Chapter 2: Introduction to Linear Programming

You may recall *unconstrained optimization* from your high school years: the idea is to find the highest point (or perhaps the lowest point) on an *objective function* (see Figure 2.1). For optimization to be required, there must be more than one solution available. In Figure 2.1, any point on the function is a solution, and because the single variable is real-valued, there are an infinite number of solutions. Some kind of optimization process is then required in order to choose the very *best* solution from among those available. What is meant by *best* depends on the problem at hand: it might mean the solution that provides the most profit, or that consumes the least of some limited resource, e.g. area in computer chip design, or fuel in delivery route design.

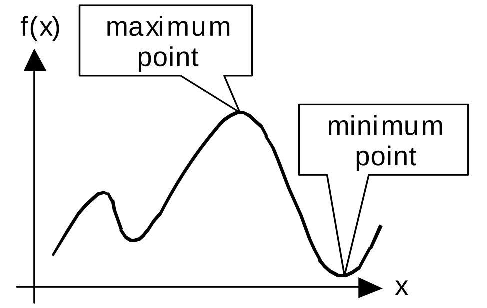

Figure 2.1: Simple unconstrained optimization.

Linear programming (LP) is the most commonly applied form of *constrained optimization*. Constrained optimization is much harder than unconstrained optimization: you still have to find the best point of the function, but now you also have to respect various constraints while doing so. For example, you must guarantee that the optimum point does not have a value above or below a prespecified limit when substituted into a given constraint function. The constraints usually relate to limited resources. The simple methods you used in high school to find peaks and valleys won't work anymore: now the best solution (the *optimum point*) may not occur at the top of a peak or at the bottom of a valley. The best solution might occur half way up a peak when a constraint prohibits movement farther up.

The main elements of any constrained optimization problem are:

- **Variables** (also called *decision variables*). The values of the variables are not known when you start the problem. The variables usually represent things that you can adjust or control, for example the rate at which to manufacture items. The goal is to find values of the variables that provide the best value of the objective function.
- **Objective function.** This is a mathematical expression that combines the variables to express your goal. It may represent profit, for example. You will be required to either maximize or minimize the objective function.
- **Constraints.** These are mathematical expressions that combine the variables to express limits on the possible solutions. For example, they may express the idea that the number of workers available to operate a particular machine is limited, or that only a certain amount of steel is available per day.

- **Variable bounds.** Only rarely are the variables in an optimization problem permitted to take on any value from minus infinity to plus infinity. Instead, the variables usually have bounds. For example, zero and 1000 might bound the production rate of widgets on a particular machine.

In *linear programming (LP)*, all of the mathematical expressions for the objective function and the constraints are linear. The *programming* in linear programming is an archaic use of the word "programming" to mean "planning". So you might think of linear programming as "planning with linear models". You might imagine that the restriction to linear models severely limits your ability to model real-world problems, but this isn't so. An amazing range of problems can be modeled using linear programming, everything from airline scheduling to least-cost petroleum processing and distribution. LP is very widely used. For example, IBM estimated that in 1970, 25% of all scientific computation was devoted to linear programming.

Linear programming is by far the most widely used method of constrained optimization. The largest optimization problems in the world are LPs having millions of variables and hundreds of thousands of constraints. With recent advances in both solution algorithms and computer power, these large problems can be solved in practical amounts of time.

Of course, there are also many problems for which LP is *not* appropriate, and part of the job for this textbook is to help you decide when to use LP and the other techniques covered here, and when *not* to use them.

A Prototype Example: The Acme Bicycle Company

It's time now to introduce you to a small example that we will be visiting numerous times throughout the book: the Acme Bicycle Company (ABC). The name follows the time-honored tradition in optimization and operations research texts of inventing bogus companies such as the "Wyndor Glass Company" (which, oddly, makes glass windows and doors), or the "Nori and Leets" Iron and Steel Company.

The Acme Bicycle Company produces two kinds of bicycles by hand: mountain bikes and street racers. Acme wishes to determine the rate at which each type of bicycle should be produced in order to maximize the profits on the sales of the bicycles. Acme assumes that it can sell all of the bicycles produced.

The physical data on the production process is available from the company engineer. A different team produces each kind of bicycle, and each team has a different maximum production rate: 2 mountain bikes per day and 3 racers per day, respectively. Producing a bicycle of either type requires the same amount of time on the metal finishing machine (a production bottleneck), and this machine can process at most a total of 4 bicycles per day, of either type. The company accountant estimates that mountain bikes are currently generating a profit of around $15 per bicycle, and racers a profit of around $10 per bicycle.

