EuroSymphony Solver

After opening Lotus Symphony Spreadsheet EuroSymphony Solver can be reached in Tools menu EuroSymphony Solver submenu. It provides four algorithms to solve optimization problems, namly Simplex, Quadratic, Non-linear and Hungarian. More details are in context help.

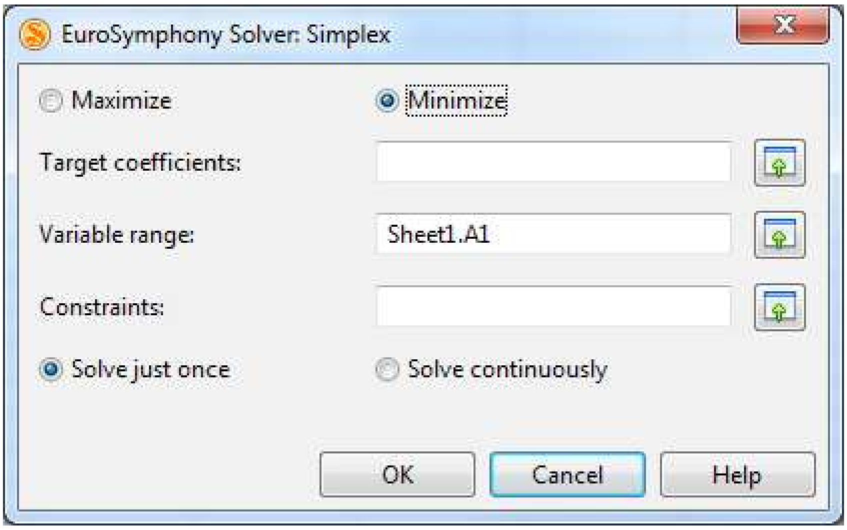
The Simplex Algorithm

The simplex algorithm is a popular technique for numerical solution of the linear programming problem.

A linear programming problem consists of a set of linear inequalities (the constraints) on a number of non-negative real variables and a linear function (the target function) on the same real variables which is to be maximized or minimized.

To use the simplex algorithm of EuroOffice Solver, pick the corresponding menu item in the *Tools* menu. In the dialog that appears you can set the cells that contain the coefficients of the target function, the cells to change (the variable cells) and the cell range describing the constraints to be observed. If some cells are selected when you open the dialog, these will be assumed to be the variable cells. Note that the variable cells will be overwritten with the solution! If there are N variables, the target range and the variable range both have to be N cells wide and 1 cell high.

The inequalities or constraints are expressed in a cell range N+2 cells wide and as high as many inequalities there are – each row represents an inequality. The first N cells in each row contain the coefficients of a function over the variable cells. The second cell from the right contains a symbol of relationship: either "<=", ">=", "=" or "==", the last two of which are synonymous. The rightmost cell contains a constant (or a formula over something other than the changing cells).



EuroSymphony Solver: Simplex

○ Maximize ● Minimize

Target coefficients:

Variable range: Sheet1.A1

Constraints:

● Solve just once ○ Solve continuously

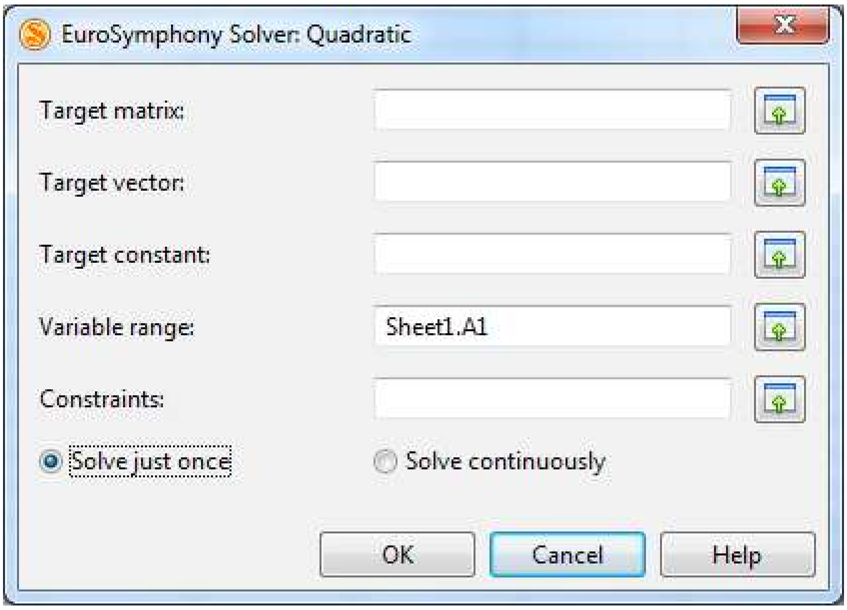
Quadratic Solver

The free version of EuroOffice Solver does not contain this solver method. It is only available in the Professional version.

