1

Operational Research Using a Spreadsheet

1.1. Foreword

In this chapter, I will present various possible ways of using Microsoft Excel for solving classical problems linked to operational research.

The version that I use is part of the Microsoft Office 2013 Suite. Most of the exercises shown can easily be adapted for other spreadsheets provided they have similar functions.

1.2. Dynamic programming

Here, we will examine a problem we first encountered in section 4.2.3 of the first volume of this series, the famous knapsack problem (KP).

I will present a solution here based on a spreadsheet linked to two procedures written in Visual Basic Application (VBA).

To begin with, we must first create the table in Figure 1.1 in a spreadsheet.

Once the input has been carried out, place the following formulae in the indicated cells:

– `E1: =COUNTA(B4:K4)`, calculates the number of objects available;

– `B24: =TOTAL(B23:T23)`, calculates the total value of the objects;

– now save your table;

– switch to VBE, create a module for your project (DEVELOPER tab, VISUAL BASIC icon, right click on the project file, INSERT then MODULE);

– open the module that has just been created, normally MODULE 1 and input the following lines of code:

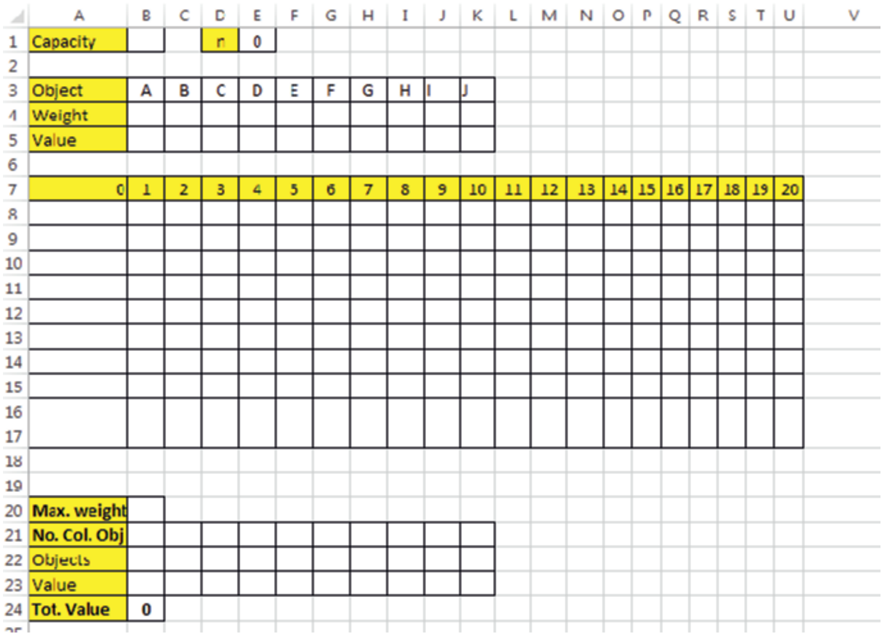


Figure 1.1. *The table to be created in the spreadsheet*

```
Sub Clear()↵
    'Clears the contents of the initial table↵
    rep = MsgBox("Do you wish to clear the tables'
contents to begin a new calculation", vbYesNo,
"Clear")↵
    If rep = vbYes Then↵
        Range("B1").ClearContents↵
        Range("B3:K5").ClearContents↵
        Range("A8:U17").ClearContents↵
        Range("B20:K23").ClearContents↵
    End If↵
End Sub↵
↵
Sub Knapsack()↵
```

```
    rep = MsgBox(“Have you entered the capacity and
values in the table above to begin a new calculation
?", vbYesNo, "Calcul")↵
    If rep = vbNo Then Exit Sub↵
    ‘Declaration of variables and tables↵
    Dim i, j, m1, m2 As Integer↵
    'Table of weights↵
    Dim p() As Integer↵
    'Table of values↵
    Dim v() As Integer↵
    Dim c As Integer↵
    ‘Allocation of the capacity to the variable c↵
    c = Cells(1, 2)↵
    Dim n As Integer↵
    ‘Allocation of the number of objects to the
variable n↵
    n = Cells(1, 5)↵
    ReDim p(n)↵
    ReDim v(n)↵
    ‘Objects matrix↵
    Dim table() As Integer↵
    ReDim table(n, c)↵
↵
    ‘Reading object weight↵
    For i = 1 To n↵
        p(i) = Cells(4, i + 1)↵
    Next↵
↵
    'Reading object weight↵
    For j = 1 To n↵
       v(j) = Cells(5, j + 1)↵
    Next↵
↵
    Initializing the table, setting its content to 0↵
    For j = 0 To c↵
        table(0, j) = 0↵
    Next↵
↵
  ‘Constructing and filling the matrix↵
    For i = 1 To n↵
        For j = 0 To c↵
            If j >= p(i) Then↵
                m1 = table(i - 1, j)↵
```

```
                m2 = table(i - 1, j - p(i)) + v(i)↵
                If m1 > m2 Then table(i, j) = m1 Else
table(i, j) = m2↵
            Else↵
                table(i, j) = table(i - 1, j)↵
            End If↵
            Cells(7 + i, j + 1) = table(i, j)↵
        Next↵
    Next↵
↵
  ‘Reading objects↵
    Dim ob(8) As String↵
    For u = 1 To n↵
       ob(u) = Cells(3, u + 1)↵
    Next↵
↵
  ‘Further reallocation of variables↵
    i = n↵
    j = c↵
↵
    'Determining maximal weight↵
    Dim k, m As Integer↵
    m = 2↵
    For k = c To 0 Step -1↵
        If table(i, j) = table(i, j - 1) Then j = j -
1↵
    Next↵
    Cells(20, 2) = j↵
↵
    'Obtaining the list of objects in the table of
results↵
    While i > 0 And j > 0↵
        If table(i, j) <> table(i - 1, j) Then↵
            j = j - p(i)↵
            Cells(22, m) = ob(i)↵
            Cells(23, m) = v(i)↵
            Cells(21, m) = i↵
            i = i - 1↵
            m = m + 1↵
        Else↵
            i = i - 1↵
        End If↵
    Wend↵
End Sub↵
```

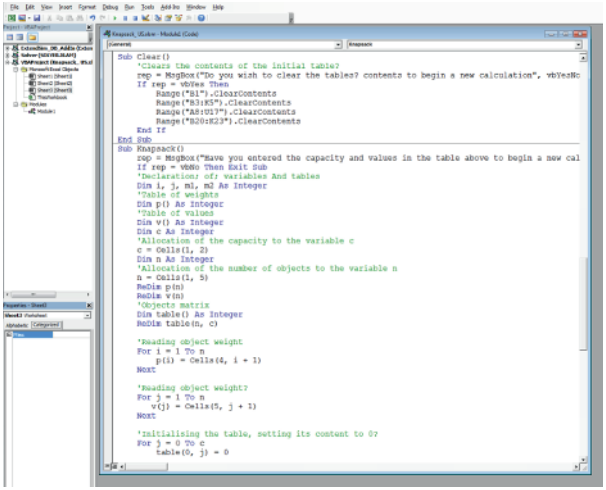


Figure 1.2. *The VBA code window in VBE attached to Module 1 of the project*

To launch these two procedures, `Clear()` and `Knapsack()`, create two buttons by inserting two forms to which you will assign each of the two macros (right click on form then ASSIGN MACRO…). Your spreadsheet should look like the one in Figure 1.3.

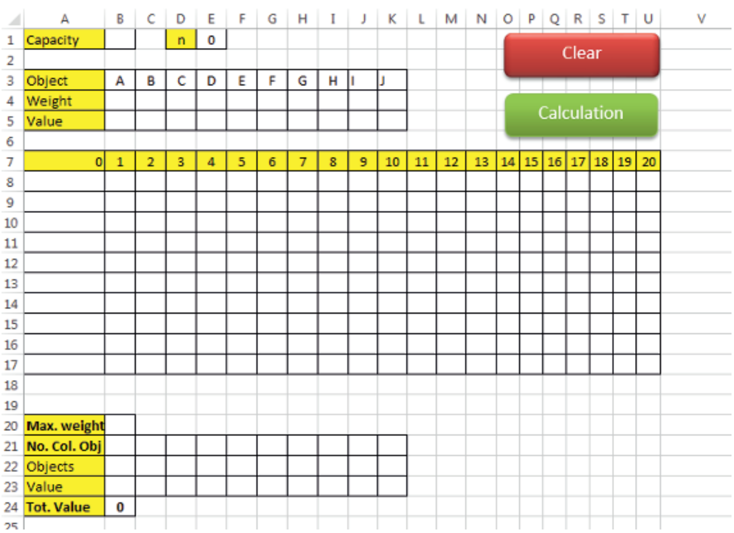


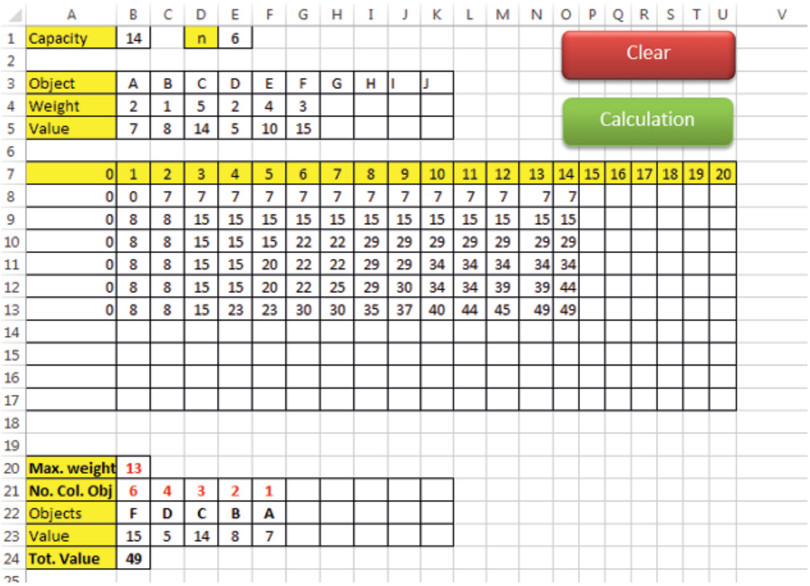
Figure 1.3. *An accounting spreadsheet containing two (Delete and Calculate) procedure buttons (macros)*

In order to use the KP calculation, all you need to do is click on the CLEAR button to erase any calculation that has already been carried out and to delete the calculated contents.

