Formulating an MP: An Overview

Prepared by: Nathaniel Grier

9 November, 2001

This document is intended to help guide you through the process of formulating an MP. It will provide some guidelines and general "rules-of-thumb". Model formulation is more of an art than a science, however, and none of these rules are hard and fast. As you will see in class, there are often multiple ways to formulate a problem and the way in which you do so can make the difference between whether you can find a solution to your problem or not.

1 The pieces of a Formulation

There are, generally, three parts which define a model.

1. The objective function.
2. The constraints on the problem.
3. The decision variables.

The first two items are the terms in which one often describes a problem. Yet they rely on the underlying assumptions of the decision variables. You will see that you are often faced with a "chicken-and-egg" problem when formulating a problem.

1.1 The Objective Function

The objective function summarizes just that: your objective of the problem. If you are maximizing profit, than it will be a function which determines the profit for a given set of decision variables. It is never an equation. Note that for simplicity sometimes we may write:

$$\text{Minimize :} \qquad p = \sum \ldots$$

but this is merely shorthand for

$$\begin{aligned} \text{Minimize :} \quad & p \\ \text{Subject to :} \quad & p = \sum \ldots \end{aligned}$$

The objective function can be very simple or can be very complex. As we will see later, we will often move pieces of the problem into the objective function (and out of the constraints) in order to make the problem tractable.

1.2 The Constraints

The constraints are the limitations placed on the problem and control the allowable solutions. If, for example, we have a limit on the amount of goods we can produce, this would be modeled by a constraint. Similarly, in the set-covering problem, we enforce the requirement that, for example, every route is operated, with an appropriate constraint. These are equations or inequalities and are usually presented with the decision variables on the left and the constant term(s) on the right. Example constraints would be:

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = b_i \qquad \forall\, i \in N$$

$$y_i + y_{i+1} \le d_i \qquad \forall\, i \in P$$

1.3 The Decision Variables

The decision variables model the "decisions" we are trying to make by solving the optimization problem. For example, if we are trying to determine the production levels which will maximize our profit, then we would have a decision variable for each of the products we produce; the value of each variable would indicate the number of each product we should produce.

In addition to direct decision variables, such as the amount of a product to produce, we may need to introduce additional, secondary variables to fully encapsulate the problem situation. For example in "location" problems – problems where you try to find the optimal location of a facility to meet some other objective – we often must introduce a binary variable indicating whether we open a plant a some location and use this to ensure that our solution does not include using a plant which we don't build. You can see why the formulation of a problem is not always straightforward — often in the process of developing the model you will realize that you must introduce additional variables to capture the restrictions of the problem.

Finally, as we alluded to in the previous paragraph, the nature of the variables also plays a role in the formulation and solution. In a linear program, all the variables are continuous and may assume any value in the set of real numbers. As negative quantities often make little sense in real-world situations, there is often a non-negativity constraint imposed (so the decision variables are in $\mathbb{R}^+$). Real-world problems often have the further constraint that we are dealing with "whole" things, and thus the decision variables are restricted to the set of integers $\mathbb{Z}$ (all or non-negative). This formulation is known as an integer program. It is important to understand that the solution to a linear program and integer program with the same constraints and objective function *is most likely not the same.* Note that whether the variables are integer or real-valued is a function of the solving technique and cannot generally be specified as a constraint of the model. The one sort of exception to this rule are binary variables – variables which can take on the value of 0 or 1. These are not allowed in a pure linear program. However in mixed programs you can have binary variables with the rest of the variables real-valued. Such restrictions are also a function of the solver you use to solve the problem - some will allow "non-traditional" mixes of variable types and others are very strict about uniformity among the variables.

1.4 Data

It is important to differentiate between data and variables. In a model with a number of letters and subscripts it is important to remember which are the variables and which are the fixed data. Generally, the data is known before you formulate a model and the task of the formulation is to relate the known data with the unknown variables. Often, though, you will find that there is not a perfect fit between your formulation in the data you have at hand. In some cases you will find yourself with extraneous data. In these cases you must choose what is appropriate and what to leave out. More frequently, however, you will find that you do not have all the data you need, or think you need. In these cases, you will have to either obtain the data you lack or try to reformulate the problem to utilize only the data you have available. In the problem sets for this class you will always be provided sufficient information to solve the problem.

2 Formulating a Model – One Approach

As mentioned earlier, there's no one correct way to develop a model formulation. Often you will find that it is an iterative process where in developing one part of the formulation you realize you must add to another part. Below I give an approach that I have found to work well for me.

