

Initialization: The Big-M Formulation

Consider the linear program:

$$\begin{array}{ll} \text{Minimize} & 4x_1 + x_2 \\ \text{Subject to:} & \\ & 3x_1 + x_2 = 3 \quad (1) \\ & 4x_1 + 3x_2 \geq 6 \quad (2) \\ & x_1 + 2x_2 \leq 3 \quad (3) \\ & x_1, x_2 \geq 0. \end{array}$$

Notice that there are several new features in this problem, namely: (i) the objective is to minimize; (ii) the first constraint is in equality form, but it does not have a candidate basic variable; and (iii) the second constraint is of the “ \geq ” type.

Two approaches are commonly adopted for the handling of minimization objective functions. The first is to convert the given objective function into one that is to be maximized. This is done by multiplying the original objective function by -1 and then maximizing the resulting expression. For this example, this means that we can replace the given objective function by:

$$\text{Maximize} \quad -4x_1 - x_2.$$

The second approach is to revise the optimality criterion in the Simplex algorithm. For maximization problems, recall that the optimality criterion is that if the coefficients of all nonbasic variables in the zeroth row of a Simplex tableau are nonnegative, then its associated basic feasible solution is optimal. If, instead, we are trying to minimize, then it is not difficult to see that we could simply revise the just-stated criterion into one that has the word “nonnegative” replaced by the word “nonpositive.” Moreover, if the current solution is not optimal, then we should select the nonbasic variable that has the most-positive coefficient in the zeroth row as the entering variable. Note, however, that the remaining aspects of the Simplex algorithm, the ratio test in particular, do not require any revision.

In our solution of this linear program, we will adopt the second approach. Hence, no action is necessary at this point.

Suppose a given constraint is in equality form with a nonnegative right-hand-side constant, as in equation (1) above. (If the right-hand-side constant of an equation is negative, the entire equation can be multiplied by -1 to convert the constant to a positive number.) What we need to do is to find out whether or not the constraint contains a candidate basic variable. If such a variable can be found, we simply declare it as the basic variable associated with that equation and move on to other constraints. In the event that such a

variable does not exist, the idea is to artificially introduce a new nonnegative variable to serve as the basic variable associated with that equation. Such a variable will be called an *artificial variable*. Since constraint (1) above does not contain a candidate basic variable, we will revise it by introducing an artificial variable, which we denote by A_1 ; this results in

$$3x_1 + x_2 + A_1 = 3.$$

We will also add $A_1 \geq 0$ into the set of nonnegativity constraints.

It is important to realize that with the introduction of an artificial variable, the resulting new equation is *not* equivalent to the original. This is because if the artificial variable assumes a positive value, then any solution that satisfies the new equation won't satisfy the original equation. For example, if we let $A_1 = 2$ in the above equation, then, since $3x_1 + x_2 + A_1 = 3$, we must have $3x_1 + x_2 = 1$, which is in contradiction with the original equation $3x_1 + x_2 = 3$. Now, as A_1 is one of the starting basic variables, it will (typically) assume a positive value at the beginning of the Simplex iterations. Hence, the starting basic feasible solution will not be a feasible solution to the original problem. Since our aim is to derive an optimal solution that satisfies the original equality constraint, not the revised constraint, we are, in the end, only interested in solutions that have $A_1 = 0$. Therefore, an important question is: How do we get rid of this artificial variable?

One answer (another answer will be given a bit later) to this question is that we can introduce a new term MA_1 , where M is a "sufficiently large" constant, into the objective function. The idea behind this approach, which is naturally called the *big- M method*, is that although the value of A_1 may be positive initially, but with this added term in the objective function, any solution that has a positive A_1 will have an associated objective-function value that is exceedingly large. Hence, as the Simplex algorithm performs its search for a solution that has the smallest objective function value, it will systematically discard or avoid solutions that have a positive A_1 . In other words, the Simplex algorithm will, by design, attempt to converge to solutions that have $A_1 = 0$, and hence are feasible to the original problem.

