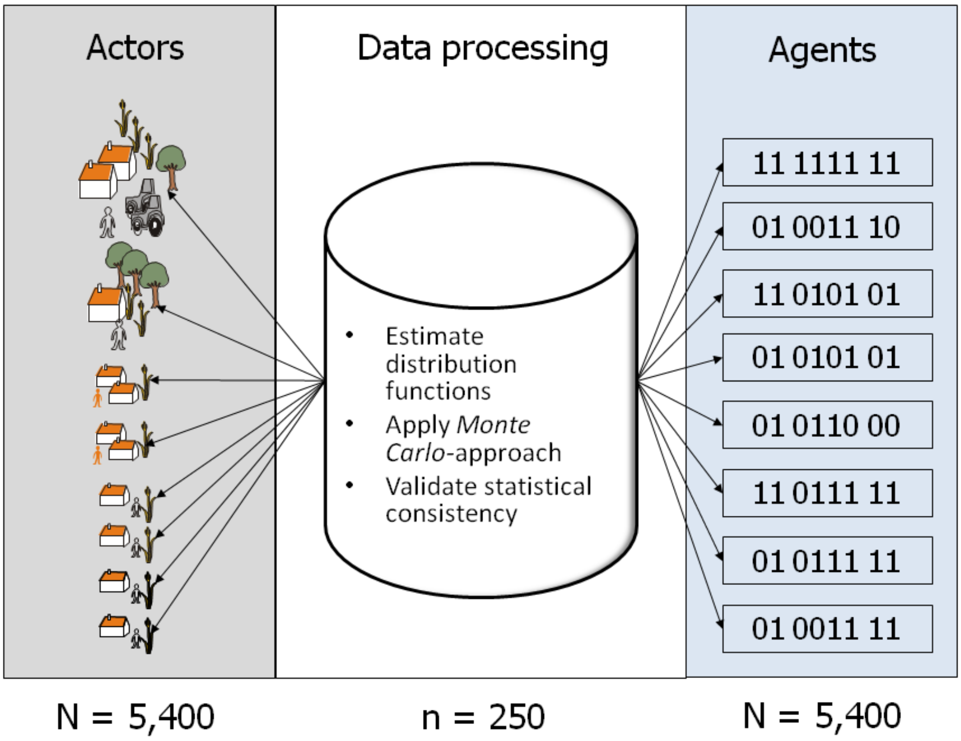
Mathematical Programming-based Multi-Agent Systems (MPMAS)

Tutorial to Building MPMAS Applications



Version 2.0 of January 17, 2018

Thomas Berger, Evgeny Latynskiy, William McClain, Teresa Walter,
Christian Troost, Sara Yazdan Bakhsh

mas@uni-hohenheim.de

Table of Contents

1 Introduction

This tutorial requires installation of the MPMAS Add-in either in MS-Excel or in LibreOffice-Ubuntu. Before you start, please download the demo version of MPMAS and make yourself familiar with the basic structure of MPMAS and the functions of its Add-in. Only then it is possible to begin working with the tutorial model.

2 Exercises for extending the model

The following exercises will each extend the model in various aspects, each time starting again with the basic model set-up:

1. Add more periods
2. Add new annual crop
3. Add more investments
4. Add symbolic objects
5. Add more soil types
6. Add more agents
7. Assign assets randomly

Please note:

- The MPMAS executable in this tutorial package is a fully functional version but is limited to 100 agents only. If you plan to use MPMAS for "serious" research, please contact mas@uni-hohenheim.de for more details. Applying MPMAS for Msc and PhD projects, however, requires advanced knowledge of farm-level linear programming as taught in Hazell and Norton (1986)[1]. Without this knowledge of LP, it will be impossible for you to develop your own empirical MPMAS application.
- Please start the exercises always from the demo version (=basic model) and try to implement them. If you look at the folder *xlsInput* (or *odsInput* in case you are working with LibreOffice-Ubuntu), you will see the MPMAS files that you are supposed to extend. With *Scenariomanager.xls*[2] you can then create your input, run your simulations and analyze the results. It is recommendable to copy the complete demo version folder and rename it accordingly (for example, *TUT_0_basic*, *TUT_1_ex*, *TUT_2_ex* etc.). The tutorial zip file contains the correct inputs and output files (=solutions) for the various exercises.

[1] Especially chapter 4 in http://ebrary.ifpri.org/cdm/ref/collection/p15738coll5/id/2319

[2] For simplicity, we write here ".xls" but refer to both .xls in Mpmas_Excel and .ods in Mpmas_LibreOffice.

Hints and tips:

- Whenever you insert a new activity in the matrix file, be careful with the lower and upper bounds!
- All cells where the value is equal to zero must explicitly state so and not be left empty. The MPMAS code ignores all empty cells, so if a zero is not included when needed, MPMAS will give you an error message. This is especially important when you enter new rows and columns!
- Whenever you enter new columns and activities in the matrix also remember to update the corresponding headers in *XResults.xls* (copy and paste from the Matrix file)!