This problem is small enough to solve without using LP, just the straightforward application of common sense. To maximize profit, start by producing the maximum number possible of the higher-profit mountain bikes and use any leftover production capacity to produce racers for additional profit. This would mean a production rate of 2 mountain bikes per day, which is the limit of the mountain bike team, yet leaves spare capacity on the metal finishing machine. This remaining capacity can be used to produce 2 racers per day, which is below the capacity of the racer production team. The total profit would then be $2\times\$15+2\times\$10=\$50$ per day.

We will be formulating and solving the Acme problem as a linear program, but there is an important lesson here: the results returned by a mathematical program should always be compared to the results predicted by common sense. If the two are in conflict, investigate. You will discover either a modeling or data error, or will learn more about the underlying process, thereby sharpening your intuition. The LP solution of the Acme problem had better turn up a daily profit of at least $50!

The first step in formulating the ABC problem as a linear program is to identify the variables. These are the items whose values you can set or otherwise control. The Acme variables are the production rates of mountain bikes (call this x_1) and racers (call this x_2). Note any bounds on the variables:

- variable nonnegativity: $x_1 \geq 0$, $x_2 \geq 0$

Using these variables, next write the objective function:

- maximize daily profit: maximize $Z=15x_1+10x_2$ (in $ per day)

It's a convention to represent the value of the objective function by Z.

Use the variables to write the constraints as well:

- mountain bike production limit: $x_1 \leq 2$ (in bikes per day)
- racer production limit: $x_2 \leq 3$ (in bikes per day)
- metal finishing machine production limit: $x_1+x_2 \leq 4$ (in bikes per day)

The first two constraints are normally considered variable bounds, but we will treat them as general constraints for now.

Note that it is customary to write LP constraints with all of the variables on the left hand side of the relationship, and the constant value on the right hand side (*rhs*). As in all LPs, all of the relationships (constraints and objective) are linear.

On a superficial level, you now have all that you need to apply a linear programming solver: a set of linear constraints (≥ type, ≤ type, or = type) and a linear objective, and some variable bounds. LP solvers are not hard to find: several are available for free via the World Wide Web, and an LP solver is even included in the Microsoft Excel spreadsheet software for PCs. For this reason, many people with only very limited understanding of LP are formulating and solving them. The difficulty arises when unexpected results are returned: then a deeper understanding of LP is essential.

Cornerpoints are Important

Because there are only two variables in the Acme Bicycle Company formulation, the problem can be sketched on the plane, as shown in Figure 2.2. The limiting value of each of the constraints is shown as a line. Each constraint eliminates part of the plane. For example, the vertical line labeled "x_1=2" is the limiting value of the inequality $x_1 \leq 2$ and all points to the right of the line violate the constraint (i.e. are *infeasible*). The areas eliminated by the constraints are shaded. The unshaded area represents points that are not eliminated by any constraint, and is called the *feasible region*. Points in the feasible region (which includes the bordering lines) satisfy *all* of the constraints.

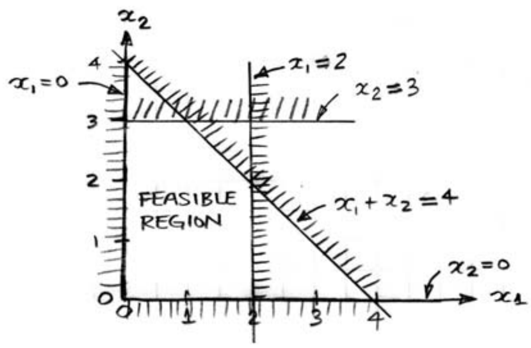

Figure 2.2: The feasible region for the Acme Bicycle Company problem.

The linear programming problem in Figure 2.2 is to find the point in the feasible region that gives the largest value of the objective function. One (silly) way to do this is to randomly choose feasible points and to calculate the value of the objective function at those points, keeping the point that gives the best value of the objective. Because there are an infinite number of points in the feasible region, this is not very effective! There is no guarantee that the best point will be found, or even that an objective function value that is close to the best possible value will be found. We need a more efficient way of searching the feasible region.

We can develop a more efficient search technique based on a couple of simple observations. First, let's plot points in the 2-dimensional $x_1 \times x_2$ plane that have the same value of the objective function. As shown in Figure 2.3, points having the same value of Z (value of the objective function) form a line. This is easy to understand if we replace Z by the specific value that we want to plot, e.g. Z=15x_1+10x_2 becomes the line 15x_1+10x_2=20 plotted in Figure 2.3.

