

THE SOLVER

Linear optimisation in Excel

The Solver tool

Excel's Solver can solve linear and many nonlinear optimisation problems.

The Solver tool

Excel's Solver can solve linear and many nonlinear optimisation problems.

1. It can minimise or maximise an objective function.

The Solver tool

Excel's Solver can solve linear and many nonlinear optimisation problems.

1. It can minimise or maximise an objective function.
2. It allows integer or binary restrictions on the decision variables.

The Solver tool

Excel's Solver can solve linear and many nonlinear optimisation problems.

1. It can minimise or maximise an objective function.
2. It allows integer or binary restrictions on the decision variables.
3. The standard Solver add-in supports up to 200 decision variables.

Four steps for a linear programme

A spreadsheet model connects the decisions to their consequences.

Four steps for a linear programme

A spreadsheet model connects the decisions to their consequences.

1. Set up a spreadsheet that calculates the quantities of interest, such as costs, profits and resource usage.

Four steps for a linear programme

A spreadsheet model connects the decisions to their consequences.

1. Set up a spreadsheet that calculates the quantities of interest, such as costs, profits and resource usage.
2. Identify the cell containing the objective function: the **objective cell** (or target cell).

Four steps for a linear programme

A spreadsheet model connects the decisions to their consequences.

1. Set up a spreadsheet that calculates the quantities of interest, such as costs, profits and resource usage.
2. Identify the cell containing the objective function: the **objective cell** (or target cell).
3. Identify the decision variables: the **changing variable cells**.

Four steps for a linear programme

A spreadsheet model connects the decisions to their consequences.

1. Set up a spreadsheet that calculates the quantities of interest, such as costs, profits and resource usage.
2. Identify the cell containing the objective function: the **objective cell** (or target cell).
3. Identify the decision variables: the **changing variable cells**.
4. Enter the **constraints** that the solution must satisfy.

Four steps for a linear programme

A spreadsheet model connects the decisions to their consequences.

1. Set up a spreadsheet that calculates the quantities of interest, such as costs, profits and resource usage.
2. Identify the cell containing the objective function: the **objective cell** (or target cell).
3. Identify the decision variables: the **changing variable cells**.
4. Enter the **constraints** that the solution must satisfy.

After clicking Solve and keeping the solution, the variable cells contain the optimal solution and the objective cell contains its value.

Factory production costs

A factory produces three types of electronic products.

Product	Quantity	Unit cost

Factory production costs

A factory produces three types of electronic products.

Product	Quantity	Unit cost
Laptops	n_a	\$250

Factory production costs

A factory produces three types of electronic products.

Product	Quantity	Unit cost
Laptops	n_a	\$250
Smartphones	n_b	\$300

Factory production costs

A factory produces three types of electronic products.

Product	Quantity	Unit cost
Laptops	n_a	\$250
Smartphones	n_b	\$300
Plasma TVs	n_c	\$500

Factory production costs

A factory produces three types of electronic products.

Product	Quantity	Unit cost
Laptops	n_a	\$250
Smartphones	n_b	\$300
Plasma TVs	n_c	\$500

$$\text{Total cost} = 250n_a + 300n_b + 500n_c.$$

The production requirements

What is the minimum cost subject to the following requirements?

The production requirements

What is the minimum cost subject to the following requirements?

At least 100 laptops. $n_a \geq 100$

The production requirements

What is the minimum cost subject to the following requirements?

At least 100 laptops. $n_a \geq 100$

At least 20 smartphones. $n_b \geq 20$

The production requirements

What is the minimum cost subject to the following requirements?

At least 100 laptops. $n_a \geq 100$

At least 20 smartphones. $n_b \geq 20$

At least 50 TVs. $n_c \geq 50$

The production requirements

What is the minimum cost subject to the following requirements?

At least 100 laptops. $n_a \geq 100$

At least 20 smartphones. $n_b \geq 20$

At least 50 TVs. $n_c \geq 50$

Exactly 350 units in total. $n_a + n_b + n_c = 350$

The production requirements

What is the minimum cost subject to the following requirements?

At least 100 laptops. $n_a \geq 100$

At least 20 smartphones. $n_b \geq 20$

At least 50 TVs. $n_c \geq 50$

Exactly 350 units in total. $n_a + n_b + n_c = 350$

$$\begin{array}{ll} \min_{n_a, n_b, n_c} & 250n_a + 300n_b + 500n_c \\ \text{subject to} & n_a \geq 100, \quad n_b \geq 20, \quad n_c \geq 50, \\ & n_a + n_b + n_c = 350. \end{array}$$

FACTORY COST OPTIMISATION

Spreadsheet example

Factory cost optimization

The starting worksheet

The initial production plan satisfies the requirements.

B5	<i>fx</i>	=SUM(B2:B4)		
	A	B	C	D
1	Product	Quantity	Production cost	Total cost
2	Laptop	100	250	25,000
3	Smartphone	20	300	6,000
4	Plasma TV	230	500	115,000
5		350		146,000

Requirements: at least 100 laptops, 20 smartphones and 50 TVs, with exactly 350 units in total.

The decision variables

Solver changes the quantities in B2:B4.