Secondly, you should input the data in the upper table (`B1` and `B3` to `K5`).

For the final stage, click on the CALCULATE button to launch the calculation and to show the results.

COMMENT 1.1.– In the spreadsheet that we have just created, you can manage up to 10 objects. Nothing will stop you from increasing this number by making the necessary modifications to the tables and the relevant VBA procedures.



Capacity	14		n	6

Object	A	B	C	D	E	F	G	H	I	J
Weight	2	1	5	2	4	3				
Value	7	8	14	5	10	15				

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	7	7	7	7	7	7	7	7	7	7	7	7	7						
0	8	8	15	15	15	15	15	15	15	15	15	15	15	15						
0	8	8	15	15	15	22	22	29	29	29	29	29	29	29						
0	8	8	15	15	20	22	22	29	29	34	34	34	34	34						
0	8	8	15	15	20	22	25	29	30	34	34	39	39	44						
0	8	8	15	23	23	30	30	35	37	40	44	45	49	49						

Max. weight	13									
No. Col. Obj	6	4	3	2	1					
Objects	F	D	C	B	A					
Value	15	5	14	8	7					
Tot. Value	49									

Figure 1.4. *The example of paragraph 4.2.3.3 calculated. The results can be seen in the bottom table*

1.3. Scheduling

In this section, I will describe two possible ways of using the Excel spreadsheet:

– as a method for calculating the critical path and the duration of the project based on a calculation using a matrix (double entry table);

– for creating a simple Gantt chart using a precedence table.

1.3.1. *Critical path calculation matrix*

Using a Microsoft Excel spreadsheet containing a matrix constructed around several calculation formulae, we will be able to determine the critical path and the duration of a project (see section 5.8 of Volume 1 [REV 17a]).

The table shown is limited to a PERT chart with at most 10 points. It can easily be adapted to a larger number of lines and columns by applying the same logic in order to create the missing calculation formulae.

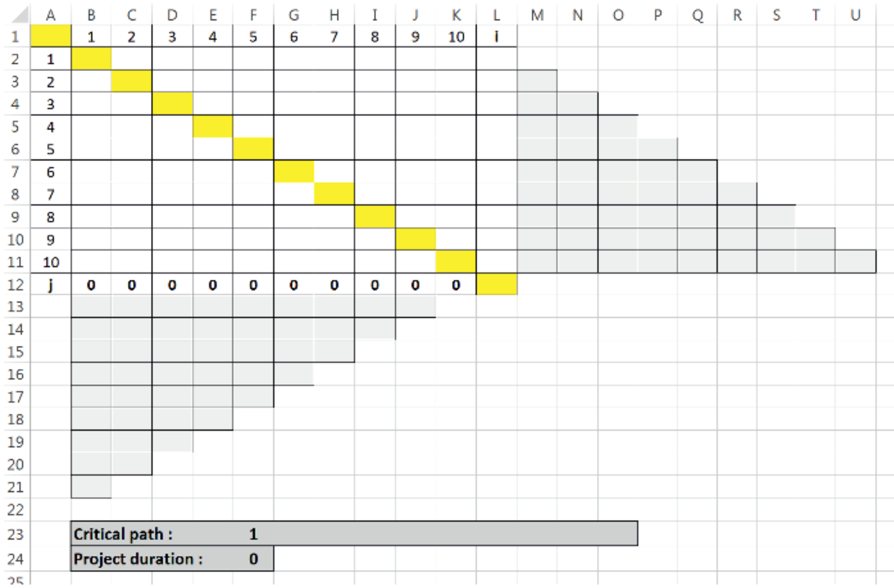
In order to present and explain how this spreadsheet functions, we will re-use Exercise 2 from section 5.14.2 of Volume 1 [REV 17a], whose matrix is shown in Table 1.1.

	1	2	3	4	5	6	7	8	i
1		2							0
2			10	6	8				2
3					0				12
4					4	10	18		8
5							5		12
6							4		18
7								4	26
8									30
j	0	2	21	8	21	22	26	30	

Table 1.1. *The double entry matrix from Exercise 2*

Let's now create an accounting spreadsheet like the one in Figure 1.5.

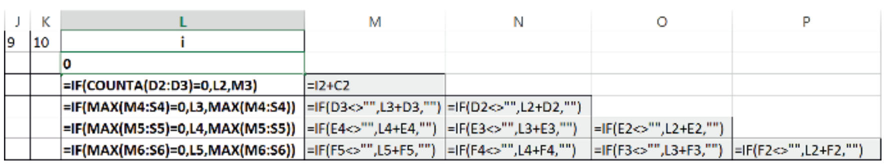
We can see that the matrix created is designed for 10 points. The grayed-out areas will contain calculation formulae as well as the columns from the beginning for the earliest start and latest end dates *i* and *j*.



	A	B	C	D	E	F	G	H	I	J	K	L
1		1	2	3	4	5	6	7	8	9	10	i
2	1											
3	2											
4	3											
5	4											
6	5											
7	6											
8	7											
9	8											
10	9											
11	10											
12	j	0	0	0	0	0	0	0	0	0	0	
23		Critical path :				1						
24		Project duration :				0						

Figure 1.5. *The table to be created in the spreadsheet*

Let's begin with the section on the right, in Figure 1.6. Here, you can see the formulae to be inputted (only cells `L3` and `P6` are shown – the others follow the same pattern).



J	K	L	M	N	O	P
9	10	i				
		0				
		=IF(COUNTA(D2:D3)=0,L2,M3)	=I2+C2			
		=IF(MAX(M4:S4)=0,L3,MAX(M4:S4))	=IF(D3<>"",L3+D3,"")	=IF(D2<>"",L2+D2,"")		
		=IF(MAX(M5:S5)=0,L4,MAX(M5:S5))	=IF(E4<>"",L4+E4,"")	=IF(E3<>"",L3+E3,"")	=IF(E2<>"",L2+E2,"")	
		=IF(MAX(M6:S6)=0,L5,MAX(M6:S6))	=IF(F5<>"",L5+F5,"")	=IF(F4<>"",L4+F4,"")	=IF(F3<>"",L3+F3,"")	=IF(F2<>"",L2+F2,"")

Figure 1.6. *The section on the right with the calculation formulae to be inputted*

We will comment on these formulae that fit with the calculation technique described in section 5.8.3 of Volume 1. All they do is automate the approach:

– `M3: =L2+C2`, adds the earliest start date for the first task to the first value encountered;

– L3: =IF(COUNTA(D2:D3)=0,L2,M3), checks for the absence of a value in D2 and D3, if the value is absent, it displays the previous earliest start date, which is L2, otherwise nothing is displayed;

– M4: =IF(D3<>"",L3+D3,""), checks for the presence of a value in D3 and in this case effects the sum L3+D3, otherwise nothing is displayed;

– N4: =IF(D2<>"",L2+D2,""), identical to the previous test for cell D2. The other group of grey cells, from M5 to U11, behaves in the same way, by taking account of each of the values present or absent in the column in question;

– L4: =IF(MAX(M4:S4)=0,L3,MAX(M4,S4)), checks the maximal value calculated in the column (the result of the calculations placed in cells M4 to S4). If it is equal to 0, no values will have been calculated, we can therefore assume that all the points have been reached, in which case the last value calculated from the beginning to the end should be copied, which is L3, otherwise the maximum is displayed. The same goes for the other formulae placed from L5 to L11 with the corresponding cells.

Let’s now look at the lower part, cells K12 to B21. As was the case with the previous example, only a few formulae are shown. The others follow the same pattern:

– K12: =L11, returns to the last value in the column of earliest start dates;

– C13: =IF(D3<>"",D12-D3,""), checks if the cell located below the diagonal is empty and if it is not, calculates the difference with the corresponding cell, which is D12-D3, otherwise nothing is displayed. This calculation is identical for all the other cells in the grey zone if the cell references are changed to obtain the appropriate result (see Figure 1.7);

– C12: =IF(MIN(C13:C21)=0,D12,MIN(C13:C21)), checks the minimal value calculated in the column (the result of the calculations in cells C13 to C20). If it is equal to 0, no values will have been calculated, we can therefore assume that the vertex has not been reached, in which case the last value calculated for the latest end date is copied, which is cell D12, otherwise the minimum is displayed. The same goes for the other formulae located in D12 to J12 with the corresponding cells;

– B12: =MIN(B13:B21), calculates the minimal value from zone B13 to B21, to display the latest end date of the first vertex in the graph.

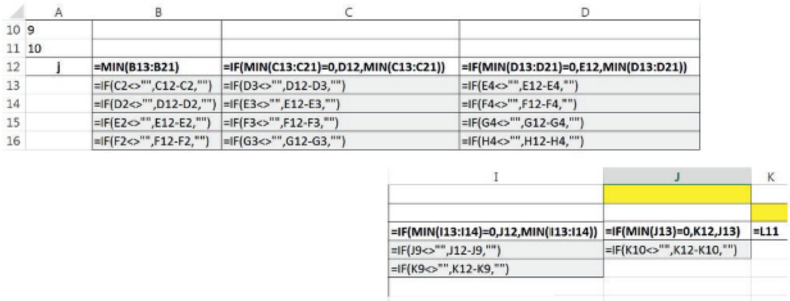


Figure 1.7. *The cells located below the CPM matrix with their formulae*

We will now concern ourselves with find the critical path through the graph node numbers. The length of the project is associated with the last value for the earliest beginning, `L11`:

– `F23: =IF(B12=L2,B1,"")`, tests if cell `B12` is equal to cell `B2` and displays the node number located in `B1`. This formula is not compulsory, it can simply be replaced by `=L2`, since the first node is always in the critical path, except in specific cases;

– `G23: =IF(MAX(M3)<>0,IF(C12=L3,C1,""),"")`, we find two interlinked tests. The first verifies that M3 is different to 0, if this was not the case, this would mean that the current node does not belong to the critical path and nothing would be displayed. If `M3` is different to 0 then we check that `C12` is equal to its corresponding cell, `L3`, on the diagonal in the matrix and the node number `C1` is displayed, otherwise nothing is displayed;

– `H23: =IF(MAX(M4:N4)<>0,IF(D12=L4;D1,""),"")`, identical to the above, but with a zone of `M4` to `N4` to test the maximum;

– the same principle is then applied for the cells from `I23` to `O23` with the corresponding cell references.