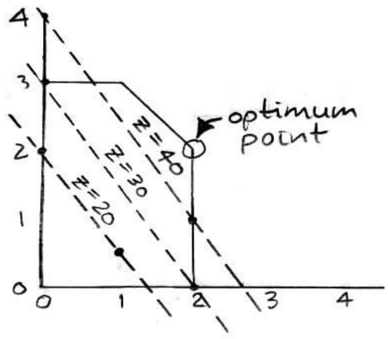

Figure 2.3: Some constant-profit lines for the ABC problem.

Figure 2.3 also shows that all of the constant-profit lines are parallel. This is because all of the constant-profit line equations differ only by the selected value of Z. If you were to calculate the slope of any constant-profit line, the Z constant disappears; hence the slopes of all of the constant-profit lines are the same. For the Acme Bicycle Company, all of the constant profit lines have the same slope, given by dx_2/dx_1 = -15/10 = -1.5.

The third important observation is that the value of *Z* is higher for the constant-profit lines towards the upper right in Figure 2.3. We will revisit this property in more detail when we cover nonlinear programming, but for now accept that this is a property of linear functions because they have a constant *gradient*.

Now we can view the linear programming problem in a different way. Picture the objective function as a constant-profit line that is floating upwards from the lower left to the upper right in Figure 2.3, increasing in value as it floats higher. Now the linear programming question is this: what is the last point in the feasible region that the objective function passes through as it floats up to infinity? From Figure 2.3 we see that the last point is (2,2) with *Z*=50. This is the solution to the LP: the feasible point that has the best value of the objective function! Another analogy is to imagine the objective function sinking from infinity in the upper right, decreasing in value until it first bumps into the feasible region. What is the first feasible point that it bumps into (which will give the best value of the objective function)? It is of course (2,2) with *Z*=50.

Here is the final and most important observation. Because lines define the feasible region, all of its external edges (or *faces*) are flat linear surfaces that join together at *cornerpoints*. Again imagine the objective function as a line sinking from infinity: where will it first bump into this feasible region defined by flat faces and cornerpoints? As you can see by inspection, the linear objective function will always first bump into the feasible region *at a cornerpoint!* This is because a cornerpoint "sticks out farthest" in the direction of the sinking objective function line, hence first contact will be at a cornerpoint, and this will define the optimum point.

In some cases, the objective function has exactly the same slope as a face of the feasible region and first contact is between the objective function and this face, as in Figure 2.4. This means that *all* of the points on that face have the same value of the objective function, and all are optimum: i.e. there are *multiple optima*. Notice, though, that if a face has first contact, then the cornerpoints of the face also have first contact. The important idea is that first contact between the objective function and the feasible region always involves at least one cornerpoint. Hence, *an optimum solution to the LP is always at a cornerpoint.*

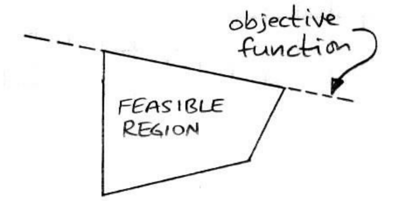

Figure 2.4: The slope of the objective function exactly matches the slope of the face of the feasible region.

This observation drastically simplifies the search for the optimum point: we need to look only at the relatively small number of cornerpoints of the feasible region instead of randomly sampling the infinite number of points on the interior of the feasible region.

This fact underlies the *simplex method* of linear programming, which we shall begin to address in the next chapter. For now just observe in Fig. 2.2 that there are only five *feasible cornerpoints* that need to be visited to find an optimum solution to the Acme Bicycle Company LP.

It is also easy to identify when there are multiple optima just by looking at the feasible cornerpoints. In two dimensions, when two feasible cornerpoints have the same optimum value of the objective function, then all of the points on the line segment joining the two cornerpoints have the same optimum value. It's worth knowing this because one of the intermediate optimum points may be preferable to the cornerpoints for nonquantifiable reasons. It is possible to have three or more feasible cornerpoints with the same optimum value of the objective function in a three dimensional problem. Imagine, for example, that the feasible region is defined by a three-dimensional tetrahedron and that the slope of the objective function plane is exactly equal to the slope of one of the faces of the tetrahedron. Now all three feasible cornerpoints of the triangular face of the tetrahedron, and all of the points on the face of the tetrahedron, will have the same optimum value of the objective function.

