Practical Guide to the Simplex Method of Linear Programming

Marcel Oliver

Revised: September 28, 2020

1 The basic steps of the simplex algorithm

Step 1: Write the linear programming problem in *standard form*

Linear programming (the name is historical, a more descriptive term would be *linear optimization*) refers to the problem of optimizing a linear *objective function* of several variables subject to a set of linear equality or inequality constraints.

Every linear programming problem can be written in the following *standard form.*

$$\text{Minimize } \zeta = \boldsymbol{c}^T \boldsymbol{x} \tag{1a}$$

subject to

$$A\boldsymbol{x} = \boldsymbol{b}\,, \tag{1b}$$

$$\boldsymbol{x} \geq \boldsymbol{0}\,. \tag{1c}$$

Here $\boldsymbol{x} \in \mathbb{R}^n$ is a vector of n unknowns, $A \in M(m \times n)$ with n typically much larger than m, $\boldsymbol{c} \in \mathbb{R}^n$ the coefficient vector of the objective function, and the expression $\boldsymbol{x} \geq \boldsymbol{0}$ signifies $x_i \geq 0$ for $i = 1, \dots, n$. For simplicity, we assume that rank $A = m$, i.e., that the rows of A are linearly independent.

Turning a problem into standard form involves the following steps.

(i) Turn Maximization into minimization and write inequalities in standard order.

This step is obvious. Multiply expressions, where appropriate, by -1.

(ii) Introduce *slack variables* to turn inequality constraints into equality constraints with nonnegative unknowns.

Any inequality of the form $a_1\, x_1 + \cdots + a_n\, x_n \leq c$ can be replaced by $a_1\, x_1 + \cdots + a_n\, x_n + s = c$ with $s \geq 0$.

(iii) Replace variables which are not sign-constrained by differences.

Any real number x can be written as the difference of nonnegative numbers $x = u - v$ with $u, v \geq 0$.

Consider the following example.

$$\text{Maximize } z = x_1 + 2\, x_2 + 3\, x_3 \tag{2a}$$

subject to

$$x_1 + x_2 - x_3 = 1\,, \tag{2b}$$
$$-2\, x_1 + x_2 + 2\, x_3 \geq -5\,, \tag{2c}$$
$$x_1 - x_2 \leq 4\,, \tag{2d}$$
$$x_2 + x_3 \leq 5\,, \tag{2e}$$
$$x_1 \geq 0\,, \tag{2f}$$
$$x_2 \geq 0\,. \tag{2g}$$

Written in standard form, the problem becomes

$$\text{minimize } \zeta = -x_1 - 2\, x_2 - 3\, u + 3\, v \tag{3a}$$

subject to

$$x_1 + x_2 - u + v = 1\,, \tag{3b}$$
$$2\, x_1 - x_2 - 2\, u + 2\, v + s_1 = 5\,, \tag{3c}$$
$$x_1 - x_2 + s_2 = 4\,, \tag{3d}$$
$$x_2 + u - v + s_3 = 5\,, \tag{3e}$$
$$x_1, x_2, u, v, s_1, s_2, s_3 \geq 0\,. \tag{3f}$$

Step 2: Write the coefficients of the problem into a *simplex tableau*

The coefficients of the linear system are collected in an augmented matrix as known from Gaussian elimination for systems of linear equations; the coefficients of the objective function are written in a separate bottom row with a zero in the right hand column.

For our example, the initial tableau reads:

x_1	x_2	u	v	s_1	s_2	s_3	
1	1	−1	1	0	0	0	1
2	−1	−2	2	1	0	0	5
1	−1	0	0	0	1	0	4
0	1	1	−1	0	0	1	5
−1	−2	−3	3	0	0	0	0

In the following steps, the variables will be divided into m *basic variables* and $n - m$ *non-basic variables*. We will act on the tableau by the rules of Gaussian elimination, where the pivots are always chosen from the columns corresponding to the basic variables.

Before proceeding, we need to choose an initial set of basic variables which corresponds to a point in the feasible region of the linear programming problem. Such a choice may be non-obvious, but we shall defer this discussion for now. In our example, x_1 and $s_1, \ldots, s_3$ shall be chosen as the initial basic variables, indicated by gray columns in the tableau above.

