Linear Programming

Introduction

Linear programming is a subset of a larger area of application called mathematical programming. The purpose of this area is to provide a means by which a person may find an optimal solution for a problem involving objects or processes with fixed 'costs' (e.g. money, time, resources) and one or more 'constraints' imposed on the objects. As an example, consider the situation where a manufacturer wishes to produce 100 pounds of an alloy which is 83% lead, 14% iron and 3% antimony. Assume he has at his disposal, five existing alloys with the following characteristics:

Alloy1	Alloy2	Alloy3	Alloy4	Alloy5	Characteristic
90	80	95	70	30	Lead
5	5	2	30	70	Iron
5	15	3	0	0	Antimony
$6.13	$7.12	$5.85	$4.57	$3.96	Cost

This problem results in the following system of equations:

$$
\begin{array}{rcrcrcrcrcl}
X_1 & + & X_2 & + & X_3 & + & X_4 & + & X_5 & = & 100 \\
0.90X_1 & + & 0.80X_2 & + & 0.95X_3 & + & 0.70X_4 & + & 0.30X_5 & = & 83 \\
0.05X_1 & + & 0.05X_2 & + & 0.02X_3 & + & 0.30X_4 & + & 0.70X_5 & = & 14 \\
0.05X_1 & + & 0.15X_2 & + & 0.03X_3 & & & & & = & 3 \\
6.13X_1 & + & 7.12X_2 & + & 5.85X_3 & + & 4.57X_4 & + & 3.96X_5 & = & Z(\min)
\end{array}
$$

The last equation is known as the 'objective' equation. The first four are constraints. We wish to obtain the coefficients of the X objects that will provide the minimal costs and result in the desired composition of metals. We could try various combinations of the alloys to obtain the desired mixture and then calculate the price of the resulting alloy but this could take a very long time!

As another example: a dietitian is preparing a mixed diet consisting of three ingredients, food A, B and C. Food A contains 81.85 grams of protein and 13.61 grams of fat and costs 30 cents per unit. Each unit of food B contains 58.97 grams of protein and 13.61 grams of fat and costs 40 cents per unit. Food C contains 68.04 grams of protein and 4.54 grams of fat and costs 50 cents per unit. The diet being prepared must contain the at least 100 grams of protein and at the most 20 grams of fat. Also, because food C contains a compound that is important for the taste of the diet, there must be exactly 0.5 units of food C in the mix. Because food A contains a vitamin that needs to be included, there should also be a minimum of 0.1 units of food A in the diet. Food B contains a compound that may be poisonous when taken in large quantities, and the diet may contain a maximum of 0.7 units of food B. How many units of each food should be used in the diet so that all of the minimal requirements are satisfied, the maximum allowances are not violated, and we have a diet which cost is minimal? To make the problem a little bit easier, we put all the information of the problem in a tableau, which makes the formulation easier.

	Protein	Fat	Cost	Minimum	Maximum	Equal
Food A	81.65	13.60	$0.30	0.10		
Food B	58.97	13.60	$0.40		0.70	
Food C	68.04	4.54	$0.50			0.50
Min.	100					
Max.		20				

The numbers in the tableau represent the number of grams of either protein and fat contained in each unit of food. For example, the 13.61 at the intersection of the row labelled "Food A" and the column labelled "Fat" means that each unit of food A contains 13.61 grams of fat.

Calculation

We must include 0.5 units of food C, which means that we include 0.5 * 4.54 = 2.27 grams of fat and 0.5 * 68.04 = 34.02 grams of protein in the diet, coming from food C. This means, that we have to get 100 - 34.02 = 65.98 grams of protein or more from Food A and B, and that we may include a maximum of 20 - 2.27 = 17.73 grams of fat from food A and B. We have to include a minimum of 0.1 units of food A in the diet, accounting for 8.17 grams of protein and 0.45 grams of fat. This means that we still have to include 65.98 - 8.17 = 57.81 grams of protein from food A and/or B, and that the maximum allowance for fat from A and/or B is now 17.73 - 0.45 = 17.28 grams. We should first look at the cheapest possibility, eg inclusion of food A for the extra required 57.81 grams of protein. If we include 57.81/81.65 = 0.708 units of food A, we have met the requirement for protein, and we have added 0.708 * 13.61 = 9.64 grams of fat, which is below the allowance of 17.28 grams which had remained. So we don't need any of the food B, which is more expensive, and which is contains less protein. The price of the diet is now $0.48. But what would we do, if food B was available at a lower price? We may or may not want to use B as an ingredient. The more interesting question is, at what price would it be interesting to use B as an ingredient insteaed of A? This could be approached by an iterative procedure, by choosing a low price for B, and see if the price for the diet would become less than the calculated price of $0.48.

Implementation in Simplex

A more sophisticated approach to these problems would be to use the Simplex method to solve the linear program. The sub-program 'Linear Programming', provided with OpenStat can be used to enter the parameters for these problems in order to solve them.

