Optimisation Modelling with Excel and CMPL2

Mike Steglich
mike.steglich@th-wildau.de

Technical University of Applied Sciences Wildau

SIMUL 2021

Prof. Dr. Mike Steglich

- **Vita**
  - Degree in Business Informatics at the Department of Economics of the Martin-Luther-University Halle-Wittenberg in 1993
  - Scientific assistant at the Department of Economics at the Martin-Luther-University Halle-Wittenberg 1993-1999
  - Dissertation (Dr. rer. pol.) on cost variance analysis with artificial neural networks at the Department of Economics of the Martin-Luther-University Halle-Wittenberg in 2001 (summa cum laude)
  - Management Accountant at the MTU Maintenance Berlin-Brandenburg GmbH (Maintenance, Repair and Overhauls of Flight-Engines) 2000-2004
  - Professor of Business Administration, Quantitative Methods and Management Accounting at the Department of Business, Computing, Law of the Technical University of Applied Sciences Wildau, since September 2004
- **Interests**
  - Mathematical Modelling, in particular Mathematical Programming Languages
  - Logistical Decisions
  - Intensively involved in the development of software for mathematical modelling and optimisation in several open-source projects

Bringing two worlds together

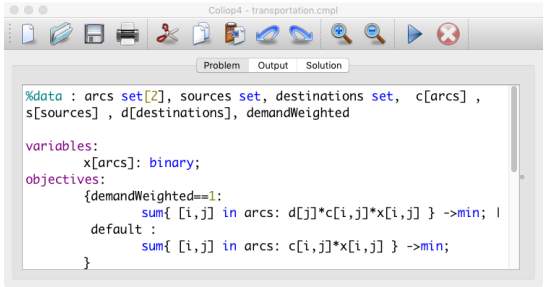
- **Manager**
  - Responsible for planning, directing and controlling in companies or other organisations
  - **Tools**

    Office software and ERP software which are **easy to use**

- **OR expert**
  - Responsible for supporting decisions to be made by the management (by using mathematical approaches)
  - **Tools**

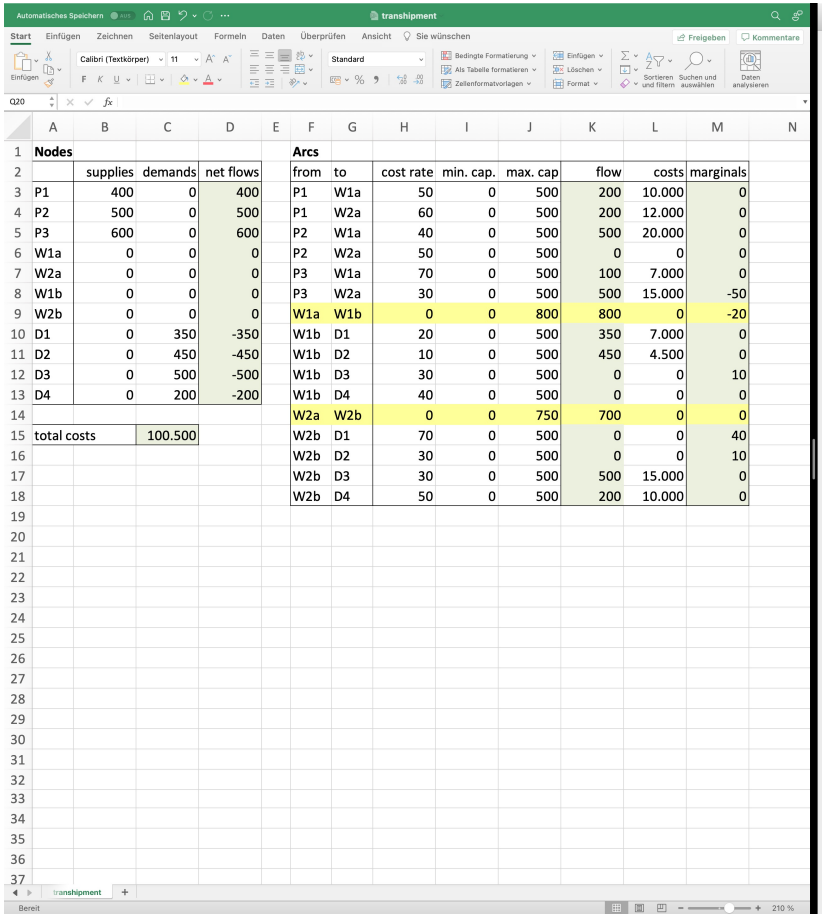
    Programming languages, mathematical programming languages and optimisation software which **requires special knowledge**.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Nodes**					**Arcs**							
2		supplies	demands	net flows		from	to	cost rate	min. cap.	max. cap	flow	costs	marginals
3	P1	400	0			P1	W1a	50	0	500		0	
4	P2	500	0			P1	W2a	60	0	500		0	
5	P3	600	0			P2	W1a	40	0	500		0	
6	W1a	0	0			P2	W2a	50	0	500		0	
7	W2a	0	0			P3	W1a	70	0	500		0	
8	W1b	0	0			P3	W2a	30	0	500		0	
9	W2b	0	0			W1a	W1b	0	0	800		0	
10	D1	0	350			W1b	D1	20	0	500		0	
11	D2	0	450			W1b	D2	10	0	500		0	
12	D3	0	500			W1b	D3	30	0	500		0	
13	D4	0	200			W1b	D4	40	0	500		0	
14						W2a	W2b	0	0	750		0	
15	total costs					W2b	D1	70	0	500		0	
16						W2b	D2	30	0	500		0	
17						W2b	D3	30	0	500		0	
18						W2b	D4	50	0	500		0	