Step 3: Gaussian elimination

For a given set of basic variables, we use Gaussian elimination to reduce the corresponding columns to a permutation of the identity matrix. This amounts to solving $A\boldsymbol{x} = \boldsymbol{b}$ in such a way that the values of the nonbasic variables are zero and the values for the basic variables are explicitly given by the entries in the right hand column of the fully reduced matrix. In addition, we eliminate the coefficients of the objective function below each pivot.

Our initial tableau is thus reduced to

x_1	x_2	$\boldsymbol{u}$	v	s_1	s_2	s_3	
1	1	**−1**	1	0	0	0	1
0	−3	**0**	0	1	0	0	3
0	*−2*	***1***	*−1*	*0*	*1*	*0*	*3*
0	1	**1**	−1	0	0	1	5
0	−1	**−4**	4	0	0	0	1

The solution expressed by the tableau is only admissible if all basic variables are non-negative, i.e., if the right hand column of the reduced tableau is free of negative entries. This is the case in this example. At the initial stage, however, negative entries may come up; this indicates that different initial basic variables should have been chosen. At later stages in the process, the selection rules for the basic variables will guarantee that an initially feasible tableau will remain feasible throughout the process.

Step 4: Choose new basic variables

If, at this stage, the objective function row has at least one negative entry, the cost can be lowered by making the corresponding variable basic. This new basic variable is called the *entering variable*. Correspondingly, one formerly basic variable has then to become nonbasic, this variable is called the *leaving variable*. We use the following standard selection rules.

(i) The *entering variable* shall correspond to the column which has the most negative entry in the cost function row. If all cost function coefficients are non-negative, the cost cannot be lowered and we have reached an optimum. The algorithm then terminates.

(ii) Once the entering variable is determined, the *leaving variable* shall be chosen as follows. Compute for each row the ratio of its right hand coefficient to the corresponding coefficient in the entering variable column. Select the row with the smallest finite positive ratio. The leaving variable is then determined by the column which currently owns the pivot in this row. If all coefficients in the entering variable column are non-positive, the cost can be lowered indefinitely, i.e., the linear programming problem does not have a finite solution. The algorithm then also terminates.

If entering and leaving variable can be found, go to Step 3 and iterate.

Note that choosing the most negative coefficient in rule (i) is only a heuristic for choosing a direction of fast decrease of the objective function. Rule (ii) ensures that the new set of basic variables remains feasible.

Let us see how this applies to our problem. The previous tableau holds the most negative cost function coefficient in column 3, thus u shall be the entering variable (marked in boldface). The smallest positive ratio of right hand column to entering variable column is in row 3, as $\frac{3}{1} < \frac{5}{1}$. The pivot in this row points to s_2 as the leaving variable. Thus, after going through the Gaussian elimination once more, we arrive at

x_1	$\boldsymbol{x_2}$	u	v	s_1	s_2	s_3	
1	$\mathbf{-1}$	0	0	0	1	0	4
0	$\mathbf{-3}$	0	0	1	0	0	3
0	$\mathbf{-2}$	1	−1	0	1	0	3
0	***3***	*0*	*0*	*0*	*−1*	*1*	*2*
0	$\mathbf{-9}$	0	0	0	4	0	13

At this point, the new entering variable is x_2 corresponding to the only negative entry in the last row, the leaving variable is s_3. After Gaussian elimination, we find

x_1	x_2	u	v	s_1	s_2	s_3	
1	0	0	0	0	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{14}{3}$
0	0	0	0	1	−1	1	5
0	0	1	−1	0	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{13}{3}$
0	1	0	0	0	$-\frac{1}{3}$	$\frac{1}{3}$	$\frac{2}{3}$
0	0	0	0	0	1	3	19

Since there is no more negative entry in the last row, the cost cannot be lowered by choosing a different set of basic variables; the termination condition applies.

Step 4: Read off the solution

The solution represented by the final tableau has all nonbasic variables set to zero, while the values for the basic variables can be can be read off the right hand column. The bottom right corner gives the negative of the objective function.

In our example, the solution reads $x_1 = \frac{14}{3}$, $x_2 = \frac{2}{3}$, $x_3 = u = \frac{13}{3}$, $s_1 = 5$, $v = s_2 = s_3 = 0$, which corresponds to $\zeta = -19$, which can be independently checked by plugging the solution back into the objective function.

As a further check, we note that the solution must satisfy (2b), (2d), and (2e) with equality and (2c) with a slack of 5. This can also be checked by direct computation.

