Mathematical Models: What and Why

Matthew J. Saltzman*

March 11, 1998

1 What Is a Model: Overview

Mathematical modeling is a technique used to gain control of complexity in real life. In science, mathematical models are often *descriptive* (so-called "laws of nature", such as Newton's gravitational law, are examples), but in management, models are just as often *prescriptive*, aiding a decision maker by pointing toward the "best" course of action. In this course we will concentrate on prescriptive models. The term *optimization* means selecting the best course of action from among many alternatives.

A *mathematical model* is a description of a real-world situation or problem using the language of mathematics. Often, the grubby details of the real-life situation are abstracted away, so many mathematical models appear to be simple, elegant, and unrealistic. It turns out that even these models can be complex and difficult to solve, but they can also be rewarding in that their solutions can be applied back to the real-world situation from which they arose. It is the modeler's responsibility to understand what properties of reality have been assumed away and to make judicious use of the model solution in the context of the original situation.

2 Why Model?

It does not take much complexity to make verbal descriptions of problems unwieldy. Even Euclid's description of Pythagoras's observation on the relationship among the sides of a right triangle shows signs of awkwardness:

> The sum of the squared lengths of the two sides of a right triangle adjacent to the right angle is equal to the square of the length of the side opposite the right angle.

Considering that all of Euclid's *Elements* was written before the advent of modern mathematical notation, it is remarkable what he was able to accomplish.

Compare the description above with a modern statement of this relationship:

> Let a and b represent the lengths of the sides of a right triangle adjacent to the right angle, and let c represent the length of the hypotenuse (the side opposite the right angle). Then
>
> $$a^2 + b^2 = c^2.$$

Once the meanings of the symbols are understood, the mathematical statement is a very concise description of the relationship under discussion; it conveys a lot of information in a small space. The math version has an additional advantage—it can be manipulated according to the rules of algebra to yield insights that are difficult (if not impossible) to discover from the verbal form. For example, given the lengths of two sides of a right triangle, we can solve the equation for the length of the third side. Although writing a verbal description of the formula is not much harder than writing a description of the original relationship, it is somewhat harder to write a convincing derivation of the new formula from the old (try it!). But with the math form, the algebraic steps necessary to derive the new formula are quite straightforward.

Consider the example in Murty [1], pages 11–12. This problem is already so complex that a completely verbal description is not even practical; instead, the pertinent data are organized into a table. The problem is to assign each of six candidates for regional sales director to one of six zones (and vice versa). The goal is to maximize the total projected annual sales across all zones. Data on projected sales under each possible pairing are contained in Table 1.

	Zone					
	1	2	3	4	5	6
Candidate 1	1	2	6	10	17	29
2	3	4	8	11	20	30
3	5	7	9	12	22	33
4	13	14	15	16	23	34
5	18	19	21	24	25	35
6	26	27	28	31	32	36

Table 1: Projected sales data for assignment example. The value in the ith row and jth column of the table is the projected annual sales volume in $million if candidate i is assigned to zone j.

Let us consider some approaches to solving this problem that we can undertake take with no further analysis. Clearly we can find the best assignment if we consider in turn each of the possible assignments of candidates to zones, evaluated the projected sales, and keep a record of the best assignment encountered to date during the search. This strategy is called *total enumeration*. Its strong point is that the optimal assignment is sure to be found, but it has a fatal weak point: the search is too time-consuming.

Actually, there are $6! = 720$ different possible assignments of candidates to zones that meet the conditions that each candidate be assigned to one zone and each zone be assigned to one candidate. Checking all 720 may be an onerous task for a human but it is not too bad for a computer. The problem comes when we consider solving similar problems with more items to be assigned. For example, if there were ten candidates and ten zones, the number of possible assignments would be $10! = 3,628,800$. This still isn't too bad for a computer. It would not be unreasonable to expect a modern desktop machine to be able to evaluate a million assignments per second, in which case the ten-candidate, ten-zone problem would take less than four seconds. But the twelve-candidate, twelve-zone problem admits $12! = 479,001,600$ different assignments, and the 24-candidate, 24-zone problem admits $24! = 620,448,401,733,239,439,360,000$ (620 sextillion) assignments. On a computer that could evaluate a billion assignments per second (about the limit of current technology), this problem would require over a billion centuries to solve. So the issue of running time is not simply about the time so solve a particular instance of a problem, but more about the impact of trying to solve larger and larger instances of problems with a similar structure. We will discuss this issue further later on in the course.