The Linear Programming Procedure

To start the Linear Programming procedure, click on the Sub-Systems menu item and select the Linear Programming procedure. The following screen will appear:

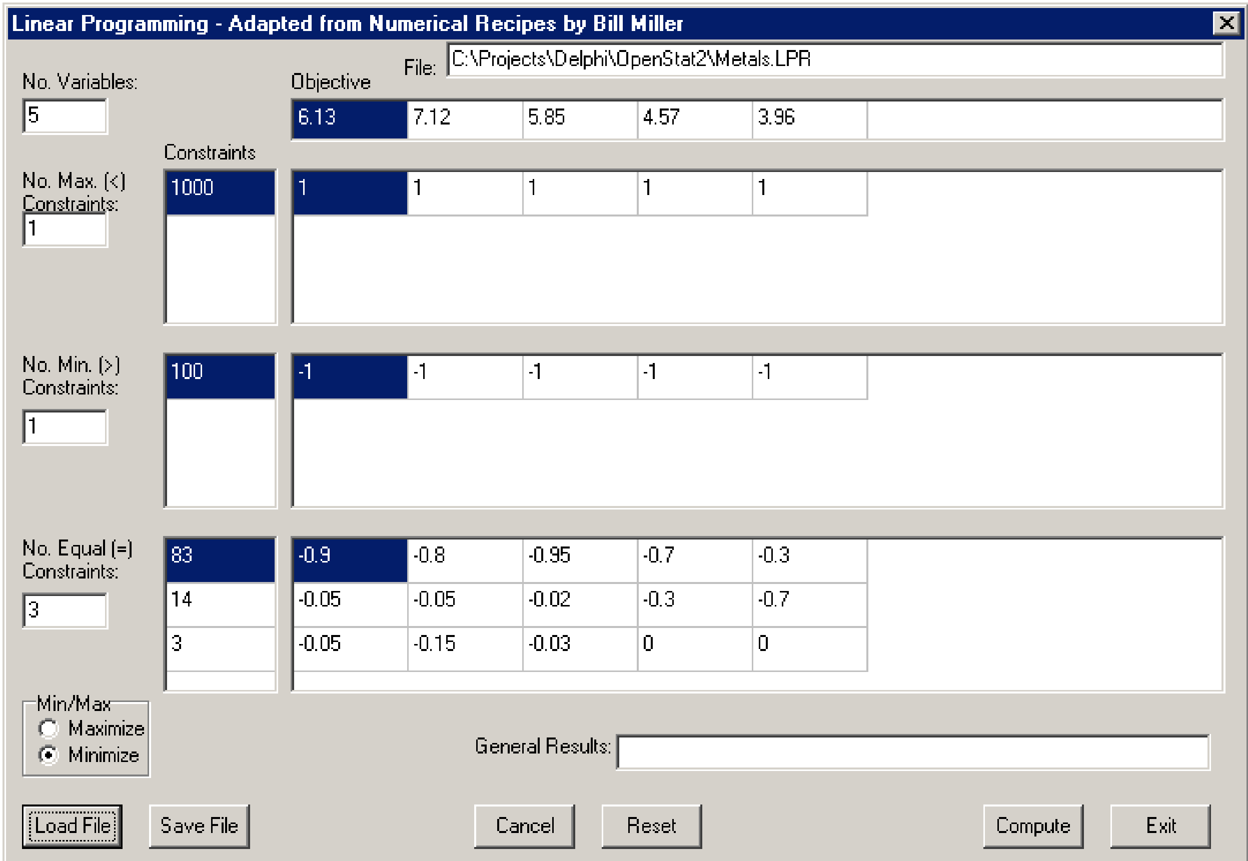

Figure XV-1 Linear Programming Dialogue Form

We have loaded a file named Metals.LPR by pressing the Load File button and selecting a file which we had already constructed to do the first problem given above. When you start a problem, you will typically enter the number of variables (X's) first. When you press the tab key to go to the next field or click on another area of the form, the grids which appear on the form will automatically reflect the correct number of columns for data entry. In the Metals problem we have 1 constraint of the 'Maximum' type, 1 constraint of the 'Minimum' type and 3 Equal constraints. When you have entered the number of each type of constraint the grids will automatically provide the correct number of rows for entry of the coefficients for those constraints. Next, we enter the 'Objective' or cost values. Notice that you do NOT enter a dollar sign, just the values for the variables - five in our example. Now we are ready to enter our constraints and the corresponding coefficients. Our first (maximum) constraint is set to 1000 to set an upper limit for the amount of metal to produce. This constraint applies to each of the variables and a value of 1.00 has been entered for the coefficients of this constraint. The one minimum constraint is entered next. In this case we have entered a value of 100 as the minimum amount to produce. Notice that the coefficients entered are ALL negative values of 1.0! You will be entering negative values for the Minimum and Equal constraints coefficients. The constraint values themselves must all be zero or greater. We now enter the Equal constraint values and their coefficients from the second through the fourth equations. Again note that negative values are entered. Finally, we click on the Minimize button to indicate that we are minimizing the objective. We then press the Compute button to obtain the following results:

```
Linear Programming Results

                   X1         X5

  z   544.8261   -0.1520   -0.7291
 Y1 1100.0000    0.0000    0.0000
```

X3	47.8261	-0.7246	1.7391
Y2	0.0000	0.0000	0.0000
X4	41.7391	-0.0870	-2.3913
X2	10.4348	-0.1884	-0.3478

The first column provides the answers we sought. The cost of our new alloy will be minimal if we combine the alloys 2, 3 and 4 with the respective percentages of 10.4, 47.8 and 41.7. Alloys 1 and 5 are not used. The z value in the first column is our objective function value (544.8).