The Underlying Assumptions in Linear Programming

The inescapable underlying assumption that is made in modeling the real world via linear programming is that a linear model is suitable. Models constructed solely from linear relationships have certain limitations. The most obvious is that some real-world phenomena are poorly modeled by lines. Nonlinear relationships such as curves or step-functions may be needed instead. If such nonlinear or discontinuous relationships are not adequately approximated by linear relationships, then you must use a technique other than linear programming.

The two linear properties of *additivity* and *proportionality* preclude curves or step-functions. The additivity property prohibits cross-product terms, e.g. $5x_1x_2$, which might represent interaction effects between the variables. For example, Acme may discover that ordering materials for both bicycles together from the same supplier lowers costs, but this effect cannot be modelled using a linear relationship.

The proportionality property requires that the value of each term in the linear function is strictly proportional to the value of the variable in the term. For example, the objective function cost of using a certain variable is always directly proportional to the level of use of the variable: it is not possible to include a start-up cost. In the Acme model, the proportionality assumption is violated if, for example, the production efficiency improves significantly as the rate of production increases.

Linear programming assumes that the variables are real-valued, meaning that they can take on fractional values. In the Acme problem we are trying to determine the *rate* of production, which can take on fractional values, e.g. produce mountain bicycles at the rate of 1.5 per day. Fractional values are not suitable in some problems, such as determining the number of people to staff a set of restaurants or the number of ships to purchase. Where at least one variable is restricted to taking on an integer value, then you must use the methods of *integer programming*, which are covered in a later chapter. For now, note that it is *not* acceptable to treat integer problems as linear problems and then just round the results to the closest integer. This may yield an infeasible solution. The true optimum in an integer programming problem can be very far away from the approximate solution obtained by linear programming.

A weakness common to all of mathematical programming is the assumption that the input data are perfectly accurate. You are assuming that the objective function coefficients (profit per bicycle for Acme), the constraint coefficients, and the constraint right hand sides (e.g. maximum team daily production of mountain bikes) are all correct. In the real world, these numbers are seldom known with accuracy. For example, how does Acme really know how much profit it makes per bicycle of either type? In large companies such a number is generally produced by the Accounting department, which uses data about average amount of material used in each bicycle, average price paid for the materials, average worker wages, yearly depreciation estimates on machines, average selling prices, etc., to *estimate* the "profit per mountain bike sold". The emphasis is on "estimate".

So how useful is the optimum result produced by the mathematical program if the input data is of poor quality? It can be extremely useful, but you have to be careful. First and foremost, don't treat the output result as if it is "the" answer: you might arrive at quite different results just by using slightly different estimates of the input parameters. For example, will you get a different result if the profit per mountain bike is estimated at $14 instead of the present $15? This is where *sensitivity analysis* is applied: using various tests to determine how sensitive the optimum result is to small changes in the values of the input parameters. It turns out that Acme should still make two each of the racers and mountain bikes per day even if the profit per mountain bike is $1 lower than estimated, but of course the total rate of profit generation will be lower. If knowing the total profit generation rate is crucial to Acme, then it is worthwhile to analyze various scenarios of profit per bicycle.

Formulation Practice: the Smalltime Mutual Funds Company

Formulating LPs well takes practice. In a classroom situation you will often know in advance that you are formulating a linear program. In contrast, in the real world you normally *don't* know the type of the problem when you begin studying it, and that makes formulation much more difficult. The only way to improve your skills in formulation is practice, practice, practice. So here's another example. Try to formulate it before looking at the solution as give below.

You are the investments manager for the Smalltime Mutual Funds Company, and are trying to determine how to invest a pool of $14 million released by cashing out some of the stock investments. Table 2.1 summarizes the information that you have about a set of five possible investments.

To be as conservative as possible, you assume that in the event of a loss by the investment, you lose all of your money. This is a fairly serious assumption, since most mutual fund investments are likely to lose some but not all of their value. On the other hand, you also assume that if there is not a loss, then the investment will grow by the growth rate shown.

For policy reasons, there are limits on how you can invest the money. You must allocate at least 35% of the total funds available to the balanced and bond investments. Of all the money put into equity, special equity and foreign investments, at least half must be in the equity investment. Finally, the expected lost capital must be less than 10%. Of course, your overall objective is to maximize the return on the original pool of money.