B2	<i>fx</i>	100		
	A	B	C	D
1	Product	Quantity	Production cost	Total cost
2	Laptop	100	250	25,000
3	Smartphone	20	300	6,000
4	Plasma TV	230	500	115,000
5		350		146,000

B2 is n_a , B3 is n_b , and B4 is n_c .

Total production

The cell B5 adds the three production quantities.

B5	<i>fx</i>	=SUM(B2:B4)		
	A	B	C	D
1	Product	Quantity	Production cost	Total cost
2	Laptop	100	250	25,000
3	Smartphone	20	300	6,000
4	Plasma TV	230	500	115,000
5		350		146,000

B5 contains a formula. The total production constraint will require its value to equal 350.

The unit costs

The production costs per unit are fixed inputs in C2:C4.

C2	<i>fx</i>	250		
	A	B	C	D
1	Product	Quantity	Production cost	Total cost
2	Laptop	100	250	25,000
3	Smartphone	20	300	6,000
4	Plasma TV	230	500	115,000
5		350		146,000

Solver varies the quantities while these unit costs remain unchanged.

The cost of laptop production

In D2, multiply the quantity by its unit cost.

D2	<i>fx</i>	=C2*B2		
	A	B	C	D
1	Product	Quantity	Production cost	Total cost
2	Laptop	100	250	25,000
3	Smartphone	20	300	6,000
4	Plasma TV	230	500	115,000
5		350		146,000

$$250 \times 100 = 25,000.$$

The cost of smartphone production

In D3, multiply the quantity by its unit cost.

D3	<i>fx</i>	=C3*B3		
	A	B	C	D
1	Product	Quantity	Production cost	Total cost
2	Laptop	100	250	25,000
3	Smartphone	20	300	6,000
4	Plasma TV	230	500	115,000
5		350		146,000

$$300 \times 20 = 6,000.$$

The cost of TV production

In D4, multiply the quantity by its unit cost.

D4	<i>fx</i>	=C4*B4		
	A	B	C	D
1	Product	Quantity	Production cost	Total cost
2	Laptop	100	250	25,000
3	Smartphone	20	300	6,000
4	Plasma TV	230	500	115,000
5		350		146,000

$$500 \times 230 = 115,000.$$

The total production cost

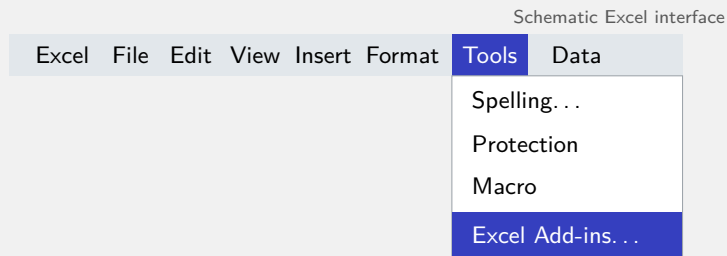
The cell D5 adds the three production costs.

D5	<i>fx</i>	=SUM(D2:D4)		
	A	B	C	D
1	Product	Quantity	Production cost	Total cost
2	Laptop	100	250	25,000
3	Smartphone	20	300	6,000
4	Plasma TV	230	500	115,000
5		350		146,000

$$25,000 + 6,000 + 115,000 = 146,000.$$

Activating Solver on a Mac

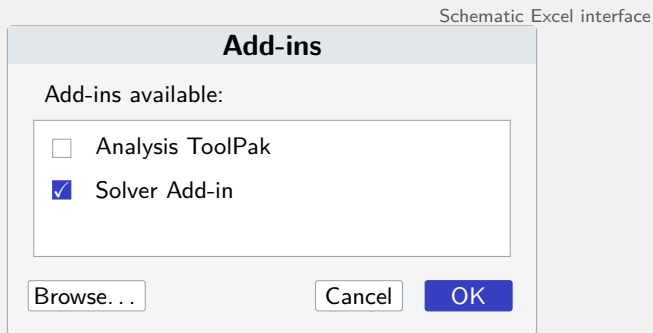
From the menu bar, select **Tools**, then **Excel Add-ins**.



The label may appear as **Add-ins** in older versions of Excel.

The Solver add-in

Select **Solver Add-in**, then click **OK**.



The Data ribbon

Open the **Data** tab in the Excel ribbon.

Schematic Excel interface

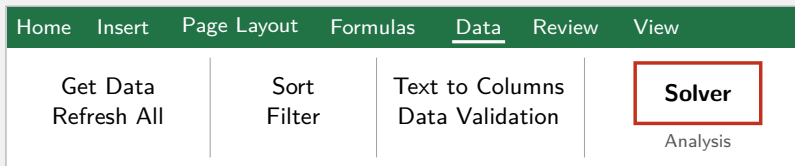
Home	Insert	Page Layout	Formulas	<u>Data</u>	Review	View
Get Data Refresh All		Sort Filter		Text to Columns Data Validation		Solver Analysis

The Solver command becomes available after the add-in is enabled.

Opening Solver

On the **Data** tab, click **Solver** in the Analysis group.

Schematic Excel interface



Solver Parameters

The dialogue links the objective, the variable cells and the constraints.

Schematic Excel interface

Solver Parameters

Set Objective:

To: ☐ Max ☒ Min ☐ Value Of:

By Changing Variable Cells:

Subject to the Constraints:

Add

Change

Delete

☒ Make Unconstrained Variables Non-Negative

Solving Method:

Close

Solve

The objective cell

The objective is the total production cost in D5.

