EDUCAÇÃO e TECNOLOGIA

Revista do Instituto Politécnico da Guarda

EDUCAÇÃO E TECNOLOGIA
Propriedade : Instituto Politécnico da Guarda
Director : **João Raimundo**
Redacção : **Serviços Centrais do IPG - Quinta do Zâmbito**
6300 Guarda * Telf. 222634 * Fax 222690
Composição : **Gabinete Editorial do IPG**
Execução Gráfica e Impressão : **Secção de Reprografia do IPG**
Depósito Legal **nº 17.981/87**
Periodicidade : **Semestral**

nº X - Julho de 1992

Capa : Vista parcial do edifício do Pólo de Seia do
Instituto Politécnico da Guarda

UM PROJECTO, UMA OBRA...

A edição deste número coincide com o final de mais um ano lectivo e outrossim com o epílogo da nova estrutura física do Instituto Politécnico da Guarda.

Símbolo da modernidade e do progresso, este Instituto é, já no presente, uma resposta credenciada às exigências das próximas décadas e uma via de futuro para os cerca de três milhares de jovens que o irão frequentar a partir de Outubro.

Será, então, ampliado neste estabelecimento de ensino superior o leque de cursos que são indispensáveis à actual e futura conjuntura de desenvolvimento regional, empresarial e industrial, cujo percurso tem de ser pautado pela necessidade de se marcar uma presença digna, activa e de qualidade no cenário europeu.

"Nómadas do mundo, teremos de ser agora sedentários conviventes nesta Europa onde sempre coubemos mal e nunca nos soubemos realizar", **como escreveu Miguel Torga.**

E esta presença tem sido bem afirmada pelo Politécnico da Guarda, através das suas múltiplas relações com estabelecimentos de ensino congéneres.

Cumpriu-se um projecto. O Instituto Politécnico é uma realidade resultante de um trabalho planificado, de uma ideia assumida, da resposta consciente a objectivos definidos, tendo subjacente a comunidade regional. O IPG é, bem poderemos dizer, uma obra impulsionada pela " força de um sonho inteiro".

João Raimundo
Presidente do IPG

SPREADSHEET MODELLING IN MANAGEMENT SCIENCE

Barrie M Baker*

Resumo: A Ciência de Gestão e Pesquisa Operacional é rica em aplicações interessantes, que poderão ser usados pelos estudantes, para praticar e desenvolver os seus conhecimentos em folhas de cálculo. Este artigo ilustra como isto pode ser feito.

É esperado que este artigo estimule os professores a utilizar as vantagens da computação moderna nos âmbitos em que os livros têm falhado, permitindo ao mesmo tempo acompanhar a revolução dos computadores.

INTRODUCTION

Students of Management Science or Operational Research are usually taught how to use spreadsheets at an early stage in their studies. These fields are rich in interesting applications which the student can use to practise and develop spreadsheet skills, simultaneously with building up knowledge of various aspects of Management Science. This paper illustrates how this can be done.

It is assumed that the student has already been given instruction on simple spreadsheet usage and that he knows a little about the menu system. The details that follow assume that the spreadsheet is Lotus 1-2-3, but a teacher will be able to adapt this to any alternative spreadsheet that is available.

The Manufacturing and Distribution Process

Figure 1 shows a simplified representation of the process

* Department of Statistics and Operational Research, Coventry University (England)

within a manufacturing firm. The firm buys in raw materials, these are processed at a factory and the finished product (or range of products) is distributed to one or more regional depots. Finally, local delivery takes place from the depot(s) to customers or towns in the region.

Various problems in Operational Research can be identified from this diagram. For example:

(i) Product mix: If a range of products is being manufactured, what quantities should be manufactured, and what combination of raw materials should be used for maximum profit? Linear programming is often used for this type of problem.

(ii) Depot location: How many depots should the firm operate with and where should they be sited, to minimise long run costs?

(iii) Vehicle routeing: Delivery from depot to customers may require several vehicles. How big should they be, which customers should each vehicle visit and what route should each vehicle then follow?

The problem of depot location lends itself to a spreadsheet analysis and this problem will be considered now.

LOCATING A SINGLE DEPOT

Supose that the locations of towns where deliveries are to be made are given by co-ordinates, $(x_j, y_j), j = 1, \ldots, n$. The weight of goods required at town j per unit time is given by w_j, and it is assumed that these weights do not vary much over time. The cost of moving one unit of the goods required at town j through one unit of distance is represented by α_j. The problem is to determine the location of a single depot, $(x_0\ y_0)$, so that long run delivery costs are minimised. Figure 2 is a pictoral representation of the situation.