This is a textbook problem, so the data is stated much more succinctly and clearly than in real world problems, which are plagued by misleading, hidden, and spurious information. Still, extracting an LP formulation from even a textbook word problem can be harder than it seems. You can test yourself by trying to answer the questions posed in the next few paragraphs before reading the answers.

type of investment	annual growth rate	probability of loss
equity	0.15	0.18
special equity	0.21	0.31
balanced	0.11	0.09
foreign	0.19	0.19
bond	0.08	0.03

Table 2.1: Investments available to the Smalltime Mutual Funds Company.

The first thing to do in an LP formulation is to identify the decision variables. Ask yourself what it is that you can control in this problem, what quantities do you need to find values for? *What are the decision variables?*

The most straightforward formulation of this problem chooses variables representing the amount of money put into each investment:

- x_1: millions of dollars put into equity investment,
- x_2: millions of dollars put into special equity investment,
- x_3: millions of dollars put into balanced investment,
- x_4: millions of dollars put into foreign investment,
- x_5: millions of dollars put into bond investment.

It is also possible to formulate this problem using variables representing the *fraction* of the total money to be put into each kind of investment. It is an awkward approach, but the results are the same in the end.

Now that you have selected variables, the second question is: *what is the objective function?* You are to “maximize the return on the money invested”, but what does this mean? Since some money is gained from interest on the investments and some money is lost, let’s say the net return is:

(expected growth from good investments) *minus* (expected losses from bad investments).

If you are not sure if this is what the managers at Smalltime mean by “maximizing the return on money invested”, then *make sure you ask!* Effective mathematical programming is not just about number-crunching, it’s about crunching the *right* numbers.

Assuming we have the correct idea about the objective, let’s now write it out in terms of the decision variables:

maximize $Z = 0.15(1-.18)x_1 + 0.21(1-.31)x_2 + 0.11(1-.09)x_3 + 0.19(1-.19)x_4 +$
$0.08(1-.03)x_5 - 0.18x_1 - 0.31x_2 - 0.09x_3 - 0.19x_4 - 0.03x_5$

The pattern is simple: the first five terms represent the income due to annual growth on the investments that do not lose money, and the second five terms represent the capital losses on the investments that lose money (remember that we assume you also get no interest on losing investments).

Now we add the constraints. Scan the problem description. *Can you identify all of the constraints?* There are five:

1. limit on proportion of total funds put into balanced and bond investments,
2. limit on proportion of funds in the equity, special equity, and foreign investments that goes into equity funds,
3. limit on expected capital losses,

And the two most frequently forgotten by students:

4. limit on total funds available,
5. nonnegativity constraints on the variables.

Now we can write out these constraints.

Limit on proportion of total funds put into balanced and bond investments:

$$(x_3+x_5)/14 \geq 0.35 \Rightarrow x_3+x_5 \geq 4.9$$

Limit on proportion of funds in the equity, special equity, and foreign investments that goes into equity funds:

$$x_1/(x_1+x_2+x_4) \geq 0.5 \Rightarrow -0.5x_1 + 0.5x_2 + 0.5x_4 \leq 0$$

Limit on expected capital losses. We will interpret this to mean (expected capital loss)/(total capital invested), so:

$$(0.18x_1 + 0.31x_2 + 0.09x_3 + 0.19x_4 + 0.03x_5)/(x_1 + x_2 + x_3 + x_4 + x_5) \leq 0.1$$
$$\Downarrow$$
$$0.08x_1 + 0.21x_2 - 0.01x_3 + 0.09x_4 - 0.07x_5 \leq 0$$

Limit on total funds available (remember that the variables are in units of millions of dollars):

$$(x_1 + x_2 + x_3 + x_4 + x_5) \leq 14$$

Nonnegativity of the variables:

$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

Are the assumptions inherent in any LP model appropriate for *this* model? The additivity and proportionality assumptions are likely correct here. Strictly speaking, the restriction to real numbers does not hold because you can't subdivide a penny, but when dealing with very large numbers, this rounding to the nearest penny is negligible. The worst assumption here is that the parameters are known for certain. Both the annual growth rate and the probability of loss are educated guesses at best. Since $14 million depends on this decision, you should very carefully examine how changes in those numbers affect your solution. You will need to do some *sensitivity analysis*, a topic addressed later in this book.

Another unrealistic assumption is that you lose all of your capital if the mutual fund loses value. There is an old saying that "the map is not the territory", or to paraphrase for applied optimization "the model is not the real world". You must be aware of the losses in accuracy inherent in the assumptions that you make during modeling. Always check any assumptions with the client to make sure they are appropriate for the task at hand.