2.1 Add more periods

In this first exercise, we will take the demo version model and add two more periods to be simulated, resulting in five total simulation periods. We will not change the prices from the last period modeled (Period 3 in the input files). The purpose here is to extend the time-horizon of the model and become familiar with adding more periods. Start with the files contained in the demo version folder *xlsinput*. You will find the solutions in the folder */TUT*_periods*.

Files to modify: Market.xls; Region.xls; ScenarioManager.xls

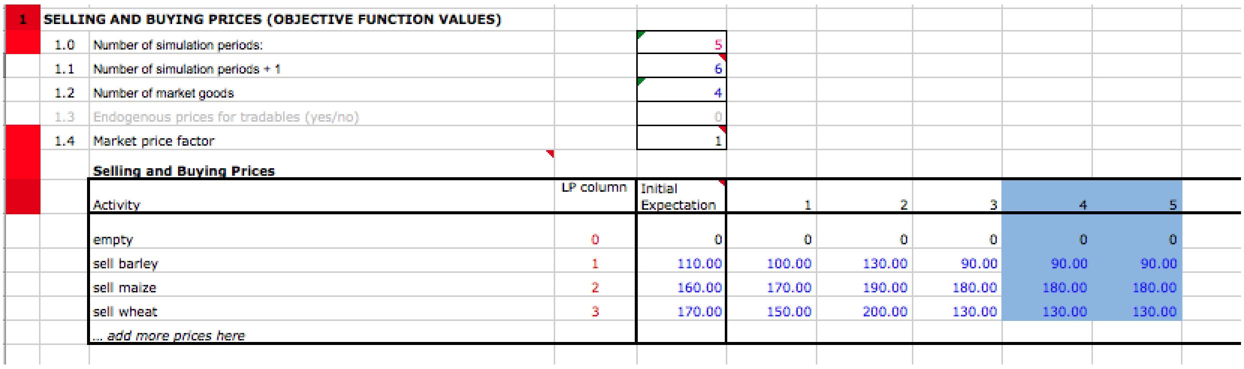
Market.xls: ***Worksheet: Market:*** First we have to indicate in the market file that new periods will be simulated and to define the prices agents earn in those periods.

- In the table "*Selling and Buying Prices*," add the number of periods on the upper right side. Then enter the prices that correspond to the new periods. Remember to take into account any formulas, e.g. if prices are linked with "*1.4 Market price factor*".

NOTE: In the basic tutorial model, all market prices are exogenous and need to be defined for each period in the market file.

- Check that the cells "*1.0 Number of simulation periods*" and "*1.1 Number of simulation periods +1*" have been updated (they should read "5" and "6," respectively).

- Save the changes



1	SELLING AND BUYING PRICES (OBJECTIVE FUNCTION VALUES)		
1.0	Number of simulation periods:		5
1.1	Number of simulation periods + 1		6
1.2	Number of market goods		4
1.3	Endogenous prices for tradables (yes/no)		0
1.4	Market price factor		1

Selling and Buying Prices

Activity	LP column	Initial Expectation	1	2	3	4	5
empty	0	0	0	0	0	0	0
sell barley	1	110.00	100.00	130.00	90.00	90.00	90.00
sell maize	2	160.00	170.00	190.00	180.00	180.00	180.00
sell wheat	3	170.00	150.00	200.00	130.00	130.00	130.00
... add more prices here							

ScenarioManager.xls: *Worksheet: Manager*: We want to make sure that our changes to simulation periods have been reflected in *ScenarioManager.xls*. If the simulation periods do not match here, MPMAS will not actually run for the correct number of periods.

- In the table at the bottom, make sure that the parameter "*Number of simulation years*" matches the new number of simulation periods
- Save the changes made in ScenarioManager
- Select "*Set path names*" in the MPMAS toolbar and make sure that current and main directory are correctly updated
- Clean any pervious input files with "*Delete all files*" in the MPMAS toolbar, select "*Create input files*", and finally "*Run MPMAS*"
- Usually, you will shortly see a flickering screen with MPMAS running in the background. If you have a fast computer, however, you will not be able to read any possible error message from MPMAS. Therefore, locate the file *BSL_.err* in the main directory and open it with a text editor (check the file's date of creation so you are sure you are reading the correct log file). In case you find the message "*Simulation experiment[BSL_ ./] completed*" all went well and you are ready to analyze your outputs with *XResults.xls*. Otherwise you will get some hints about errors in your input files and how to fix them.

xResults.xls:

- Open with macros enabled and check if the correct number of periods has been entered
- Start importing the "k" files (RHS and LHS values of all agents), "u" files (solution vectors and objective functions) and "p" files (performance data) by typing into the respective cell. Select "*XResults*" in the MPMAS toolbar and repeat for all file types.
- Each time, you should notice the Add-In is working, but again this can be for less than a second in case you have a fast computer.
- Insert table or pivot table so that you can filter and inspect the results for individual agents.