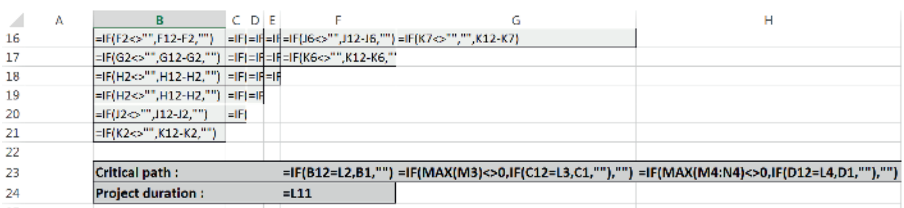


Figure 1.8. *Calculating the critical field and display showing the duration*

See Figure 1.9 for the results obtained after inputting the figures from the table referred to earlier.

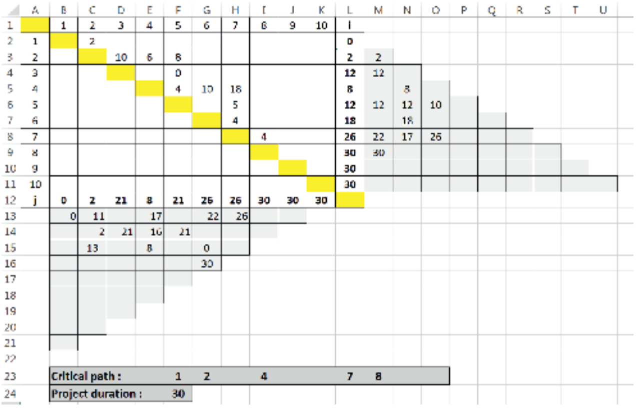


Figure 1.9. *Spreadsheet with all cells filled in the matrix. Note the values of I (earliest start) and j (latest end) as well as the length of the project and the critical path that passes through nodes: 1, 2, 4, 7 and 8*

1.3.2. *Classic Gantt chart*

In order to create this chart, we will use the public address system exercise dealt with in section 5.5 of Volume 1. In the precedence table (Table 1.2) only the columns for task and duration (in minutes) have been retained with the addition of two new columns, start and end.

Task	Start	Duration	End
A	0	120	120
B	120	20	145
C	120	15	140
D	120	20	140
E	140	10	155
F	140	15	155
G	155	25	180
H	140	20	180
I	180	15	195

Table 1.2. *Tasks, starts, durations and ends for the sonorization exercise*

This table is inserted into a Microsoft Excel spreadsheet and we then add a variance column that will calculate the difference between "End" and "Start" (in the cell `E2: =D2-B2`) in order to obtain the matrix in Figure 1.10.

Task	Start	Duration	End	Variation
A	0	120	120	120
B	120	20	145	25
C	120	15	140	20
D	120	20	140	20
E	140	10	155	15
F	140	15	155	15
G	155	25	180	25
H	140	20	180	40
I	180	15	195	15

Figure 1.10. *The matrix corresponding to the task table with each of its columns*

Next, select the table, then carry out the following group of operations:

– INSERT tab, CHARTS toolbar, select INSERT A BAR CHART, select STACKED BARS in 2D BAR (Figure 1.11);

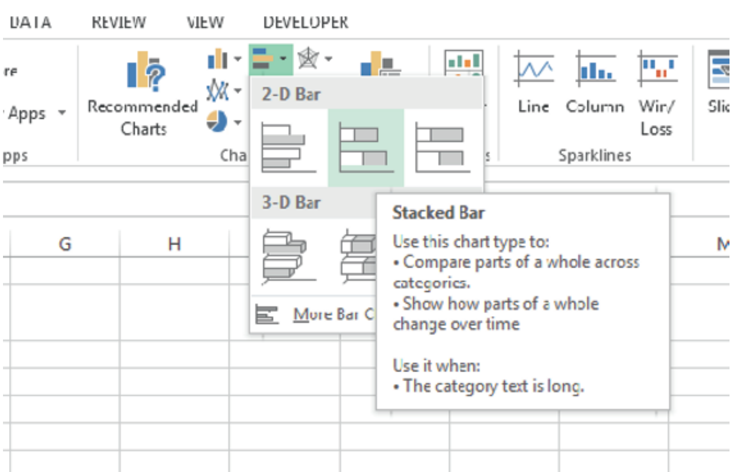


Figure 1.11. *Microsoft Excel: insert a 2D STACKED BAR CHART*

– finally, move the graph and resize it (Figure 1.12);

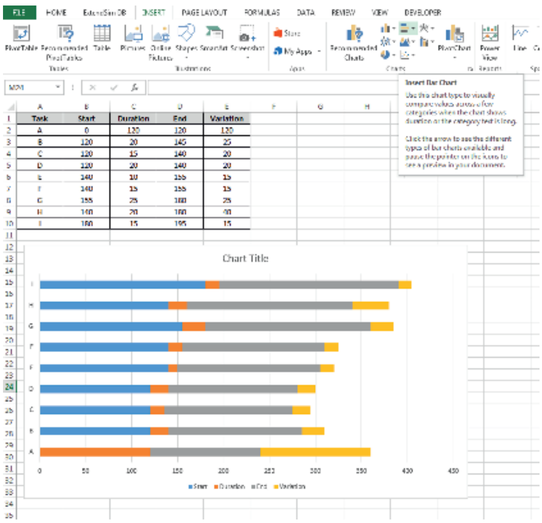


Figure 1.12. *The resized graph, positioned under the matrix*

– click on the y-axis;

– right-click, select FORMAT AXIS (Figure 1.13);

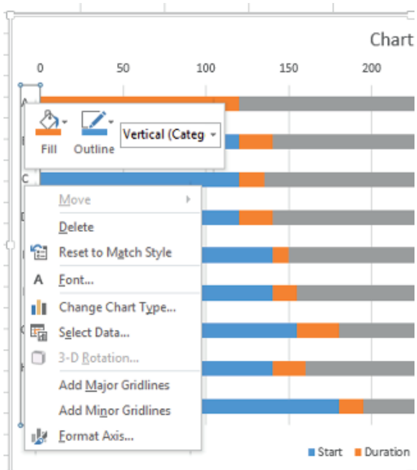


Figure 1.13. *The contextual menu (right click) to format the axis*

– in the FORMAT AXIS window, select CATEGORIES IN REVERSE ORDER (Figure 1.14);

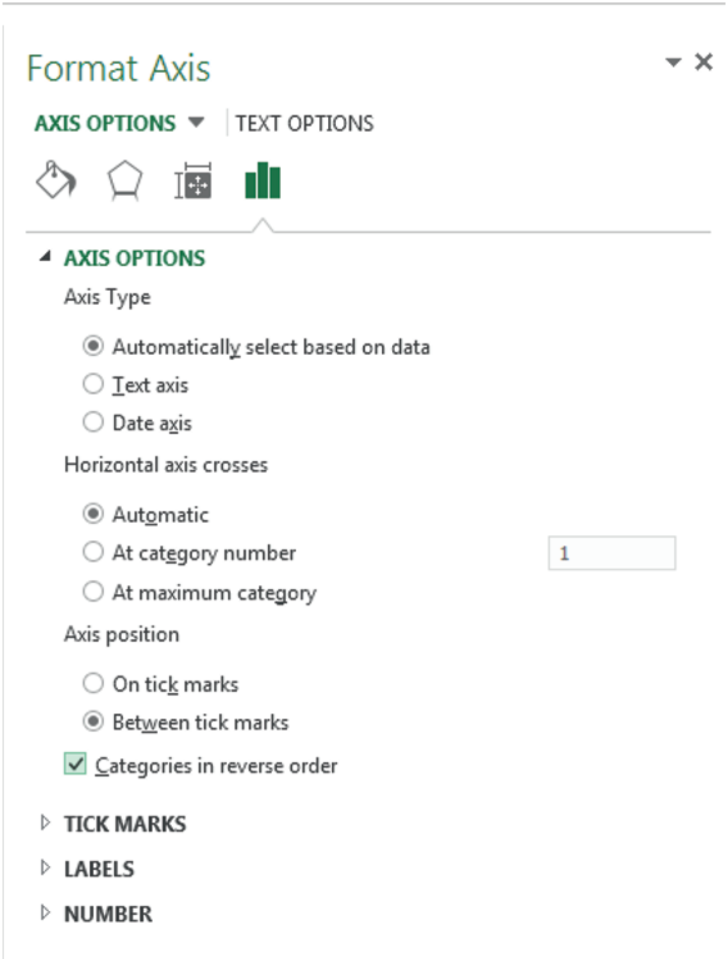


Figure 1.14. *The window for configuring the axis in Microsoft Excel 2013, including the checkbox: values in reverse order (in axis options)*

– click on the x axis to select it;

– in the FORMAT AXIS window, open drop-down LABELS and select HIGH in LABEL POSITION;

– enter 10 and 5 respectively in AXIS OPTIONS as MAJOR and MINOR UNITS (Figure 1.15);

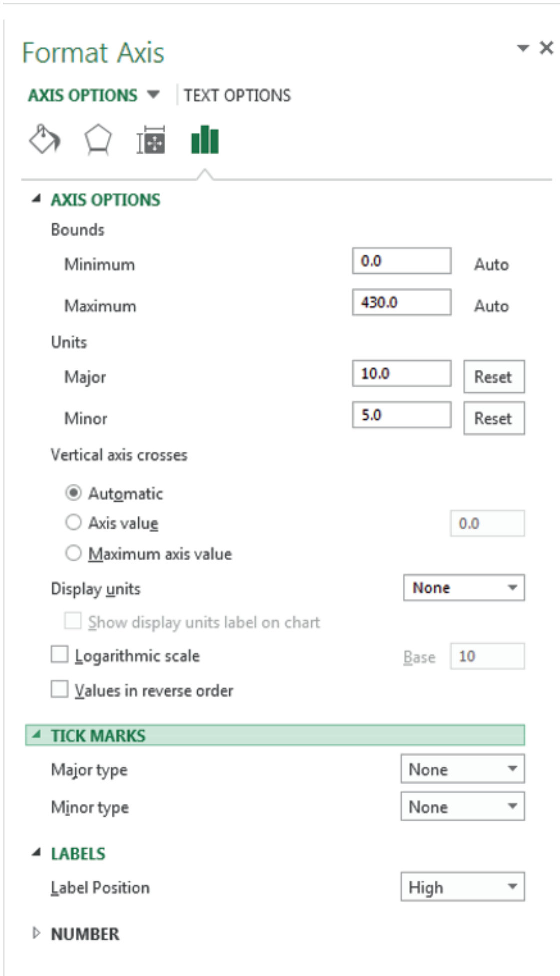


Figure 1.15. *Configuring axes (10 and 5) and labels (high)*

– right click on the middle of the graph and click on SELECT DATA;