Bringing two worlds together

- **Cmpl2 with CmplXlsData**
  - Combining Excel with Cmpl (<Coliop|Coin> Mathematical Programming Language)
  - CmplXlsData was introduced with CMPL version 2.0 and is CMPL's interface for reading sets and parameters from an Excel file and for writing optimisation results to an open Excel file.



Nodes

	supplies	demands	net flows
P1	400	0	400
P2	500	0	500
P3	600	0	600
W1a	0	0	0
W2a	0	0	0
W1b	0	0	0
W2b	0	0	0
D1	0	350	-350
D2	0	450	-450
D3	0	500	-500
D4	0	200	-200
total costs		100.500	

Arcs

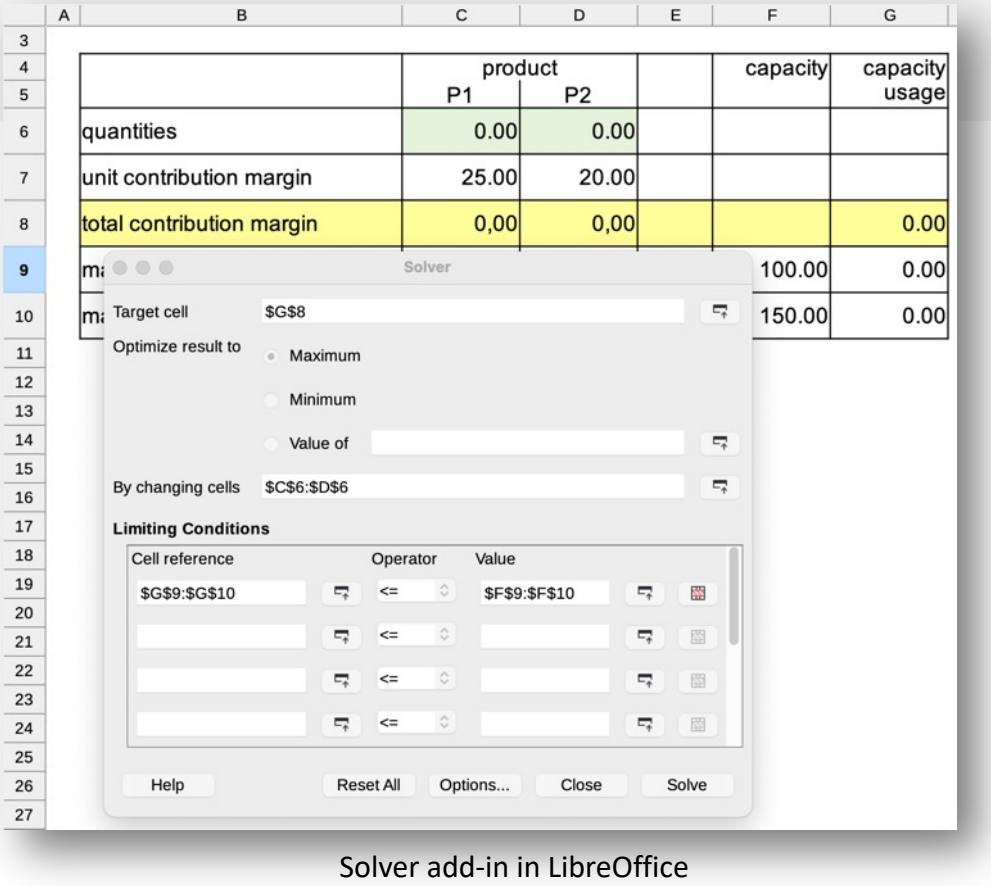
from	to	cost rate	min. cap.	max. cap	flow	costs	marginals
P1	W1a	50	0	500	200	10.000	0
P1	W2a	60	0	500	200	12.000	0
P2	W1a	40	0	500	500	20.000	0
P2	W2a	50	0	500	0	0	0
P3	W1a	70	0	500	100	7.000	0
P3	W2a	30	0	500	500	15.000	-50
W1a	W1b	0	0	800	800	0	-20
W1b	D1	20	0	500	350	7.000	0
W1b	D2	10	0	500	450	4.500	0
W1b	D3	30	0	500	0	0	10
W1b	D4	40	0	500	0	0	0
W2a	W2b	0	0	750	700	0	0
W2b	D1	70	0	500	0	0	40
W2b	D2	30	0	500	0	0	10
W2b	D3	30	0	500	500	15.000	0
W2b	D4	50	0	500	200	10.000	0

```
%xlsdata : nodes set, s[nodes], d[nodes], edges set[2], c[edges],
minCap[edges], maxCap[edges]

var:
    { [i,j] in edges: x[i,j] : real[minCap[i,j]..maxCap[i, j]];  }

obj:
    costs: sum { [i,j] in edges: c[i,j] * x[i,j] } ->min;

con:
{ i in nodes :
    netFlow[i]:   sum{ j in edges *> [i,*] : x[i,j] } -
                  sum{ j in edges *> [*,i] : x[j,i] } = s[i] - d[i];
    }
```

```
Transhipment.cmpl
Transhipment.xdat
```

