

Optimization with Excel Solver

SUMMER SCHOOL, 06/09/2018 MISKOLC

UNIVERSITY OF MISKOLC

INSTITUTE OF LOGISTICS

DR. LÁSZLÓ KOTA



Solver in MS Excel

The Solver in Excel not a Microsoft product

Originally created by Frontline Systems in 1991, MS Excel 3.0

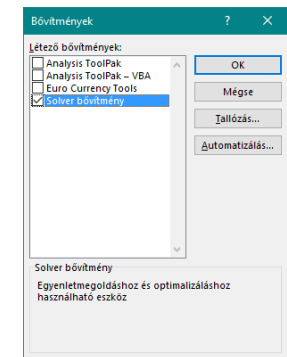
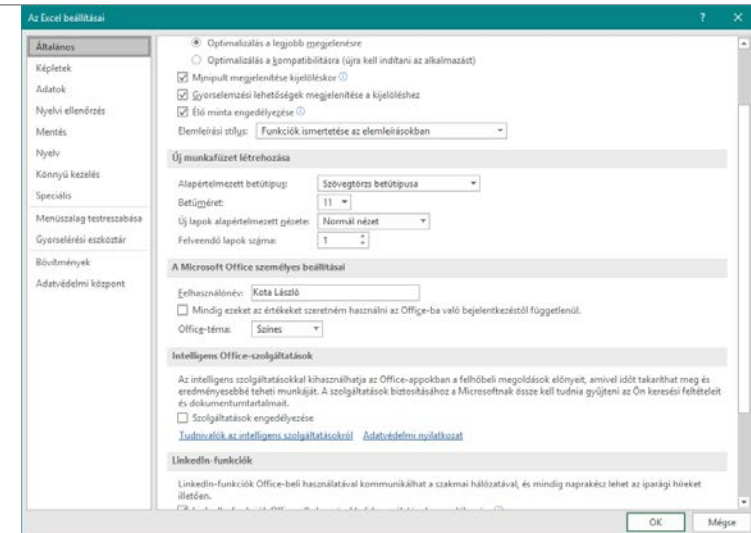
Well integrated

Improved in 2010, able to handle INDEX, VLOOKUP functions

Hidden until not enabled

Enable in: File/Settings/Extensions/Excel Extensions/

Jump/Solver Extension



Limitations

Limits on decision variable

- 200 decision variables (both linear and non linear problems)

Limits on constraints

- 100 constraints

Frontline's Solver: (licenses vary from \$250 (Basic) to \$5000 (Comprehensive) per year)

- 2000 decision variables for linear, 500 non linear problems
- 8000 in premium, but built in engines can handle virtually unlimited
- 1000 constraint for linear, 250 for non linear in premium, several thousands in larger scale engines

Basics

Optimization: Selection of a best element with regard to some constraints from some set of available alternatives

Have to define:

Target function

Decision variable

Constraints

A Solver paramétereit

Célérték beállítása: SJS4

Cél: Max Min Érték: 0

Változócellák módosításával:

Vonatkozó korlátozások:

Nem korlátozott változók nemnegatívá tétele

Válasszon egy megoldási módszert: Nemlineáris ÁRG

Megoldási metódus

A sima nemlineáris Solver-problémákhoz válassza a nemlineáris ÁRG motort. Lineáris Solver-problémákhoz válassza az LP szimplex motort, a nem sima Solver-problémákhoz pedig az evolutív motort.

Súgó Megoldás Bezárás

Solver Methods

In the Solver engine you can choose:

- Simplex LP
 - Simplex method linear programming
- Nonlinear GRG (Generalized Reduced Gradient)
 - Published in 1973 and 1975
- Evolutionary method

Fine tuning

- Exit criteria
- Multiple points
- Random generator initialization
- Population size
- Mutation rate

The image shows the 'A Solver paramétereit' dialog box in Excel. The 'Célérték beállítása:' field is set to 'SJS4'. The 'Cél:' section has 'Max' selected. The 'Értéke:' field is set to '0'. The 'Változócellák módosításával:' field is empty. The 'Vonatkozó korlátozások:' list is empty, with buttons for 'Hozzáadás', 'Csere', 'Törlés', 'Alaphelyzet', and 'Betöltés/mentés'. The checkbox 'Nem korlátozott változók nemnegatívá tétele' is checked. The 'Válasszon egy megoldási módszert:' dropdown is set to 'Nemlineáris ÁRG'. The 'Megoldási módszer' section contains the text: 'A sima nemlineáris Solver-problémákhoz válassza a nemlineáris ÁRG motort. Lineáris Solver-problémákhoz válassza az LP szimplex motort, a nem sima Solver-problémákhoz pedig az evolutív motort.' At the bottom, there are buttons for 'Súgó', 'Megoldás', and 'Bezárás'.