2.1 Determining the Objective Function

While the objective function is a function of your decision variables and subject to any constraints, I find it works best to formulate the objective function so that you know what you are working toward in the rest of the problem. Also, the objective is often readily obtainable from the problem statement.

At this stage it's not necessary to write the problem algebraically, especially if the objective is quite complex. If our objective is to minimize the costs we incur our objective function might look something like this:

$$\text{Minimize :} \qquad \sum_i cost_i$$

How $cost_i$ is determined may be easy or quite complicated. In this simple production case we might incur a unit cost for each type of good so $cost_i = num_i \cdot unitCost_i$. In the case of the package-flow problem, as we will see later in the semester, our objective is essentially to minimize costs as well, but the formulation of the objective function is much more complex.

2.2 Determine the Decision Variables

Now that you have a handle on the objective function, we turn to deciding on the decision variables. Hopefully the decision variables will fall out directly from the objective function. In the example above, our decision variable was num_i. For ease of writing the model, though, we usually use single letter variables. Traditionally letters at the end of the alphabet, often starting with x, are used as decision variables, reserving those near the beginning and middle for constants.

Furthermore, as you have already noticed, we tend to use subscripts to make up for the fact that there are a limited number of letters. This also allows us to present the problem in a very compact way. So if we use x_i in place of num_i, and let $c_i = unitCost_i$, we end up with the very compact objective function:

$$\text{Minimize :} \qquad \sum_i x_i c_i$$

We also use subscripts to represent relationships between two things. In flow problems, for example, we often denote the flow between some point i and another point j by x_{ij}. Note that x_{ij} is distinct from x_{ji}; while they may be the same or equal in some cases, in many others the ordering of the subscripts is quite important.

At this point you also want to introduce any auxiliary variables if you can see, from the objective, that they will be necessary. For example, if you are performing a flow assignment, the primary costs arise from the individual flows multiplied by the unit costs. But if there is some additional fixed cost for using a link you must introduce an appropriate variable.

2.3 Determining the Constraints

In the simplest of problems there may be no constraints other than variable value constraints, such as the non-negativity or binary constraints. In many cases, however, it is the constraints which complete the problem.

The first step is to write, in terms of your decision variables, any constraints which are inherent to the problem. If, for instance, you know that you cannot produce more than a certain amount of good i, or that the total number of goods produced must be equal to some constant, you would include it here. In our previous example, if, for each good i, we could produce no more than b_i units, we would model that as follows:

$$x_i \leq b_i \qquad \forall\ i$$

At this point, you may find that you have completely encapsulated the model: the objective function fully includes all attributes of the problem (all costs, for example), and the constraints completely model all limitations. You may, however, realize that there are constraints which you cannot model given the variables you have chosen. At this point you need to reexamine the problem and decide how to model those constraints, introducing variables as necessary.

If you have unusually complicated constraints, this will often necessitate the introduction of additional variables. If you do add extra variables, or ones which do not seem directly apparent from the problem statement, it is important that you then add the appropriate constraints to bound the new variables.

Finally, it is important to not that in order for (most) solvers to solve your formulation, it (and in particular the constraints) *must be linear*. We will talk more about this below, but for now you should now that functions such as max{...} or if-then constraints are not linear.

3 Two Examples

As I've said several times, MP formulation is more of an art than a science. To that end, here are a few examples which we go through step-by-step.

3.1 Example: Product Distribution

A producer has just completed a limited production run of its new product. It has a number of product testers to whom it wishes to distribute a sample of the product as soon as possible. But in order to minimize the cost of this distribution it has decided to use its existing supply network. This network G consists of a set of transfer points, or nodes, N between which the distribution is non-stop. Let $|N| = n$; we want to deliver a sample to each of the $n - 1$ cities in our distribution network. We denote the set of these connections as A, and the travel time along such a connection, starting at some point i and connecting to a point j, by τ_{ij}. If we number the plant as node 0, provide a formulation of the problem given that we must deliver only one sample to each destination.

Additionally, how would you model the following constraints:

(a) In order to have as small an impact on existing shipments as possible, you impose a limit such that a given city-pair route (i, j) can can be used by at most u_{ij} sample products. In this case your objective is to minimize the total travel time.

(b) Instead of one plant producing all your samples, you now have two, each of which produced $\frac{n-2}{2}$ of the total samples; let the two plants be located in cities 0 and 1. Reformulate your model to minimize the total travel time.