Known solutions for optimisation in spreadsheets

- **Spreadsheet add-ins**
  - E.g., Excel solver [1] and its commercial version by Frontline [2]. Solver in LibreOffice/Calc [3], OpenSolver [4] [5], Frontline's add-in for Google Sheets [6], Excel add-in Evolver by Palisade [7], Lindo's Whats'sBest! [8], XLOPTIM by Addinsoft and LocalSolver [9].



Solver add-in in LibreOffice

  - Advantages
    - Data and models can be easily combined and shared.
    - Interactive work is possible.
  - Disadvantages
    - Formulation of models in the form of cell references is not suitable for complex models.
    - Debugging models is also rather complicated [10].
    - Some of the add-ins are only available for Windows (e.g., What'sBest! and Evolver) and not for macOS.

AMPL [11], MPL [12], AIMMS [13], GAMS [14], OPL [15], MOSEL [16] and SAS [17]

Known solutions for optimisation in spreadsheets

- **Algebraic modelling languages with spreadsheet interfaces (partially via VBA)**
  - E.g., AMPL [11], MPL [12], AIMMS [13], GAMS [14], OPL [15], MOSEL [16] and SAS [17]

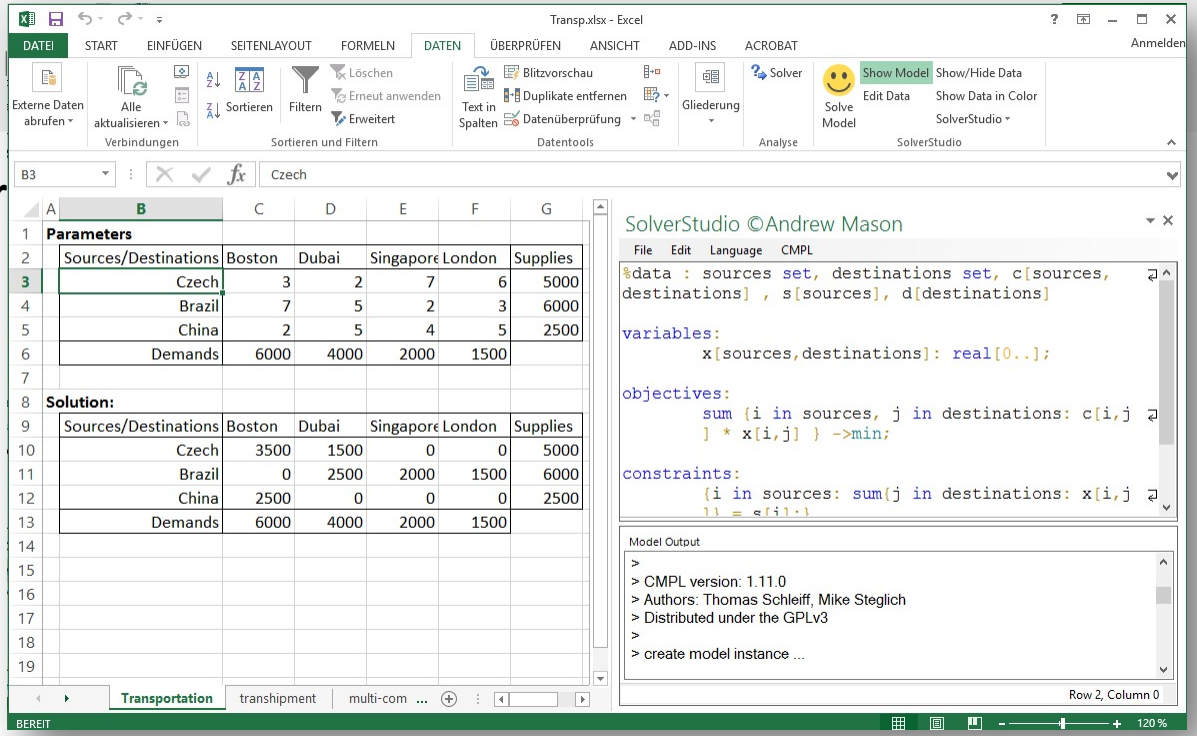
  - Advantages
    - Combination of the flexibility and capabilities of the languages with spreadsheet data
    - Use the possibilities of a spreadsheet program to further process a solution
  - Disadvantages
    - No interactive work (spreadsheet files cannot be used by other processes while they are being written and thus cannot be opened)
    - Partially complex syntax

```
*=== Import from Excel using GDX utilities

*=== First unload to GDX file (occurs during compilation phase)
$call gdxxrw.exe results.xls par=Level rng=sheet1!A1:D3

*=== Now import data from GDX
Parameter Level(i,j);
$gdxin results.gdx
$load Level
$gdxin

*=== Fix variables to values from Excel file
x.FX(i,j) = Level(i,j);
display Level, x.L;
```