The quadratic solver algorithm can be used to solve quadratic optimization problems. The problem is similar to the linear programming problem, but the target function that is minimized can be quadratic. A quadratic optimization problem consists of a set of linear inequalities (the constraints) on a number of non-negative real variables and a quadratic function (the target function) on the same real variables which is to be minimized.

To use the quadratic solver pick the corresponding menu item in the Tools menu. In the dialog that appears you can set the cells that contain the coefficients of the target function, the cells to change (the variable cells) and the cell range describing the constraints to be observed. If some cells are selected when you open the dialog, these will be assumed to be the variable cells. Note that the variable cells will be overwritten with the solution!

If there are N variables, the target function is described by an N by N matrix for the quadratic component, a linear range that is N cells wide and 1 cell high and a single cell for a constant. The variable range has to be N cells wide and 1 cell high as well. The inequalities or constraints are expressed in a cell range N+2 cells wide and as high as many inequalities there are – each row represents an inequality. The first N cells in each row contain the coefficients of a function over the variable cells. The second cell from the right contains a symbol of relationship. For the quadratic solver it can only be "<=". The rightmost cell contains a constant (or a formula over something other than the changing cells).



EuroSymphony Solver: Quadratic

Target matrix:

Target vector:

Target constant:

Variable range: Sheet1.A1

Constraints:

● Solve just once ○ Solve continuously

Non-linear Solver

The free version of EuroOffice Solver does not contain this solver method. It is only available in the Professional version.

The non-linear solver algorithm can be used to solve a number of non-linear optimization problems. The problems solved consist of a set of linear inequalities (the constraints) on a number of non-negative real variables and an arbitrary function (the target function) on the same real variables which is to be maximized or minimized.

To use the non-linear solver pick the corresponding menu item in the Tools menu. In the dialog that appears you can set the cell that contains the target function, the cells to change (the variable cells) and the cell range describing the constraints to be observed. If some cells are selected when you open the dialog, these will be assumed to be the variable cells. Note that the variable cells will be overwritten with the solution!

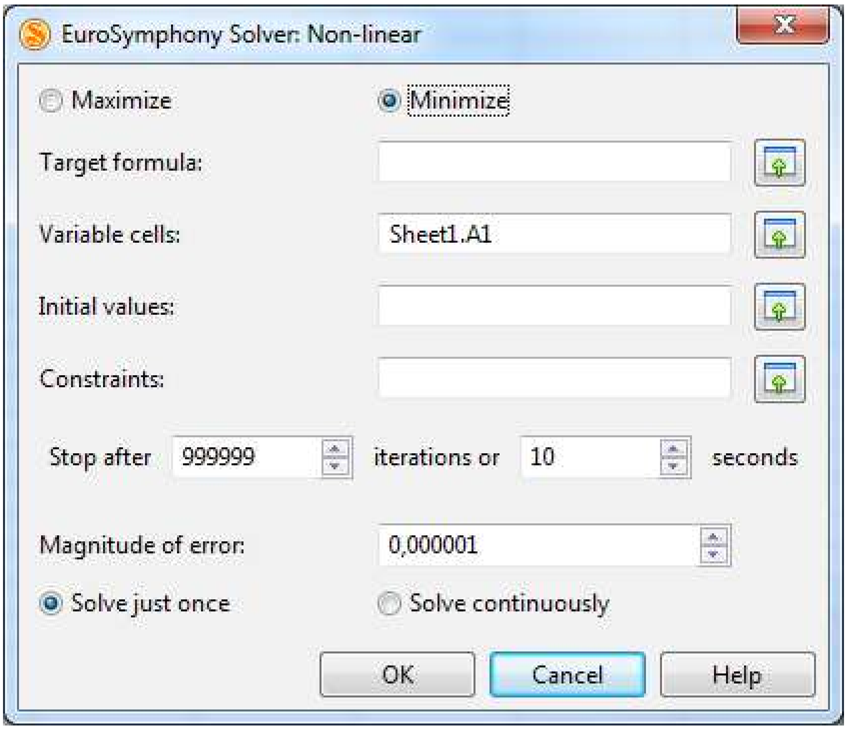
The target function is given as a regular OpenOffice.org formula, complete with the "=" sign at the start. You can use any function available in OpenOffice.org, but note that the solver will only perform well with target functions that change their values gradually as their parameters change.