– a window for selecting a data source will open. Tick “Duration” and “End Date” (you can also remove these using the DELETE button) then validate by clicking the OK button (Figure 1.16);

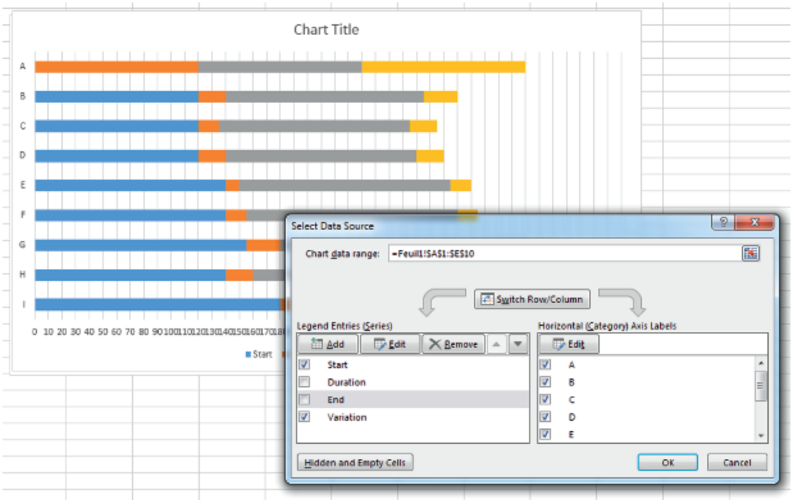


Figure 1.16. *The fields "length" and "end" selected in the window displaying the data source*

– click on the legend and delete it (press DEL or backspace on Mac);

– right click on the "Start" bar on the graph, specify NO OUTLINE and NO FILL by clicking on the corresponding icons (OUTLINE and FILL) in the pop-up tool bar (Figure 1.17);

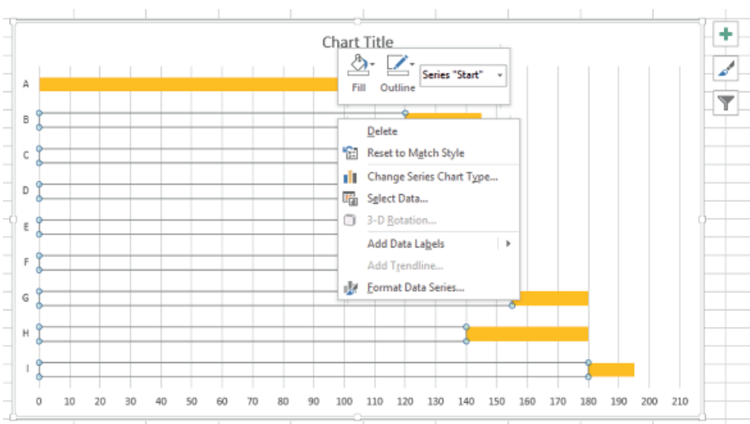
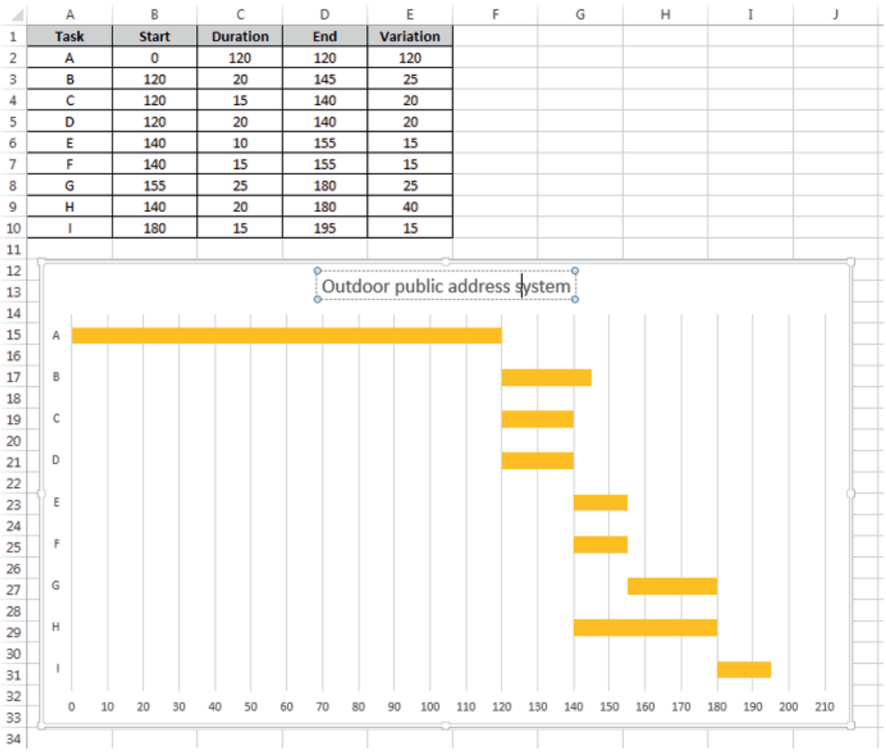


Figure 1.17. *The menu and the pop-up tool bar, obtained by right-clicking on the "Start" bar. Note the tools OUTLINE and FILL above the menu*

– double-click on the title to change it and type “Outdoor public address system”;

– to finish, save your work (Figure 1.18).



Task	Start	Duration	End	Variation
A	0	120	120	120
B	120	20	145	25
C	120	15	140	20
D	120	20	140	20
E	140	10	155	15
F	140	15	155	15
G	155	25	180	25
H	140	20	180	40
I	180	15	195	15

Figure 1.18. *The finished Gantt chart and its associated table in a Microsoft Excel 2013 spreadsheet*

1.3.3. *Gantt chart with a calendar*

In cases where dates have to be managed according to a calendar that can show workdays and holidays, we must consider excluding the latter from the project timeline.

Table 1.3 gives an example involving these conditions.

Installing an air conditioning system (simplified agenda) (see Table 1.3).

Task	Appointment	Duration (days)	Previous
A	Needs analysis (electrical equipment, piping, fittings, coolant gas and various other components)	1	–
C	Order and delivery of electrical equipment	5	A
D	Order and delivery of the group exchanger and diffusers	8	A
E	Order and delivery of coolant gas	4	A
F	Assembly, installation and laying of the group exchanger	2	D
G	Installation and laying of diffusers	4	D
H	Laying and connecting pipes	2	B, F, G
I	Electrical connection	1	C, F, G
J	Filling the installation, pressurization and clean-up	1	E, H
K	Switching on, testing installation and tuning	1	H, I, J
L	Tidying up the work site	1	K

Table 1.3. *The precedence table for the installation of an air-conditioning system*

For this second exercise, we will consider the project start date as being Monday March 2nd 2015.

We will begin by entering a table in Microsoft Excel similar to the one shown in Figure 1.19. The dates for the start and the end have been calculated in a standard way using a PERT or MPM type method (the MPM chart is shown in Figure 1.20).

	A	B	C	D	E	F
1	Project start date :		03/02/15			
2						
3	Task	Start	Duration	End	Variation	
4	A	03/02/15	1	03/03/15	1	
5	B	03/03/15	10	03/17/15	14	
6	C	03/03/15	5	03/10/15	7	
7	D	03/03/15	8	03/13/15	10	
8	E	03/03/15	4	03/09/15	6	
9	F	03/11/15	2	03/13/15	2	
10	G	03/11/15	4	03/17/15	6	
11	H	03/15/15	2	03/17/15	2	
12	I	03/15/15	1	03/16/15	1	
13	J	03/17/15	1	03/18/15	1	
14	K	03/18/15	1	03/19/15	1	
15	L	03/19/15	1	03/20/15	1	

Figure 1.19. *The table to be created in Microsoft Excel (here the critical tasks are in gray and in bold)*

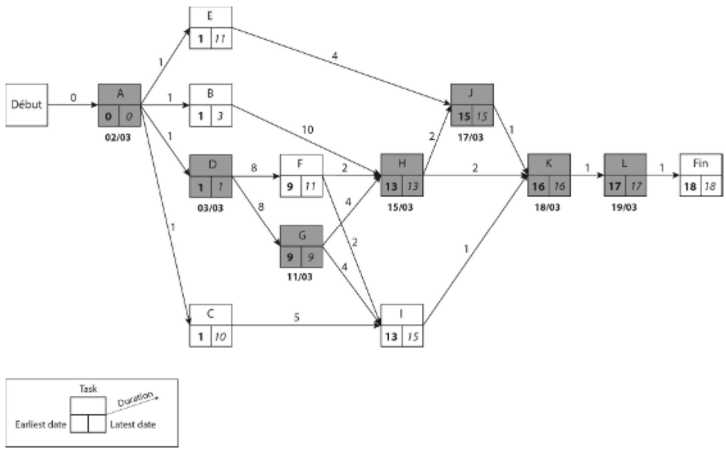


Figure 1.20. *The MPM chart corresponding to the project: installation of an air conditioning system*

In the cell D4 (End), we have the formula =B4+C4, that we will fill downward as far as D15.

In the cell E4 (Variance), we have the formula =D4-B4, that we will fill downward as far as E15.

It should be noted that the end dates do not take into account a week where only the days from Monday to Friday will be workdays. To compensate for this we can replace the formula, located in D4, by =SERIES.WORK.DAY(B4;C4). This gives the table shown in Figure 1.21.