GAMMS-Excel interface

Known solutions for optimisation in spr

- **Excel add-ins for algebraic modelling languages**
  - AIMMS Excel add-in [20], SolverStudio [10] [21].
  - Microsoft's Solver Foundation was an interesting offer which is evidently not being continued [22].



SolverStudio

  - Advantages
    - Combination of the flexibility and capabilities of the languages with spreadsheet data
    - Use the possibilities of a spreadsheet program to further process a solution
    - Interactivity in the process of formulating, solving and interpreting an optimisation problem,
  - Disadvantages
    - Partially complex user interface (AIMMS)
    - Only available for Microsoft Windows
    - SolverStudio does not seem to have been continued seriously.

CMPL

- **CMPL** (<Coliop|Coin> Mathematical Programming Language) is a mathematical programming language and a system for mathematical programming and optimisation of linear optimisation problems.
- CMPL executes CBC (default), CPLEX, GLPK, Gurobi and SCIP directly to solve the generated model instance.
- The CMPL syntax is similar in formulation to the original mathematical model but also includes syntactic elements from modern programming languages. CMPL is intended to combine the clarity of mathematical models with the flexibility of programming languages.[24][25]

- Cmpl is a **COIN-OR project** initiated by the Technical University of Applied Sciences Wildau.
- The CMPL distribution is **Open Source** and available for most of the relevant operating systems (Windows, macOS, Linux).

- For more details, see: **http://coliop.org** and **https://github.com/coin-or/Cmpl**.

CmplXlsData

- Introduced with CMPL version 2.0 (2021).
- CmplXlsData is CMPL's interface for reading sets and parameters from an Excel file and for writing optimisation results to an open Excel file.
- Available on Windows and macOS
- Implemented with Python3 using the (open-source) *Python for Excel library* by xlwings [28].

- Working steps
  - Maintaining the data in Excel
  - Specification of the data to be read from Excel and the solution elements to be written to Excel in a CmplXlsData file
  - Formulation of the CMPL model including the CMPL header entry `%xlsData` to link the model and CmplXlsDataFile

- A CmplXlsData file is a plain text file that contains the definition of parameters and sets with the cell addresses of their values in the specified Excel file in a particular syntax. Additionally, the optimisation results to be written to Excel with their cell addresses can be specified in this file.

CmplXlsData

- A CmplXlsData file contains the three sections @source, @input and @output.

- The @source section is intended to specify the Excel file and optionally the sheet to be used to read sets and parameters and to write the optimisation results.

`@source`	Section for specifying the Excel file and the default sheet
`%file <`*`fileName`***`>`**	Name of the Excel file
*`[`***`%sheet <`***`sheetName`***`>`***`]`*	Optional argument to specify the name of the active sheet In each entry for the inputs and the outputs, the sheet can be specified directly.

CmplXlsData

- In the @input section, the sets and parameters to be read into the CMPL model have to be specified with their cell ranges.