It is apparent that, if we want to be able to solve problems like this on a regular basis, we need a method that is more efficient than total enumeration. One plausible technique is the so-called *greedy* method. It works like this: we could select, among all possible assignments of an individual candidate to an individual zone, the one that yields the largest sales volume. In this case, candidate 6 is assigned to zone 6, yielding a volume of \$36 million. This choice is "greedy" in the sense that the best single match is chosen, without regard to the effect on other choices that need to be made. The process is then repeated with the remaining candidates and zones, until all candidates have been assigned to distinct zones. The resulting assignment is displayed in Table 2.

The greedy method is pretty clearly efficient (compared to total enumeration), but it suffers from a different fatal flaw: it is not guaranteed to find the optimal assignment. In fact, in this example, the greedy method leads to the worst possible assignment among all the available ones! (Gerald Thompson of Carnegie Mellon University coined the term "pessimal" to refer to the worst solution.) We could

Candidate	Zone	Volume
1	1	1
2	2	4
3	3	9
4	4	16
5	5	25
6	6	36
	Total	91

Table 2: Greedy solution to the sample assignment problem. Volume is in $millions.

Candidate	Zone	Volume
1	4	10
2	5	20
3	6	33
4	1	13
5	3	21
6	2	27
	Total	124

Table 3: Optimal solution to the sample assignment problem. Volume is in $millions.

certainly come up with methods that rely on more sophisticated criteria for making the greedy choice or for "fixing up" a solution constructed by a greedy method, but all such *ad hoc* techniques suffer from a failure to guarantee optimality.

Clearly, if we want efficiency *and* optimality, we need some more sophisticated analysis. But this discussion is getting rather long-winded already (and this isn't even that complicated a problem). If we want to get more sophisticated, we are going to have to get mathematical. (For the record, the optimal solution to this problem is given in Table 3.)

3 The Components of an Optimization Model

3.1 The Symbol Dictionary

The most important distinction between the two statements of the Pythagorean theorem given above is the introduction of symbols. We have symbols that stand for quantities (a, b, c) and symbols that represent operations ($+$, $\cdot^2$) and relations ($=$). Our goal now is to come up with a mathematical representation of the assignment problem given above.

If we begin by attempting to assign symbols to the quantities we are given (not a bad place to start, since we would like to be able to discuss the model independent of any particular estimates of sales volume), we quickly run into trouble. There are a lot of numbers—more than there are symbols if we restrict our symbol set to lower-case English letters. To get around this problem, we now introduce the notion of a *vector*. The idea is to use a single symbol to stand for an *attribute* of an object. If we have a set of such objects, all with different values for the attribute, we can distinguish the values associated with different objects in the set by using an index or subscript. The use of vectors not only saves symbols, it introduces another level of abstraction. For the assignment problem, we will see that we can discuss a single model for assignments involving arbitrary numbers of candidates and zones as well as arbitrary projected sales volumes.

For example, consider the set of students in this class. For convenience, let's assign a number to each student in some sequence (say alphabetically by name), so that if there are n students, they are numbered from 1 to n. Then we could denote the height (in inches) of student 1 as h_1, that of student 2 as h_2, and in general, the height of student i would be denoted h_i for each $i = 1, 2, \ldots, n$. We call the symbol i used in this fashion an *index variable* over the set of students. The symbol h with no subscript designates the list of heights *in order by subscript*, written as a column:

$$h = \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{pmatrix}.$$

We refer to such a column-wise list as a *vector*, and the individual h_is as its *components*. It is often convenient to transpose the column-wise list to a row-wise list:

$$h = (h_1, h_2, \ldots, h_n)^T.$$

The superscript T means that the vector h is the transpose of the row-wise list $(h_1, h_2, \ldots, h_n)$. The reason for this technicality will become clear when we review linear algebra.