If there are N variables, the variable range has to be N cells wide and 1 cell high. The range for the initial values has to be N cells wide and 1 cell high as well. The algorithm will start out from these values, so they have to satisfy the constraints. If the target function has multiple local optima, the initial values have to be near the optimum aimed at for the algorithm to perform well. Note that if you use the "Solve just once" setting, it is fine to use the same cells as both variable cells and initial value cells, but if you use the "Solve continuously" setting to create a continuous solver you have to use a separate cell range for initial values.

The inequalities or constraints are expressed in a cell range N+2 cells wide and as high as many inequalities there are – each row represents an inequality. The first N cells in each row contain the coefficients of a linear function over the variable cells. The second cell from the right contains a symbol of relationship: either "<=", ">=", "=" or "==", the last two of which are synonymous. The rightmost cell contains a constant (or a formula over something other than the changing cells).

You can set the error magnitude of the algorithm. This setting influences how accurate the solution is going to be and also how fast it will be found. If the numbers in your calculations are of unusually large or small magnitude, then change this setting accordingly.

You can also set a limit on how much time the optimization is allowed to take. It is usually convenient to give this setting in seconds, but if you want consistent behaviour across computers of varying CPU strengths, you can set a limit in terms of maximum number of iterations.



EuroSymphony Solver: Non-linear

○ Maximize ● Minimize

Target formula:

Variable cells: Sheet1.A1

Initial values:

Constraints:

Stop after 999999 iterations or 10 seconds

Magnitude of error: 0,000001

● Solve just once ○ Solve continuously

Hungarian Algorithm

The free version of EuroOffice Solver does not contain this solver method. It is only available in the Professional version.

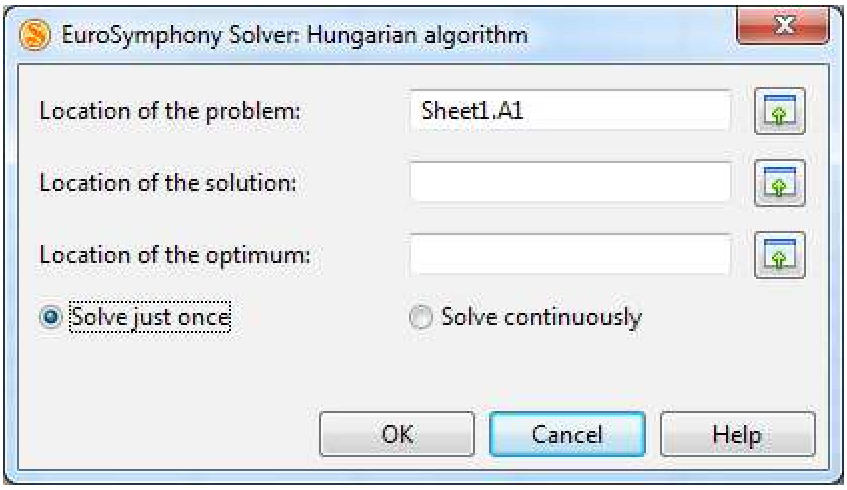
The Hungarian algorithm is a combinatorial optimization algorithm which solves assignment problems in polynomial time. The assignment problem is one of the fundamental combinatorial optimization problems in the branch of optimization or operations research in mathematics. It consists of finding a maximum weight matching in a weighted bipartite graph. The form of the problem that is solved by the implementation in EuroOffice Solver is as follows:

There are a number of suppliers and a number of consumers. There is an amount of goods supplied by each supplier and an amount of goods consumed by each consumer. The number of suppliers and consumers can be different, but the total amount of goods supplied and consumed must be equal. There is a function assigning a number to each pair of suppliers and consumers that is the cost of transporting a single item of goods from the given supplier to the given consumer. It is required to create a function that assigns a number to each pair of suppliers and consumers that is the number of goods to be transported from the given supplier to the given consumer so that for every supplier the sum of the goods transported from it to the consumers is equal to the number of goods supplied by it, for every consumer the sum of goods transported to it is equal to the number of goods that it is willing to consume and that the total cost of transportation is minimal.