It is assumed that an approximate measure of long run cost is given by

$$H = \sum_{j=1}^{n} \alpha_j w_j d_j \qquad (1)$$

where

$$d_j = ((x_0 - x_j)^2 + (y_0 - y_j)^2)^{1/2} \qquad (2)$$

the straight line distance between the depot and town j.

There are 2 well known methods of finding the co-ordinates, (x_0, y_0), that will give the minimum value of H.

The Mechanical Analogue.

A map of the region containing the delivery points is pinned or pasted to the top of a table. A hole is drilled through the table at each delivery point and a piece of string is passed through each hole. Each piece of string is tied to a ring or washer above the table, and at the end below the table a weight is attached, $w_j = k\alpha_j w_j$, for some suitable constant, k. If friction is ignored, the system comes to rest at a position which minimises potential energy, and it is easy to show that the position of the ring or washer coincides with the optimal depot location.

Although interesting and amusing, this method has some obvious drawbacks. The following method is much quicker and more accurate.

The Numeric-Analytic Method.

Re-writing the cost function as

$$H = \Sigma \alpha_j w_j ((x_0 - x_j)^2 + (y_0 - y_j)^2)^{1/2}, \qquad (3)$$

this is seen to be a function of 2 variables, that can be minimised using calculus.

Differentiating partially with respect to x_0, equating to zero and re-arranging, leads to

$$x_0 = \frac{\Sigma \alpha_j w_j x_j / d_j}{\Sigma \alpha_j w_j / d_j} \qquad (4)$$

and similarly, differentiating partially with respect to y_0 leads to

$$y_0 = \frac{\Sigma \alpha_j w_j y_j / d_j}{\Sigma \alpha_j w_j / d_j} \qquad (5)$$

It is left to the reader to fill in the intermediate steps to confirm these 2 results. These steps, and the remaining standard checks required to confirm that this solution corresponds to a minimum value for H, are given by Eilon et al [1].

The catch is that the values x_0 and y_0 are implicitly contained within the right hand sides of these expressions, since these 2 values are required in order to evaluate each d_j. Hence, the optimal depot location cannot be determined explicitly using this method, although an optimal solution would have to satisfy the expressions which have been derived.

Consequently, an iterative method has to be used. An initial depot location, (x_0, y_0), must be chosen, enabling the d_j values to be calculated. Then equations (4) and (5) can be used to calculate an improved location for the depot. Using this new location, the d_j values can be re-calculated, and this process can be repeated until there is no significant change in the position of the depot.

Intuitively, we can think of the optimal depot location being calculated as a weighted average of the x_j values and a weighted average of the y_j values. The weight used for each town, j, is proportional to $\alpha_j w_j$ and inversely proportional to its distance from the final depot location.

Up to now, the cost of moving goods from factory to depot has been ignored. However, the above analysis has never mentioned the direction of flow of goods. Therefore, the factory can be included exactly like any other customer, but with an associated weight of goods equal to Σw_j.

This is depicted in figure 3.

SETTING UP THE SPREADSHEET

Table 1, shows how a spreadsheet could be designed for an example with 22 towns. The column headings in the table are self-explanatory. It should be noted that to obtain the Greek letter α in Lotus 1-2-3 requires special techniques which a novice might find difficult. Therefore, a student attempting to set up a similar spreadsheet would probably need to substitute an alternative, such as "a" in place of α.

Columns A, B, D, and E contain the data provided for the example in rows 7-29, except in B29, where @sum (B7.B28) is entered. The other columns contain formulae which must be entered in row 7 and then copied downwards. The only formula which could present any difficulty is in column F, where an absolute cell address must be used. Calculating distances from the depot co-ordinates in H3 and I3, the formula required is @sqrt ((D7-H$3) ^2 + (E7-I$3) ^2).

@sum (G7. G29) is entered into G31 and then copied across to columns H, I and J. With the first choice of the depot location entered into H3 and I3, the improved depot location is calculated in H2 and I2 using +G31/I31 and +H31/I31 respectively. J3 contains the formula +J31, so that the effect on cost can be monitored easily.

Now, as noted in cells A1..E2, the co-ordinates for the new depot location are copied from row 2 to row 3, so that all distances are re-calculated from this location, and a further improvement to the depot location is obtained. The effect on cost is seen alongside. Note that / Range Value is required, rather than /

Range Copy, since we do not wish to copy formulae into cells H3 and I3.