Returning to the example assignment problem, we see that there are two sets of objects to consider: the set of candidates and the set of zones. Let i be an index variable over the set of candidates ($i = 1, 2, \ldots, 6$) and let j be an index variable over the set of zones ($j = 1, 2, \ldots, 6$).[1] In this problem, the objects in these *ground*

[1] Note that even though i and j can take on values from the same set, they really represent different things, and should not be confused.

sets do not possess attributes of their own that are relevant to the problem. The attributes of interest are associated with *pairs* of objects, one from each set. This fact suggests that we form a new set of ordered pairs of objects, one from the set of candidates and one from the set of zones. We call a set such as this, which is derived from another set or sets a *derived set*. Mathematically, we denote this particular derived set by

$$\{(i,j) \mid i = 1, 2, \ldots, 6 \text{ and } j = 1, 2, \ldots 6\},$$

and we will give it the name P.

The members of the set P have attributes of interest to us. First, associated with each pair in P is a projected sales volume. We denote this value c_{ij}, where the choice of the letter c is arbitrary, and the two subscripts i and j indicate the member (i,j) of P with which the value is associated.[2] Since the sales volumes are exogenous (that is, they are fixed by outside circumstances), we call the components of c *parameters*.

Attributes whose values we control are called *decision variables*. The other attribute of importance to us is such a variable, namely, whether the pair (i,j) is a part of the assignment that we will implement or not. We name this attribute x (again, the choice is arbitrary, although x, y, and z are popular choices for the names of variables). The question remains: what values should the components of x assume? For reasons that will become clear shortly, yes/no decisions such as those involved here are usually represented by the values 1 and 0.[3] Thus, we define

$$x_{ij} = \begin{cases} 1 & \text{if candidate } i \text{ is assigned to zone } j, \\ 0 & \text{otherwise.} \end{cases}$$

At this point, we have all the pieces in place for the first major section of a mathematical model, namely, a *dictionary of symbols*. For this problem, we might summarize the discussion thus far as follows:

- $i = 1, 2, \ldots 6$ is an index variable over the set of candidates.
- $j = 1, 2, \ldots 6$ is an index variable over the set of zones.
- c_{ij} is the predicted annual sales volume (in $millions) if candidate i is assigned to zone j.

---

[2] By convention, the components of a vector indexed by two or more subscripts are listed in order with the rightmost subscript cycling the fastest. Thus

$$c = (c_{11}, c_{12}, \ldots, c_{16}, c_{21}, c_{22}, \ldots, c_{65}, c_{66})^T.$$

[3] A variable whose value is restricted to be 0 or 1 is called *binary*.

- $x_{ij} = \begin{cases} 1 & \text{if candidate } i \text{ is assigned to zone } j, \\ 0 & \text{otherwise.} \end{cases}$

This dictionary gives the model its connection to the real phenomenon being modeled. Through our understanding of the meanings of the symbols, we are able to write down mathematical relations among the symbols that correspond to the real relationships among the real objects and attributes being modeled.

One important note about the symbol dictionary that is often overlooked by authors and modeling-language implementors: all attributes of objects have *units* associated with them in the symbol dictionary. The values of the parameters or variables associated with these attributes must be expressed in the same units as appear in the dictionary, and arithmetic with these quantities must make sense: quantities added together or compared must have the same units, and products of quantities take on the units defined as the products of the units of the individual quantities. Careful attention to the arithmetic of units can help you ensure that models you construct make sense.

3.2 Objective and Constraints

Along with the symbol dictionary, a mathematical model includes mathematical descriptions of relations among the symbols that mirror the relationships among the objects and attributes being modeled. In the Pythagorean Theorem example, the symbol dictionary contained the definitions of a, b, and c (with the unstated assumption that the lengths could be expressed in any sensible units, as long as the units were the same for all three quantities), and the relationship was expressed by the equation $a^2 + b^2 = c^2$.

In optimization models, two particular relationships must be described. First, we must set down conditions on the values of decision variables that define acceptable solutions (*constraints*), and we need to define a measure of the quality of a proposed solution (the *objective*), so that we may compare solutions. Our goal will be to find an acceptable solution that maximizes or minimizes (depending on the context) the value of the objective. We call a set of values for the decision variables a *solution.* A solution that satisfies all the constraints is called *feasible*, and one that does not is called *infeasible.* (Also, a set of constraints for which a feasible solution exists is itself called *feasible* and one for which no feasible solution exists is called *infeasible*). A feasible solution whose objective function value is as large as that of any other solution (when the goal is to maximize) is called *optimal.* Likewise, a feasible solution whose objective function value is as small as that of any other solution (when the goal is to minimize) is also called optimal.

