Linear Optimisation Example

FILE: Linear.cln

Introduction

A common problem of resource optimisation uses a technique called Linear Programming (LP). These days LP problems are solved by spreadsheets as a side function to the main solver utility. CleanSheet is no different in this, its Solution Search object can detect and solve LP problems.

When you open this file, the sheet is designed on 3 layers to help the explanation. Select Scale to Fit on the Scale dialogue, the whole sheet will be visible.

This example is explained in much more detail than the others. As well as explaining how the sheet works, it explains the steps that went into its design.

Scenario

A manufacturer of fertilizer develops 3 new products, code names X1, X2 and X3. These 3 fertilizers are made from different mixtures of 4 basic ingredients: Nitrates, Phosphates, Potash and Fillers.

The tons used of each ingredient for each product is listed in the following table. It should be clear that this table makes 100 tons of each ingredient since 10+10+20+60 is 100.

	Nitrate	Phosphates	Potash	Filler
X1	10	10	20	60
X2	10	20	10	60
X3	20	10	10	60

The ingredients cost different amounts to buy: Nitrates £150 per ton, Phosphates £60 per ton, Potash £120 per ton and Filler £10 per ton.
A limit on mining and shipping means that we can obtain a maximum supply of each ingredient as follows: 1200 tons per month of Nitrates, 2000 tons per month Phosphates, 2200 tons per month Potash, and a virtually unlimited supply of Fillers.
The prices we charge for our three fertilizers are X1 £83, X2 £81 and X3 £81.
The cost of manufacturing the ingredients into finished products is £11 per ton.

Given that we can sell whatever we make, what quantity of X1, X2 and X3 should we make in the next month to make the maximum profit?

Layer 0 - Known Inputs

The first step in any sheet is to place objects containing the information we know. For this we have laid out 4 tables and 1 Input box into which the known data is entered.

Ingredients (Tons)

Ratios	Nitrate	Phosphates	Potash	Filler
X1	10	10	20	60
X2	10	20	10	60
X3	20	10	10	60

Price per ton

Nitrate	Phosphate	Potash	Filler
£150.00	£60.00	£120.00	£10.00

Shipping Limit

Nitrate	Phosphate	Potash	Filler
1200	2000	2200	1000000

Retail Price

X1	£83.00
X2	£81.00
X3	£81.00

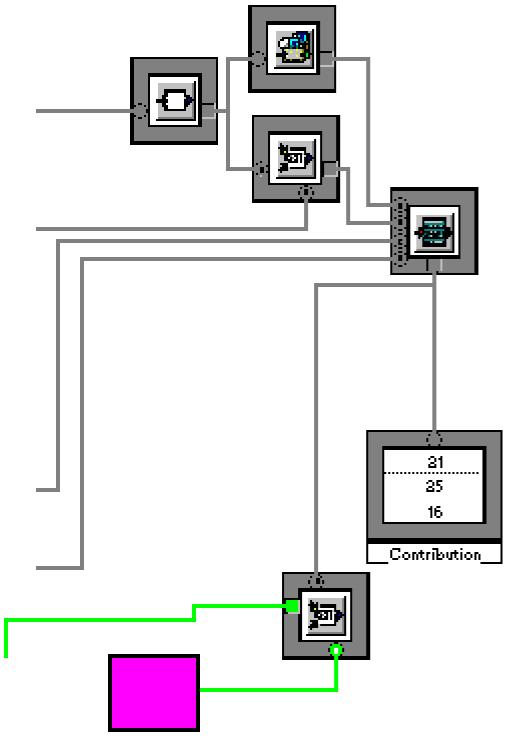
Manufacturing Cost

£11

Layer 1 - The Contribution to Profits

The next step for an LP problem is to calculate the profit we would make if we sold a particular number of units. How many units? This is determined by the Solution Search object, which we haven't put into the sheet yet. During the development of this sheet an Input Table was placed where the purple square is on the diagram. This was a column containing 3 entries which we used to check this section of the sheet. This trick may help you in developing your own sheets.

The basic contribution to profit, is the total value of sales minus the cost of the ingredients and minus the manufacturing cost.

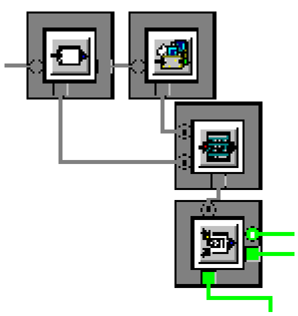


Click on each object you need help with.

Layer 2 - the Amount of Ingredients Used

Suppose we make ten tons each of X1,X2 and X3. Then we have used up Nitrates, Phosphates, Potash and Filler - but how much?