D5	<i>fx</i>	=SUM(D2:D4)		
	A	B	C	D
1	Product	Quantity	Production cost	Total cost
2	Laptop	100	250	25,000
3	Smartphone	20	300	6,000
4	Plasma TV	230	500	115,000
5		350		146,000

Set Objective: \$D\$5 **To:** Min

The changing variable cells

Minimise the cost by varying the numbers of units produced.

B2	<i>fx</i>	100		
	A	B	C	D
1	Product	Quantity	Production cost	Total cost
2	Laptop	100	250	25,000
3	Smartphone	20	300	6,000
4	Plasma TV	230	500	115,000
5		350		146,000

By Changing Variable Cells: \$B\$2:\$B\$4

The constraints and solving method

Schematic Excel interface

Solver Parameters

Set Objective:

To: ☐ Max ☒ Min ☐ Value Of:

By Changing Variable Cells:

Subject to the Constraints:

\$B\$2 >= 100

\$B\$3 >= 20

\$B\$4 >= 50

\$B\$5 = 350

Add

Change

Delete

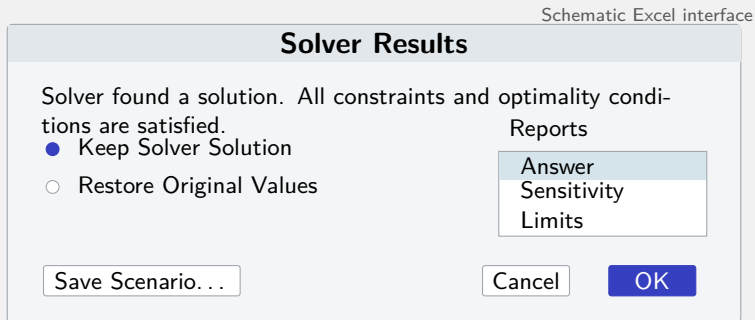
☒ Make Unconstrained Variables Non-Negative

Solving Method:

Choose **Simplex LP**: the objective and all constraints are linear. Then click **Solve**.

Keeping the Solver solution

When Solver reports a solution, keep the solution in the worksheet.



Select the **Answer** report to inspect the final values and the constraints.

The optimal production plan

D5	<i>fx</i>	=SUM(D2:D4)		
	A	B	C	D
1	Product	Quantity	Production cost	Total cost
2	Laptop	280	250	70,000
3	Smartphone	20	300	6,000
4	Plasma TV	50	500	25,000
5		350		101,000

Answer report

Cell	Quantity	Original	Final	Status
D5	Total cost	146,000	101,000	Minimum
B2	Laptops	100	280	Continuous
B3	Smartphones	20	20	Continuous
B4	Plasma TVs	230	50	Continuous
Cell	Constraint	Final	Slack	Status
B5	Total = 350	350	0	Binding
B2	Laptops $\geq$ 100	280	180	Not binding
B3	Smartphones $\geq$ 20	20	0	Binding
B4	TVs $\geq$ 50	50	0	Binding

Feasible integer solutions satisfy:

$$n_a, n_b, n_c \in \mathbb{Z}$$

$$n_a \geq 100, \quad n_b \geq 20, \quad n_c \geq 50$$

$$n_a + n_b + n_c = 350$$

Objective (minimize):

$$250n_a + 300n_b + 500n_c$$

Assume a feasible integer solution with $n_b > 20$.

Assume a feasible integer solution with $n_b > 20$.

Exchange move:

$$n'_a = n_a + 1, \quad n'_b = n_b - 1, \quad n'_c = n_c$$

Feasibility preserved: sum stays 350; $n'_b \geq 20$ because $n_b > 20$.

Assume a feasible integer solution with $n_b > 20$.

Exchange move:

$$n'_a = n_a + 1, \quad n'_b = n_b - 1, \quad n'_c = n_c$$

Feasibility preserved: sum stays 350; $n'_b \geq 20$ because $n_b > 20$.

Objective change:

$$\begin{aligned} \Delta &= 250(n_a + 1) + 300(n_b - 1) + 500n_c \\ &\quad - (250n_a + 300n_b + 500n_c) \end{aligned}$$

Assume a feasible integer solution with $n_b > 20$.

Exchange move:

$$n'_a = n_a + 1, \quad n'_b = n_b - 1, \quad n'_c = n_c$$

Feasibility preserved: sum stays 350; $n'_b \geq 20$ because $n_b > 20$.

Objective change:

$$\begin{aligned} \Delta &= 250(n_a + 1) + 300(n_b - 1) + 500n_c \\ &\quad - (250n_a + 300n_b + 500n_c) \end{aligned}$$

$$\Delta = 250 - 300 = -50 < 0$$

Assume a feasible integer solution with $n_b > 20$.

Exchange move:

$$n'_a = n_a + 1, \quad n'_b = n_b - 1, \quad n'_c = n_c$$

Feasibility preserved: sum stays 350; $n'_b \geq 20$ because $n_b > 20$.