A Solution

The problem above is actually a version of the classic shortest paths problems. There are a number of ways to formulate this problem but we give here the most common. Our objective is to minimize the travel time of each sample. If all the travel times are non-negative and the network is uncapacitated, this is equivalent to minimizing the total travel time. Since travel time may not be as intuitive, it helps to think of it as the cost (in dollars, for example). Given that we are thinking of "shortest paths", a first inclination might be to delineate each and every path. However, as the number of such paths in combinatorial, we would not be able to easily find a solution to the problem in a large network. Instead, if we view the problem in terms of arcs, the problem becomes readily solvable. In this case we get an objective function of the following form:

$$\text{Minimize :} \qquad \sum packages\_on\_each\_arc \cdot travel\_time\_on\_arc$$

We see that the travel time on each path is accounted for; we have only "regrouped" the terms in the above formulation. The only variable we need introduce, then, is the flow on each arc, which is traditionally denoted x_{ij}. Thus our objective is now:

$$\text{Minimize :} \qquad \sum_{(i,j) \in A} x_{ij} \tau_{ij}$$

The only constraint we have is that each destination receive one package and one package only. We can model this using the classical "balance" constraint. That is we sum up all the flow entering a node and subtract it from the sum of the flow leaving a node and force this to equal some value b_i for each node i. Since we need not deliver any samples to the plant, but we're delivering a total

From ↓ \ To →	Region 1	Region 2	Region 3
New York	20	40	50
Los Angeles	48	15	26
Chicago	26	35	18
Atlanta	24	50	35

Table 1: Shipping costs, in dollars, to send 1 unit from a given warehouse to a given region, for Example 3.2.

of $n-1$ samples to the other cities, so $b_0 = n-1$. For every other city, however, we want one more sample to end up there than to leave, so for $i \neq 0, b_i = -1$. While it is convention to define the balance such that the source nodes have a positive b_i and the sinks a negative value, it is completely fine to reverse the two. Finally, we can state our constraint as follows:

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = b_i \qquad \forall\, i \in N$$

We must, of course, remember to include the restriction that $x_{ij} \geq 0$. More accurately, we should state that $x_{ij} \in \mathbb{Z}^+$ since it does not make much sense to half of a sample down one path and half down another.

We can model the additional constraints as follows:

(a) In this case while our objective has changed, our objective function need not – it still works to minimize total travel time. To model the capacity constraint, we need only introduce an upper bound constraint such that:

$$x_{ij} \leq u_{ij} \qquad \forall\, (i,j) \in A$$

Because we did not place any additional constraints, this one additional constraint is sufficient.

(b) In this case our objective and objective function remain the same as the previous case, so we need not reexamine those. Moreover, we need not change our model at all. Because we formulated the problem in terms of flows on arcs and utilized the balance constraints, we need only change b_0 and b_1 to reflect their new values – $\frac{n-2}{2}$ – while all the other balance constraints are unchanged. Think, though, how much more work we would have to do if we had formulated this problem in terms of paths. We would have to enumerate all the paths from the second plant and calculate their distances in order to solve this modified version of the problem. Remember: it may not be easy to develop a formulation which is robust, but the extra time can more than repay itself in future savings.

3.2 Example: Warehouse Location

A company is considering opening warehouse(s) in four cities: New York, Los Angeles, Chicago and Atlanta. Each warehouse can ship 100 units per week. The weekly fixed cost of keeping each warehouse open is \$400 for New York, \$500 for Los Angeles, \$300 for Chicago and \$150 for Atlanta. Region 1 requires 80 units per week, Region 2 requires 70 units per week and Region 3 requires 40 units per week. The shipping costs costs are shown in Table 1.

(a) Formulate the problem to meet weekly demands at minimum cost.

(b) How can the following conditions be imposed?:

1. If the New York warehouse is opened, the Los Angeles warehouse must be opened.
2. At most two warehouses can be opened.
3. Either the Atlanta or the Los Angeles warehouse must be opened (but not both).

A Solution

The first step would be to characterize the objective function. While it may not be immediately apparent, with a little thought we come up with the following:

$$\text{Minimize :} \qquad \sum_{i \in W} Cost_i \cdot Open_i + \sum_{i \in W} \sum_{j \in R} transpCost_{ij} \cdot flow_{ij}$$

where W is the set of warehouse locations and R is the set of regions. This objective function encapsulates all our costs. In this case our "pseudovariables" above translate directly into real variables. We let y_i be a binary variable representing whether we open the warehouse or not. And we let x_{ij} represent the flow from warehouse i to region J. We can then denote the weekly cost of operating warehouse i by c_i and the unit transportation cost by t_{ij}.