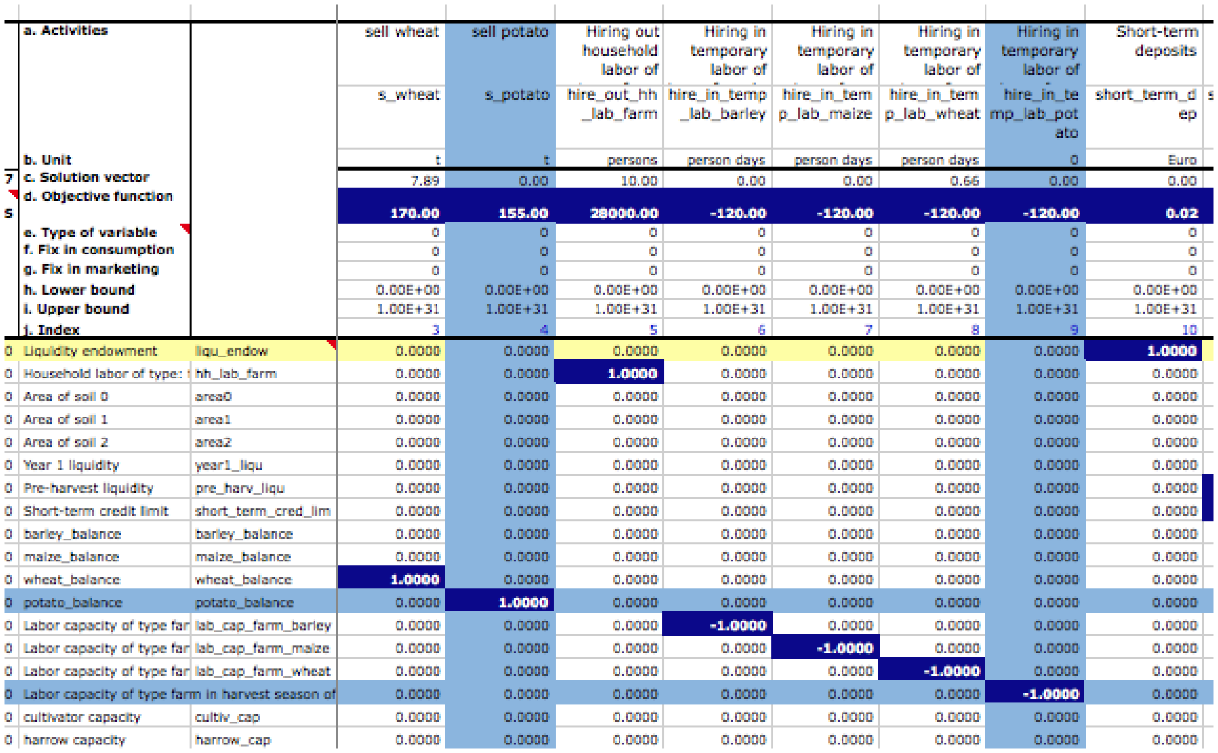
2.2 Add new annual crop

In this exercise we will take the basic model and add one additional crop (potatoes) and all of its resulting activities. This means we must define all the activities (selling potato, hiring labor for potato harvest, and growing potato on all three soil types). We must also define the prices the farm agents can receive for selling potatoes. Start with the demo version files contained in the folder *xlsinput*. You will find the solutions in the folder *TUT*_annual_crop*.

Files to modify: Matrix.xls; Market.xls

Matrix.xls: In this file we will add all of the activities and constraints associated with adding a new crop to the agent decision matrix. In this, we want to define the different activities (selling, hiring labor and growing), as well as the labor and supply/demand constraints of the potato growing activity. Be sure to pay close attention to the instructions, as any mistakes in the LP matrix file may result in errors when MPMAS runs its simulation.

- Insert a new column for the selling activity (e.g. "*Sell potato*"). Put the initial expectation for the price of potatoes in the row "*d. Objective function*" (in this example it is 155€, this value can often be found in the row "*Initial Expectation*" in the file *Market.xls*). (Remember: MPMAS receives prices for each simulation period from the *Market.xls* file. Here we are including the initial expectation for the price. Below, we will define the prices for each selling period).
- Define a cell name for the column/selling activity index (e.g. "*iact_sell_potato*"). As discussed above, using the 'Name Manager' feature of Excel allows for more stable cross-references to cells.
- Insert another column for the activity of "*Hiring in temporary labor of type: farm in harvest seasons of potato*".
- Finally insert three columns to 'produce' potato on each available soil type (production costs amount to 1272€ in this example).
- Take care that in the upper part of the matrix file, under "*1.2 Other information about the matrix,"* the column "*Number nonzero activities*" changes correspondingly. (You must change the reference to the index name of your inserted column so that it will be counted as a nonzero activity)
- Insert one row for "*Potato balance,*" and another for "*Labor capacity of type farm in harvest season of potato*".
- Write the technical coefficients into the corresponding cells, especially to relate land, labor, machinery, etc. demand with their availability, as well as the production activity with the selling activity.
- Compare the blue shaded parts of the Matrix file:

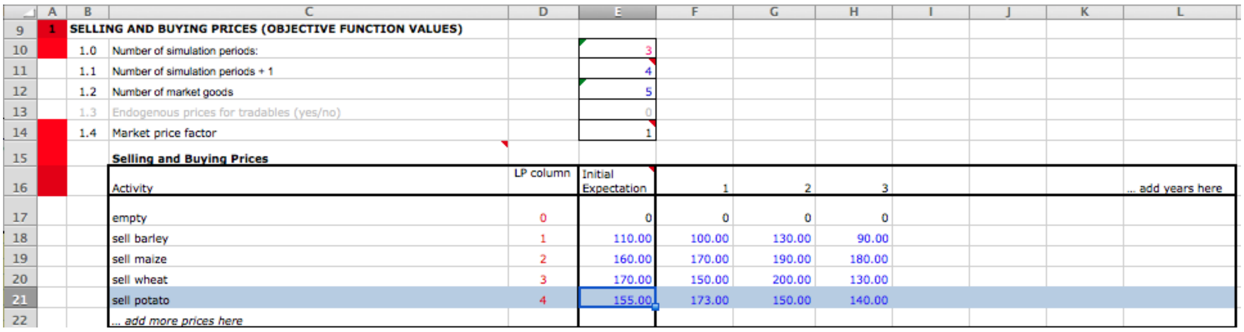


	a. Activities		sell wheat	sell potato	Hiring out household labor of	Hiring in temporary labor of	Hiring in temporary labor of	Hiring in temporary labor of	Hiring in temporary labor of	Short-term deposits
			s_wheat	s_potato	hire_out_hh_lab_farm	hire_in_temp_lab_barley	hire_in_temp_lab_maize	hire_in_temp_lab_wheat	hire_in_temp_lab_potato	short_term_dep
	b. Unit		t	t	persons	person days	person days	person days	0	Euro
7	c. Solution vector		7.89	0.00	10.00	0.00	0.00	0.66	0.00	0.00
5	d. Objective function		170.00	155.00	28000.00	-120.00	-120.00	-120.00	-120.00	0.02
	e. Type of variable		0	0	0	0	0	0	0	0
	f. Fix in consumption		0	0	0	0	0	0	0	0
	g. Fix in marketing		0	0	0	0	0	0	0	0
	h. Lower bound		0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	i. Upper bound		1.00E+31	1.00E+31	1.00E+31	1.00E+31	1.00E+31	1.00E+31	1.00E+31	1.00E+31
	j. Index		3	4	5	6	7	8	9	10
0	Liquidity endowment	liqu_endow	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
0	Household labor of type: f	hh_lab_farm	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0	Area of soil 0	area0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0	Area of soil 1	area1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0	Area of soil 2	area2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0	Year 1 liquidity	year1_liqu	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0	Pre-harvest liquidity	pre_harv_liqu	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0	Short-term credit limit	short_term_cred_lim	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0	barley_balance	barley_balance	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0	maize_balance	maize_balance	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0	wheat_balance	wheat_balance	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0	potato_balance	potato_balance	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0	Labor capacity of type far	lab_cap_farm_barley	0.0000	0.0000	0.0000	-1.0000	0.0000	0.0000	0.0000	0.0000
0	Labor capacity of type far	lab_cap_farm_maize	0.0000	0.0000	0.0000	0.0000	-1.0000	0.0000	0.0000	0.0000
0	Labor capacity of type far	lab_cap_farm_wheat	0.0000	0.0000	0.0000	0.0000	0.0000	-1.0000	0.0000	0.0000
0	Labor capacity of type farm in harvest season of		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-1.0000	0.0000
0	cultivator capacity	cultiv_cap	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0	harrow capacity	harrow_cap	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Market.xls: In this file we want to make sure that the new activity of selling potatoes and their respective selling prices are updated. Remember, since we returned to the base model our simulation periods should be back to three!

- In section "*SELLING AND BUYING PRICES*" the new selling activity has to be introduced (e.g. sell potato). Enter the initial expectation and product selling price for each of the considered periods. Make sure that you relate the entry to the Market price factor (see the other prices).

- Relate the corresponding cell in LP column to the correct matrix index, e.g. “*=Matrix.xls!iact_sell_potato*” (see how useful it is to define names for our Excel cells?)



	A	B	C	D	E	F	G	H	I	J	K	L
9	1	SELLING AND BUYING PRICES (OBJECTIVE FUNCTION VALUES)										
10		1.0	Number of simulation periods:		3							
11		1.1	Number of simulation periods + 1		4							
12		1.2	Number of market goods		5							
13		1.3	Endogenous prices for tradables (yes/no)		0							
14		1.4	Market price factor		1							
15			Selling and Buying Prices									
16			Activity	LP column	Initial Expectation	1	2	3				... add years here
17			empty	0	0	0	0	0				
18			sell barley	1	110.00	100.00	130.00	90.00				
19			sell maize	2	160.00	170.00	190.00	180.00				
20			sell wheat	3	170.00	150.00	200.00	130.00				
21			sell potato	4	155.00	173.00	150.00	140.00				
22			... add more prices here									