Solve a simple problem

Five suppliers

Minimum order

Maximum order

Constant price

Linear transportation cost

Integer criteria

The screenshot shows an Excel spreadsheet with a Solver Parameters dialog box open. The spreadsheet contains a table with the following data:

| | Supplier 1 | Supplier 2 | Supplier 3 | Supplier 4 | Supplier 5 | |
|---------------------|------------|------------|------------|------------|------------|--------------------|
| Min order | 50 | 20 | 50 | 20 | 20 | |
| Max order | 100 | 80 | 150 | 120 | 170 | |
| Price | 3 | 1 | 2 | 7 | 2 | |
| Transportation cost | 6 | 1 | 2 | 4 | 9 | |
| Order | 100 | 80 | 150 | 120 | 50 | Total quantity 500 |
| Cost | 900 | 160 | 600 | 1320 | 550 | Total Cost 3530 |

The Solver Parameters dialog box is configured as follows:

- Cell Reference: \$E\$10
- To: Max, Min, or Value Of: 0
- Variable Cells: \$E\$13:\$E\$15
- Constraints: \$E\$13:\$E\$13 <= \$E\$7:\$E\$7, \$E\$13:\$E\$13 = \$E\$4:\$E\$4, \$E\$13:\$E\$13 <= \$E\$6:\$E\$6, \$E\$13 = 500
- Make Unconstrained Variables Non-Negative: checked
- Select a Solving Method: Simplex LP
- Options: Make Unconstrained Variables Non-Negative, Select a GRG Nonlinear engine, Select Solver Load/Save options, Select Solver engine for problem, Select Solver engine for problem

Solver with custom functions

Five suppliers

Non linear price

- Price is a step function, due to discount
 - Starting price
 - Price decrease
 - Steps count

Non linear transportation cost

- Transport const per vehicle can be different at every supplier
- Truck capacity can be different at every supplier

Non linear functions are programmed with the built in Visual Basic editor

Custom functions with Programming

The screenshot shows the Microsoft Excel Solver Parameters dialog box overlaid on a spreadsheet. The spreadsheet contains a table with columns for Supplier 1, Supplier 2, Supplier 3, Supplier 4, and Supplier 5. The rows include Min quantity, Max quantity, Price, Transport cost total, Price start, Price step, Price dec, Max step, Transport cost / vehicle, and Vehicle capacity. The Solver dialog box is open, showing the objective cell as \$E\$20, the variable cells as \$E\$12:\$I\$20, and constraints including \$E\$12:\$I\$20 >= \$E\$7:\$I\$7, \$E\$12:\$I\$20 = egész, \$E\$12:\$I\$20 <= \$E\$6:\$I\$6, and \$I\$20 = 500. The 'Evolatív' (Evolutionary) method is selected.

| | Supplier 1 | Supplier 2 | Supplier 3 | Supplier 4 | Supplier 5 | |
|--------------------------|------------|------------|------------|------------|------------|-----------------------|
| Min quantity | 50 | 20 | 50 | 20 | 20 | |
| Max quantity | 100 | 80 | 130 | 120 | 170 | |
| Price | 297 | 800 | 2589 | 1400 | 3120 | |
| Transport cost total | 120 | 20 | 60 | 40 | 360 | |
| Price start | 3 | 10 | 20 | 70 | 20 | |
| Price step | 100 | 100 | 100 | 50 | 40 | |
| Price dec | 1 | 1 | 1 | 1 | 1 | |
| Max step | 2 | 6 | 10 | 20 | 15 | |
| Transport cost / vehicle | 60 | 10 | 20 | 40 | 90 | |
| Vehicle capacity | 50 | 50 | 50 | 50 | 50 | |
| Order | 99 | 80 | 131 | 20 | 170 | Total quantity 500 |
| Cost | 417 | 820 | 2649 | 1440 | 3480 | Total Cost 8806 |

Custom functions with Programming

Visual Basic editor ALT+F11

Any custom and built in function can be used

Even DLL modules

Can slow the optimization

```
Public Function GetPrice(quantity_in As Integer, price_in As Integer,
step_in As Integer, dec_in As Integer, step_max As Integer) As Integer

    Dim price_total As Integer
    price_total = 0
    For x = 0 To quantity_in - 1
        If (Int(x / step_in) <= (step_max - 1)) Then
            price_total = price_total + (price_in - (Int(x / step_in) *
dec_in))
        Else
            price_total = price_total + (price_in - (step_max - 1) *
dec_in)
        End If
    Next x
    GetPrice = price_total
End Function

Public Function GetTransport(quantity_in As Integer, price_in As Integer,
cap_in As Integer) As Integer
    If quantity_in = 0 Then
        GetTransport = 0
        Exit Function
    End If
    Dim transport As Integer
    transport_num = 1 + Int(quantity_in / cap_in)
    GetTransport = transport_num * price_in
End Function
```

Using Solver on bigger problems

A problem which solve the olive oil distribution in large regions

Target function

$$\sum_{i=1}^I \sum_{k=1}^K (p_i - c_{ik}) Y_{ik} - \sum_{i=1}^I (oc * oil_i - pc_i) X_i - t * toil$$

Divided into two parts

180 input variables

140 decision variables

7 constraint rows, includes matrix constraints

Using Solver on bigger problems