We now turn to identifying the constraints necessary to model those presented in the problem statement. First, we are given the constraint that the flow out of each warehouse must be no more than 100 units per week. We can model this as follows:

$$\sum_j x_{ij} \leq 100 \qquad \forall\, i \in W$$

Next we must ensure that the demand is met in each region, and can do so with the following constraint, letting b_j represent the demand in region j:

$$\sum_i x_{ij} = b_j \qquad \forall\, j \in R$$

Finally we must somehow tie everything together, ensuring that the flow out of a warehouse is only non-zero if we decide to open that warehouse – that is if y_i is 1. The most efficient way to do this is by modifying the first constraint as follows:

$$\sum_j x_{ij} \leq 100 \cdot y_i \qquad \forall\, i \in W$$

The resulting constraint is still linear and satisfies the requirement that the flow out be 0 if the warehouse is not opened: if y_i is 0, the combined flow out of warehouse i must be 0; if we open warehouse i, the only restriction on its flow is that the total flow must be less than 100 units per week.

The above constraints rely on two implicit constraints on the variable values which it is important to state:

$$x_{ij} \in \mathbb{Z}^+,\ y_i \in \{0, 1\}$$

These last two constraints are quite important as otherwise we might end up with a solution which makes no "real-world" sense. To summarize, then, our formulation is:

$$\begin{aligned}
\text{Minimize :} \quad & \sum_{i \in W} c_i \cdot y_i + \sum_{i \in W} \sum_{j \in R} t_{ij} \cdot x_{ij} & \\
\text{Subject to :} \quad & \sum_j x_{ij} \leq 100 \cdot y_i & \forall\, i \in W \\
& \sum_i x_{ij} = b_j & \forall\, j \in R \\
& x_{ij} \in \mathbb{Z}^+, y_i \in \{0, 1\} &
\end{aligned}$$

We now turn to modeling the additional constraints.

(a) We are asked how to ensure that if the New York warehouse is opened then the Los Angeles warehouse is also opened. So if $y_{NYC} = 1$, then $y_{LA} = 1$; but if $y_{NYC} = 0$ there is no restriction on y_{LA}. If we view this in terms of the decision variables, we see that y_{LA} must always be greater than y_{NYC}. This translates into the following constraint which answers the question:

$$y_{NYC} \leq y_{LA}$$

(b) We are asked to develop a constraint which will limit the total number of warehouses opened to no more than two. Again, we can make use of the decision variables y_i. If we limit the number of warehouses to 2, this is the same as limiting the sum of the y_i to 2. This, then, is our constraint:

$$\sum_i y_i \leq 2$$

(c) Finally, we are told that either the Atlanta or Los Angeles warehouse must be opened but not both. Viewing the constraint in terms of the decision variables y_i, we see that this is equivalent to saying that the sum of y_{LA} and y_{ATL} is 1. Again, this is our constraint and answers the problem:

$$y_{LA} + y_{ATL} = 1$$

In these last few constraints, we saw, once again, how important to develop a robust model formulation. In our case we were able to make a number of changes to the problem, but each required only a small change in the formulation of the model.

A Set Notation Review

A set is some collection of distinct objects (i.e. no repetition). Some common sets are:

$\mathbb{R}$	The set of real numbers
$\mathbb{Z}$	The set of integers
$\emptyset$	The null (empty) set — this is a set with no elements

Note that we will refer to the non-negative elements of a set with a superscript '+'; thus we refer to the non-negative real numbers by $\mathbb{R}^+$.

Some notation we use to help us define sets includes:

$\in$	"Is an element of": $x \in S$ means that x is one of the objects in the set S
$\{\ \}$	"The set containing": We usually list the members of a set between brackets (see example below)
: or $\mid$	"Such that": We use this operator to help define which elements are included in a set. For instance, if we are given a set S we can create a subset such that all elements are non-negative. This new set would be written as $\{x \in S : x \geq 0\}$ (The set consisting of all non-negative elements of S).

Some operators on sets:

$\lvert S \rvert$	"Ordinality of S": this is the size of, or number of elements in, the set S
$S \cup T$	"The union of S and T": this is the collection of all elements in either set S or set T
$S \cap T$	"The intersection of S and T": this is the collection of all elements in both S and T
$S \backslash T$	"S minus T": this is the set of elements in S but not in T

Some relations between sets are expressed as:

$S \subset T$	"S is a subset of T": all elements in S are contained in T
$S = T$	"S is equal to T": all elements in S are contained in T and vice versa

Some additional shorthand notation you will see:

$\exists$	"There exists"
$\forall$	"For all"