2 Initialization

For some problem it is not obvious which set of variables form a *feasible* initial set of basic variables. For large problems, a trial-and-error approach is prohibitively expensive due the rapid growth of $\binom{n}{m}$, the number of possibilities to choose m basic variables out of a total of n variables, as m and n become large. This problem can be overcome by adding a set of m artificial variables which form a trivial set of basic variables and which are penalized by a large coefficients ω in the objective function. This penalty will cause the artificial variables to become nonbasic as the algorithm proceeds.

We explain the method by example. For the problem

$$\text{minimize } z = x_1 + 2\,x_2 + 2\,x_3 \tag{4a}$$

subject to

$$x_1 + x_2 + 2\,x_3 + x_4 = 5\,, \tag{4b}$$
$$x_1 + x_2 + x_3 - x_4 = 5\,, \tag{4c}$$
$$x_1 + 2\,x_2 + 2\,x_3 - x_4 = 6\,, \tag{4d}$$
$$\boldsymbol{x} \geq \mathbf{0}\,, \tag{4e}$$

we set up a simplex tableau with three artificial variables which are initially basic:

a_1	a_2	a_3	x_1	x_2	x_3	x_4	
1	0	0	1	1	2	1	5
0	1	0	1	1	1	-1	5
0	0	1	1	2	2	-1	6
ω	ω	ω	1	2	2	0	0

We proceed as before by first eliminating the nonzero entries below the pivots:

a_1	a_2	a_3	x_1	x_2	$\boldsymbol{x_3}$	x_4	
1	*0*	*0*	*1*	*1*	***2***	*1*	*5*
0	1	0	1	1	**1**	-1	5
0	0	1	1	2	**2**	-1	6
0	0	0	$1-3\omega$	$2-4\omega$	$\boldsymbol{2-5\omega}$	ω	-16ω

Since, for ω large, $2-5\omega$ is the most negative coefficient in the objective function row, x_3 will be entering and, since $\frac{5}{2} < \frac{6}{2} < \frac{5}{1}$, a_1 will be leaving. The Gaussian elimination step then yields

a_1	a_2	a_3	x_1	$\boldsymbol{x_2}$	x_3	x_4	
$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\mathbf{\frac{1}{2}}$	1	$\frac{1}{2}$	$\frac{5}{2}$
$-\frac{1}{2}$	1	0	$\frac{1}{2}$	$\mathbf{\frac{1}{2}}$	0	$-\frac{3}{2}$	$\frac{5}{2}$
−1	*0*	*1*	*0*	***1***	*0*	*−2*	*1*
$-1+\frac{5}{2}\omega$	0	0	$-\frac{1}{2}\omega$	$\boldsymbol{1-\frac{3}{2}\omega}$	0	$-1+\frac{7}{2}\omega$	$-5-\frac{7}{2}\omega$

Now x_2 is entering, a_3 is leaving, and we obtain

a_1	a_2	a_3	$\boldsymbol{x_1}$	x_2	x_3	x_4	
1	0	$-\frac{1}{2}$	$\mathbf{\frac{1}{2}}$	0	1	$\frac{3}{2}$	2
0	*1*	$-\frac{1}{2}$	$\boldsymbol{\frac{1}{2}}$	*0*	*0*	$-\frac{1}{2}$	*2*
−1	0	1	**0**	1	0	−2	1
ω	0	$-1+\frac{3}{2}\omega$	$\boldsymbol{-\frac{1}{2}\omega}$	0	0	$1+\frac{1}{2}\omega$	$-6-2\omega$

The new entering variable is x_1 while the criterion for the leaving variable is tied between a_2 and x_3. Since we want the artificial variable to become nonbasic, we take a_2 to be leaving. (Choosing x_3 as the leaving variable would lead us to the same solution, albeit after a few extra steps.) We obtain

a_1	a_2	a_3	x_1	x_2	x_3	x_4	
1	−1	0	0	0	1	2	0
0	2	−1	1	0	0	−1	4
−1	0	1	0	1	0	−2	1
ω	ω	$-1+\omega$	0	0	0	1	−6

The termination condition is now satisfied, and we see that the solution is $z = 6$ with $x_1 = 4$, $x_2 = 1$, $x_3 = 0$, $x_4 = 0$.

We close with two remarks.