	A	B	C	D	E	F
1	**Project start date :**		**03/02/15**			
2						
3	**Task**	**Start**	**Duration**	**End**	**Variation**	
4	**A**	**03/02/15**	**1**	**03/03/15**	**1**	
5	**B**	03/03/15	10	**03/17/15**	14	
6	**C**	03/03/15	5	**03/10/15**	7	
7	**D**	**03/03/15**	**8**	**03/13/15**	**10**	
8	**E**	03/03/15	4	**03/09/15**	6	
9	**F**	03/11/15	2	**03/13/15**	2	
10	**G**	**03/11/15**	**4**	**03/17/15**	**6**	
11	**H**	**03/15/15**	**2**	**03/17/15**	**2**	
12	**I**	03/15/15	1	**03/16/15**	1	
13	**J**	**03/17/15**	**1**	**03/18/15**	**1**	
14	**K**	**03/18/15**	**1**	**03/19/15**	**1**	
15	**L**	**03/19/15**	**1**	**03/20/15**	**1**	

Figure 1.21. *The table recalculated for workdays*

In order to properly display the graph, it will be necessary to change the format of the dates in the columns "Start" and "End" by replacing the date format with the standard format:

– select cells `B4` to `B15`, then, while holding down CTRL, cells `D4` to `D15`;

– right click, FORMAT CELLS…, NUMBER tab, GENERAL category.

This gives the number of days since 1st January 1900 in each of the cells.

	A	B	C	D	E	F
1	Project start date :		03/02/15			
2						
3	Task	Start	Duration	End	Variation	
4	A	42065	1	42066	1	
5	B	42066	10	42080	14	
6	C	42066	5	42073	7	
7	D	42066	8	42076	10	
8	E	42066	4	42072	6	
9	F	42074	2	42076	2	
10	G	42074	4	42080	6	
11	H	42078	2	42080	2	
12	I	42078	1	42079	1	
13	J	42080	1	42081	1	
14	K	42081	1	42082	1	
15	L	42082	1	42083	1	

Figure 1.22. *The "Start" and "End" columns in STANDARD format*

Next, select the table, then carry out the following operations:

– INSERT tab, CHARTS toolbar, scroll down to BAR, select STACKED BAR in 2D BAR;

– move the chart and resize it;

– click on the y-axis;

– right click, select Excel US syntax;

– in the FORMAT AXIS window, tick CATEGORIES IN REVERSE ORDER;

– click on the x-axis to select it;

– in the FORMAT AXIS window, click on the icon in the shape of a bar chart;

– enter 42065 (the value of the cell B4, for 03/02/2015, the start date for the project) in AXIS OPTIONS as MINIMUM;

– next, enter 7 and 1 respectively as MAJOR and MINOR UNITS;

– pull down LABELS and select HIGH in LABEL POSITION;

– pull down NUMBER, select DATE in CATEGORY then *3/14/2001 in TYPE;

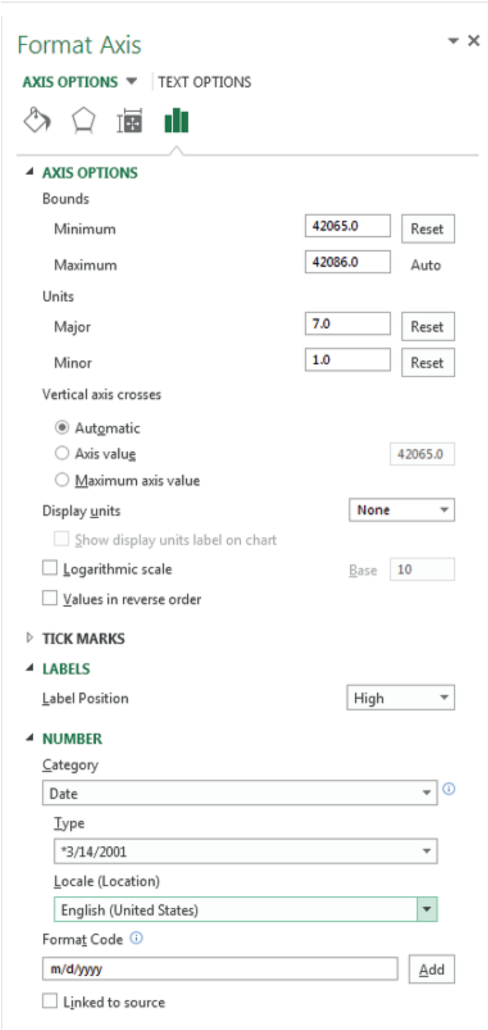


Figure 1.23. *AXIS OPTIONS, LABELS and NUMBER*

– in the FORMAT AXIS window, click on cross-shaped icon (to the left of the bar chart);

– pull down ALIGNMENT and enter -45° in the field CUSTOM ANGLE in order to pivot the dates on the x-axis;

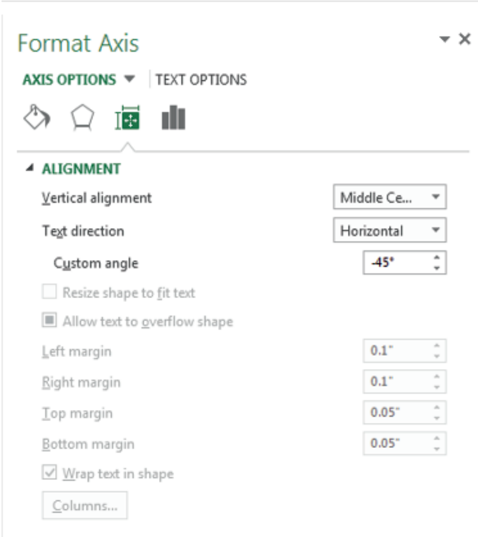


Figure 1.24. *AXIS OPTIONS, ALIGNMENT -45°*

– right click on the bottom of the chart and click on SELECT DATA;

– a 'Data Sources' window will open. Tick "Duration" and "End" (you can also remove these using the DELETE button) then validate by clicking the OK button;

– click on the legend and delete it (hold down DEL or back space on a Mac);

– right click on the "Start" bar on the chart, specify NO OUTLINE and NO FILL by clicking on the corresponding icons (OUTLINE AND FILL) in the pop-up tool bar;

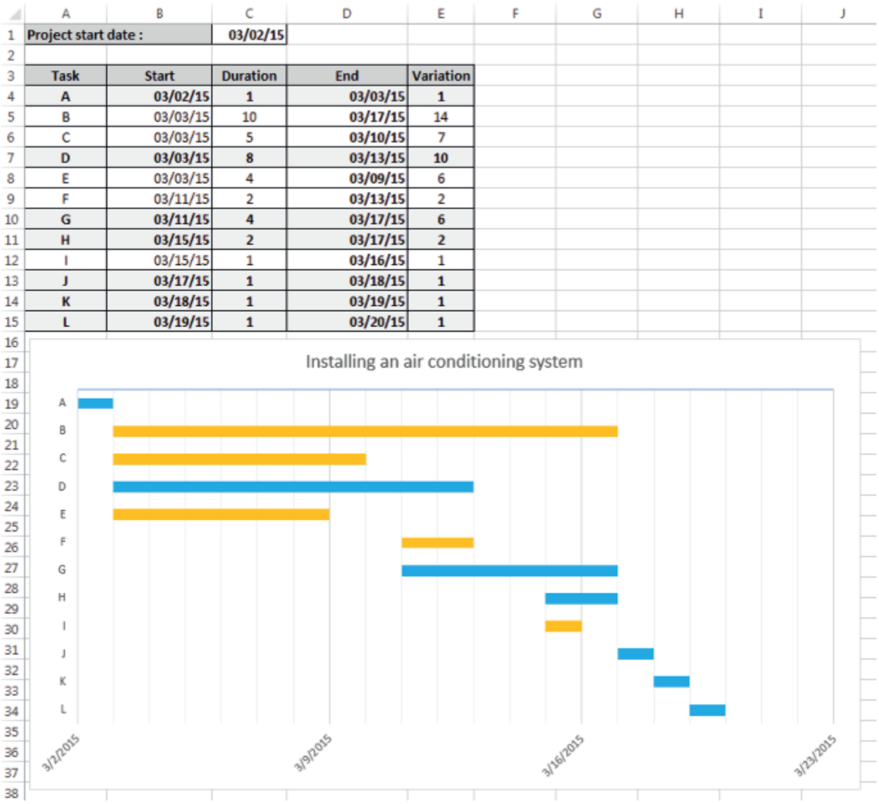
– right click on a date on the x axis, ADD A MINOR GRIDLINE;

– double click on the title to change it and type "Installing an air conditioning system";

– next, repeat a date format for the "Start" and "End" columns;

– to finish, save your work;

– you can change the colors of the chart by putting, for example, critical tasks in a different color.



Project start date :		03/02/15		
Task	**Start**	**Duration**	**End**	**Variation**
A	03/02/15	1	03/03/15	1
B	03/03/15	10	03/17/15	14
C	03/03/15	5	03/10/15	7
D	03/03/15	8	03/13/15	10
E	03/03/15	4	03/09/15	6
F	03/11/15	2	03/13/15	2
G	03/11/15	4	03/17/15	6
H	03/15/15	2	03/17/15	2
I	03/15/15	1	03/16/15	1
J	03/17/15	1	03/18/15	1
K	03/18/15	1	03/19/15	1
L	03/19/15	1	03/20/15	1

Figure 1.25. *The finished table and chart corresponding to the Gantt chart for the project*

The exercises for creating a Gantt chart that have previously been examined can be adapted for larger projects. The table will be larger but the principles and the practical tasks remain the same.

1.4. Maximal flows

In order to calculate the maximal flow available in a network between the source and the sink, we will use the Excel solver alongside two VBA procedures and a spreadsheet. In the solution shown here, the number of edges is limited to 25 and the number of vertices is limited to 20.

Begin by creating the spreadsheet in Figure 1.26.

