# Linear Programming for Optimization

Mark A. Schulze, Ph.D.
Perceptive Scientific Instruments, Inc.

## 1. Introduction

### 1.1 Definition

*Linear programming* is the name of a branch of applied mathematics that deals with solving optimization problems of a particular form. Linear programming problems consist of a linear *cost function* (consisting of a certain number of variables) which is to be minimized or maximized subject to a certain number of *constraints*. The constraints are linear inequalities of the variables used in the cost function. The cost function is also sometimes called the *objective function*. Linear programming is closely related to linear algebra; the most noticeable difference is that linear programming often uses inequalities in the problem statement rather than equalities.

### 1.2 History

Linear programming is a relatively young mathematical discipline, dating from the invention of the *simplex method* by G. B. Dantzig in 1947. Historically, development in linear programming is driven by its applications in economics and management. Dantzig initially developed the simplex method to solve U.S. Air Force planning problems, and planning and scheduling problems still dominate the applications of linear programming. One reason that linear programming is a relatively new field is that only the smallest linear programming problems can be solved without a computer.

### 1.3 Example

(Adapted from [1].) Linear programming problems arise naturally in production planning. Suppose a particular Ford plant can build Escorts at the rate of one per minute, Explorer at the rate of one every 2 minutes, and Lincoln Navigators at the rate of one every 3 minutes. The vehicles get 25, 15, and 10 miles per gallon, respectively, and Congress mandates that the average fuel economy of vehicles produced be at least 18 miles per gallon. Ford loses $1000 on each Escort, but makes a profit of $5000 on each Explorer and $15,000 on each Navigator. What is the maximum profit this Ford plant can make in one 8-hour day?

The *cost function* is the profit Ford can make by building $x$ Escorts, $y$ Explorers, and $z$ Navigators, and we want to maximize it:

$$-1000x + 5000y + 15000z \tag{1.3.1}$$

The *constraints* arise from the production times and Congressional mandate on fuel economy. There are 480 minutes in an 8-hour day, and so the production times for the vehicles lead to the following limit:

$$x + 2y + 3z \leq 480 \tag{1.3.2}$$

The average fuel economy restriction can be written:

$$25x + 15y + 10z \geq 18(x + y + z) \tag{1.3.3}$$

which simplifies to:

$$7x - 3y - 8z \geq 0 \tag{1.3.4}$$

There is an additional implicit constraint that the variables are all non-negative: $x, y, z \geq 0$.

This production planning problem can now be written succinctly as:

$$\begin{aligned}\text{Maximize}\quad & -1000x + 5000y + 15000z \\ \text{subject to}\quad & x + 2y + 3z \leq 480 \\ & 7x - 3y - 8z \geq 0 \\ & x, y, z \geq 0\end{aligned} \tag{1.3.5}$$

The solution to this problem is $x$=132.41, $y$=0, and $z$=115.86, yielding a cost function value of 1,605,517.24. Note that for some problems, non-integer values of the variables may not be desired. Solving a linear programming problem for integer values of the variables only is called *integer programming* and is a significantly more difficult problem. The solution to an integer programming problem is not necessarily close to the solution of the same problem solved without the integer constraint. In this example, the optimal solution if $x$, $y$, and $z$ are constrained to be integers is $x$=132, $y$=1, and $z$=115 with a resulting cost function value (profit) of $1,598,000.

## 2. Terminology

A linear programming problem is said to be in “standard form” when it is written as:

$$\begin{aligned}\text{Maximize}\quad & \sum_{j=1}^{n} c_j x_j \\ \text{subject to}\quad & \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1 \ldots m \\ & x_j \geq 0, \qquad j = 1 \ldots n\end{aligned} \tag{2.1}$$

The problem has $m$ variables and $n$ constraints. It may be written using vector terminology as:

$$\begin{aligned}\text{Maximize}\quad & \mathbf{c}^{\mathbf{T}}\mathbf{x} \\ \text{subject to}\quad & \mathbf{Ax} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}\end{aligned} \tag{2.2}$$

Note that a problem where we would like to minimize the cost function instead of maximize it may be rewritten in standard form by negating the cost coefficients $c_j$ ($\mathbf{c}^{\mathbf{T}}$).