@input	Section for specifying sets and parameters to be read into CMPL
%*name* **<***cell***>**	A scalar parameter name is assigned a single string or number available in Excel at the specified cell.
%*name* **set[[***rank***]]** ↵ **<***cellRange***>**	Definition of an n-tuple set A set definition starts with the name followed by the keyword set. For n-tuple sets with $n > 1$ the rank of the set is to be specified enclosed by square brackets. The set is assigned the entries available in Excel in the cells specified in the cell range reference.
%*name***[***set[,set1,* ↵ *...]***]** ↵ **<***cellRange***>**	Definition of a parameter array The specification of a parameter array starts with the name followed by one or more sets, over which the array is defined. The data entries can be strings or numbers and have to be found at the specified cell range reference in Excel.

CmplXlsData

- The @output section specifies the optimisation result elements to be written to the Excel file. These results are displayed directly in the Excel file.

```
@output
```

Section for specifying the optimisation results to be written to Excel

```
%name.activity ↵
 <cell>
%name.type <cell>
%name.lowerBound ↵
 <cell>
%name.upperBound ↵
 <cell>
%name.marginal ↵
 <cell>
```

Singleton variable or constraint

For a singleton variable or constraint named `name`, the activity, type, limits and dual values can be written to Excel in the `cell`. The name is followed by a dot and one of the keywords (`activity`, `type`, `lowerbound`, `upperbound`, `marginal`) for the information to be written to Excel.

```
%name[set[,set1, ↵
...].activity ↵
  <cellRange>
%name[set[,set1, ↵
...].type ↵
  <cellRange>
%name[set[,set1, ↵
...].lowerBound ↵
<cellRange>
...
```

Arrays of variables or constraints

A complete array of variables or constraints named `name`, the activity, type, limits and dual values can be written to Excel in the specified cell range.
The specification of an array of variables or constraints starts with the name followed by one or more sets, over which the array is defined. This is followed by a dot and one of the keywords for the attributes `activity`, `type`, `lowerbound`, `upperbound`, `marginal` of the result information to be written to Excel.

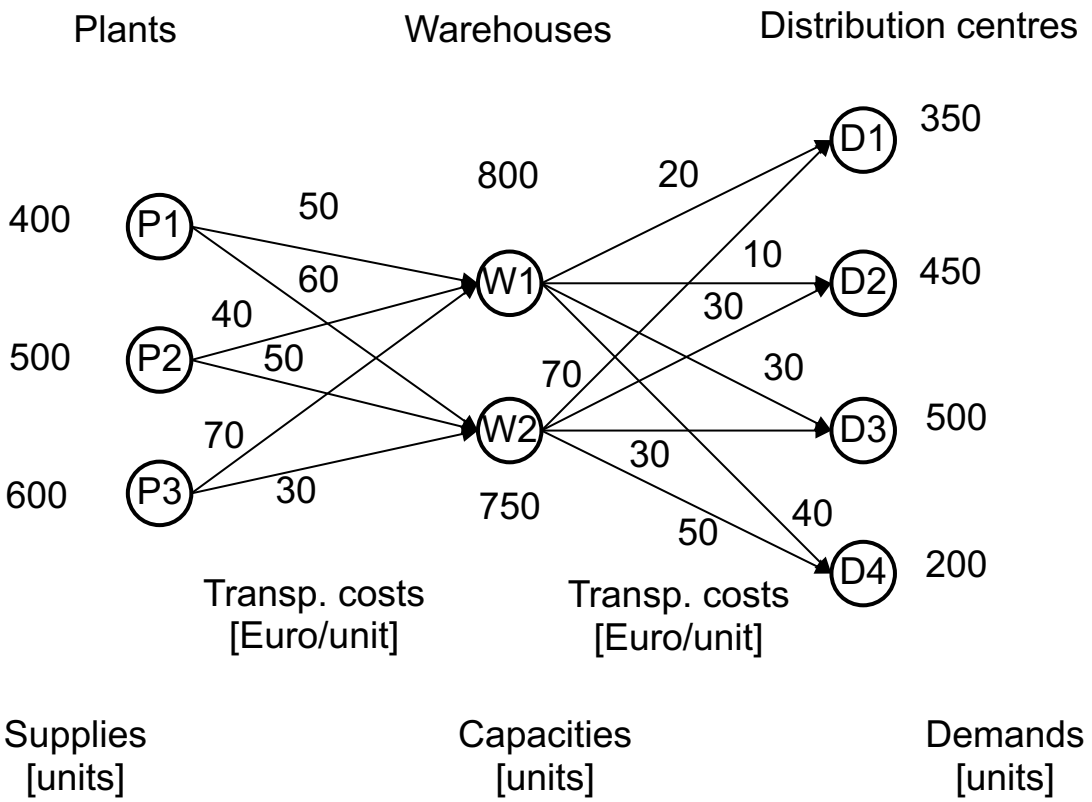
CmplXlsData

- The @output section specifies the optimisation result elements to be written to the Excel file. These results are displayed directly in the Excel file.