Objective change:

$$\begin{aligned} \Delta &= 250(n_a + 1) + 300(n_b - 1) + 500n_c \\ &\quad - (250n_a + 300n_b + 500n_c) \end{aligned}$$

$$\Delta = 250 - 300 = -50 < 0$$

- ▶ So any feasible integer solution with $n_b > 20$ is not optimal.
- ▶ **Therefore, at an integer optimum we must have $n_b = 20$.**

Assume a feasible integer solution with $n_c > 50$.

Exchange move:

$$n'_a = n_a + 1, \quad n'_b = n_b, \quad n'_c = n_c - 1$$

Feasibility preserved: sum stays 350; $n'_c \geq 50$ because $n_c > 50$.

Objective change:

$$\begin{aligned} \Delta &= 250(n_a + 1) + 300n_b + 500(n_c - 1) \\ &\quad - (250n_a + 300n_b + 500n_c) \end{aligned}$$

$$\Delta = 250 - 500 = -250 < 0$$

- ▶ So any feasible integer solution with $n_c > 50$ is not optimal.
- ▶ **Therefore, at an integer optimum we must have $n_c = 50$.**

From previous steps:

At any integer optimum: $n_b = 20$ and $n_c = 50$

Sum constraint forces:

$$n_a = 350 - 20 - 50 = 280$$

Objective at $(280, 20, 50)$:

$$250 \cdot 280 + 300 \cdot 20 + 500 \cdot 50 = 101000$$

- Thus the integer optimum is $(n_a, n_b, n_c) = (280, 20, 50)$, and it is unique.

BINARY PROJECT SELECTION

Where should the company build to maximise profit?

A property developer's decisions

A property developer is considering projects in six cities: **Rome, Paris, Madrid, Berlin, Milan and Athens.**

In each city, the company may build at most one unit of each type:

- ▶ Residential
- ▶ Commercial
- ▶ Mercantile
- ▶ Educational
- ▶ Food Storage
- ▶ Industrial

There are $6 \times 6 = 36$ possible projects. Each requires a yes/no decision.

Revenues, costs and profit

For each possible project, we know its construction time and the monthly construction cost of its building type.

Project cost = months required $\times$ cost per month.

A completed unit earns the revenue specified for its city. In this example, this revenue is the same for all building types in that city.

Revenues are total amounts received per completed unit. All monetary amounts are in euros and are treated as directly comparable.

The objective is to maximise

$$\text{Profit} = \text{Total revenues} - \text{Total construction costs.}$$

The resource limits

1. The total number of selected projects must be at most 40.
2. The sum of construction months over all selected projects must be at most 80.
3. Every decision must be binary: 0 or 1.

The 80 months represent an **aggregate budget of construction effort**, for example 80 months of work by one shared construction team.

A redundant constraint

There are only 36 possible projects. Hence a limit of 40 units cannot exclude any binary selection. We retain it to illustrate a slack constraint.

Build the worksheet: monthly costs

Create a new worksheet and name it **Projects**. Enter the labels in row 2 and the monthly costs in row 3, exactly as shown.

B3:G3		<i>fx</i>	15000				
	A	B	C	D	E	F	G
1							
2	Building type	Residential	Commercial	Mercantile	Educational	Food Storage	Industrial
3	EUR / month	15,000	25,000	3,000	22,000	15,000	3,000

Each column is a **building type**. Widen the columns until all labels fit; use Wrap Text for Food Storage. Enter numbers without EUR and format them with a thousands separator and no decimal places.

Build the worksheet: unit revenues

In the same worksheet, enter the cities in B6:G6 and the revenue per completed unit in B7:G7.

B7:G7		<i>fx</i>	95000					
	A		B	C	D	E	F	G
5								
6	City		Rome	Paris	Madrid	Berlin	Milan	Athens
7	EUR / unit		95,000	100,000	90,000	83,000	98,000	75,000

Each column in this block is a **city**. For example, every completed project in Rome earns EUR 95,000.

Build the worksheet: construction months

Enter the headings in row 10, the building types in A11:A16, and the construction times in B11:G16.

B11:G16		<i>fx</i>	2				
	A	B	C	D	E	F	G
10	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens
11	Residential	2	5	5	4	6	8
12	Commercial	4	2	4	6	6	6
13	Mercantile	5	5	2	8	7	5
14	Educational	4	6	8	2	4	5
15	Food Storage	6	6	6	4	2	5
16	Industrial	8	8	5	5	5	2

The same row and column order will be used for the decisions and their costs. These construction times are fixed inputs.

Build the worksheet: the decision cells

Copy A10:G10 to A19:G19, and A11:A16 to A20:A25. Fill B20:G25 with zeros.

B20:G25		<i>fx</i>	0				
	A	B	C	D	E	F	G
19	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens
20	Residential	0	0	0	0	0	0
21	Commercial	0	0	0	0	0	0
22	Mercantile	0	0	0	0	0	0
23	Educational	0	0	0	0	0	0
24	Food Storage	0	0	0	0	0	0
25	Industrial	0	0	0	0	0	0

These are the 36 cells Solver will change: **1** selects a project and **0** excludes it.

Enter a trial selection

To check the formulas, change B20, C20, C21, F22, G22 and D25 to 1. Leave the other decisions at 0.