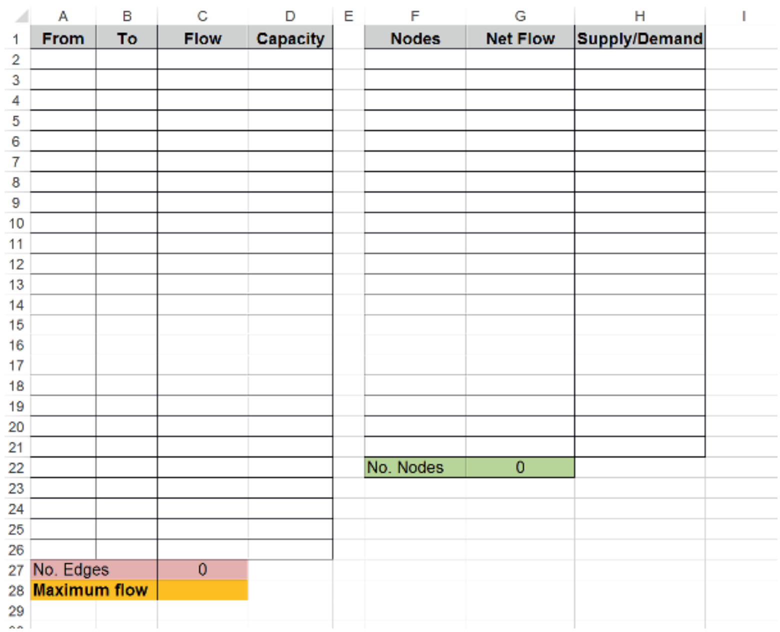


Figure 1.26. *The spreadsheet to be created in Microsoft Excel*

Next, enter the following calculations:

– `G2: =IF(F2="","",SUMIF(From,F2,Flow)-SUMIF(To,F2, Flow))`, if the cell in the column "Nodes" is empty then nothing is displayed; if not the difference is calculated: the total of the flows equal to F2 in the column "From" – the total of the flows equal to F2 in the column "To". This will determine the net flow;

– "From" and "To" are cell ranges that will be named later using a VBA procedure;

– Copy `G2` downward as far as cell `G21`;

– `C27: =COUNTA(A2:A26)`, counts the number of edges;

– `G22: =COUNTA(F2:F21)`, counts the number of vertices;

– `C28: =G2`, displays the maximal flow;

– then save your table;

– switch to VBE, create a module for your project (DEVELOPER tab, VISUAL BASIC icon, right click on the project file, INSERT then MODULE);

– open the module that has just been created, ordinarily MODULE 1 and input the following lines of code:

```
Sub Create_tables()↵
Dim CountArc, CountNodes As String↵
  ‘Clear tables↵
    'If  click on No then clear otherwise quit↵
    If MsgBox(“Do you wish to clear the table content?",
vbYesNo, "Clear") = vbYes Then↵
    ‘Clears the cell content↵
        Range("A2:D26,F2:F21,H3:H21").Select↵
        Selection.ClearContents↵
        'Deletes the cell background color↵
        Range(“A2:D26,F2:G21,H3:H21").Select↵
        Selection.Interior.ColorIndex = 0↵
        Range("C28").Select↵
        'Relaunch automatic sheet recalculation↵
        Application.Calculation =
xlCalculationAutomatic↵
    Else↵
        Exit Sub↵
    End If↵
  ‘Name and color allocation↵
    CountArc = InputBox(“How many edges does your graph
have?", “Number of edges")↵
    'If click on Cancel then quit↵
    If CountArc = "" Then Exit Sub↵
    Range(Cells(2, 1), Cells(1 + CountArc, 1)).Name =
"From"↵
    Range(Cells(2, 1), Cells(1 + CountArc,
3)).Interior.ColorIndex = 40↵
    Range(Cells(2, 2), Cells(1 + CountArc, 2)).Name =
"To"↵
    Range(Cells(2, 2), Cells(1 + CountArc,
3)).Interior.ColorIndex = 44↵
    Range(Cells(2, 3), Cells(1 + CountArc, 3)).Name =
"Flow"↵
    Range(Cells(2, 3), Cells(1 + CountArc,
3)).Interior.ColorIndex = 36↵
    Range(Cells(2, 4), Cells(1 + CountArc, 4)).Name =
"Capacity"↵
    Range(Cells(2, 4), Cells(1 + CountArc,
4)).Interior.ColorIndex = 27↵
```

```
    CountNode = InputBox(“How many vertices does your
graph have?", “Number of nodes")↵
    'If click on Cancel then quit↵
    If CountNode = "" Then Exit Sub↵
    Range(Cells(2, 6), Cells(1 + CountNode, 6)).Name =
"Node"↵
    Range(Cells(2, 6), Cells(1 + CountNode,
6)).Interior.ColorIndex = 17↵
    Range(Cells(3, 7), Cells(CountNode, 7)).Name =
"NetFlow"↵
    Range(Cells(2, 7), Cells(1 + CountNode,
7)).Interior.ColorIndex = 37↵
    Range(Cells(3, 8), Cells(CountNode, 8)).Name =
"SupplyDemand"↵
    Range(Cells(3, 8), Cells(CountNode,
8)).Interior.ColorIndex = 8↵
    MsgBox (“You can now enter your" & CountArc & " arcs
with their capacity in the FROM and TO column. Then
enter the list of your " & CountNode & "nodes in NODES
column.")↵
    Range(Cells(3, 8), Cells(CountNode, 8)) = 0↵
End Sub↵
↵

'Launch the solver to find the solution↵
Sub Resolve()↵
    ‘Reset solver to zero↵
    SolverReset↵
    ‘Defining solver parameters↵
    SolverOk SetCell:="$C$28", MaxMinVal:=1, ValueOf:=0,
ByChange:="$C$2:$C$19", _↵
        Engine:=2, EngineDesc:="Simplex LP"↵
    SolverDelete CellRef:="$C$2:$C$19", Relation:=1,
FormulaText:="Capacity"↵
    SolverOk SetCell:="$C$28", MaxMinVal:=1, ValueOf:=0,
ByChange:="$C$2:$C$19", _↵
        Engine:=2, EngineDesc:="Simplex LP"↵
    SolverAdd CellRef:="$C$2:$C$19", Relation:=1,
FormulaText:="Capacity"↵
    SolverOk SetCell:="$C$28", MaxMinVal:=1, ValueOf:=0,
ByChange:="$C$2:$C$19", _↵
        Engine:=2, EngineDesc:="Simplex LP"↵
    SolverDelete CellRef:="$G$3:$G$10", Relation:=2,
FormulaText:="SupplyDemand"↵
    SolverOk SetCell:="$C$28", MaxMinVal:=1, ValueOf:=0,
ByChange:="$C$2:$C$19", _↵
        Engine:=2, EngineDesc:="Simplex LP"↵
```

```
    SolverAdd CellRef:="$G$3:$G$10", Relation:=2,
FormulaText:="SupplyDemand"↵
    SolverOk SetCell:="$C$28", MaxMinVal:=1, ValueOf:=0,
ByChange:="$C$2:$C$19", _↵
        Engine:=2, EngineDesc:="Simplex LP"↵
    SolverOk SetCell:="$C$28", MaxMinVal:=1, ValueOf:=0,
ByChange:="$C$2:$C$19", _↵
        Engine:=2, EngineDesc:="Simplex LP"↵
    'Launch automatic calculation of results by the
solver↵
    SolverSolve (True)↵
End Sub↵
```

In order to launch these two procedures, `Create_tables()` and `Resolve()`, create two buttons by inserting two forms to which you will assign each of the two macros (right click on form then ASSIGN a MACRO). Your spreadsheet should look like the one in Figure 1.27.

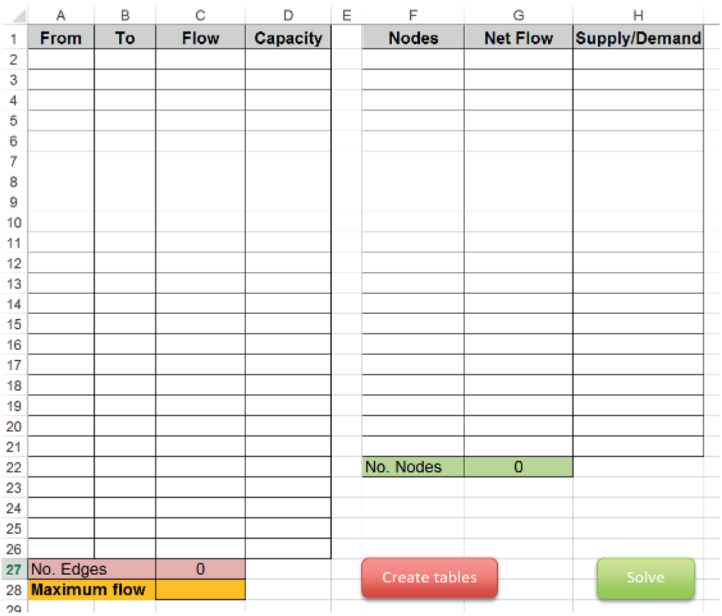


Figure 1.27. *The spreadsheet with its two buttons ("Create tables" and "Solve")*

In order to use the calculation of maximal flows, all you need to do is click on the CREATE TABLES button to destroy any calculation that has already been carried out and delete the calculated contents. Two dialogue boxes will be displayed one after the other asking you how many edges you want followed by the number of vertices in your graph.

Enter the data in the table on the left by mentioning, for each arc, the peak of departure in the column “From” and its vertice (node) of arrival in the column “To” followed by its capacity in the column “Capacity”.

Now add the flows. I would advise entering the minimal value of the capacity for all of the flows.

Next, input all the vertices in the table on the right, in the “Vertices” column, beginning with the source (ordinarily S) and ending with the sink (ordinarily T). The order for the other vertices is irrelevant.

It should be noted that the entry fields have been colored in order to make it easier to input data.

Lastly, click on the SOLVE button to begin the calculation of the solution by the solver and to display the results.

In Figure 1.28, I re-use the data from Exercise 1 in Chapter 6 (Volume 1, section 6.5.1), and you can see all edges and their capacities (18), all the vertices (10) and the starting flow, which is 2000.

	A	B	C	D	E	F	G	H
1	From	To	Flow	Capacity		Nodes	Net Flow	Supply/Demand
2	S	P1	2000	10000		S	6000	
3	S	P2	2000	8000		P2	0	0
4	S	P3	2000	6000		P3	2000	0
5	P1	P2	2000	4000		R1	0	0
6	P1	R1	2000	6000		R2	-2000	0
7	P2	R1	2000	4000		R3	0	0
8	P2	R2	2000	2000		R4	-2000	0
9	P2	R3	2000	2000		R5	-2000	0
10	P3	P2	2000	4000		P1	2000	0
11	P3	R3	2000	6000		T	-4000	
12	R1	R2	2000	4000				
13	R1	R4	2000	6000				
14	R2	R4	2000	4000				
15	R2	R5	2000	3000				
16	R3	R2	2000	6000				
17	R3	R5	2000	2000				
18	R4	T	2000	12000				
19	R5	T	2000	4000				
20								
21								
22						No. Nodes	10	
23								
24								
25								
26								
27	No. Edges		18			Create tables		Solve
28	Maximum flow		6000					
29								

Figure 1.28. *The data from Exercise 1 in Chapter 6 entered into the spreadsheet*

Figure 1.29 shows the result of the calculation launched by clicking on the SOLVE button.