Run the simulation

2.3 Add more investments

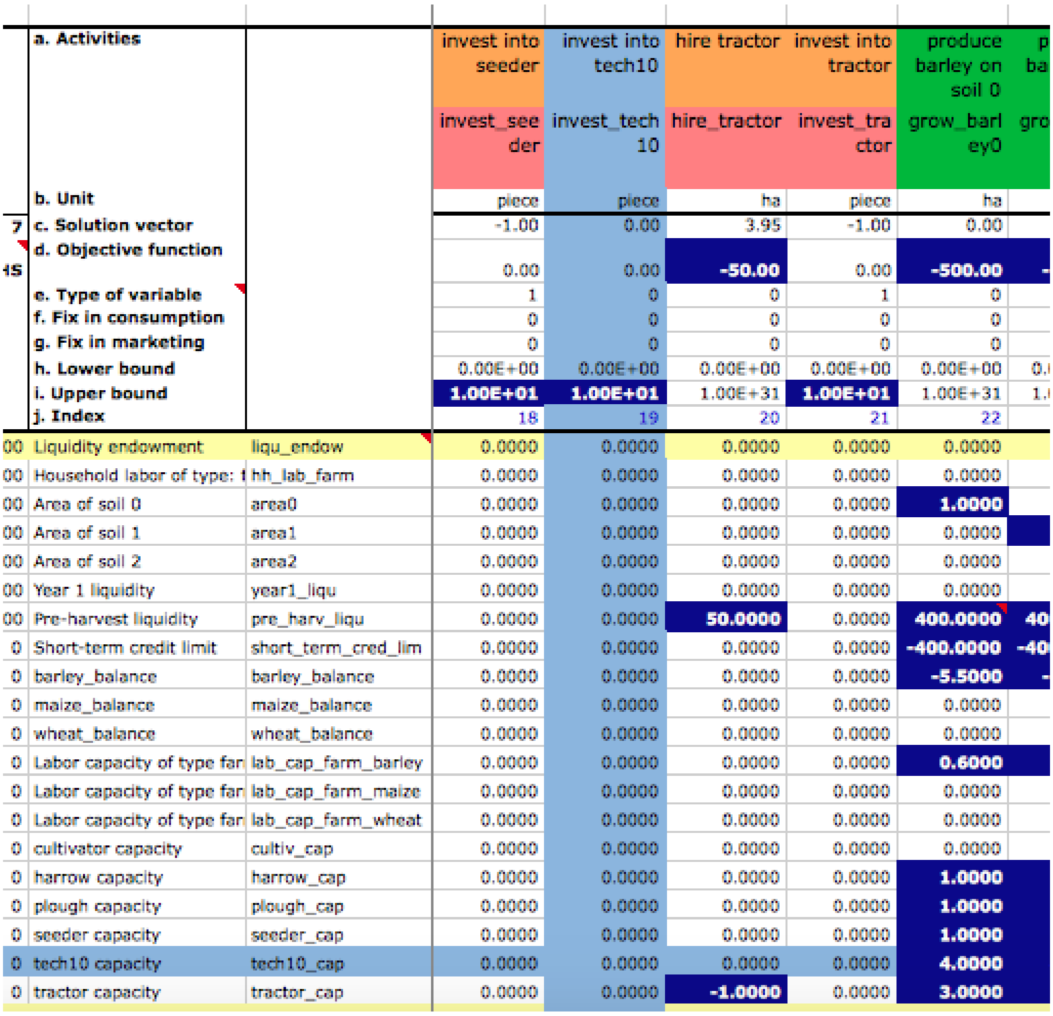
In our original model, our agents had the option to invest in different technologies (e.g. cultivator, harrow, plough, seeder, or tractor). Now we are going to add one more investment activity for a cereal dryer that is needed during harvest campaigns for winter cereal crops (weather conditions are otherwise too risky in barley and wheat). The dryer called “Tech10” costs 8000€, has a lifetime of 8 years and is capable of working 400 ha per year. Agent 300 will be the only farm household who has this machine available in the beginning of the first period. All the other features are considered the same as the other investments. Start with the model contained in *./TUT**. The solutions can be found in *./TUT*_investment.*

NOTE: Before inserting one investment type in any of the following files, be sure that they are inserted in the same order in relation to the other investment types. In this example, we insert it before “Tractor” and after “Seeder”. It's important that all the input from our different files line up, so make sure to include it always in the correct order.