Any vector $x$ satisfying the constraints of the linear programming problem is called a *feasible solution* of the problem. Every linear programming problem falls into one of three categories:

1. *Infeasible*. A linear programming problem is *infeasible* if a feasible solution to the problem does not exist; that is, there is no vector $x$ for which all the constraints of the problem are satisfied.
2. *Unbounded*. A linear programming problem is *unbounded* if the constraints do not sufficiently restrain the cost function so that for any given feasible solution, another feasible solution can be found that makes a further improvement to the cost function.
3. *Has an optimal solution*. Linear programming problems that are not infeasible or unbounded have an optimal solution; that is, the cost function has a unique minimum (or maximum) cost function value. This does not mean that the values of the variables that yield that optimal solution are unique, however.

The basic algorithm most often used to solve linear programming problems is called the *simplex method*. Over the past 50 years, it has been developed to the point that good computer programs using the simplex method and its relatives (the *revised simplex method* and the *network simplex method*) can solve virtually any bounded, feasible linear programming problem of reasonable size in a reasonable amount of time. Only in the past ten years have other methods of solving linear programming problems (so-called *interior point methods*) developed to the point where they can be used to solve practical problems.

# 3. The Simplex Method

## 3.1 How It Works

The simplex method has two basic steps, often called "phases." The first phase is to find a feasible solution to the problem. For small problems, or larger problems of certain forms, this is not at all difficult. Often, a trivial solution such as $x = 0$ is a feasible solution, as in the production planning problem described earlier. We will omit the details of solving the first phase to find a feasible solution for now.

After a feasible solution to the problem is found, the simplex method works by iteratively improving the value of the cost function. This is accomplished by finding a variable in the problem that can be increased, at the expense of decreasing another variable, in such a way as to effect an overall improvement in the cost function. This can be visualized graphically as moving along the edges of a feasible set from corner to corner. A two-dimensional example follows.

## 3.2 Geometric Interpretation of the Simplex Method

Consider the linear programming problem:

$$\begin{array}{ll} \text{Maximize} & x + y \\ \text{subject to} & 2x + y \le 14 \\ & -x + 2y \le 8 \\ & 2x - y \le 10 \\ & x \ge 0,\ y \ge 0 \end{array} \tag{3.2.1}$$

The *feasible set* of this problem can be graphed in two dimensions as shown in Figure 1. The non-zero constraints $x \ge 0$ and $y \ge 0$ confine the feasible set to the first quadrant. The other three constraints are lines in the $x$-$y$ plane, as shown. The cost function, $x + y$, can be represented as a line of slope –1 *with any intercept*. The value of the intercept of the cost function line is the value of the cost function for any solution that lies along that line. The heavy line in Figure 1 represents the optimal solution of the problem, since it is the line with slope –1 with the maximum intercept (10) that intersects the feasible set. The value of the cost function for this optimal solution is 10, and the cost function line $x + y = 10$ has exactly one point in the feasible set, $x = 4$, $y = 6$.

The simplex method works by finding a feasible solution, and then moving from that point to any vertex of the feasible set that improves the cost function. Eventually a corner is reached from which any movement does not improve the cost function. This is the optimal solution. In this example, $x = 0$ and $y = 0$ is a trivial feasible solution, and has a cost function value of 0. This is vertex A in Figure 1. From this point, we can either move to point B or point E. Point E (0, 4) increases the cost function to 4, while point B (5, 0) increases it to 5. Since point B gives us more improvement, we will select it as our first iteration (although we could just as well have chosen E and in fact would reach the optimal solution more quickly if we did). The value of $x$ increases from 0 to 5, while $y$ remains 0.