The values shown in H3, I3 and J3 in Table 1 are the result of several iterations, when no significant change is occurring with each iteration.

The operation just described provides an ideal opportunity for the novice spreadsheet user to write his first macro. This simply automates the keystrokes required to carry out one iteration. The {goto} command at the start of the macro positions the cursor on cell H2. A press of the return key is indicated with~. The remaining keystrokes are then entered as shown. Note that a quote symbol must be used to enter the second line of the macro as a label, otherwise pressing "/" will bring up the menu. This macro can be entered on the spreadsheet in any convenient space. The

```
\I {goto} h2~
   /rv {right}~
   {down} ~
```

The Macro

macro must then be named, using /Range Name Create, highlighting the first cell of the set of macro commands as the range to be named, and then entering a name. Depending on the version of Lotus being used, the name may have to be restricted to a single character, such as I for iterate, preceded by a \ (back-slash). It is common practice to place the label \I alongside the first macro command as a reminder.

To run the macro, simply hold down the ALT key and press I. The change in cost will be observed, and the macro can be re-run as many times as necessary by repeatedly pressing ALT I.

A slight weakness to this spreadsheet is that it is possible for the depot position to coincide exactly with the location of a town at any iteration. A zero value for a d_j will result and division by zero will cause an error. This is not a frequent occurrence in single depot location unless the user chooses to start from one of the town locations, and the user can ignore this weakness. If such an error arises, the problem can be bypassed simply by altering one of the values in H3 or I3 by a small amount. Alternatively, the possibility of such an error can be avoided as described below under Multi-depot Location.

Having reached a stage where the cost and depot location do not change significantly from one iteration to the next, it is worth seeing how much the cost would be affected if a slightly different position for the depot was used. This is important because, in practice, it would always be necessary to find a feasible site close to the theoretically best position.

Table 2 shows the cost for different combinations of x_0 and y_0. To obtain such a table, first enter a suitable set of values for y_0 along the left hand edge and for x_0 along the top edge in a suitable space in the spreadsheet. These 2 sets of values can be entered easily using / Data Fill. The cost is shown in cell J3, so the formula +J3 is entered in the top left hand corner of the required table, where the value 2147948 is shown. Then select / Data Table 2, and highlight the range for the table, including the y0 and x0 values along the edges. For input cell 1, enter I3, and for input cell 2, enter H3. This produces the complete table as shown.

As each feasible site is considered, the corresponding cost can be read from Table 2. It is seen that the percentage increase in cost is quite small if the final site is reasonably close to the optimal solution. This also suggests that small variations in demand over time will not be too important.

The next step is to produce a graphical representation of the town and depot locations. To include the depot, enter formulae +H3 and +I3 into D30 and E30 respectively. Then select / Graph Type XY. For the X range, highlight D7..D30, for the A range, highlight E7..E29, and for the B range, select E30. Finally, select options Format Graph Symbols Quit Quit View. Figure 4 shows the graph obtained. Once the graph has been set up with these steps, it can be re-displayed at any time simply by pressing the function key F10.

MULTI-DEPOT LOCATION

The numeric-analytic method can be extended to multi-depot location with the following steps.

1. Decide on the number of depots and choose their initial locations.

2. Allocate each town to a depot with smallest transportation cost.

3. Apply the adjustment formulae to each depot, using the subset of towns allocated to that depot and the factory with the corresponding subtotal of weights.

4. If any significant change in any depot location occurs at step 3, return to step 2.

Figure 5 depicts the situation when 2 depots are used. On completion of the above steps, each depot will be located optimally relative to the factory and the towns that have been allocated to it.

Figure 6 shows the process by which each town is allocated to a depot with smallest transportation cost.

Let d_{0i} = distance from the factory, 0, to depot i,
d_{ij} = distance from depot i to town j,
α_j = cost of moving one unit of the goods required at town j through one unit of distance,
β_i = cost of moving one unit of goods sent from factory to depot i through one unit of distance.

Then if town j is allocated to depot i, the total transportation cost for each unit required at town j is

$$\beta_i d_{0i} + \alpha_j d_{ij}. \qquad (6)$$

At step 2, this quantity must be calculated for every combination of depot and customer, and then each customer is allocated to the depot that gives the smallest cost. At step 3, the position of each depot changes slightly, and as a result it may be more economical for some towns to switch to different depots. Thus step 2 is repeated at every iteration.