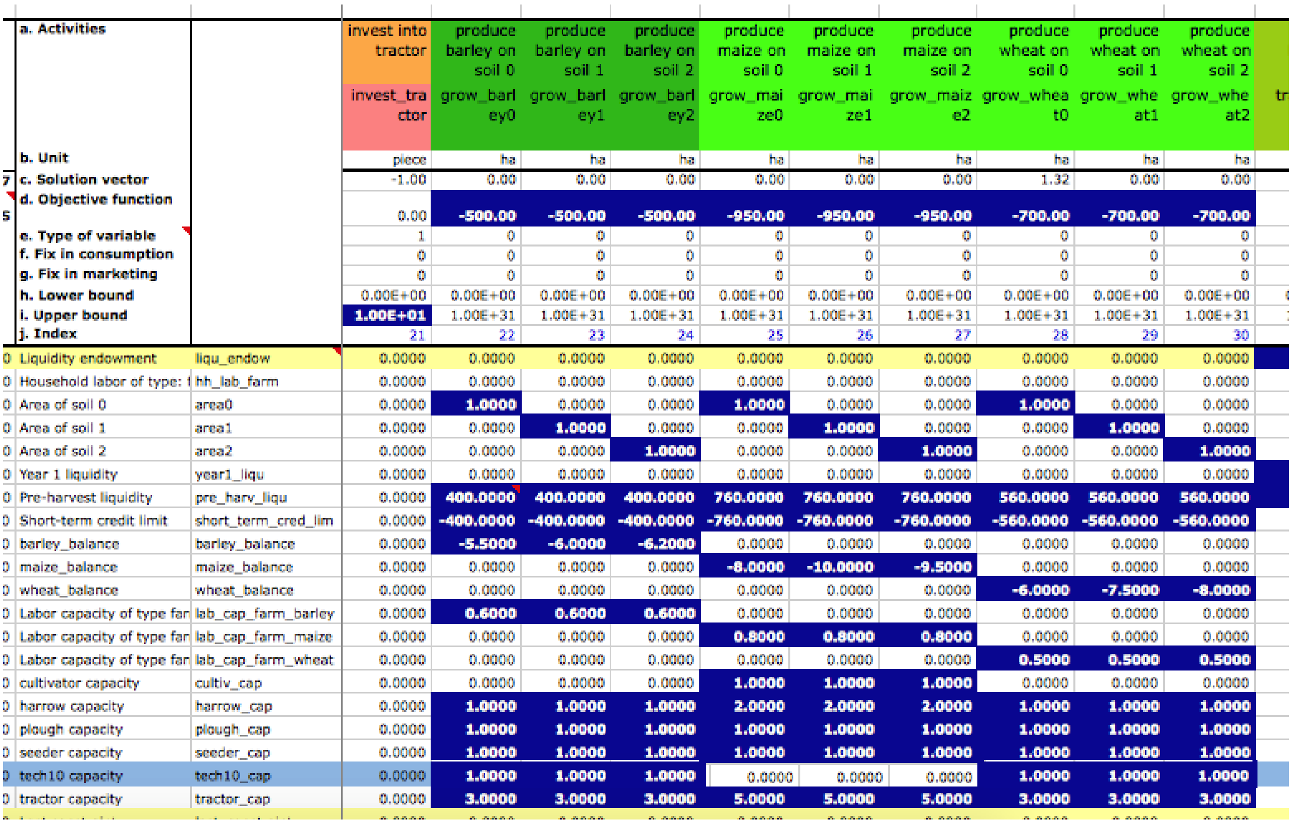
Files to modify: Matrix.xls; Network.xls; Population.xls

Matrix.xls: We first want to add the investment activity to the agent decision matrix, including information about how the technology is used to produce crops.

- Insert one new column/activity with the activity name “*Invest into tech10*”
- Be sure to give the index cell a name using Excel's 'Name Manager' (e.g. “a_tech10_invest”)
- Be sure that the values given in the rows “*c. Solution vector*”, “*d. Objective function*”, “*e. Type of variable*”, “*f. Fix in consumption*”, “*g. Fix in market*”, “*h. Lower bounds*” and “*j. Upper bounds*” are the same as the other investment activities.
- Insert one new row/constraint. Give the index cell a name (e.g. “c_tech10”). Be sure that the cells for “*1 Type*,” “*2 Range*,” “*3 LHS*,” “*4 Sign*” and “*5 RHS*” have the same values and formulas as the other constraint cells.
- Insert the capacity values for the activities for drying barley and wheat after harvest (in this example you must insert “1” for each soil type). You do not have to write coefficients in your investment activity, as this will be done by MPMAS during run-time.