The Standard Form LP

Linear programs can have objective functions that are to be maximized or minimized, constraints that are of three types (≤, ≥, =), and variables that have upper and lower bounds. An important subset of the possible LPs is the **standard form LP**. A standard form LP has these characteristics:

- the objective function must be maximized,
- all constraints are ≤ type,
- all constraint right hand sides are nonnegative,
- all variables are restricted to nonnegativity.

A standard form LP is the simplest form of linear program, so we will begin our study of how to solve LPs using them. The most significant property of a standard form LP is that the origin (all variables set to zero) is always a feasible cornerpoint. This is because all standard form LPs have the kind of shape illustrated in Figure 2.5. Knowing this initial feasible cornerpoint greatly simplifies the search for the optimum. After studying how to solve standard form LPs, we will return to the problem of optimizing LPs that are not in standard form.

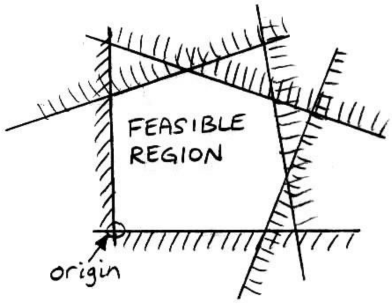

Figure 2.5: The origin is always a feasible cornerpoint in a standard form LP.

In an algebraic representation, a standard form LP having *m* functional constraints and *n* variables looks like this:

- **Objective function:** maximize $Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n$
  where the c_j, the coefficients in the objective function, represent the increase or decrease in Z, the objective function value, per unit increase in x_j. For the Acme Bicycle Company, Z is the daily profit, and the c_i are the contributions to profit made by the mountain bikes (c_1) and the racers (c_2).
- m **functional constraints**, so called because they take a functional form:

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_{1n} \leq b_1$$
$$a_{12}x_1 + a_{22}x_2 + \ldots + a_{2n}x_{2n} \leq b_2$$
$$\vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leq b_{\mathrm{m}}$$

  where the b_i are the resource limits, and the a_{ij} are the coefficients of the functional constraint equations, expressing the usage resource i consumed by activity j. For the Acme Bicycle Company, the b_i are limits on mountain bike production (b_1), on racer production (b_2) and on the metal finishing machine (b_3).
- n **nonnegativity constraints**: $x_1 \geq 0$, $x_2 \geq 0$, …, $x_{\mathrm{n}} \geq 0$.

Don’t forget to explicitly include the nonnegativity constraints when writing out a problem formulation. You don’t want to allow negative values for the variables accidentally: in the Acme example, this would mean that you could perhaps make money by disassembling bicycles and selling the materials back to the suppliers! Fortunately, most commercial LP solvers will assume nonnegativity if you don’t mention it, but while you are learning the subject, show that you have considered the variable bounds by explicitly writing them out. There are some formulations in which negative variables are allowed, for example when the variable represents change from the current level, as in the level of water in a reservoir.

In Practice

It is easy to get started using linear programming on real problems. Modern LP solvers are generally coupled with a user-friendly front-end which permits easy input of the model and browsing of the results. The solvers use a variety of input formats, so choose a solver that includes an input format that suits the way you work:

- A straight algebraic representation of the problem, with each constraint written out explicitly, such as in this book.
- A spreadsheet representation, generally with columns for the variables and rows for the constraints.
- An algebraic language that allows use of summations and indices to write the model very compactly. One statement in the algebraic language may generate numerous individual constraints for submission to the solver.
- A proprietary input format.

Algebraic modeling languages are the best developed input format, and are most used in practice. The ability to use summations and indices means that large industrial-scale models can be written in a concise form that is easier to debug. Many of the languages permit direct connection to databases, so the details of very large models can be easily changed without altering the overall form of the problem. For example, you may have a very stable overall model of your worldwide petroleum refining system, but need to change prices and demand information on a regular basis. This is straightforward to do with an algebraic modeling language.

Algebraic modeling languages also act as generic front-ends to solvers. In other words, you can write the model in the modeling language, and then choose from among the solvers that are attached to the modeling system. You might want a different solver because it uses a different algorithm, or because your model has gradually grown so large that a more powerful solver is required. With modeling languages, these kinds of changes are not a problem.

Web Resources

The following web page has links to web pages where you can find free student-edition linear programming solvers, on-line solvers, and other helpful resources:

http://www.sce.carleton.ca/faculty/chinneck/StudentOR.html