%objName <*cell*>	Writes the name of the objective function to Excel in the specified cell
%objSense <*cell*>	Writes the objective sense
%objValue <*cell*>	Writes the objective function value
%objStatus <*cell*>	Writes the status of the objective function
%nrOfVars <*cell*>	Writes the number of the variables
%nrOfCons <*cell*>	Writes the number of the constraints
%solverName <*cell*>	Writes the name of the solver
%solverMsg <*cell*>	Writes a message of the solver

CMPL example - Transhipment Problem

- The optimal transportation of a homogeneous good has to be planned. The objective is to minimise the total transportation costs.
- The capacity of each possible road is restricted to 500 units due to the vehicles pool.



Example - Transhipment Problem

- Transhipment problems can be formulated as linear programmes using the minimum cost flow model [29].
- Consider a network that can be described as a directed network $G = (N, A)$ where N is a set of nodes and A is a set of edges joining pairs of nodes.

$$\sum_{(i,j)\in A} c_{ij} \cdot x_{ij} \rightarrow \min!$$

$$s.t.$$

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = s_i - d_i \qquad ;i \in N$$

$$x_{ij}^l \le x_{ij} \le x_{ij}^u \qquad ;(i,j) \in A$$

Parameters:

N	Set of the nodes
A	Set of the edges
c_{ij}	Unit transportation costs (or distance, time) between node i and node j,$(i,j) \in A$
s_i	Supply of source $i \in N$
d_i	Demand of destination $i \in N$
$x^l{}_{ij}$	Lower bound of the flow between node i and node j, $x^l{}_{ij} \ge 0$
$x^u{}_{ij}$	Upper bound of the flow between node i and node j

Variables:

x_{ij}	Flow between node i and node j

Example - Transhipment Problem

- Excel file transhipment.xlsx / sheet transhipment

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Nodes**					**Arcs**							
2		supplies	demands	net flows		from	to	cost rate	min. cap.	max. cap	flow	costs	marginals
3	P1	400	0			P1	W1a	50	0	500		0	
4	P2	500	0			P1	W2a	60	0	500		0	
5	P3	600	0			P2	W1a	40	0	500		0	
6	W1a	0	0			P2	W2a	50	0	500		0	
7	W2a	0	0			P3	W1a	70	0	500		0	
8	W1b	0	0			P3	W2a	30	0	500		0	
9	W2b	0	0			W1a	W1b	0	0	800		0	
10	D1	0	350			W1b	D1	20	0	500		0	
11	D2	0	450			W1b	D2	10	0	500		0	
12	D3	0	500			W1b	D3	30	0	500		0	
13	D4	0	200			W1b	D4	40	0	500		0	
14						W2a	W2b	0	0	750		0	
15	total costs					W2b	D1	70	0	500		0	
16						W2b	D2	30	0	500		0	
17						W2b	D3	30	0	500		0	
18						W2b	D4	50	0	500		0	

Example - Transhipment Problem

CmplXlsData file transhipment.xdat

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Nodes**					**Arcs**							
2		supplies	demands	net flows		from	to	cost rate	min. cap.	max. cap	flow	costs	marginals
3	P1	400	0			P1	W1a	50	0	500		0	
4	P2	500	0			P1	W2a	60	0	500		0	
5	P3	600	0			P2	W1a	40	0	500		0	
6	W1a	0	0			P2	W2a	50	0	500		0	
7	W2a	0	0			P3	W1a	70	0	500		0	
8	W1b	0	0			P3	W2a	30	0	500		0	
9	W2b	0	0			W1a	W1b	0	0	800		0	
10	D1	0	350			W1b	D1	20	0	500		0	
11	D2	0	450			W1b	D2	10	0	500		0	
12	D3	0	500			W1b	D3	30	0	500		0	
13	D4	0	200			W1b	D4	40	0	500		0	
14						W2a	W2b	0	0	750		0	
15	total costs					W2b	D1	70	0	500		0	
16						W2b	D2	30	0	500		0	
17						W2b	D3	30	0	500		0	
18						W2b	D4	50	0	500		0	

```
@source
   %file < transhipment.xlsx >
   %sheet < transhipment>

@input
   %edges set[2] < F3:G18 >
   %nodes set < A3:A13 >
   %c[edges] < H3:H18 >
   %d[nodes] < C3:C13 >
   %s[nodes] < B3:B13 >
   %minCap[edges] < I3:I18  >
   %maxCap[edges] < J3:J18 >

@output
   %x[edges].activity < K3:K18 >
   %x[edges].marginal < M3:M18 >
   %netFlow[nodes].activity < D3:D13 >
   %objValue < C15 >
```