	A	B	C	D	E	F	G	H
1	From	To	Flow	Capacity		Nodes	Net Flow	Supply/Demand
2	S	P1	6000	10000		S	14000	
3	S	P2	8000	8000		P2	0	0
4	S	P3	0	6000		P3	0	0
5	P1	P2	0	4000		R1	0	0
6	P1	R1	6000	6000		R2	0	0
7	P2	R1	4000	4000		R3	0	0
8	P2	R2	2000	2000		R4	0	0
9	P2	R3	2000	2000		R5	0	0
10	P3	P2	0	4000		P1	0	0
11	P3	R3	0	6000		T	-14000	
12	R1	R2	4000	4000				
13	R1	R4	6000	6000				
14	R2	R4	4000	4000				
15	R2	R5	2000	3000				
16	R3	R2	0	6000				
17	R3	R5	2000	2000				
18	R4	T	10000	12000				
19	R5	T	4000	4000				
20								
21								
22						No. Nodes	10	
23								
24								
25								
26								
27	No. Edges		18					
28	**Maximum flow**		**14000**					
29								

Create tables

Solve

Figure 1.29. *The solution from exercise 1 with the set of results displayed*

1.5. Transport model

There are a variety of logistical problems in transport: often, you need to standardize distances, the costs and the constraints for supplying customers.

Below, you will find two classic examples that will be solved using the Microsoft Excel solver.

Using this as a basis, it will be easy to use each of these solutions to adapt them to your own problems.

1.5.1. *Customer delivery*

A railway company has to transport two batches of palletized products from two departure stations (GD1 and GD2) to three destination stations (GA1, GA2 and GA3) spread out across France. The goods will pass through intermediary marshalling yards (GT1, GT2, GT3, GT4 and GT5).

The number of pallets available in the outgoing stations and the storage capacities of the end stations are known, as are the transport capacities of the inter-station railway lines.

Table 1.4 groups together the shared information for each station.

Pallet flow										
→	**DS1**	**DS2**	**MY1**	**MY2**	**MY3**	**MY4**	**MY5**	**ES1**	**ES2**	**ES3**
Pallets available	**105**	**75**								
DS1		–	60	45	36	–	–	–	–	–
DS2	–		–	18	66	–	-	–	–	–
MY1	–	–	–	30	–	45	–	–	–	–
MY2	–	–	–	–	–	–	–	30	45	45
MY3	–	–	–	–	–		66	–	–	–
MY4	–	-	–	–	–	–	–	21	30	–
MY5	–	-	–	–	–		–	–	30	30
Storage capacities of the arrival stations								**45**	**45**	**60**

Table 1.4. *Summary of palletized product batch flow*

If we examine the problem in greater detail, we notice that there is a problem involving maximal flow with constraints being exceeded.

We will tackle this problem by using a simple double-entry table as well as the solver.

First, create and fill a spreadsheet, the table shown in Figure 1.30.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2														
3		DS : Departure station				MY : Marshalling yard				ES : End station			Flow :	0
4														
5		S	DS1	DS2	MY1	MY2	MY3	MY4	MY5	ES1	ES2	ES3	T	
6	S	0	105	75	0	0	0	0	0	0	0	0	0	
7	DS1	0	0	0	60	45	36	0	0	0	0	0	0	
8	DS2	0	0	0	0	18	66	0	0	0	0	0	0	
9	MY1	0	0	0	0	30	0	45	0	0	0	0	0	
10	MY2	0	0	0	0	0	0	0	0	30	45	45	0	
11	MY3	0	0	0	0	0	0	0	66	0	0	0	0	
12	MY4	0	0	0	0	0	0	0	0	21	30	0	0	
13	MY5	0	0	0	0	0	0	0	0	0	30	30	0	
14	ES1	0	0	0	0	0	0	0	0	0	0	0	45	
15	ES2	0	0	0	0	0	0	0	0	0	0	0	45	
16	ES3	0	0	0	0	0	0	0	0	0	0	0	60	
17	T	0	0	0	0	0	0	0	0	0	0	0	0	
18														

Figure 1.30. *The table to be created in a Microsoft Excel spreadsheet*

All of the data from the statement is shown here. For this purpose, the table has an S column and a T column, which represent the source and the sink, thus formalizing the number of pallets available at the departure station (DS1: 105 and DS: 75) and the storage capacity of the end stations (ES1: 45, ES2: 45 and ES3: 60).

Below the first table, add a second like the one shown in Figure 1.31.

18														
19														
20		S	DS1	DS2	MY1	MY2	MY3	MY4	MY5	ES1	ES2	ES3	T	Total
21	S	0	0	0	0	0	0	0	0	0	0	0	0	
22	DS1	0	0	0	0	0	0	0	0	0	0	0	0	0
23	DS2	0	0	0	0	0	0	0	0	0	0	0	0	0
24	MY1	0	0	0	0	0	0	0	0	0	0	0	0	0
25	MY2	0	0	0	0	0	0	0	0	0	0	0	0	0
26	MY3	0	0	0	0	0	0	0	0	0	0	0	0	0
27	MY4	0	0	0	0	0	0	0	0	0	0	0	0	0
28	MY5	0	0	0	0	0	0	0	0	0	0	0	0	0
29	ES1	0	0	0	0	0	0	0	0	0	0	0	0	0
30	ES2	0	0	0	0	0	0	0	0	0	0	0	0	0
31	ES3	0	0	0	0	0	0	0	0	0	0	0	0	0
32	T	0	0	0	0	0	0	0	0	0	0	0	0	
33	Total		0	0	0	0	0	0	0	0	0	0		
34														

Figure 1.31. *The second table positioned below the first*

It will be filled as indicated below:

– line 33 will contain the total of each of the columns;

– in cell C33, enter : =TOTAL(C21:C32);

– next, copy this formula down to cell L33;

– column N will contain the total of each of the lines;

– in Cell N22, enter : =TOTAL(B22:M22);

– next, reproduce this below until N31;

– in cell N3, enter the total of the wells to maximize : =TOTAL(B31:M31);

– launch the solver, DATA tab, SOLVER tool in the tool ribbon. If it is not visible, consult Appendix 1 of this volume to begin its installation;

– configure the solver as shown in Figure 1.32;

– SET OBJECTIVE : $N$3;

– BY CHANGING VARIABLE CELLS : $B$21:$M$32;

– SUBJECT TO THE CONSTRAINTS : $B$21:$M$32<=$B$6:$M$17;

– SUBJECT TO THE CONSTRAINTS : $C$33:$L$33=$N$22:$N$31;

– Tick the box MAKE UNCONSTRAINTED VARIABLES NON-NEGATIVE

– SELECT A SOLVING METHOD : Simplex PL;

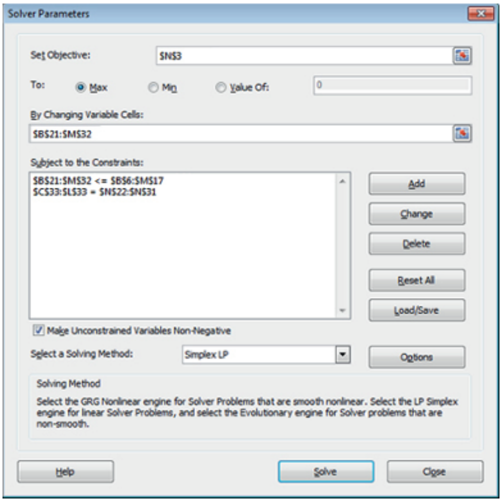


Figure 1.32. *Configuring the solver*

– Next, click on the bottom SOLVE. In a few moments, the solver will display the result window;

– You will then be able to SAVE SOLVER SOLUTION or RESET INITIAL VALUES (to start again) by ticking the appropriate box and clicking on the OK button.

This will give you the values from Figure 1.33 in the second table that will specify the number of pallets to be transported between each of the points, taking all restrictions into account.

	S	DS1	DS2	MY1	MY2	MY3	MY4	MY5	ES1	ES2	ES3	T	Total
S	0	105	45	0	0	0	0	0	0	0	0	0	
DS1	0	0	0	57	45	3	0	0	0	0	0	0	105
DS2	0	0	0	0	18	27	0	0	0	0	0	0	45
MY1	0	0	0	0	30	0	27	0	0	0	0	0	57
MY2	0	0	0	0	0	0	0	0	30	33	30	0	93
MY3	0	0	0	0	0	0	0	30	0	0	0	0	30
MY4	0	0	0	0	0	0	0	0	15	12	0	0	27
MY5	0	0	0	0	0	0	0	0	0	0	30	0	30
ES1	0	0	0	0	0	0	0	0	0	0	0	45	45
ES2	0	0	0	0	0	0	0	0	0	0	0	45	45
ES3	0	0	0	0	0	0	0	0	0	0	0	60	60
T	0	0	0	0	0	0	0	0	0	0	0	0	
Total		105	45	57	93	30	27	30	45	45	60		

Figure 1.33. *The results calculated by the solver*

You can see that the total number of pallets arriving at the destination does not exceed 150 (45+45+60), ensuring that the storage capacity of the arrival stations is not exceeded.

1.5.2. *Minimum-cost transport*

In this type of problem, the aim is to transport goods at a minimal cost while taking into account restrictions such as customer demand and warehouse storage capacity.

Let's consider three factories (U1, U2 and U3) located in France who wish to deliver to their clients (C1 to C5) spread out across Europe. The transport costs (per ton: 1,000 kg) are specified in Table 1.5.

Factory stock	C1	C2	C3	C4	C5
U1	$300.00	$2,400.00	$300.00	$1,500.00	$1,200.00
U2	$1,500.00	$1,500.00	$900.00	$1,800.00	$2,100.00
U3	$600.00	$900.00	$1,500.00	$2,700.00	$2,400.00

Table 1.5. *Transport costs between factories and customers*

In terms of stock, each factory has a volume of available product estimated at being 7,200 kg for U1, 4,800 kg for U2 and 7,800 kg for U3.

The five clients want the following quantities: C1, 3,600 kg; C2, 3,900 kg; C3, 4,350 kg; C4, 3,750 kg and C5, 4,200 kg.