B20	<i>fx</i>	1					
	A	B	C	D	E	F	G
19	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens
20	Residential	1	1	0	0	0	0
21	Commercial	0	1	0	0	0	0
22	Mercantile	0	0	0	0	1	1
23	Educational	0	0	0	0	0	0
24	Food Storage	0	0	0	0	0	0
25	Industrial	0	0	1	0	0	0

This temporary selection contains six projects. It is a check of the worksheet, not the optimal plan.

Count the projects in each city

Type **Total** in A26. In B26, enter =SUM(B20:B25), then fill right through G26.

B26	<i>fx</i>	=SUM(B20:B25)					
	A	B	C	D	E	F	G
19	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens
20	Residential	1	1	0	0	0	0
21	Commercial	0	1	0	0	0	0
22	Mercantile	0	0	0	0	1	1
23	Educational	0	0	0	0	0	0
24	Food Storage	0	0	0	0	0	0
25	Industrial	0	0	1	0	0	0
26	Total	1	2	1	0	1	1

Drag the small fill handle from B26 to G26. Excel adjusts the column: C26 becomes =SUM(C20:C25).

Count the projects by building type

Type **Total** in H19. In H20, enter =SUM(B20:G20), then fill down through H25.

H20	<i>fx</i>	=SUM(B20:G20)						
	A	B	C	D	E	F	G	H
19	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens	Total
20	Residential	1	1	0	0	0	0	2
21	Commercial	0	1	0	0	0	0	1
22	Mercantile	0	0	0	0	1	1	2
23	Educational	0	0	0	0	0	0	0
24	Food Storage	0	0	0	0	0	0	0
25	Industrial	0	0	1	0	0	0	1
26	Total	1	2	1	0	1	1	

Drag the fill handle down to H25. The row number changes automatically: H21 becomes =SUM(B21:G21).

Count all selected projects

In H26, add the six city totals. Enter the formula shown in the formula bar.

H26	<i>fx</i>	=SUM(B26:G26)						
	A	B	C	D	E	F	G	H
19	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens	Total
20	Residential	1	1	0	0	0	0	2
21	Commercial	0	1	0	0	0	0	1
22	Mercantile	0	0	0	0	1	1	2
23	Educational	0	0	0	0	0	0	0
24	Food Storage	0	0	0	0	0	0	0
25	Industrial	0	0	1	0	0	0	1
26	Total	1	2	1	0	1	1	6

The trial selection gives **6 projects**. This cell will be constrained to be at most 40. Summing H20:H25 gives the same result.

Add the total construction months

Type **Months** in I19. In I26, enter
=SUMPRODUCT(B11:G16,B20:G25).

I26	<i>fx</i>	=SUMPRODUCT(B11:G16,B20:G25)							
	A	B	C	D	E	F	G	H	I
19	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens	Total	Months
20	Residential	1	1	0	0	0	0	2	
21	Commercial	0	1	0	0	0	0	1	
22	Mercantile	0	0	0	0	1	1	2	
23	Educational	0	0	0	0	0	0	0	
24	Food Storage	0	0	0	0	0	0	0	
25	Industrial	0	0	1	0	0	0	1	
26	Total	1	2	1	0	1	1	6	26

Each decision is multiplied by its construction time. For this selection, $2 + 5 + 2 + 7 + 5 + 5 = \mathbf{26 \text{ months}}$. Solver will require I26 to be at most 80.

Create the project-cost block

Copy A19:G19 to A28:G28, and A20:A25 to A29:A34. The empty cells will contain **formulas**.

B29:G34		fx					
	A	B	C	D	E	F	G
28	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens
29	Residential						
30	Commercial						
31	Mercantile						
32	Educational						
33	Food Storage						
34	Industrial						

We will build one row at a time. Keep the same building types and cities in the same order as the decision block.

Calculate the first project cost

For Residential in Rome, multiply the decision, the months and the monthly cost. Enter `=B20*B11*$B$3` in B29.

Decision			Months			Monthly cost		
B20	<i>fx</i>	1	B11	<i>fx</i>	2	B3	<i>fx</i>	15000
	A	B		A	B		A	B
20	Residential	1	11	Residential	2	3	EUR / month	15,000

B29	<i>fx</i>	=B20*B11*\$B\$3						
	A	B	C	D	E	F	G	
28	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens	
29	Residential	30,000						

The result is $1 \times 2 \times 15,000 = \mathbf{30,000}$ euros. The dollar signs keep the monthly-cost cell fixed when the formula is copied.

Copy the Residential cost across

Select B29 and drag its fill handle right to G29. The formula shown below is the copied formula in C29.

C29	<i>fx</i>	=C20*C11*\$B\$3					
	A	B	C	D	E	F	G
28	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens
29	Residential	30,000	75,000	0	0	0	0
30	Commercial						
31	Mercantile						
32	Educational						
33	Food Storage						
34	Industrial						

The decision and month references move to the Paris column. \$B\$3 stays fixed because this entire row is Residential.

Calculate the Commercial costs

In B30, enter `=B21*B12*$C$3`. Then fill right through G30.

B30	<i>fx</i>	<code>=B21*B12*\$C\$3</code>					
	A	B	C	D	E	F	G
28	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens
29	Residential	30,000	75,000	0	0	0	0
30	Commercial	0	50,000	0	0	0	0
31	Mercantile						
32	Educational						
33	Food Storage						
34	Industrial						

This row uses the Commercial monthly cost in `$C$3`: EUR 25,000.