To use the Hungarian algorithm of EuroOffice Solver pick the corresponding menu item in the Tools menu. In the dialog that appears you can set the cell range that describes the problem, the cell range to output the solution to and a single cell to output the value of the optimal solution to. If there is a range selected when you open the dialog it will be assumed to be the problem range. If there is no range selected then the range with numbers around the cursor will be guessed to be the problem range. The problem range for a problem of S suppliers and C consumers is S+1 cells tall and C+1 cells wide. The first cell of the first row is ignored. The rest of the first row contains the amounts of goods the consumers are willing to consume. The rest of the first column contains the amounts of goods the suppliers supply.

The rest of the table contains the transportation cost function. If transportation is impossible between a supplier and a consumer then enter "X" in the cell corresponding to the supplier and consumer instead of a numerical value for the cost function.

You have to set either both the solution range and the optimal value cell or just one of them. If you set the solution range it has to be S cells tall and C cells wide for a problem of S suppliers and C consumers. The cells of this range will be filled with the numbers of the solution indicating how many goods should be transported between the given pair of supplier and consumer. Note that any values currently in these cells will be overwritten! The same goes for the cell that is to contain the optimal cost.



EuroSymphony Solver: Hungarian algorithm

Location of the problem: Sheet1.A1

Location of the solution:

Location of the optimum:

● Solve just once ○ Solve continuously

Using Continuous Solvers

Once you have set up the Solver dialog box, you have the choice of either solving the problem just once or creating a continuous solver. In both cases the problem will be solved by changing the variable cells. But if you have chosen to create a continuous solver, it will continue monitoring the target formula, the cells describing the initial values and the constraint range and whenever they change, the variable cells will immediately be updated with the new solution. This is very useful for quickly trying out different scenarios. The continuous solver will even be saved into the OpenDocument Spreadsheet file and will continue to function when loaded into an OpenOffice.org application that has the EuroOffice Solver extension installed.

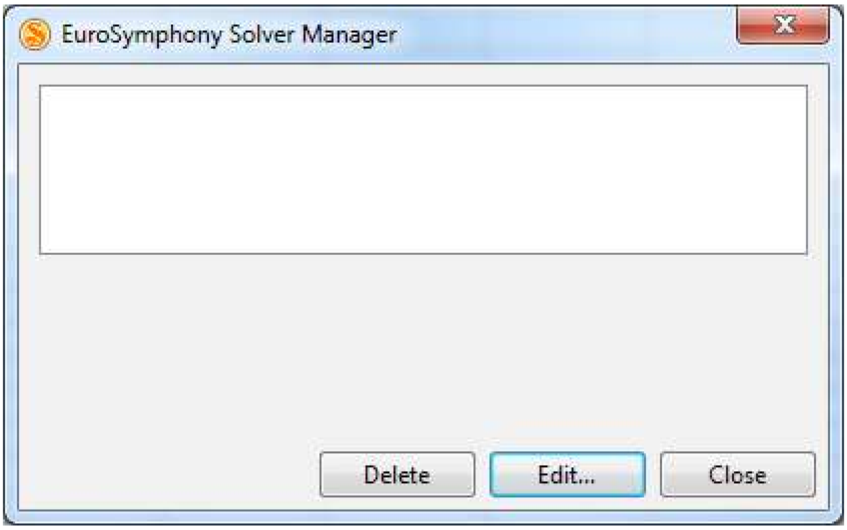
If you have the variable range of a continuous solver selected when you open the non-linear solver dialog from the menu, the dialog will be filled with the properties of the solver. If you create a continuous solver with the same variable range as an already existing continuous solver then you will be asked if you want to overwrite the existing solver.

Solver Manager

The Solver Manager is a dialog for listing, modifying and deleting continuous solvers.

You can also use this dialog to create a new continuous solver that is set up similarly to an already running one. To do this choose to edit an existing solver, modify any settings you would like to (you must change the variable range, or the problem range for the Hungarian algorithm solver) and click OK.

You can also access and modify a continuous solver by selecting its variable range (or the problem range for the Hungarian algorithm solver) and clicking the menu item corresponding to the type of the continuous solver. You can not delete a continuous solver this way though.


EuroSymphony Solver Manager
Delete
Edit...
Close