As a numerical example, consider the solutions $(x_1, x_2, A_1) = (1/3, 2, 0)$ and $(x_1, x_2, A_1) = (1/3, 1, 1)$, which satisfy the original and the revised equation (1), respectively. The corresponding objective-function values of these two solutions can be evaluated as: $4 \times (1/3) + 1 \times 2 + M \times 0 = 10/3$ and $4 \times (1/3) + 1 \times 1 + M \times 1 = (7/3) + M$. Notice that the outcome of the first evaluation does not involve M . It follows that the second evaluation will have a greater value, provided that M is sufficiently large (any M that is strictly greater than 1, specifically). Since a comparison between any solution with $A_1 = 0$ and any other solution with $A_1 > 0$ will always result in this order (when M is sufficiently large), the Simplex algorithm will attempt to weed out any solution with a positive A_1 .

We now turn our attention to constraint (2), which is of the “ \geq ” type. To create an equivalent equality, we will reverse what we do in the case of a “ \leq ” constraint. That is, we will subtract a nonnegative *surplus variable*, denoted by s_1 , from the left-hand side of that constraint and let the resulting expression equal to the right-hand-side constant. This yields

$$4x_1 + 3x_2 - s_1 = 6.$$

Next, observe that none of the three variables on the left-hand side of this new equation can serve as a candidate basic variable. Therefore, similar to what we did in equation (1), we will further revise this equation by introducing another artificial variable, which we denote by A_2 . This results in

$$4x_1 + 3x_2 - s_1 + A_2 = 6.$$

In addition, we will also introduce a new term MA_2 into the objective function.

Finally, since the last constraint is of the “ \leq ” type, we simply add a slack variable, denoted by s_2 , to its left-hand side to convert it into an equality.

In summary, we have converted the given linear program into the following form:

$$\begin{array}{ll} \text{Minimize} & 4x_1 + x_2 + MA_1 + MA_2 \\ \text{Subject to:} & \\ & 3x_1 + x_2 + A_1 = 3 \quad (1) \\ & 4x_1 + 3x_2 - s_1 + A_2 = 6 \quad (2) \\ & x_1 + 2x_2 + s_2 = 3 \quad (3) \\ & x_1, x_2, A_1, s_1, A_2, s_2 \geq 0. \end{array}$$

At this point, the objective function is still not in conformance with the standard form. Following what we did in our first example, we now define a new variable z to serve as our objective function, which is to be minimized; and we will introduce a zeroth constraint, namely

$$z - 4x_1 - x_2 - MA_1 - MA_2 = 0,$$

into the constraint set. Furthermore, notice that the artificial variables A_1 and A_2 , which are targeted to serve as basic variables in equations (1) and (2), also participate in this new constraint. Since this is not allowed in the standard form, we will have to eliminate them. This is done in two steps. First, we multiply equation (1) by M and add the outcome into equation (0); this eliminates MA_1 . Next, we multiply equation (2) by M and add the outcome into the equation obtained in the first step. These two steps result in the new equation (0) below.

$$z + (7M - 4)x_1 + (4M - 1)x_2 - Ms_1 = 9M.$$

With these further revisions, we finally arrive at

$$\begin{array}{l}
 \text{Minimize } z \\
 \text{Subject to:} \\
 z + (7M - 4)x_1 + (4M - 1)x_2 - Ms_1 = 9M \quad (0) \\
 3x_1 + x_2 + A_1 = 3 \quad (1) \\
 4x_1 + 3x_2 - s_1 + A_2 = 6 \quad (2) \\
 x_1 + 2x_2 + s_2 = 3 \quad (3) \\
 x_1, x_2, A_1, s_1, A_2, s_2 \geq 0,
 \end{array}$$

which is now ready for the Simplex algorithm.