Do not fill B29 down: the fixed monthly-cost reference must change for each building type.

Calculate the Mercantile costs

In B31, enter `=B22*B13*$D$3`. Then fill right through G31.

B31	<i>fx</i>	<code>=B22*B13*\$D\$3</code>					
	A	B	C	D	E	F	G
28	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens
29	Residential	30,000	75,000	0	0	0	0
30	Commercial	0	50,000	0	0	0	0
31	Mercantile	0	0	0	0	21,000	15,000
32	Educational						
33	Food Storage						
34	Industrial						

This row uses the Mercantile monthly cost in `$D$3`: EUR 3,000.

Calculate the Educational costs

In B32, enter =B23*B14*\$E\$3. Then fill right through G32.

B32	<i>fx</i>	=B23*B14*\$E\$3					
	A	B	C	D	E	F	G
28	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens
29	Residential	30,000	75,000	0	0	0	0
30	Commercial	0	50,000	0	0	0	0
31	Mercantile	0	0	0	0	21,000	15,000
32	Educational	0	0	0	0	0	0
33	Food Storage						
34	Industrial						

This row uses the Educational monthly cost in \$E\$3: EUR 22,000.

Calculate the Food Storage costs

In B33, enter =B24*B15*\$F\$3. Then fill right through G33.

B33	<i>fx</i>	=B24*B15*\$F\$3					
	A	B	C	D	E	F	G
28	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens
29	Residential	30,000	75,000	0	0	0	0
30	Commercial	0	50,000	0	0	0	0
31	Mercantile	0	0	0	0	21,000	15,000
32	Educational	0	0	0	0	0	0
33	Food Storage	0	0	0	0	0	0
34	Industrial						

This row uses the Food Storage monthly cost in \$F\$3: EUR 15,000.

Calculate the Industrial costs

In B34, enter =B25*B16*\$G\$3. Then fill right through G34.

B34	<i>fx</i>	=B25*B16*\$G\$3					
	A	B	C	D	E	F	G
28	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens
29	Residential	30,000	75,000	0	0	0	0
30	Commercial	0	50,000	0	0	0	0
31	Mercantile	0	0	0	0	21,000	15,000
32	Educational	0	0	0	0	0	0
33	Food Storage	0	0	0	0	0	0
34	Industrial	0	0	15,000	0	0	0

This row uses the Industrial monthly cost in \$G\$3: EUR 3,000.

Add the total construction cost

Type **Costs**, **Revenues** and **Profit** in F36:F38. In G36, enter `=SUM(B29:G34)`.

G36	<i>fx</i>	=SUM(B29:G34)					
	A	B	C	D	E	F	G
35							
36						Costs	206,000
37						Revenues	
38						Profit	

Widen columns F and G so the labels and amounts fit. The project-cost formulas sum to **EUR 206,000**; G37:G38 will be completed next.

Match unit revenues to city totals

The two rows below use the same city order. For each city, revenue equals the revenue per unit multiplied by its project count.

B7:G7		<i>fx</i>	95000				
	A	B	C	D	E	F	G
6	City	Rome	Paris	Madrid	Berlin	Milan	Athens
7	EUR / unit	95,000	100,000	90,000	83,000	98,000	75,000

B26:G26		<i>fx</i>	=SUM(B20:B25)				
	A	B	C	D	E	F	G
25	Industrial	0	0	1	0	0	0
26	Total	1	2	1	0	1	1

For Rome: $95,000 \times 1$. For Paris: $100,000 \times 2$. SUMPRODUCT adds the six city contributions.

Add the total revenues

In G37, enter `=SUMPRODUCT(B7:G7,B26:G26)`. The formula pairs each unit revenue with the corresponding city total.

G37	<i>fx</i>	<code>=SUMPRODUCT(B7:G7,B26:G26)</code>					
	A	B	C	D	E	F	G
35							
36						Costs	206,000
37						Revenues	558,000
38						Profit	

For the trial selection, total revenues are **EUR 558,000**. Both ranges run from Rome in column B to Athens in column G.

Complete the profit cell

In G38, enter =G37-G36: total revenues minus total construction costs.

G38	<i>fx</i>	=G37-G36					
	A	B	C	D	E	F	G
35							
36						Costs	206,000
37						Revenues	558,000
38						Profit	352,000

The check is $558,000 - 206,000 = \mathbf{352,000}$ euros. G38 is the objective cell that Solver will maximise.

Check the completed worksheet

The trial values should now agree with the checks below. If they do, the worksheet is ready for Solver.

H26:I26		fx	=SUM(B26:G26)						
	A	B	C	D	E	F	G	H	I
19	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens	Total	Months
20	Residential	1	1	0	0	0	0	2	
21	Commercial	0	1	0	0	0	0	1	
22	Mercantile	0	0	0	0	1	1	2	
23	Educational	0	0	0	0	0	0	0	
24	Food Storage	0	0	0	0	0	0	0	
25	Industrial	0	0	1	0	0	0	1	
26	Total	1	2	1	0	1	1	6	26

Checks: H26 = 6 projects; I26 = 26 months.

