal{I}_k = \{i : (d_k^{(B)})_i < 0\}$, , the positive components of the pivot column should be used to compute the ratio $(x_0^{(B)})_i/(-d_0^{(B)})_i$ where $x_0^{(B)}$ is the far-right column in the table. The minimum ratio is $i^* = 2$. The second basic variable, x_5, should be exchanged with x_3 to become a nonbasic variable.

Tabular Form of the Simplex Method

Basic variables		Nonbasic variables				
x_2	x_5	x_1	x_3	x_4	$B^{-1}b$	
1	$-\frac{1}{6}$	$-\frac{1}{6}$	0	$-\frac{1}{6}$	$\frac{1}{3}$	
0	$\frac{1}{3}$	$-\frac{5}{3}$	1	$-\frac{2}{3}$	$\frac{1}{3}$	
0	0	11	$-\frac{3}{2}$	$\frac{9}{2}$	$-\frac{9}{2}$	$\leftarrow \lambda_k^T b$

- To transform x_3 into the second basic variable, we use row operations to transform the pivot column into the i^*th coordinate vector. Here we can add $-1/6$ times the second row to the first row, and then multiply the second row by $1/3$

Tabular Form of the Simplex Method

Basic variables		Nonbasic variables				
x_2	x_3	x_1	x_5	x_4	$B^{-1}b$	
1	0	$-\frac{1}{6}$	$-\frac{1}{6}$	$-\frac{1}{6}$	$\frac{1}{3}$	
0	1	$-\frac{5}{3}$	$\frac{1}{3}$	$-\frac{2}{3}$	$\frac{1}{3}$	
0	0	$\frac{17}{2}$	$\frac{1}{2}$	$\frac{7}{2}$	-4	$\leftarrow \lambda_k^T b$

- We interchange the columns associated with variable x_3 and x_5 to form the updated basic and nonbasic variables, and then add $3/2$ times the second row to the last row to eliminate the nonzero Lagrange multiplier associated with variable x_3.
- The Lagrange multipliers $\hat{\mu}_1$ in the last row of the tale are all positive and hence x_1 is the unique minimizer. Vector x_1 is specified by $x_1^{(B)} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \end{bmatrix}^T$ in the far0right column and $x_1^{(N)} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$.

Tabular Form of the Simplex Method

- In the conjunction with the composition of the basic and nonbasic variables, $x_1^{(B)}$ and $x_1^{(N)}$ yield

$$x_1 = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 \end{bmatrix}^T$$

At x_1, the lower-right corner of the table gives the minimum of the objective function as $f(x_1) = 4$.

Reference

1. Joaquim R. R. A. Martins, Andrew Ning, "*Engineering Design Optimization*," Cambridge University Press, 2021.
2. Mykel J. kochenderfer, and Tim A. Wheeler, "*Algorithms for Optimization*," The MIT Press, 2019.
3. Ashok D. Belegundu, Tirupathi R. Chandrupatla, "*Optimization Concepts and Applications in Engineering*," Cambridge University Press, 2019.
4. Laurent Lessarn , "*Introduction to Optimization*," Lecture Note, University of Wisconsin–Madison.
5. Andreas Antoniou and Wu-Sheng Lu, "*Practical Optimization: Algorithms and Engineering Applications*," 2^{nd} edition, Springer, 2021.