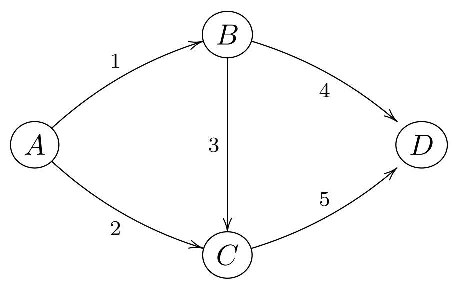
- When using a computer to perform the simplex algorithm numerically, ω should be chosen large (one or two orders of magnitude larger than any of the other coefficients in the problem) but not too large (to avoid loss of significant digits in floating point arithmetic).

- If not all artificial variables become nonbasic, ω must be increased. If this happens for any value of ω, the feasible region is empty.

- In the final tableau, the penalty parameter ω can only appear in artificial variable columns.

3 Duality

The concept of duality is best motivated by an example. Consider the following transportation problem. Some good is available at location A at no cost and may be transported to locations B, C, and D according to the following directed graph:



On each of the unidirectional channels, the unit cost of transportation is c_j for $j = 1, \ldots, 5$. At each of the vertices b_α units of the good are sold, where $\alpha = B, C, D$. How can the transport be done most efficiently?

A first, and arguably most obvious way of quantifying efficiency would be to state the question as a minimization problem for the total cost of transportation. If x_j denotes the amount of good transported through channel j, we arrive at the following linear programming problem:

$$\text{minimize } c_1\, x_1 + \cdots + c_5\, x_5 \tag{5a}$$

subject to

$$x_1 - x_3 - x_4 = b_B\,, \tag{5b}$$
$$x_2 + x_3 - x_5 = b_C\,, \tag{5c}$$
$$x_4 + x_5 = b_D\,. \tag{5d}$$

The three equality constraints state that nothing gets lost at nodes B, C, and D except what is sold.

There is, however, a second, seemingly equivalent way of characterizing efficiency of transportation. Instead of looking at minimizing the cost of transportation, we seek to maximize the income from selling the good. Letting y_α denote the unit price of the good at node $\alpha = A, \ldots, D$ with $y_A = 0$ by assumption, the associated linear programming problem is the following:

$$\text{maximize } y_B\, b_B + y_C\, b_C + y_D\, b_D \tag{6a}$$

subject to

$$y_B - y_A \le c_1\,, \tag{6b}$$

$$y_C - y_A \le c_2\,, \tag{6c}$$

$$y_C - y_B \le c_3\,, \tag{6d}$$

$$y_D - y_B \le c_4\,, \tag{6e}$$

$$y_D - y_C \le c_5\,. \tag{6f}$$

The inequality constraints encode that, in a free market, we can only maintain a price difference that does not exceed the cost of transportation. If we charged a higher price, then "some local guy" would immediately be able to undercut our price by buying from us at one end of the channel, using the channel at the same fixed channel cost, then selling at a price lower than ours at the high-price end of the channel.

Setting

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_5 \end{pmatrix}, \qquad \boldsymbol{y} = \begin{pmatrix} y_B \\ y_C \\ y_D \end{pmatrix}, \qquad \text{and } A = \begin{pmatrix} 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \tag{7}$$

we can write (5) as the abstract *primal problem*

$$\text{minimize } \boldsymbol{c}^T \boldsymbol{x} \tag{8a}$$

$$\text{subject to } A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \ge \boldsymbol{0}\,. \tag{8b}$$

Likewise, (6) can be written as the *dual problem*

$$\text{maximize } \boldsymbol{y}^T \boldsymbol{b} \tag{9a}$$

$$\text{subject to } \boldsymbol{y}^T A \le \boldsymbol{c}^T\,. \tag{9b}$$

We shall prove in the following that the minimal cost and the maximal income coincide, i.e., that the two problems are equivalent.

Let us first remark this problem is easily solved without the simplex algorithm: clearly, we should transport all goods sold at a particular location through the cheapest channel to that location. Thus, we might perform a simple search for the cheapest channel, something which can be done efficiently by combinatorial algorithms such as Dijkstra's algorithm [2]. The advantage of the linear programming perspective is that additional constraints such as channel capacity limits can be easily added. For the purpose of understanding the relationship between the primal and the dual problem, and for understanding the significance of the dual formulation, the simple present setting is entirely adequate.