In tabular form, the above problem becomes:

| Basic Variable | z | x_1 | x_2 | A_1 | s_1 | A_2 | s_2 | |
|----------------|-----|----------|----------|-------|-------|-------|-------|------|
| | 1 | $7M - 4$ | $4M - 1$ | 0 | $-M$ | 0 | 0 | $9M$ |
| A_1 | 0 | 3 | 1 | 1 | 0 | 0 | 0 | 3 |
| A_2 | 0 | 4 | 3 | 0 | -1 | 1 | 0 | 6 |
| s_2 | 0 | 1 | 2 | 0 | 0 | 0 | 1 | 3 |

Notice that the introduction of the artificial variables allows us to conveniently declare the basis associated with this tableau as A_1 , A_2 , and s_2 (listed on the left margin). Therefore, the initial basic feasible solution is $(x_1, x_2, A_1, s_1, A_2, s_2) = (0, 0, 3, 0, 6, 3)$, with a corresponding objective-function value of $9M$. Since M is “big,” the coefficients of x_1 and x_2 in R_0 , namely $7M - 4$ and $4M - 1$, are both positive, implying that the current solution is not optimal. Moreover, a big M also implies that $7M - 4$ is strictly larger than $4M - 1$. Hence, x_1 is the entering variable, and the x_1 -column is the pivot column.

A comparison of the three ratios $3/3$, $6/4$, and $3/1$ shows that R_1 is the pivot row, and hence A_1 is the leaving variable. This also identifies the entry “3,” located at the intersection of the pivot column and the pivot row, as the pivot element.

We now execute a pivot. An examination of the x_1 -column shows that we need to go through the following row operations: $[-(7M - 4)/3] \times R_1 + R_0$, $(1/3) \times R_1$, $(-4/3) \times R_1 + R_2$, and $(-1/3) \times R_1 + R_3$. These four sets of operations will produce new versions of equations (0), (1), (2), and (3), respectively; and these equations constitute the new tableau below.

| Basic Variable | z | x_1 | x_2 | A_1 | s_1 | A_2 | s_2 | |
|----------------|-----|-------|--------------|---------------|-------|-------|-------|----------|
| | 1 | 0 | $(5M + 1)/3$ | $-(7M - 4)/3$ | $-M$ | 0 | 0 | $2M + 4$ |
| x_1 | 0 | 1 | $1/3$ | $1/3$ | 0 | 0 | 0 | 1 |
| A_2 | 0 | 0 | $5/3$ | $-4/3$ | -1 | 1 | 0 | 2 |
| s_2 | 0 | 0 | $5/3$ | $-1/3$ | 0 | 0 | 1 | 2 |

Since the coefficient of x_2 in R_0 is positive, this tableau is not optimal, and hence more iterations are necessary. We will not complete the remaining iterations, since they are now straightforward.

Several comments are in order.

At the end of the above iteration, the artificial variable A_1 is no longer in the basis; that is, its value has been driven to zero by the algorithm. Since artificial variables are not part of the original problem, they can be discarded from further consideration as soon as they leave the basis. This serves to reduce the amount of computation. In fact, we can do even better by discarding the leaving artificial variable (A_1 , in this example) at the start (as opposed to the end) of the pivot. One should be careful not to discard any of the original variables, however.

Any solution that contains a positive value for any of the artificial variable is not feasible to the original problem. For example, the basic feasible solution associated with the above tableau is $(x_1, x_2, s_1, A_2, s_2) = (1, 0, 0, 2, 2)$, where we have removed A_1 . Since $A_2 = 2$ is positive, the current solution is not feasible to the original problem. This will continue to be the case until all artificial variables are driven out.

In general, it is possible for a given linear program not to have any feasible solution. In such a case, the Simplex algorithm will not be able to succeed in driving out all of the artificial variables. Thus, if the algorithm terminates with an optimal solution that has at least one of the artificial variables being positive, then the original problem is *infeasible*.

Throughout our computation, we did not assign a specific value for M . We simply treated it as a large number *operationally*. This means that whenever M is compared against another number, we will let M be the larger of the two. This seems convenient, but can pose a challenge in a computer implementation of the algorithm.

If our objective is to *maximize* $4x_1 + x_2$, then, instead of introducing MA_1 and MA_2 into the objective function, we should introduce $-MA_1$ and $-MA_2$.