Next, we will examine the second problem in which the nutritionist desires to minimize costs for the optimal food mix. We will click the Reset button on the form to clear our previous problem and load a previously saved file labeled 'Nutrition.LPR'. The form appears below:

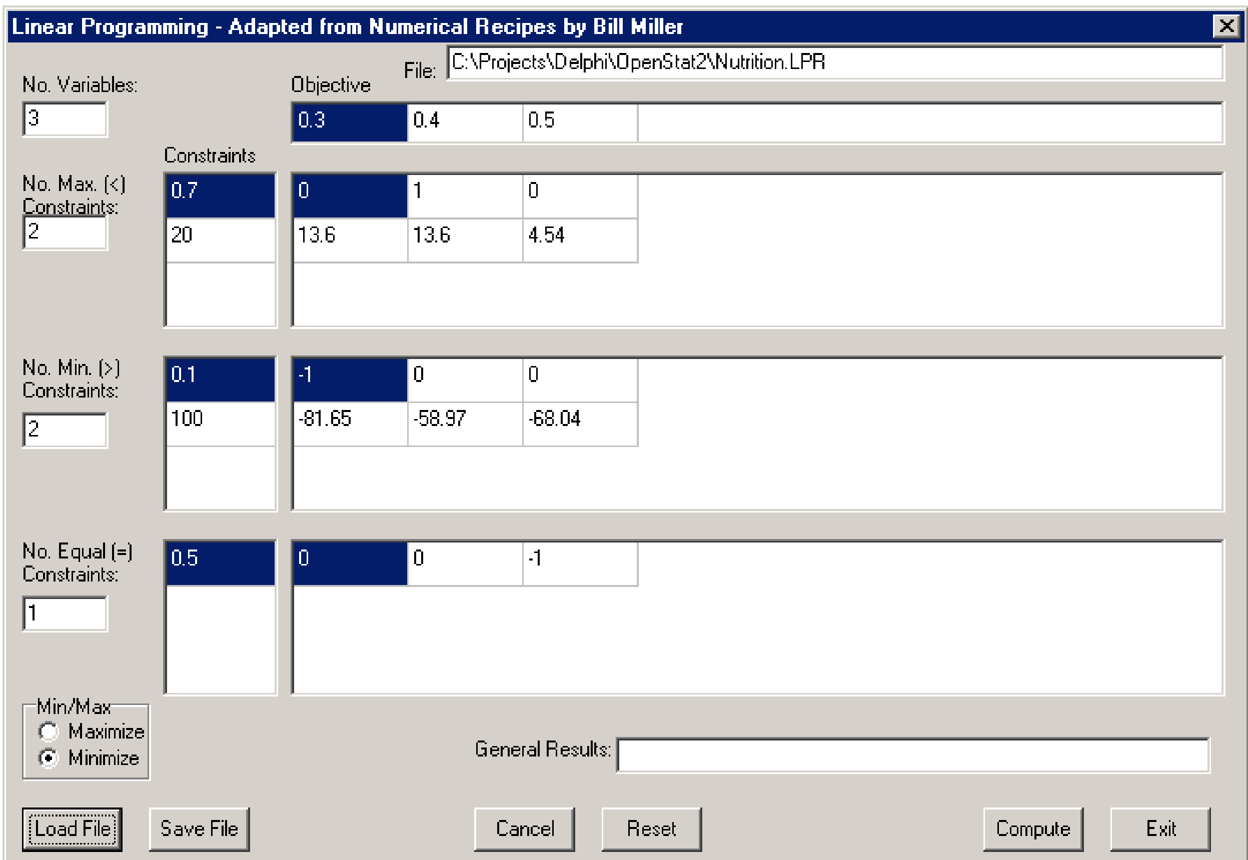

Figure XV-2 Example Specifications for a Linear Programming Problem

Again note that the minimum and equal constraint coefficients entered are negative values. When the compute button is pressed we obtain the following results:

Linear Programming Results

		Y4	X2
z	0.4924	-0.0037	-0.1833
Y1	0.7000	0.0000	1.0000
Y2	33.2599	0.1666	3.7777
X1	0.8081	0.0122	-0.7222

```
 Y3     0.7081     0.0122   -0.7222
 X3     0.5000     0.0000    0.0000
```

In this solution we will be using .81 parts of Food A and .5 parts of Food C. Food B is not used.

The Linear Programming procedure of this program is one adapted from the Simplex program in the Numerical Recipes book listed in the bibliography (#56). The form design is one adapted from the Linear Programming program by Ane Visser of the AgriVisser consulting firm.