The unknowns $\boldsymbol{x}$ in the primal formulation of the problem not only identify the vertex of the feasible region at which the optimum is reached, but they also act as sensitivity parameters with regard to small changes in the cost coefficients $\boldsymbol{c}$. Indeed, when the linear programming problem is nondegenerate, i.e. has a unique optimal solution, changing the cost coefficients from $\boldsymbol{c}$ to $\boldsymbol{c} + \Delta\boldsymbol{c}$ with $|\Delta\boldsymbol{c}|$ sufficiently small will not make the optimal vertex jump to another corner of the feasible region, as the cost depends continuously on $\boldsymbol{c}$. Thus, the corresponding change in cost is $\Delta\boldsymbol{c}^T\boldsymbol{x}$. If x_i is nonbasic, the cost will not react at all to small changes in c_i, whereas if x_i is large, then the cost will be sensitive to changes in c_i. This information is often important because it gives an indication where to best spend resources if the parameters of the problem—in the example above, the cost of transportation—are to be improved.

Likewise, the solution vector $\boldsymbol{y}$ to the dual problem provides the sensitivity of the total income to small changes in $\boldsymbol{b}$. Here, $\boldsymbol{b}$ is representing the number of sales at the various vertices of the network; if the channels were capacity constrained, the channel limits were also represented as components of $\boldsymbol{b}$. Thus, the dual problem is providing the answer to the question "if I were to invest in raising sales, where should I direct this investment to achieve the maximum increase in income?"

The following theorems provide a mathematically precise statement on the equivalence of primal and dual problem.

Theorem 1 (Weak duality)**.** *Assume that $\boldsymbol{x}$ is a feasible vector for the primal problem* (8) *and $\boldsymbol{y}$ is a feasible vector for the dual problem* (9)*. Then*

(i) $\boldsymbol{y}^T\boldsymbol{b} \le \boldsymbol{c}^T\boldsymbol{x}$*;*

(ii) if (i) holds with equality, then $\boldsymbol{x}$ and $\boldsymbol{y}$ are optimal for their respective linear programming problems;

(iii) the primal problem does not have a finite minimum if and only if the feasible region of the dual problem is empty; vice versa, the dual problem does not have a finite maximum if and only if the feasible region of the primal problem is empty.

The proof is simple and shall be left as an exercise.

To proceed, we say that $\boldsymbol{x}$ is a *basic feasible solution* of $A\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{x} \ge 0$ if it has at most m nonzero components. We say that it is *nondegenerate* if it has exactly m nonzero components. If, in the course of performing the simplex algorithm, we hit a degenerate basic feasible solution, it is possible that the objective function row in the simplex tableau contains negative coefficients, yet the cost cannot be lowered because the corresponding basic variable is already zero. This can lead to cycling and thus non-termination of the algorithm. We shall not consider the degenerate case further.

When $\boldsymbol{x}$ is a nondegenerate solution to the primal problem (8), i.e., $\boldsymbol{x}$ is nondegenerate basic feasible and also optimal, then we can be assured that the simplex method terminates with all coefficients in the objective function row nonnegative. (If they were not, we could immediately perform at least one more step of the algorithm with strict decrease in the cost.) In this situation, we can use the simplex algorithm as described to prove the following stronger form of the duality theorem.

Theorem 2 (Strong duality)**.** *The primal problem* (8) *has an optimal solution* $\boldsymbol{x}$ *if and only if the dual problem* (9) *has an optimal solution* $\boldsymbol{y}$*; in this case* $\boldsymbol{y}^T\boldsymbol{b} = \boldsymbol{c}^T\boldsymbol{x}$.

Proof. We only give a proof in the case when the solution to the primal problem is non-degenerate. It is based on a careful examination of the termination condition of the simplex algorithm. Assume that $\boldsymbol{x}$ solves the primal problem. Without loss of generality, we can reorder the variables such that the first m variables are basic, i.e.