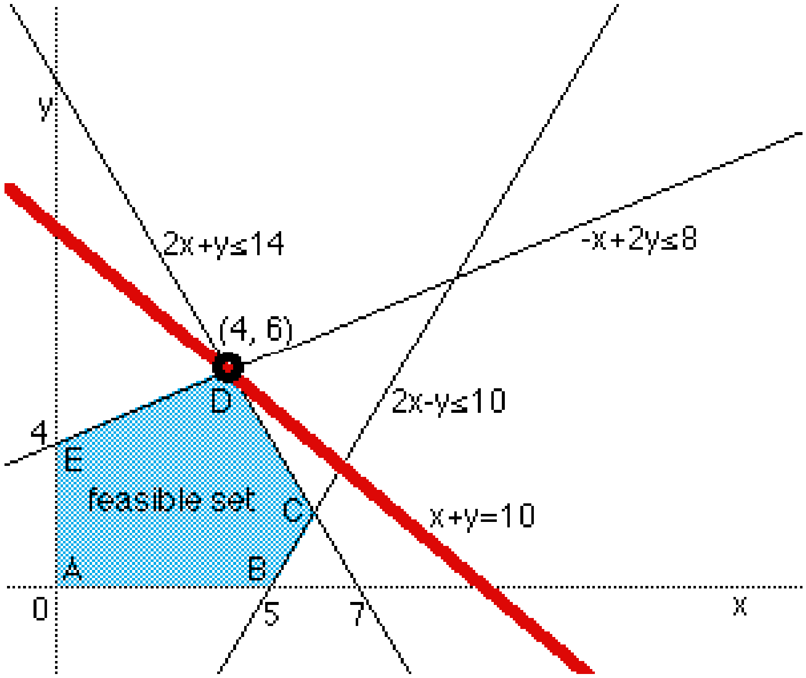

**Figure 1.** Graphical solution of a linear programming problem.

From point B, we check whether a move to point C is advantageous. (We know that moving back to A is not.) Point C (6, 2) has a cost function value of 8, which is an improvement. Therefore we increase $x$ from 5 to 6, which means that because of the constraints $2x - y \le 10$ and $-x + 2y \le 8$, we must also increase $y$ from 0 to 2. From point C, we now check whether a move to point D improves the situation. The cost function at D (4, 6) is 10, so we accept the move and increase $y$ from 2 to 6. This means that $x$ must decrease from 6 to 4 because of the constraints. Now, since a move to either point E (cost function value of 4) or point C (cost function value of 8) decreases the cost function, we know that point D is the optimal solution to this problem.

### 3.3 Algebraic Solution using the Simplex Method

The same problem illustrated graphically above can be solved using only algebraic manipulations. First, we rewrite equation (3.2.1) adding *slack variables* to convert the inequality constraints into equalities:

$$\begin{array}{lllllllll}
\text{Maximize} & x & + & y & & & & = & z \\
\text{subject to} & 2x & + & y & + & r & & = & 14 \\
 & -x & + & 2y & & + & s & = & 8 \\
 & 2x & - & y & & & + \; t & = & 10 \\
 & \multicolumn{7}{l}{x, y, r, s, t \ge 0} & 
\end{array} \qquad (3.3.1)$$

The simplex method starts by fixing enough of the variables at 0 (their lower bound) and “remove” them from the problem so that the system $\mathbf{Ax} = \mathbf{b}$ is square. In this case, with the slack variables added there are 5 variables and 3 constraints, so we need to set two variables to 0 and remove their coefficients from the problem to make the matrix $\mathbf{A}$ into a 3x3 matrix. The solution of the system in this instance represents one of the vertices of the feasible set, as illustrated geometrically in Figure 1. An obvious feasible solution to this problem is $x = 0$, $y = 0$. These are the two variables set to zero and “removed” from the problem. Such variables are called *non-basic* variables. The solution to this system is then

$r = 14$, $s = 8$, $t = 10$. These three variables are called the *basic variables* for this solution.