An important feature of multi-depot location is that local optimal solutions can occur, and the solution obtained by the above steps depends on the initial depot locations that are chosen. Therefore, it is necessary to repeat the whole process using several different sets of initial depot locations, and the best of the local optima that are obtained is used. It cannot be guaranteed that the best of these local optima will be a global optimum.

Setting up the Spreadsheet

It is best to set up a spreadsheet that will handle several depots, say 5. Then the same spreadsheet can be used for smaller numbers of depots simply by setting β_i to a large number, for i greater than the number of depots. Such a spreadsheet is inevitably much larger than for the single depot case, as shown in Tables 3 to 6.

Table 3 shows how the top section of the spreadsheet has been modified to show the old and new co-ordinates of the 5 depots, together with the values of β_i. In this case, only 2 of the 5 possible depots are being used. An @if is used with the calculation of the new depot co-ordinates, so that the adjustment formula will be applied only when the denominator is non-zero, and otherwise an asterisk is displayed.

Table 4 shows how extra columns have been inserted to calculate distances between the towns and each of the depots, and to calculate the unit transportation costs from factory to town.

Rows 29 to 33 represent 5 copies of the factory, the weights supplied to each depot and the distances from the 5 depots. Cells A29 to A33 actually show the values of $\beta 1$ to $\beta 5$, and contain the formulae +U4, +W4, +Y4, +AA4 and +AC4. Cells D35.. E39 contain simple formulae to repeat the depot locations in table 3, to enable graph plotting as in the single depot case.

Table 5 represents the columns immediately to the right of those in table 4. Column P contains a simple @min formula for columns G, I, K, M and O. Columns Q and R contain nested @if formulae to identify the depot giving the minimum cost and to record the distance between that depot and the town. Then the remaining columns in table 5 contain simple @if formulae to calculate the terms in the adjustment formulae for each depot. The @if enables a zero to be recorded wherever a depot does not supply a customer, so that a simple @sum at the bottom of each column will give the correct total for each depot.

Table 6 shows the last 6 columns of the spreadsheet. Column AI is just the product of columns C and R for rows 7 to 28. But in rows 29 to 33, column AI contains the products of column C with columns F, H, J, L and N respectively. Thus, an @sum for column AI gives the total cost. The last 4 columns each contain a simple @if to enable the weight of goods to be totalled separately for each depot. Thus, B29 contains the formula @sum (AJ7.. AJ28), and the formulae for B30 to B33 are entered similarly.

Finally, column S is used to avoid the possibility of dividing by zero, if any of the depot locations ever coincides exactly with a town. This usually occurs only if a depot has just one town allocated to it. Recall that an optimal depot location is a weighted sum of the co-ordinates of the towns supplied by that depot, with the weight for each co-ordinate being inversely proportional to the distance of the town from the depot. If the distance is zero, the weight for that town becomes infinite. A simple way of avoiding the problem is to pretend that the depot and town are separated by a very small distance, so that a very large weight is used, rather than an infinite weight. Thus, column S contains a simple @if to replace a zero value in column R with the value shown at the head of column S, 0.1 in this case. Then in columns T to AH, the value in column S is used for the distance, instead of the value in column R. From a theoretical standpoint, this technique of dealing with zero dj is not the most precise, but it works satisfactorily in view of the limited circumstances under which zero dj arises, and it is much simpler on the spreadsheet than the theoretical ideal.

The macro is modified as shown. The first line of the macro positions the cursor so that the cost will be displayed as each iteration takes place. The second line of the macro positions the cursor correctly for the start of the copying process on the third line. The range of cells to be highlighted for this process depends on the number of depots, and this is calculated in a cell just above the start of the macro. This cell is given the range name COVER, as indicated by the label placed in the cell to the left of it.

```
COVER 2*r3-1
      \I {goto}p1~
         {right 5} {down}
         /rv {right cover}~
         {down}~
```

Modified Macro

Clearly, it will take the student much longer to set up the multi-depot spreadsheet compared to the single depot case. Careful use of absolute cell addresses must be used to enable formulae to be copied wherever possible. There may be some cells where the best tactic is to copy the formula from a nearby cell and then edit it, rather than enter the whole formula.

SOME INVESTIGATIONS FOR THE STUDENT

When the student has set up the spreadsheet for the single depot case, it is interesting to see what effect different starting positions have, and to see how the depot location changes with each iteration. The macro can be modified so that at each iteration, the depot co-ordinates are recorded on the spreadsheet. When the optimal depot location has been reached, it is then possible to produce a graph showing all the depot locations throughout the iterative process. Such a graph should be produced for several different starting positions, including some which are obviously far from optimal. There are various ways of modifying the macro to achieve this.