Example - Transhipment Problem

$$\sum_{(i,j)\in A} c_{ij} \cdot x_{ij} \rightarrow \min!$$

$$s.t.$$

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = s_i - d_i \qquad ;i \in N$$

$$x_{ij}^l \le x_{ij} \le x_{ij}^u \qquad ;(i,j) \in A$$

CMPL file transhipment.cmpl

```
%xlsdata : nodes set, s[nodes], d[nodes], edges set[2], c[edges],
minCap[edges], maxCap[edges]


var:
  { [i,j] in edges: x[i,j] : real[minCap[i,j]..maxCap[i, j]]; }

obj:
  costs: sum { [i,j] in edges: c[i,j] * x[i,j] } ->min;


con:
  { i in nodes :
        netFlow[i]:sum{ [i,j] in edges : x[i,j] } -
                   sum{ [j,i] in edges : x[j,i] } = s[i] - d[i];
  }
```

Example - Transhipment Problem

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Nodes**					**Arcs**							
2		supplies	demands	net flows		from	to	cost rate	min. cap.	max. cap	flow	costs	marginals
3	P1	400	0	400		P1	W1a	50	0	500	200	10.000	0
4	P2	500	0	500		P1	W2a	60	0	500	200	12.000	0
5	P3	600	0	600		P2	W1a	40	0	500	500	20.000	0
6	W1a	0	0	0		P2	W2a	50	0	500	0	0	0
7	W2a	0	0	0		P3	W1a	70	0	500	100	7.000	0
8	W1b	0	0	0		P3	W2a	30	0	500	500	15.000	-50
9	W2b	0	0	0		W1a	W1b	0	0	800	800	0	-20
10	D1	0	350	-350		W1b	D1	20	0	500	350	7.000	0
11	D2	0	450	-450		W1b	D2	10	0	500	450	4.500	0
12	D3	0	500	-500		W1b	D3	30	0	500	0	0	10
13	D4	0	200	-200		W1b	D4	40	0	500	0	0	0
14						W2a	W2b	0	0	750	700	0	0
15	total costs		100.500			W2b	D1	70	0	500	0	0	40
16						W2b	D2	30	0	500	0	0	10
17						W2b	D3	30	0	500	500	15.000	0
18						W2b	D4	50	0	500	200	10.000	0

Summary

- It is reasonable to connect spreadsheet programs with optimisation software to combine the modelling capabilities of optimisation software with the data maintained in spreadsheets.
- Advantages and disadvantages of known solutions:
  - Add-ins in spreadsheet programs such as the Excel solver add-in allow interactive work, although modelling with cell ranges does not seem suitable for complex models.
  - Data interfaces to spreadsheets of algebraic modelling languages, which are excellent for modelling complex problems, do not allow interactive work.
  - Approaches that combine modelling languages with Excel in the form of an Excel add-in and thus combine interactive work with excellent modelling possibilities. Unfortunately, these are only available for Windows and some of them seem to have been discontinued.
- This facts led to the motivation to create CmplXlsData, which is CMPL's interface to Excel. It is an easy-to-use interface between this modelling language and Excel, which allows interactive work and is available for Windows and macOS.

Literature

[1] Microsoft, “Define and solve a problem by using Solver,” 2021. [Online]. Available: https://support.microsoft.com/en-us/office/define-and-solve-a-problem-by-using-solver-5d1a388f-079d-43ac-a7eb-f63e45925040. [retrieved: July 2021].

[2] Frontline, “Excel Solver – Overview and Example” 2021. [Online]. Available: https://www. solver.com/excel-solver-overview-and-example. [retrieved: July 2021].

[3] LibreOffice, “Solver,” 2021. [Online]. Available: https://help.libreoffice.org/7.0/en-US/text/scalc/01/solver.html?DbPAR=CALC. [retrieved: July 2021].

[4] OpenSolver, “About OpenSolver,” 2021. [Online]. Available: https://opensolver.org. [retrieved: July 2021].

[5] A. J. Mason, “OpenSolver – An Open Source Add-in to Solve Linear and Integer Progammes in Excel,” in *Operations Research Proceedings 2011*, Berlin and Heidelberg, pp. 401-406, 2012.

[6] Frontline, “Solver - Add-on for Google Sheets,” 2021. [Online]. Available: https://workspace.google.com/marketplace/app/solver/539454054595. [retrieved: July 2021].

[7] Palisade, “Evolver - Innovative Optimization for Spreadsheets,” 2021. [Online]. Available: https://www. palisade.com/evolver/default.asp. [retrieved: July 2021].