Now, we rewrite the problem with the constraints as equations for the basic variables $r$, $s$, and $t$:

$$\begin{array}{lcl} z &=& \phantom{14 -{}} x + y \\ r &=& 14 - 2x - y \\ s &=& 8 + x - 2y \\ t &=& 10 - 2x + y \\ \multicolumn{3}{l}{x, y, r, s, t \geq 0} \end{array} \tag{3.3.2}$$

To begin iterating the simplex method, we look for a way to increase the cost function, $z$. Looking at the non-basic variables $x$ and $y$, we try to increase one of them while holding the other at zero. However, the amount we can increase $x$ is limited by the non-negativity constraints on the basic variables. Looking at the reformulated constraints shown above, we see that for the first constraint for $r \geq 0$, we must have $x \leq 7$. The second constraint does not limit $x$ since an increase in $x$ increases $s$, while the third constraint imposes the restriction that $x \leq 5$. Therefore, we choose to increase $x$ from zero to 5, and recalculate the values of $r$, $s$, and $t$ from the above equations: $r = 4$, $s = 13$, and $t = 0$. Since $x$ goes from zero to non-zero, it is said to *enter the basis* and is called the *entering variable* for this iteration. Since $t$ goes from non-zero to zero, it is said to *leave the basis* and is called the *leaving variable*. Our basis now consists of the variables $r$, $s$, and $x$ and the variables $t$ and $y$ are non-basic and therefore zero. Our objective function $z$ now has a value of 5.

We now rewrite the entire problem so that the objective function and constraints are expressed only as functions of the non-basic variables $t$ and $y$, by rearranging the constraint equation for $t$ from above to be $x = 5 - 0.5t + 0.5y$ and substituting into the other equations:

$$\begin{array}{lcl} z &=& 5 - 0.5t + 1.5y \\ r &=& 4 + t - 2y \\ s &=& 13 - 0.5t - 1.5y \\ x &=& 5 - 0.5t + 0.5y \\ \multicolumn{3}{l}{x, y, r, s, t \geq 0} \end{array} \tag{3.3.3}$$

Now we iterate the simplex method again. To increase $z$, we must increase $y$ this time. The constraint on $r$ limits $y$ to be less than 2, the constraint on $s$ limits $y$ to be less than 8.67, and the constraint on $x$ does not limit $y$. Therefore, we choose $y$ as the entering variable with a value of 2, and $r$ to be the leaving variable. The equation for $r$ rearranges to $y = 2 –0.5r + 0.5t$. Substituting in yields:

$$\begin{array}{lcl} z &=& 8 - 0.75r + 0.25t \\ y &=& 2 - 0.5r + 0.5t. \\ s &=& 10 + 0.75r - 1.25t \\ x &=& 6 - 0.25r - 0.25t \\ \multicolumn{3}{l}{x, y, r, s, t \geq 0} \end{array} \tag{3.3.4}$$

Note that when the problem is in this form, we can read the current values of the basic variables and the objective function from the constants just to the right of the equals signs. The variables on the right of the equations are the non-basic variables and all have value zero. To increase $z$ for the next iteration, we have no other choice than to increase $t$. The constraint for $s$ limits $t$ to 8, while the constraint for $x$ limits $t$ to 24. Therefore, $t$ enters with a value of 8 and $s$ leaves the basis. The rearranged equation is $t = 8 + 0.6r - 0.8s$. Substituting, we have:

$$\begin{aligned} z &= 10 - 0.6r - 0.2s \\ y &= 6 - 0.2r - 0.4s \\ t &= 8 + 0.6r - 0.8s \\ x &= 4 - 0.4r - 0.4s \\ &x, y, r, s, t \geq 0 \end{aligned} \tag{3.3.5}$$

Now the objective function $z$ has value 10. Examining the equation for $z$, we see that we cannot raise either $r$ or $s$ above zero without decreasing $z$. This means there are no more advantageous changes we can make to the solution, so it must be optimal. Therefore, the solution to this linear programming problem is $x = 4$, $y = 6$, with an objective function value of 10.