3.2.1 Constraints for the Assignment Model

The first constraint is easy. Because in the symbol dictionary, x_{ij} is only defined to have meaning if its value is 0 or 1, we must include in our model this constraint on the allowable values of x_{ij}:

$$x_{ij} = 0 \text{ or } 1 \quad \text{for } i = 1, 2, \ldots, 6 \text{ and } j = 1, 2, \ldots, 6. \tag{1}$$

With the inclusion of this constraint, we restrict our attention to vectors x of binary values, but this is not enough to describe the solutions we intend to consider. For example, the vector with $x_{ij} = 0$ for all values of i and j is feasible for (1), but it corresponds to the case where no assignment is made. The vector with $x_{ij} = 1$ for all values of i and j is also feasible, but it corresponds to assigning each candidate to manage all six zones and each zone to all six candidates. Clearly, neither of these solutions would be acceptable. In fact, we want to reject any solution in which any candidate is assigned to manage more than one zone, and any solution in which any zone is assigned to more than one candidate. (Note that these are independent conditions: we could assign candidate 1 to manage all six zones and fire the others without violating the constraint of one manager per zone, or we could assign all six candidates to zone 1 without violating the constraint of one zone per candidate.)

Concentrating on the first condition (one zone per candidate), we want to write a mathematical relation (equation or inequality, for example) or set of relations, such that any solution that satisfies the constraint represents an assignment that meets the corresponding condition. It is also desirable that the relation be expressed using basic mathematical operations, and not, say, conditional tests on the values of variables. That is, we want to rule out confusing expressions of the constraint on candidate 1, such as

> If $x_{11} = 1$ then $x_{1j} = 0$ for $j = 2, 3, 4, 5, 6$. Otherwise, if $x_{12} = 1$ then $x_{1j} = 0$ for $j = 1, 3, 4, 5, 6$. Otherwise, ... etc.

One way to accomplish our goal is to observe that the number of zones assigned to candidate 1 can be computed by summing the values of the components of x whose first subscript is 1. Thus our constraint can be expressed as

$$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} = 1,$$

or alternatively

$$\sum_{j=1}^{6} x_{1j} = 1.$$

Now we need to duplicate this constraint for each candidate, replacing the subscript 1 with the subscripts 2 through 6. We can represent this set of constraints compactly as

$$\sum_{j=1}^{6} x_{ij} = 1 \quad i = 1, 2, \ldots 6. \tag{2}$$

A similar analysis gives us the constraints for the condition of one candidate per zone:

$$\sum_{i=1}^{6} x_{ij} = 1 \quad j = 1, 2, \ldots 6. \tag{3}$$

3.2.2 Objective for the Assignment Model

The values for the vector x that satisfy (1), (2), and (3) represent the 720 assignments of candidates to zones that would be considered acceptable, i.e., those that meet the one candidate-one zone restriction. From among these, we want to choose the "best" assignment. In order to make this choice, we need to have a way to compare two assignments and decide which is better. In this example, we have seen that we can compare two assignments by comparing the total sales volume across all zones, where the sales volume in each zone is determined by the choice of candidate to manage that zone: the larger the total volume, the better the assignment. Our goal now is to express this quality measure as a function of the decision variables.

We observe that a particular c_{ij} contributes its value to the total sales volume if and only if candidate i is assigned to zone j in the solution being considered. In that case, the value of the corresponding x_{ij} is 1, and for all pairs (i, j) for which candidate i is not assigned to zone j, the value of the corresponding x_{ij} is 0. Thus,

$$c_{ij}x_{ij} = \begin{cases} c_{ij} & \text{if candidate } i \text{ is assigned to zone } j, \\ 0 & \text{otherwise,} \end{cases}$$

and the sum of all the $c_{ij}x_{ij}$ terms is equal to the total sales volume for that solution. Thus we can write the objective function as

$$\sum_{i=1}^{6}\sum_{j=1}^{6} c_{ij}x_{ij}, \tag{4}$$

with the solution to be chosen from among all feasible solutions so as to maximize the value of (4). A model that consists of an objective function to be maximized or minimized, together with some constraints is called a *mathematical program*.