[8] L. S. Inc., “ What'sBest! 17.0 - Excel Add-In for Linear, Nonlinear, and Integer Modeling and Optimization,” 2021. [Online]. Available: https://www.lindo.com/index.php/ products/what-sbest-and-excel-optimization. [retrieved: July 2021].

[9] Addinsoft, “The leading Optimization Solver for Microsoft Excel®,” 2021. [Online]. Available: https://www.xloptim.com/en. [retrieved: July 2021].

[10] A. J. Mason, “SolverStudio: A New Tool for Better Optimisation and Simulation Modelling in Excel,” *INFORMS Transactions on Education,* vol. 14, no. 1, pp. 45-52 , 2013.

[11] AMPL, “AMPL Direct Spreadsheet Interface,” 2021. [Online]. Available: https://ampl.com/resources/new-features/spreadsheets/. [retrieved: July 2021].

[12] M. Software, “MPL for Windows Manual - Import Data from Excel Spreadsheet,” 2021. [Online]. Available: http://www. maximalsoftware.com/mplman/mpw07060.html. [retrieved: July 2021].

Literature

[13] A. B.V., “AIMMS Excel Library - AXLL,” 2021. [Online]. Available: https://how-to.aimms.com/Articles/85/85-using-axll-library.html. [retrieved: July 2021].

[14] GAMS, “Data Exchange with Microsoft Excel,” 2021. [Online]. Available: https://www.gams.com/latest/docs/UG_DataExchange_Excel.html. [retrieved: July 2021].

[15] IBM, “ILOG CPLEX Optimization Studio/ 20.1.0 / Spreadsheet Input/Output,” 2021. [Online]. Available: https://www.ibm.com/docs/en/icos/20.1.0?topic=sources-spreadsheet-inputoutput. [retrieved: July 2021].

[16] FICO, “The Excel interface,” 2021. [Online]. Available: https://www.fico.com/fico-xpress-optimization/docs/latest/mosel/mosel_data/dhtml/secsetup_sec_secexcelsetup.html. [retrieved: July 2021].

[17] V. DelGobbo, “Integrating SAS® and Microsoft Excel: Exploring the Many Options Available to You,” SAS Institute Inc., https://www.sas.com/content/dam/ SAS/support/en/sas-global-forum-proceedings/2019/2991-2019.pdf, 2019 [retrieved: July 2021].

[18] M. Software, “OptiMax Component Library,” 2021. [Online]. Available: http://www.maximalsoftware.com/optimax/. [retrieved: July 2021].

[19] FICO, “Launching Mosel from Excel using VBAFICO Xpress Optimization Examples Repository /,” 2021. [Online]. Available: https://examples.xpress.fico.com/ example.pl?id=excelmosel1. [retrieved: July 2021].

[20] AIMMS B.V., “AIMMS - The Excel Add-In User’s Guide,” https://download.aimms.com/aimms/download/references/AIMMS_excel.pdf, 2016. [retrieved: July 2021]

[21] A. J. Mason, “SolverStudio,” 2021. [Online]. Available: https://solverstudio.org. [retrieved: July 2021].

[22] Microsoft, “Solver Foundation,” 2014. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/msdn10/hh145003(v=msdn.10). [retrieved: July 2021].

[23] .NET Foundation, “IronPython - the Python programming language for .NET,” 2021. [Online]. Available: https://ironpython.net. [retrieved: July 2021]

[24] M. Steglich and T. Schleiff, “CMPL,” 2021. [Online]. Available: http://coliop.org. [retrieved: July 2021]

[25] M. Steglich and T. Schleiff, “CMPL: Coliop Mathematical Programming Language,” *Wildauer Schriftenreihe - Entscheidungsunterstützung und Operations Research,* vol. 1, 2010.

Literature

[26] COIN-OR, 2021. [Online]. Available: https://www.coin-or.org. [retrieved: Sep 2021]

[27] M. Steglich, “CMPLServer - An open source approach for distributed and grid optimisation,” *AKWI Anwendungen und Konzepte der Wirtschaftsinformatik,* no. 4, pp. 9-21, 2016.

[28] xlwings, “xlwings - Python for Excel,” Zoomer Analytics GmbH, 2021. [Online]. Available: https://www.xlwings.org. [retrieved: July 2021]

[29] F. Hillier and G. Lieberman, Introduction to Operations Research (International edition), vol. 10th ed., New York et al.: McGraw-Hill, 2015.

[30] M. Steglich, D. Feige, and P. Klaus, Logistik-Entscheidungen: Modellbasierte Entscheidungsunterstützung in der Logistik mit LogisticsLab, Berlin and Boston: De Gruyter, 2016.

Thank you for your attention.

If you have any questions, please feel free to ask.