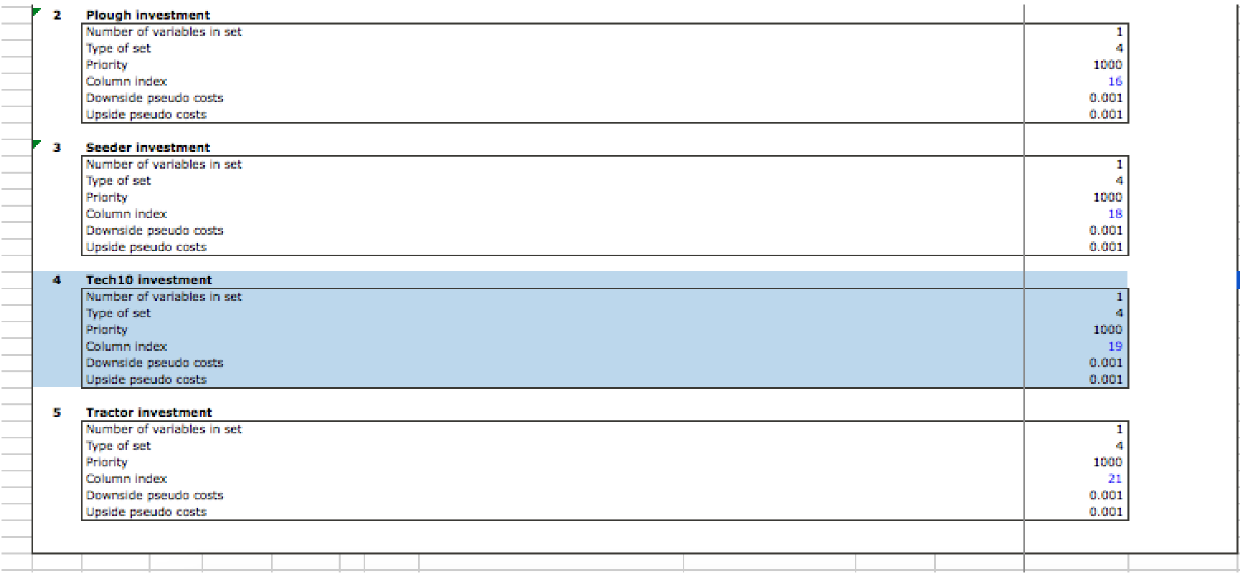


	a. Activities		invest into seeder invest_seeder	invest into tech10 invest_tech10	hire tractor hire_tractor	invest into tractor invest_tractor	produce barley on soil 0 grow_barley0	p ba gro
	b. Unit		piece	piece	ha	piece	ha	
7	c. Solution vector		-1.00	0.00	3.95	-1.00	0.00	
45	d. Objective function		0.00	0.00	-50.00	0.00	-500.00	-
	e. Type of variable		1	0	0	1	0	
	f. Fix in consumption		0	0	0	0	0	
	g. Fix in marketing		0	0	0	0	0	
	h. Lower bound		0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.(
	i. Upper bound		1.00E+01	1.00E+01	1.00E+31	1.00E+01	1.00E+31	1.(
	j. Index		18	19	20	21	22	
00	Liquidity endowment	liqu_endow	0.0000	0.0000	0.0000	0.0000	0.0000	
00	Household labor of type: f	hh_lab_farm	0.0000	0.0000	0.0000	0.0000	0.0000	
00	Area of soil 0	area0	0.0000	0.0000	0.0000	0.0000	1.0000	
00	Area of soil 1	area1	0.0000	0.0000	0.0000	0.0000	0.0000	
00	Area of soil 2	area2	0.0000	0.0000	0.0000	0.0000	0.0000	
00	Year 1 liquidity	year1_liqu	0.0000	0.0000	0.0000	0.0000	0.0000	
00	Pre-harvest liquidity	pre_harv_liqu	0.0000	0.0000	50.0000	0.0000	400.0000	40
0	Short-term credit limit	short_term_cred_lim	0.0000	0.0000	0.0000	0.0000	-400.0000	-40
0	barley_balance	barley_balance	0.0000	0.0000	0.0000	0.0000	-5.5000	-
0	maize_balance	maize_balance	0.0000	0.0000	0.0000	0.0000	0.0000	
0	wheat_balance	wheat_balance	0.0000	0.0000	0.0000	0.0000	0.0000	
0	Labor capacity of type far	lab_cap_farm_barley	0.0000	0.0000	0.0000	0.0000	0.6000	
0	Labor capacity of type far	lab_cap_farm_maize	0.0000	0.0000	0.0000	0.0000	0.0000	
0	Labor capacity of type far	lab_cap_farm_wheat	0.0000	0.0000	0.0000	0.0000	0.0000	
0	cultivator capacity	cultiv_cap	0.0000	0.0000	0.0000	0.0000	0.0000	
0	harrow capacity	harrow_cap	0.0000	0.0000	0.0000	0.0000	1.0000	
0	plough capacity	plough_cap	0.0000	0.0000	0.0000	0.0000	1.0000	
0	seeder capacity	seeder_cap	0.0000	0.0000	0.0000	0.0000	1.0000	
0	tech10 capacity	tech10_cap	0.0000	0.0000	0.0000	0.0000	4.0000	
0	tractor capacity	tractor_cap	0.0000	0.0000	-1.0000	0.0000	3.0000	