If the student progresses to the multi-depot case, then start with 2 depots and try various different starting positions. In particular, try starting with both depots on a north-south line roughly through the centre, one depot well to the north, the other to the south. Then try starting both depots on an east-west line and compare the results.

Then try increasing the number of depots and again, try some different starting positions. Compare the best costs found with 1, 2, 3, 4 and 5 depots. Discuss how the final decision will be made on the number of depots to use.

A ROUTEING EXERCISE

The problem of vehicle routeing was mentioned near the start of this paper. At the heart of this is the famous travelling salesman problem: find a route that visits each of a given number of cities and returns the traveller to his starting point with minimum total distance. This is well known to be a difficult problem to solve optimally when there are many cities involved. However, the student can be introduced to the problem and can practise his spreadsheet skills with the following exercises, which are fairly straightforward.

1. A salesman is based at point A, with co-ordinates (0,0). He is going to visit the following locations in the order shown and then return to base.

	B	C	D	E	F	G	H	I	J
X:	7	21	35	39	58	54	67	49	19
Y:	24	37	30	50	42	28	23	10	5

(i) Assuming straight line distances, how far will the salesman travel?

(ii) Produce a diagram depicting the salesman's journey.

(iii) The salesman has just been told to include location z (47,36) on his tour. He wants to increase his total distance as little as possible. Where should Z be inserted in the tour and how much does this increase the distance travelled?

2. The following day, the locations to be visited are:

	K	L	M	N	O	P	Q	R	S
X:	20	74	94	22	93	45	16	4	32
Y:	17	49	70	15	29	4	91	23	50

Suggest an order for visiting these locations to give minimum total distance, starting and finishing at A. What is the distance travelled?

For exercise 1, the co-ordinates can be entered into columns B and C in the order that they are given, but with city A included at both the top and bottom of the list. Column A can be used to label the cities. Then in column D, row 2, a simple square root formula can be entered to calculate the distance between cities A and B. This formula can be copied downwards, and an @sum at the bottom of column D gives the total distance travelled.

The diagram is straightforward to produce. The default setting for the format of an XY graph will join the points with lines in the order that they are entered into columns B and C, thus showing the route followed by the salesman. / Graph Options Format Graph Both Quit Quit View can be used to confirm this format.

Now enter the co-ordinates of city Z in a suitable space on the spreadsheet. At the top of column E, enter a formula for the distance of Z from city A. Use an absolute cell adress for Z, so that the formula can be copied down the column. Then in F2, enter the formula +E1+E2-D2. This is the distance between A and Z plus the distance between B and Z, minus the distance between A and B, which is the extra distance if Z is visited between A and B. Copy this formula down through column F and the smallest value shows the best place to insert Z in the tour, assuming that the other cities are still visited in the same order.

Exercise 2 can be solved by inspection once the graph has been displayed. The idea is to sort the points into the chosen order so that the route can be displayed and the distance can be calculated using the formulae entered for exercise 1.

First, insert a new column at the left of the spreadsheet for sorting on. This becomes column A. The co-ordinates from exercise 1 and the city labels can be overwritten without changing the rest of the spreadsheet. Make a mental note of the co-ordinates of K and then press function key F10 to display the graph. The points will be joined in the order that they have been entered, A, K, L, ..., S, A. Thus it is possible to identify the correct letter for each point on the screen, having noted the co-ordinates of K so as to ensure that you are working round the route in the right direction. Now write down the numbers 1 to 10 and against them the letters in the order that you think they should be visited. Return to the spreadsheet and in column A, enter the appropriate number against each of the letters in column B. Use / Data Sort to re-order the cities so that the numbers in column A are in ascending order. Press F10 to see the route chosen.

CONCLUSION

The examples presented here will give students practice at using a range of spreadsheet capabilities. They serve the dual purpose of introducing students to some important topics in Management Science. Experience has shown that students find this an enjoyable process.

There are many more applications for spreadsheet modelling in Operational Research and Management Science, in areas such as Forecasting, Simulation and Dynamic Programming for example. It is hoped that this paper will stimulate teachers to make full use of modern computing facilities in areas where text books have failed to keep pace with the computing revolution.

REFERENCE

1. Distribution Management, S. Eilon, C. D. T. Watson-Gandy, N. Christofides, Griffin London, 1971.