We need to know how much of each ingredient we have used in order that we don't try to use more per month than we can obtain. Layer 2 of this sheet calculates a row array showing how much of each ingredient we have used.



Click on each object you need help with.

Finishing Off

We have all the essential ingredients for the LP problem, now we must add the Solution Search object to solve this sheet. For this example we will put Solution Search on Layer 0, since this is the layer we work with most when we use the sheet.

- We set the number of constraint inputs to 3
- We set the number of output to 1, although we are changing 3 values, we are actually changing 1 array containing 3 values, so we only need one output.

Constraints

Constraint Input A	We feed in the output from the Solution Search object. It is important that all the output values be positive, since we cannot manufacture negative quantities of X1, X2 or X3.
Constraint Input B	We feed in the amount of Ingredients used we calculated on Layer 2.
Constraint Input C	We feed in the shipping limit, the limit of each ingredient that we can obtain in a month.
A >= 0	The amount we manufacture must be positive. For example {{10,10,10}} >=0 is TRUE because all of the elements are greater than zero.
B <= C	B, the ingredients used, must be less than C, the maximum ingredients we can get hold of. B and C are both arrays of the same type, so this is TRUE only if all the elements of B are less than or equal to their *corresponding* element in C.

Outputs to Vary

{{0,0,0}}	This is the only output, it is a column array which contains the amounts of each product we are to manufacture. Its start value is {{0,0,0}} notice the two sets of curly brackets indicating a column array.

Solution to Find

Our aim is to maximise the contribution to profit that we calculated on layer 1. To do this we connect this result to the optimum inlet on the Solution Search object.

Solve the Sheet

To solve the sheet, switch to Use Mode and click on the Solution Search object.

Conclusion

The Solution Search object finds a solution {{4000,8000,0}} which means that we should manufacture 4000 tons of X1, 8000 tons of X2 and no X3 to make a maximum profit of £284000.

Over Engineered

You may think there is a lot to this sheet and you would be right. This sheet is heavily over-engineered and for good reason:

- If there were more than 4 ingredients, you would simply widen the table and put in the extra values, the sheet will adjust.
- If there were more than 3 fertilizers being made, simply extend the tables and put the details in for the extra fertilizers.

- If the manufacturing costs vary for each fertilizer, change the Object Input Box for an Input Table with a column of manufacturing costs.
- For this example, the ingredients add up to the same for each fertilizer, this sheet does not make use of this assumption in the calculation.

It should be clear that this sheet can be the basis of a lot of LP problems. Simply modify the tables on Layer 0 and modify the Solution Search object's setting, the rest of the sheet will adjust automatically.

This is an Input Table into which we have placed the ratios of ingredients used. Notice that the Column and Row headings we have used on are not part of the data, the input table has been set to remove these.

This is an Input Table, into which we have placed the Price per Ton for each of the 4 ingredients. Notice that we've laid the ingredients out in columns. Keeping the products X1, X2 and X3 as the rows and the ingredients as the columns makes the sheet easier to follow.

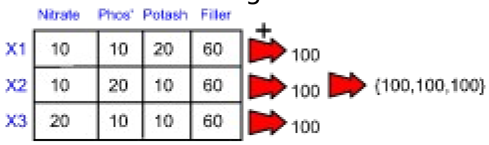
This is an Input Table into which the Shipping limits have been placed. The Filler is available in almost limitless quantities, hence its limit is well above the others.

This is an Input Table containing the Retail Price that we sell the fertilizers at. We have entered this as a column, to be consistent with the columns in the Ingredients table.

This is an Object Input Box, which is used to enter single values rather than arrays. Here it contains the cost of manufacturing per ton.

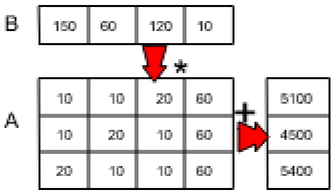
This is a Junction object, because we have used several layers for this sheet, this Junction makes it easier to take two pipes across the layers.

This is a General Toolbox Object. It is adding along the rows of ingredients to calculate how much of each of fertilizers X1, X2 and X3 we are making for each batch.



Since we are making 10+10+20+60 =100 tons of X1 and 100 tons of X2 and X3, the result from this object is {100,100,100}.

This is a Sum Product object. This is used to calculate the price of the ingredients for each of our fertilizers X1, X2 and X3. This is going to be our price per hundred tons, we will scale it later to be price per ton.



Input A is the table of how much ingredients are used for each fertilizer, Input B is the price of each ingredient.

This gives us a price of £5100 for X1, £4500 for X2 and £5499 for X3.

This is a Calculator Object, we are using it on this sheet to calculate the contribution made by each of our three fertilizers X1, X2 and X3. Notice that we are not using a Cog Object, Reporter/Announcer or any other complex object - it simply isn't necessary.