## 4. The Revised Simplex Method

The revised simplex method is the name of an implementation of the simplex method that uses a slightly different method of updating the problem at each iteration: instead of updating the problem using the results of the previous iteration, it uses the original data. Each iteration is still the same, but the revised simplex method is more computationally efficient, especially for large and sparse problems. It is described in detail in the book by Chvátal [2].

## 5. Pitfalls

There are three major pitfalls that present themselves when solving linear programming problems by the simplex method. They are:

1. *Initialization.* This is the problem of finding an initial feasible solution with which to start the simplex method.
2. *Iteration.* There may be difficulties in choosing an entering or leaving variable.
3. *Termination.* This is the problem of ensuring that the simplex method terminates and does not merely continue through an endless sequence of iterations without ever reaching an optimal solution.

### 5.1 Initialization

For many linear programming problems, finding an initial feasible solution is trivial; often, an all-zero solution is feasible and may be used as a starting point for the simplex method. If an initial feasible solution is not readily available, one can be found by solving an auxiliary linear programming problem formulated by subtracting additional variables from the original problem's constraints and changing the cost function to minimize the sum of these new variables. The auxiliary problem is solved, and when all of the added variables are zero, the values of the original variables represent an initial feasible solution. The original linear programming problem is then solved using this initial solution. If the solution to the auxiliary problem does not yield a solution where the new variables are all zero, then a feasible solution to the original problem does not exist. This strategy is known as the *two-phase simplex method*. The first phase sets up and solves the auxiliary problem, and the second phase solves the problem itself.

### 5.2 Iteration

There is usually no difficulty in selecting an entering or leaving variable for each iteration of the simplex method. There is often more than one candidate for the entering or leaving variable, but any variable which satisfies the requirements for entering or leaving the basis may be chosen subject to termination considerations (discussed in the next section). When choosing a leaving variable, the only pitfall that endangers finding a solution is *unboundedness*, when the entering variable is not constrained in value by any of the candidate leaving variables. This means there is no candidate for the leaving variable (which is meant to be the basic variable which imposed the most stringent bound on the increase of the entering variable). The problem in this case is unbounded and has no optimal solution.

Another difficulty that may be encountered during iteration is the phenomenon of *degeneracy*. Degeneracy is not harmful to the simplex method, but it does have some annoying consequences; notably, it can cause the simplex method to go through many successive iterations that do not change the value of the solution (and therefore do not change the value of the cost function either). Degeneracy arises from solutions where one or more basic variables are zero, called *degenerate solutions*. When a basic variable with zero value is replaced in the basis with an entering variable that is limited to zero value, the solution does not change value at all (although the variables in the basis do change) and the iteration just performed is called a *degenerate iteration*. This is illustrated by the following example.

$$\begin{array}{lllllll}
z & = & 4 & + 2x & - y & - 4s \\
r & = & 0.5 & & & - 0.5s \\
t & = & & - 2x & + 4y & + 3s \\
u & = & & x & - 3y & + 2s \\
\multicolumn{6}{l}{x, y, r, s, t, u \geq 0}
\end{array} \tag{5.2.1}$$

Iterating this problem and choosing $x$ as the entering variable and $t$ as the leaving variable, we get:

$$\begin{array}{lllllll}
z & = & 4 & + 3y & - s & - t \\
r & = & 0.5 & & - 0.5s & \\
x & = & & 2y & + 1.5s & - 0.5t \\
u & = & & - y & + 3.5s & - 0.5t \\
\multicolumn{6}{l}{x, y, r, s, t, u \geq 0}
\end{array} \tag{5.2.2}$$

Note that for both solutions, the objective value $z$ is 4 and the variable $r = 0.5$ while all other variables are zero. This is a degenerate iteration. Eventually, the simplex method becomes “unstuck” and the iterations are non-degenerate. In the above example, the next iteration is also degenerate, but the iteration after that is not and yields the optimal solution.

Although degeneracy itself does not endanger the simplex method, it does have consequences for termination that are discussed in the next section. Virtually all practical linear programming problems are degenerate.