3.3 The Complete Model for the Assignment Problem

Taking our abstraction one small step further, we observe that even the number of candidates and zones to be assigned could be made a parameter, which we call n.

Note that the number of candidates must agree with the number of zones, and of course the number of components of the vector c must be n^2.

Now the symbol dictionary, the data in Table 1, and (1)–(4) together specify a complete mathematical programming model of the assignment problem in this example:

Symbol dictionary

- n is the number of candidates and zones to be assigned.
- $i = 1, 2, \ldots n$ is an index variable over the set of candidates.
- $j = 1, 2, \ldots n$ is an index variable over the set of zones.
- c_{ij} is the predicted annual sales volume (in \$millions) if candidate i is assigned to zone j.
- $x_{ij} = \begin{cases} 1 & \text{if candidate } i \text{ is assigned to zone } j, \\ 0 & \text{otherwise.} \end{cases}$

Math program

$$
\begin{array}{ll}
\text{Maximize} & \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\
\text{subject to} & \displaystyle\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1, 2, \ldots n \\
 & \displaystyle\sum_{i=1}^{n} x_{ij} = 1 \quad j = 1, 2, \ldots n \\
 & x_{ij} = 0 \text{ or } 1 \quad i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, n.
\end{array}
$$

Data

- $n = 6$
-

$c_{ij} =$	$j =$ 1	2	3	4	5	6
$i = 1$	1	2	6	10	17	29
2	3	4	8	11	20	30
3	5	7	9	12	22	33
4	13	14	15	16	23	34
5	18	19	21	24	25	35
6	26	27	28	31	32	36

```
set CANDIDATE;
set ZONE;

param revenue{CANDIDATE, ZONE} >= 0;

var Assign{CANDIDATE, ZONE} binary;

maximize Total_revenue:
        sum{i in CANDIDATE, j in ZONE} revenue[i, j] * Assign[i, j];

subject to Zone_limit{i in CANDIDATE}: sum{j in ZONE} Assign[i, j] = 1;

subject to Cand_limit{j in ZONE}: sum{i in CANDIDATE} Assign[i, j] = 1;
```

Figure 1: AMPL model of the sample assignment problem.

3.4 Computer Modeling Languages

The AMPL package supports the expression of mathematical models for computer solution by solvers such as CPLEX and MINOS. An AMPL model of our sample assignment problem is reproduced in Figures 1 and 2. Compare the sections and their contents with the model in mathematical notation above. We will compare these notations (mathematics and AMPL) in more detail in class.

4 Where Are We Now?

Now that we have a mathematical formulation of our assignment problem, what is it good for? As with the Pythagorean theorem, we now have a model expressed in mathematical terms, with symbols representing real-world parameters and decision variables, and relationships expressed mathematically. With the Pythagorean theorem, we could easily see how to use our model to solve for an unknown third side, given the lengths of any two sides. In the case of the assignment model, the way to use the model to efficiently find the optimal assignment is not readily apparent. But at least we have the problem in a form that can be analyzed.

After we get some more practice setting up models from their descriptions and using the computer to get solutions, and after we review some basic mathematical concepts that we'll need, we will turn our attention to working with mathematical programming models to find conditions that we can test efficiently to see if a proposed solution is optimal, and methods that will lead us to optimal solutions.

```
set CANDIDATE := Ann Bob Carol Debbie Ed Frank;
set ZONE := NE NC NW SE SC SW;

param revenue: NE NC NW SE SC SW :=
        Ann      1  2  6 10 17 29
        Bob      3  4  8 11 20 30
        Carol    5  7  9 12 22 33
        Debbie  13 14 15 16 23 34
        Ed      18 19 21 24 25 35
        Frank   26 27 28 31 32 36;
```

Figure 2: AMPL model of the sample assignment problem.

These ideas may not seem too important at first blush (after all, that's what we have computers for), but we will see that these basic ideas are extremely important in gaining insights into our models beyond the obvious solutions.

References

[1] K. G. Murty, *Operations Research: Deterministic Optimization Models*, Prentice Hall, 1995.