The equation to do the calculation is: **c- b/{a}-d**

{a}	Is the quantity of each fertilizer we are making per batch, the result from the General Toolbox object was a 1 dimensional array {100,100,100}, we need it as a two dimensional column array {{100,100,100}}, hence we have enclosed it in curly brackets.
b	Is the cost of making one batch of fertilizer
b/{a}	Is the cost per ton to manufacture each fertilizer. Notice that when you divide two arrays of the same type, it divides corresponding elements.
c	Is the table of retail prices
d	Is the manufacturing cost

The finished calculation looks like this:

{{83,81,81}} - {{5100,4500,5400}} / {{100,100,100}} - 11

CleanSheet performs most of the operations on corresponding elements

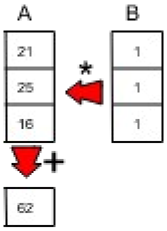
{{83-5100/100-11 , 81-4500/100-11 , 81-5400/100-11}}

Notice that subtracting 11 from an array is equivalent to subtracting 11 from each of the array's elements.

The end result is {{21,25,16}} which is the contribution to profit that one ton of each of X1, X2 and X3 makes.

This is an Output Table object, used here to check the profit contribution made by each product.

This is a Sum Product Object, it is used here to multiply the contribution array by the number of tons produced to obtain a total profit.



Input B is the amount (in tons) of each of X1, X2 and X3 we are making, as this comes from the Solution Search object, we have substituted an Input Table and used the test values 1,1,1.

If we make one ton of each, then the profit is £62.

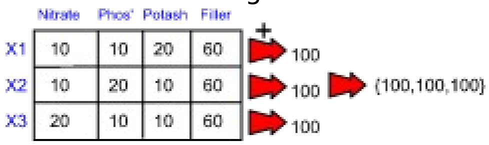
Notice that the Sum Product object returns a 1 dimensional array, so the result is actually **{62}**.

To test this section of the sheet we have to have some trial results. For this we use an Input Table placed at this point in the sheet. In the finished sheet, this table will come from the Solution Search object.

For our test we used a column containing {{1,1,1}} i.e., one ton of each product is manufactured.

This is a Junction Object, used to make connections easier.

This is a General Toolbox Object. It is adding along the rows of ingredients to calculate how much of each of fertilizers X1, X2 and X3 we are making for each batch.



Since we are making 10+10+20+60 =100 tons of X1 and 100 tons of X2 and X3, the result from this object is {100,100,100}.

This is a Calculator Object, it contains the equation:
transpose(transpose(b)/a).

a	This is the quantity we make in a batch {100,100,100}, returned by the General Toolbox object as a 1 dimensional (row) array.
transpose(b)	This is the mixture of ingredients transposed to match the row array **a**, in this example {{10,10,20,60}, {10,20,10,60}, {20,10,10,60}}. If you examine just the first sub array {10,10,20,60} you will see that this is the ingredients used to make X1.
transpose(b)/a	This breaks down as follows {{10,10,20,60}/100 , {10,20,10,60}/100, {20,10,10,60}/100} which becomes {{0.1,0.1,0.2,0.6}, {0.1,0.2,0.1,0.6}, {0.1,0.2,0.1,0.6}} in other words 1 ton of X1 contains 0.1 tons of Nitrates, 0.1 tons of Phosphates, 0.2 tons of Potash and 0.6 tons of filler.

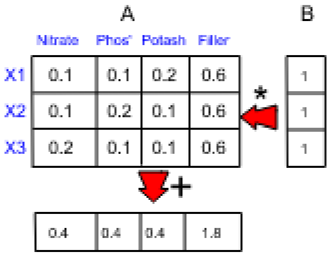
The final transpose produces an array which is the same as the Ingredients array, but scaled so that the ingredients make one ton of each of X1, X2, X3.

	Nitrate	Phos'	Potash	Filler
X1	0.1	0.1	0.2	0.6
X2	0.1	0.2	0.1	0.6
X3	0.2	0.1	0.1	0.6

This is a Sum Product object. The two inputs into this object are:

- The Ingredients list, scaled to make one ton of each product.
- The quantity in tons to make for each product. This second value comes from the Solution Search, as this object has not been placed on the sheet yet, you may wish to place an Input Table size 1 column by 3 rows to test this section separately.

This Sum Product object multiples along the rows and sums the columns to obtain the amount of each of Nitrates, Phosphates, Potash and Filler we have used.



In this example, we have made 1 ton of each of X1, X2 and X3 and used 0.4 tons each of Nitrates, Phosphates and Potash and 1.8 tons of Filler.