G36 = 206,000; G37 = 558,000; G38 = 352,000.

Only B20:G25 contains decisions; the totals and financial result contain formulas. Save your workbook.

Select the decision range

Select B20:G25, then right-click and choose **Define Name**.
Name only the decision cells, excluding the totals.

B20:G25		<i>fx</i>	1						
	A	B	C	D	E	F	G	H	I
19	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens	Total	Months
20	Residential	1	1	0	0	0	0	2	
21	Commercial	0	1	0	0	0	0	1	
22	Mercantile	0	0	0	0	1	1	2	
23	Educational	0	0	0	0	0	0	0	
24	Food Storage	0	0	0	0	0	0	0	
25	Industrial	0	0	1	0	0	0	1	
26	Total	1	2	1	0	1	1	6	26

The next dialog assigns the name **Decisions** to this 6×6 range.

Give the decisions a name

In the New Name dialog, enter **Decisions**. Check the worksheet name and the absolute cell references, then click **OK**.

New Name

Name:	Decisions
Scope:	Workbook
Refers to:	=Projects!\$B\$20:\$G\$25

The name now refers to **Projects!\$B\$20:\$G\$25**. It can be used directly in the Solver settings.

Solver settings for the project selection

Schematic Excel interface

Solver Parameters

Set Objective:

To: ☒ Max ☐ Min

By Changing Variable Cells:

Subject to the Constraints:

\$H\$26 <= 40
\$I\$26 <= 80
Decisions = binary

Add

Change

Delete

☒ Make Unconstrained Variables Non-Negative

Solving Method:

Open **Data > Solver**. Use **Add** for each constraint; choose **bin** for the decision range.

Why this is a linear binary model

Let x_{ij} be the decision for building type i in city j . Write c_i for the monthly cost, t_{ij} for construction time and r_j for revenue per completed unit.

$$\begin{aligned} \max_x \quad & \sum_{i=1}^6 \sum_{j=1}^6 (r_j - c_i t_{ij}) x_{ij} \\ \text{subject to} \quad & \sum_{i=1}^6 \sum_{j=1}^6 x_{ij} \leq 40, \\ & \sum_{i=1}^6 \sum_{j=1}^6 t_{ij} x_{ij} \leq 80, \\ & x_{ij} \in \{0, 1\}. \end{aligned}$$

Each decision is multiplied only by fixed data. This is a **linear model with binary variables**.

Solver options for a binary optimum

Open **Options**, then the **All Methods** tab.

Solver Options: All Methods

☐ Ignore Integer Constraints

Integer Optimality (%):

Keep the binary restrictions active. Setting Integer Optimality to 0% asks Solver to establish the best objective value, rather than stop with a positive allowed optimality gap.

Click **OK**, then **Solve**. Check the completion message before choosing **Keep Solver Solution**.

Source: Frontline Systems, Solver options.

A finite search can still be difficult

With 36 binary choices, there are

$$2^{36} = 68,719,476,736$$

possible selections before imposing the resource limits.

The set is finite, but checking every selection would be a large task. Solver uses bounds and a search procedure to avoid checking them all.

A 0% optimality tolerance can require more computation than a positive tolerance. Time or iteration limits may interrupt the search.

A feasible solution and a proof of optimality are different outcomes. Always inspect the Solver termination message.

Source: Frontline Systems, integer constraints and optimality tolerance.

The optimal binary decisions

After **Solve**, keep the solution. These are the values in B20:G25.

B20:G25 f_x 1									
	A	B	C	D	E	F	G	H	I
19	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens	Total	Months
20	Residential	1	1	0	0	0	0	2	
21	Commercial	0	1	0	0	0	0	1	
22	Mercantile	1	1	1	1	1	1	6	
23	Educational	0	0	0	1	0	0	1	
24	Food Storage	0	0	0	0	1	0	1	
25	Industrial	1	1	1	1	1	1	6	
26	Total	3	4	2	3	3	2	17	78

The formulas report **17 projects** in H26 and **78 months** in I26. This is the unique optimal binary plan.

Construction costs of the optimal plan

The formulas in B29:G34 update automatically for the selected projects. All costs are in euros.

B29	<i>fx</i>	=B20*B11*\$B\$3					
	A	B	C	D	E	F	G
28	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens
29	Residential	30,000	75,000	0	0	0	0
30	Commercial	0	50,000	0	0	0	0
31	Mercantile	15,000	15,000	6,000	24,000	21,000	15,000
32	Educational	0	0	0	44,000	0	0
33	Food Storage	0	0	0	0	30,000	0
34	Industrial	24,000	24,000	15,000	15,000	15,000	6,000

In G36, =SUM(B29:G34) gives the total construction cost: **EUR 424,000**.

Calculating the optimal profit

The three formulas already entered in G36:G38 now display the optimal financial result.

G38	<i>fx</i>	=G37-G36					
	A	B	C	D	E	F	G
35							
36						Costs	424,000
37						Revenues	1,558,000
38						Profit	1,134,000

G36: =SUM(B29:G34); G37: =SUMPRODUCT(B7:G7,B26:G26).

Maximum profit: **EUR 1,558,000 – EUR 424,000 = EUR 1,134,000.**

Why are two months left unused?