It is now necessary to define what mass of product each of the three factories must send to each of the five customers in order to meet demand, taking into account the stock available and minimizing transport costs.

First, create the two tables in Figure 1.34 in a Microsoft Excel spreadsheet.

	A	B	C	D	E	F	G	H
1		Transport costs						
2		C1	C2	C3	C4	C5	Factory stock (kg)	
3	U1	$ 300.00	$2,400.00	$ 300.00	$1,500.00	$1,200.00	7200	
4	U2	$1,500.00	$1,500.00	$ 900.00	$1,800.00	$2,100.00	4800	
5	U3	$ 600.00	$ 900.00	$1,500.00	$2,700.00	$2,400.00	7800	
6	Order (Kg)	3600	3900	4350	3750	4200		
7								
8		C1	C2	C3	C4	C5	Quantity to be send	
9	U1						0	
10	U2						0	
11	U3						0	
12	Quantity to be received (kg)	0	0	0	0	0		
13								
14	Objective	0						
15								
16								
17								

Figure 1.34. *The two tables for calculating costs*

Next, apply the following operations:

– in cell `B12`, enter the calculation : `=TOTAL(B9:B11)`;

– copy this formula to the right as far as F12;

– in G9, enter : =TOTAL(B9:F9);

– next, copy this total downwards as far as G11;

– in cell B14, we will define the calculation objective for the solver. This is the total that we must minimize;

– enter : =SUMPRODUCT(B3:F5;B9:F11);

– open the solver, DATA tab, SOLVER tool in the ribbon. If it is not visible, consult Appendix 1 for the installation procedure;

– enter the parameters as shown in Figure 1.35;

– SET OBJECTIVE: $B$14;

– BY CHANGING VARIABLE CELLS: $B$9:$F$11;

– SUBJECT TO THE CONSTRAINTS: $B$12:$F$12=$B$6:$F$6;

– SUBJECT TO THE CONSTRAINTS: $G$9:$G$11=$G$3:$G$5;

– Tick MAKE UNCONSTRAINED VARIABLES NON-NEGATIVE;

– SELECT A SOLVING METHOD: Simplex PL;

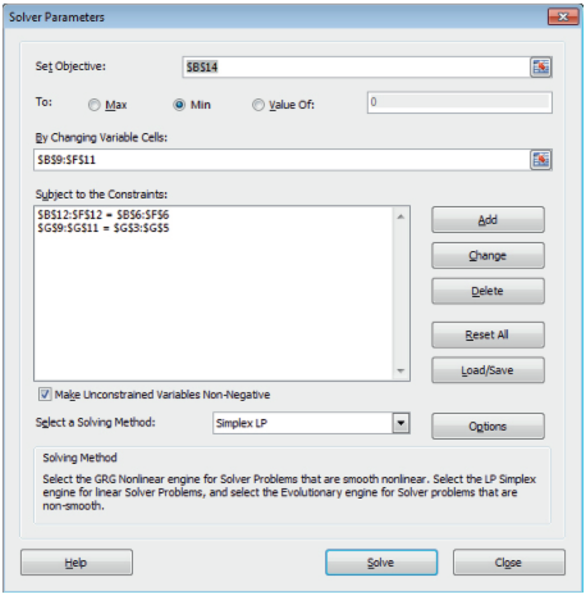


Figure 1.35. *Configuration of the solver*

– next, click on the SOLVE button and wait a few seconds;

– a window will open. You will then have the option to SAVE THE SOLVER SOLUTION or RESET TO ORIGINAL VALUES (to start again) by ticking the appropriate box and clicking on the OK button.

The lower table displays the quantities to be dispatched to each of the factories of each of the customers. We can see that the restrictions relating to customer orders and to factory stocks have been taken into account.

	A	B	C	D	E	F	G	H
1		Transport costs						
2		C1	C2	C3	C4	C5	Factory stock (kg)	
3	U1	$ 300.00	$2,400.00	$ 300.00	$1,500.00	$1,200.00	7200	
4	U2	$1,500.00	$1,500.00	$ 900.00	$1,800.00	$2,100.00	4800	
5	U3	$ 600.00	$ 900.00	$1,500.00	$2,700.00	$2,400.00	7800	
6	Order (Kg)	3600	3900	4350	3750	4200		
7								
8		C1	C2	C3	C4	C5	Quantity to be send	
9	U1	0	0	3300	0	3900	7200	
10	U2	0	0	1050	3750	0	4800	
11	U3	3600	3900	0	0	300	7800	
12	Quantity to be received (kg)	3600	3900	4350	3750	4200		
13								
14	Objective	19755000						
15								

Figure 1.36. *The results obtained from using the solver*

1.6. Linear programming

I will now show you how to use the Microsoft Excel solver to solve classical linear programming problems that would be solved on paper using a simplex.

We will create a spreadsheet capable of holding six constraints relating to six variables, by responding to maximization and minimization requests.

It will be very easy to increase the amount of data to manage by applying the same calculation formulae on a larger table.

1.6.1. *Creating a calculation table*

Build and fill the table in Figure 1.37 in a spreadsheet, then carry out the following:

	A	B	C	D	E	F	G	H	I	J	K
1	Variables	x1	x2	x3	x4	x5	x6				
2	bi										
3											
4	Constraints										
5	n1							0	<=		
6	n2							0	<=		
7	n3							0	<=		
8	n4							0	<=		
9	n5							0	<=		
10	n6							0	<=		
11											
12	z	20	30								
13											
14	Max z	0									
15											

Figure 1.37. *The table that will host our future linear programs*

– In cell H5, enter the formula: `=TOTALPRODUCT (B5:G5;$B$2:$G$2)`. This formula carries out the total of the products between the cells in line 5 (constraint n°1) and line 2 (b_i);

– Next, copy this formula downward as far as H10;

– In cell B14, enter the formula : `=TOTALPRODUCT(B12:G12;B2:G2)`. This formula calculates the total of the products between the cells in line 12 (z) and line 2 (b_i);

– Save your spreadsheet.

1.6.2. *Entering data*

To fill in our table, we will re-use the example from Chapter 8 (Volume 1, section 8.3.2), which has three constraints and the following profit, z:

$$\begin{cases} 3x_1 + 1{,}5x_2 + 2x_3 \leq 6000 \\ 6x_1 + 4x_2 + 4x_3 \leq 10000 \\ x_1 + x_2 + x_3 \leq 3500 \end{cases}$$

$$z = 8x_1 + 3{,}5x_2 + 6x_3$$

Enter this data in your spreadsheet in order to obtain the table in Figure 1.38.

	A	B	C	D	E	F	G	H	I	J	K
1	Variables	x1	x2	x3	x4	x5	x6				
2	bi										
3											
4	Constraints										
5	n1	3	1,5	2				0	<=	6000	
6	n2	6	4	4				0	<=	10000	
7	n3	1	1	1				0	<=	3500	
8	n4							0	<=		
9	n5							0	<=		
10	n6							0	<=		
11											
12	z	8	3,5	6							
13											
14	Max z	0									
15											

Figure 1.38. *The data from the example, entered into our table*

COMMENT 1.2.– The indicators of inequality (<=) located in column I in our table are only given as an illustrative example – they are not relevant in the calculation.

1.6.3. *Using the solver*

Open the solver, DATA tab, SOLVER tool in the ribbon. If it is not visible, consult Appendix 1 for the installation procedure.

Complete the different fields for the solver:

– SET OBJECTIVE : `$B$14`;

– Click on the option MAX (for this example we are dealing with an example of maximization) ;

– BY CHANGING VARIABLE CELLS : `$B$2:$G$2`;

– SUBJECT TO THE CONSTRAINTS : `$B$2:$G$2>=0`; `$H$10<=$I$10`; `$H$5<=$J$5` ; `$H$6<=$J$6`; `$H$7<=$J$7` ; `$H$8<=$J$8` ; `$H$9<=$J$9`;

COMMENT 1.3.– The constraints are not all necessary in our example, but the solver has been configured by default in order to solve a problem involving six inequalities.

– Tick the box : MAKE UNCONSTRAINED VARIABLES NON-NEGATIVE;

– SELECT A SOLVING METHOD: Simplex PL;

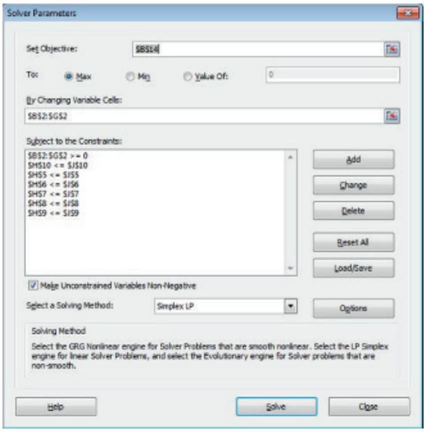


Figure 1.39. *The different parameters entered in the solver dialogue window*

– click on the SOLVE button;

– after a few moments the results will be displayed in the table and the SOLVER RESULTS window will be displayed.

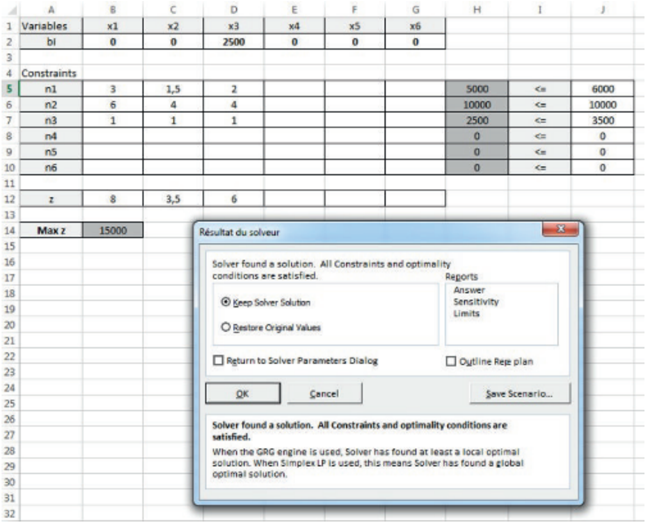


Figure 1.40. *Calculated results and the SOLVER RESULTS window*

The results found in Chapter 8 (Volume 1, section 8.3.7) are confirmed: $z = 15000$; $x1=0$; $x2=0$ and $x3 = 2500$.