### 5.3 Termination

The simplex method terminates by finding that the problem is infeasible or unbounded, or by finding the optimal solution. The only other possibility is that the simplex method *cycles*; that is, it goes through an endless, repeating sequence of non-optimal solutions (which must all have the same cost function value). Note that cycling can only occur in the presence of degeneracy, since each iteration of a cycle must be degenerate. There are several strategies for selecting either the entering or leaving variable (or both) that ensure that cycling does not occur. The simplest of these is the “smallest-subscript” rule, which states that ties in the choice of entering and leaving variables are always broken in favor of the variable with the smallest subscript.

## 6. Speed

Studies on practical linear programming problems of various sizes have shown that the simplex method typically terminates after between $1.5m$ and $3m$ iterations, where $m$ is the number of constraints in the problem. The general consensus is that the number of iterations increases linearly with the number of constraints but only logarithmically with the number of variables. Theoretical studies of the efficiency of the simplex method are much less satisfactory, however. The worst case of the simplex method requires $2^n$-1 iterations, where *n* is the number of variables. However, problems where this has been demonstrated are considered pathologic, and a different choice of entering and leaving variables reduces the number of iterations from $2^n$-1 to one! Therefore, the most practical way to ensure termination in a reasonable number of iterations is to pay careful attention to rules for selecting the entering and leaving variables. Theoretically satisfactory algorithms for solving linear programming problems are now available (known generally as *interior point methods*) but they are not useful for solving practical problems. The simplex method, despite its theoretical shortcomings, is still the method of choice.

## 7. Duality

Every linear programming problem where we seek to maximize the objective function gives rise to a related problem, called the *dual problem*, where we seek to minimize the objective function. The two problems interact in an interesting way: every feasible solution to one problem gives rise to a bound on the optimal solution in the other problem. If one problem has an optimal solution, so does the other problem and the two objective function values are the same. The equations below show a problem in standard form with *n* variables and *m* constraints on the left, and its corresponding dual problem on the right.

$$\begin{array}{ccc} \max \mathbf{c}^{\mathbf{T}}\mathbf{x} & & \min \mathbf{b}^{\mathbf{T}}\mathbf{y} \\ \text{s.t. } \mathbf{A}\mathbf{x} \le \mathbf{b} & \Longleftrightarrow & \text{s.t. } \mathbf{A}^{\mathbf{T}}\mathbf{y} \ge \mathbf{c} \\ \mathbf{x} \ge \mathbf{0} & & \mathbf{y} \ge \mathbf{0} \\ \begin{pmatrix} n \text{ variables} \\ m \text{ constraints} \end{pmatrix} & & \begin{pmatrix} m \text{ variables} \\ n \text{ constraints} \end{pmatrix} \end{array} \tag{7.1}$$

If the original or *primal* problem has the optimal solution $\mathbf{x}^*$, its dual problem has an optimal solution $\mathbf{y}^*$ and $\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*$. If the primal problem is infeasible or unbounded, then the dual problem is infeasible or unbounded.

Duality is mostly of theoretical importance, although some linear programming problems may be solved much more easily by converting them to their dual form.

## 8. Conclusions

Linear programming is an important branch of applied mathematics that solves a wide variety of optimization problems. It is widely used in production planning and scheduling problems. Many recent advances in the field have come from the airline industry where aircraft and crew scheduling have been great improved by the use of linear programming. It has also been used to solve a variety of assignment problems, such as the karyotyping problem where 46 chromosomes are assigned to 24 classes. Although the revised simplex method is not theoretically satisfactory from a computational point of view, it is by far the most widely used method to solve linear programming problems and only rarely are its limitations encountered in practical applications. The biggest advantage of linear programming as an optimization method is that it always achieves the optimal solution if one exists.

## 9. References

[1] G. Strang, *Linear Algebra and its Applications*, 3rd ed. San Diego: Harcourt Brace Jovanovich, 1988.

[2] V. Chvátal, *Linear Programming*. New York: W. H. Freeman, 1983.