For a constraint of the form “use $\leq$ capacity”, its *slack* is the unused capacity.

$$\text{Unit slack} = 40 - 17 = 23,$$

$$\text{Month slack} = 80 - 78 = 2.$$

- ▶ The limit of 40 units is redundant: there are only 36 possible binary projects.
- ▶ Although some projects take two months, **all six two-month projects are already selected.**
- ▶ Every omitted project requires at least four months. The two spare months cannot fund another whole project.

This explains why adding a project is impossible. Optimality also requires checking that *replacing* selected projects cannot improve profit.

A binary result with a 5% tolerance

In **Options > All Methods**, set **Integer Optimality (%)** to 5, then solve. This is an example of a feasible result.

B20:G25		f_x	1						
	A	B	C	D	E	F	G	H	I
19	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens	Total	Months
20	Residential	1	1	1	1	0	0	4	
21	Commercial	0	1	0	0	0	0	1	
22	Mercantile	1	1	1	1	0	1	5	
23	Educational	0	0	0	1	0	0	1	
24	Food Storage	0	0	0	0	1	0	1	
25	Industrial	1	1	1	1	1	1	6	
26	Total	3	4	3	4	2	2	18	80

G36: EUR 538,000 costs; G37: EUR 1,633,000 revenues.

G38: **EUR 1,095,000 profit.**

What does the 5% setting mean?

Integer Optimality (%) is a stopping tolerance for the gap between a feasible integer solution and Solver's bound on the optimum.

- ▶ A positive tolerance is meaningful for an integer problem: Solver may stop before proving exact optimality.
- ▶ Binary decisions remain binary. The tolerance does *not* allow fractional buildings.
- ▶ A 5% tolerance does not require a 5% loss; Solver may still find the exact optimum.

For the two binary plans shown, the actual profit loss is

$$1,134,000 - 1,095,000 = \text{EUR } 39,000, \quad \frac{39,000}{1,134,000} \approx 3.44\%.$$

The second plan uses all 80 months, but earns less profit. A smaller slack does not imply a better solution.

Relaxing the binary restrictions

To study a different, continuous problem, replace

$$x_{ij} \in \{0, 1\} \quad \text{with} \quad 0 \leq x_{ij} \leq 1.$$

Keep the same objective and the same resource constraints.

The continuous relaxation

Fractional project quantities are now mathematically allowed. The model still permits at most one unit of each building type in each city.

In Solver, select **Ignore Integer Constraints**. Add explicit bounds `Decisions >= 0` and `Decisions <= 1`.

This is a deliberate change of model, separate from changing **Integer Optimality (%)**.

The continuous optimum

With $0 \leq x_{ij} \leq 1$, one optimal relaxed solution gives these values in B20:G25.

E24	f_x	0.75							
	A	B	C	D	E	F	G	H	I
19	Building type	Rome	Paris	Madrid	Berlin	Milan	Athens	Total	Months
20	Residential	1	0	0	1	0	0	2	
21	Commercial	0	1	0	0	0	0	1	
22	Mercantile	1	1	1	1	1	1	6	
23	Educational	0	0	0	1	0	0	1	
24	Food Storage	0	0	0	0.75	1	0	1.75	
25	Industrial	1	1	1	1	1	1	6	
26	Total	3	3	2	4.75	3	2	17.75	80

G36: EUR 454,000 costs; G37: EUR 1,603,250 revenues.

G38: **EUR 1,149,250 relaxed profit.**

A higher value, but a different feasible set

Every binary plan is feasible for the continuous relaxation. The converse is false.

$$P_{\text{binary}}^* \leq P_{\text{relaxed}}^*.$$

The relaxed optimum is therefore an **upper bound** on achievable binary profit:

$$\text{EUR } 1,134,000 < \text{EUR } 1,149,250.$$

The extra EUR 15,250 comes from allowing fractional projects; it is not an attainable improvement for the original binary problem.

The relaxed solution is not unique

In Berlin, Residential and Food Storage each require 4 months and earn EUR 23,000 profit per unit. Swapping their values 1 and 0.75 gives another relaxed optimum.

Why we cannot just round the solution

The displayed relaxed plan has $x_{\text{Food Storage, Berlin}} = 0.75$. A whole project requires four months and earns EUR 23,000 profit.

- ▶ **Round up to 1:** an additional $0.25 \times 4 = 1$ month is needed. Total time becomes 81 months, so the plan is infeasible.
- ▶ **Round down to 0:** release $0.75 \times 4 = 3$ months and lose $0.75 \times 23,000 = 17,250$ euros.

Rounded-down time = 77 months,

Rounded-down profit = EUR 1,132,000.

The rounded-down plan is feasible, but falls EUR 2,000 short of the binary optimum. Solving the binary model matters.

Comparing the three solutions

Compare the objective value and feasibility for the model being solved.

Binary optimum. 17 units, 78 months, **EUR 1,134,000 profit**. This is the unique optimum of the original problem.

Binary result with a 5% tolerance. 18 units, 80 months, **EUR 1,095,000 profit**. Feasible but suboptimal; unit slack: 22, month slack: 0.

Continuous optimum. 17.75 units, 80 months, **EUR 1,149,250 profit**. Solves the relaxation; unit slack: 22.25, month slack: 0. The fractional decision violates the original binary requirement.