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_B \\ \boldsymbol{0} \end{pmatrix} \tag{10}$$

and that the final simplex tableau reads

$$\left(\begin{array}{cc|c} I & R & \boldsymbol{x}_B \\ \boldsymbol{0}^T & \boldsymbol{r}^T & -z \end{array}\right) . \tag{11}$$

The last row represents the objective function coefficients and z denotes the optimal value of the objective function. We note that the termination condition of the simplex algorithm reads $\boldsymbol{r} \geq \boldsymbol{0}$. We now partition the initial matrix A and the coefficients of the objective function $\boldsymbol{c}$ into their basic and nonbasic components, writing

$$A = \begin{pmatrix} B & N \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{c} = \begin{pmatrix} \boldsymbol{c}_B \\ \boldsymbol{c}_N \end{pmatrix} . \tag{12}$$

Finally, it can be shown that the elementary row operations used in the Gaussian elimination steps of the simplex algorithm can be written as multiplication by a matrix from the left, which we also partition into components compatible with the block matrix structure of (11), so that the transformation from the initial to the final tableau can be written as

$$\begin{pmatrix} M & \boldsymbol{v} \\ \boldsymbol{u}^T & \alpha \end{pmatrix} \left(\begin{array}{cc|c} B & N & \boldsymbol{b} \\ \boldsymbol{c}_B^T & \boldsymbol{c}_N^T & 0 \end{array}\right) = \left(\begin{array}{cc|c} MB + \boldsymbol{v}\boldsymbol{c}_B^T & MN + \boldsymbol{v}\boldsymbol{c}_N^T & M\boldsymbol{b} \\ \boldsymbol{u}^T B + \alpha \boldsymbol{c}_B^T & \boldsymbol{u}^T N + \alpha \boldsymbol{c}_N^T & \boldsymbol{u}^T \boldsymbol{b} \end{array}\right) . \tag{13}$$

We now compare the right hand side of (13) with (11) to determine the coefficients of the left hand matrix. First, we note that in the simplex algorithm, none of the Gaussian elimination steps on the equality constraints depend on the objective function coefficients (other than the path taken from initial to final tableau, which is not at issue here). This immediately implies that $\boldsymbol{v} = \boldsymbol{0}$. Second, we observe that nowhere in the simplex algorithm do we ever rescale the objective function row. This immediately implies that $\alpha = 1$. This leaves us with the following set of matrix equalities:

$$MB = I \,, \tag{14a}$$

$$M\boldsymbol{b} = \boldsymbol{x}_B \,, \tag{14b}$$

$$\boldsymbol{u}^T B + \boldsymbol{c}_B^T = \boldsymbol{0}^T \,, \tag{14c}$$

$$\boldsymbol{u}^T N + \boldsymbol{c}_N^T = \boldsymbol{r}^T \,. \tag{14d}$$

so that $M = B^{-1}$ and $\boldsymbol{u}^T = -\boldsymbol{c}_B^T B^{-1}$. We now claim that

$$\boldsymbol{y}^T = \boldsymbol{c}_B^T B^{-1} \tag{15}$$

solves the dual problem. We compute

$$\begin{aligned} \boldsymbol{y}^T A &= \boldsymbol{c}_B^T B^{-1} \begin{pmatrix} B & N \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{c}_B^T & \boldsymbol{c}_B^T B^{-1} N \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{c}_B^T & \boldsymbol{c}_N^T - \boldsymbol{r}^T \end{pmatrix} \\ &\leq \begin{pmatrix} \boldsymbol{c}_B^T & \boldsymbol{c}_N^T \end{pmatrix} = \boldsymbol{c}^T \,. \end{aligned} \tag{16}$$

This shows that $\boldsymbol{y}$ is feasible for the dual problem. Moreover,

$$\boldsymbol{y}^T \boldsymbol{b} = \boldsymbol{c}_B^T B^{-1} \boldsymbol{b} = \boldsymbol{c}_B^T \boldsymbol{x}_B = \boldsymbol{c}^T \boldsymbol{x} \,. \tag{17}$$

Thus, by weak duality, $\boldsymbol{y}$ is also optimal for the dual problem.

The reverse implication of the theorem follows from the above by noting that the bi-dual is identical with the primal problem. □

References

[1] G.B. Dantzig, M.N. Thapa, "Linear Programming," Springer, New York, 1997.

[2] "Dijkstra's algorithm." Wikipedia, The Free Encyclopedia. April 4, 2012, 06:13 UTC.
http://en.wikipedia.org/w/index.php?title=Dijkstra%27s_algorithm&oldid=485462965

[3] D. Gale, *Linear Programming and the simplex method*, Notices of the AMS **54** (2007), 364–369.
http://www.ams.org/notices/200703/fea-gale.pdf

[4] P. Pedregal, "Introduction to optimization," Springer, New York, 2004.