	a. Activities		invest into tractor invest_tractor	produce barley on soil 0 grow_barley0	produce barley on soil 1 grow_barley1	produce barley on soil 2 grow_barley2	produce maize on soil 0 grow_maize0	produce maize on soil 1 grow_maize1	produce maize on soil 2 grow_maize2	produce wheat on soil 0 grow_wheat0	produce wheat on soil 1 grow_wheat1	produce wheat on soil 2 grow_wheat2	tr
	b. Unit		piece	ha	ha	ha	ha	ha	ha	ha	ha	ha	
7	c. Solution vector		-1.00	0.00	0.00	0.00	0.00	0.00	0.00	1.32	0.00	0.00	
5	d. Objective function		0.00	-500.00	-500.00	-500.00	-950.00	-950.00	-950.00	-700.00	-700.00	-700.00	
	e. Type of variable		1	0	0	0	0	0	0	0	0	0	
	f. Fix in consumption		0	0	0	0	0	0	0	0	0	0	
	g. Fix in marketing		0	0	0	0	0	0	0	0	0	0	
	h. Lower bound		0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	(
	i. Upper bound		1.00E+01	1.00E+31	1.00E+31	1.00E+31	1.00E+31	1.00E+31	1.00E+31	1.00E+31	1.00E+31	1.00E+31	:
	j. Index		21	22	23	24	25	26	27	28	29	30	
0	Liquidity endowment	liqu_endow	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0	Household labor of type: f	hh_lab_farm	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0	Area of soil 0	area0	0.0000	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000	
0	Area of soil 1	area1	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000	0.0000	
0	Area of soil 2	area2	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000	
0	Year 1 liquidity	year1_liqu	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0	Pre-harvest liquidity	pre_harv_liqu	0.0000	400.0000	400.0000	400.0000	760.0000	760.0000	760.0000	560.0000	560.0000	560.0000	
0	Short-term credit limit	short_term_cred_lim	0.0000	-400.0000	-400.0000	-400.0000	-760.0000	-760.0000	-760.0000	-560.0000	-560.0000	-560.0000	
0	barley_balance	barley_balance	0.0000	-5.5000	-6.0000	-6.2000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0	maize_balance	maize_balance	0.0000	0.0000	0.0000	0.0000	-8.0000	-10.0000	-9.5000	0.0000	0.0000	0.0000	
0	wheat_balance	wheat_balance	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-6.0000	-7.5000	-8.0000	
0	Labor capacity of type far	lab_cap_farm_barley	0.0000	0.6000	0.6000	0.6000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0	Labor capacity of type far	lab_cap_farm_maize	0.0000	0.0000	0.0000	0.0000	0.8000	0.8000	0.8000	0.0000	0.0000	0.0000	
0	Labor capacity of type far	lab_cap_farm_wheat	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.5000	0.5000	0.5000	
0	cultivator capacity	cultiv_cap	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000	1.0000	0.0000	0.0000	0.0000	
0	harrow capacity	harrow_cap	0.0000	1.0000	1.0000	1.0000	2.0000	2.0000	2.0000	1.0000	1.0000	1.0000	
0	plough capacity	plough_cap	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	
0	seeder capacity	seeder_cap	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	
0	tech10 capacity	tech10_cap	0.0000	1.0000	1.0000	1.0000	0.0000	0.0000	0.0000	1.0000	1.0000	1.0000	
0	tractor capacity	tractor_cap	0.0000	3.0000	3.0000	3.0000	5.0000	5.0000	5.0000	3.0000	3.0000	3.0000	

- Add one investment in Table 14 “*INTEGERS SETS/SOS*,” below the matrix. You can copy the information from the other investments.
- Be sure that you give each investment the correct number: in this case Tech10 is “4” and Tractor is “5.” Link the “*Column index*” cell to the “*Activity index*” cell you named before, in the Matrix table: *“=a_tech10_invest*” (another reason why naming our cells makes sense!)

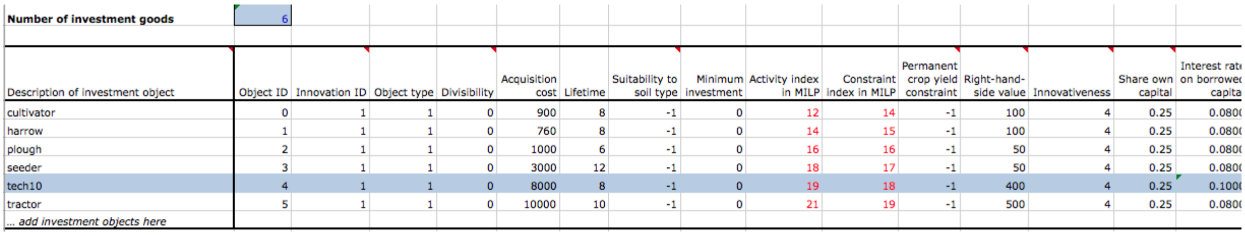


2	**Plough investment**	
	Number of variables in set	1
	Type of set	4
	Priority	1000
	Column index	16
	Downside pseudo costs	0.001
	Upside pseudo costs	0.001
3	**Seeder investment**	
	Number of variables in set	1
	Type of set	4
	Priority	1000
	Column index	18
	Downside pseudo costs	0.001
	Upside pseudo costs	0.001
4	**Tech10 investment**	
	Number of variables in set	1
	Type of set	4
	Priority	1000
	Column index	19
	Downside pseudo costs	0.001
	Upside pseudo costs	0.001
5	**Tractor investment**	
	Number of variables in set	1
	Type of set	4
	Priority	1000
	Column index	21
	Downside pseudo costs	0.001
	Upside pseudo costs	0.001

- Save the changes.

Network.xls: *Worksheet: Network0* Now we want to include our investment in *Network.xls*, which manages the diffusion of investments across our agent population. Remember we only have one network in our model.

- Insert a row in the table “*Number of investment goods*”.
- Define the “*Object ID*” number (4) and name the cell (e.g. “network_obj_4”)
- Change the tractor ID number (from 4 to 5, as you did above in *Matrix.xls*).
- Insert values for “*Acquisition cost*,” “*Lifetime*” and “*Right-hand-side value*” (given at the beginning of the exercise).
- Link the cells: *“Activity index in MILP*,” and “*Constraint index in MILP*,” to the index cells in the matrix (activity index and constraint index respectively). Create the “*Object name*” with no more than 10 characters.
- Be sure that the cell “*Number of Investment Goods*” is correct (it should update automatically)
- Save the changes



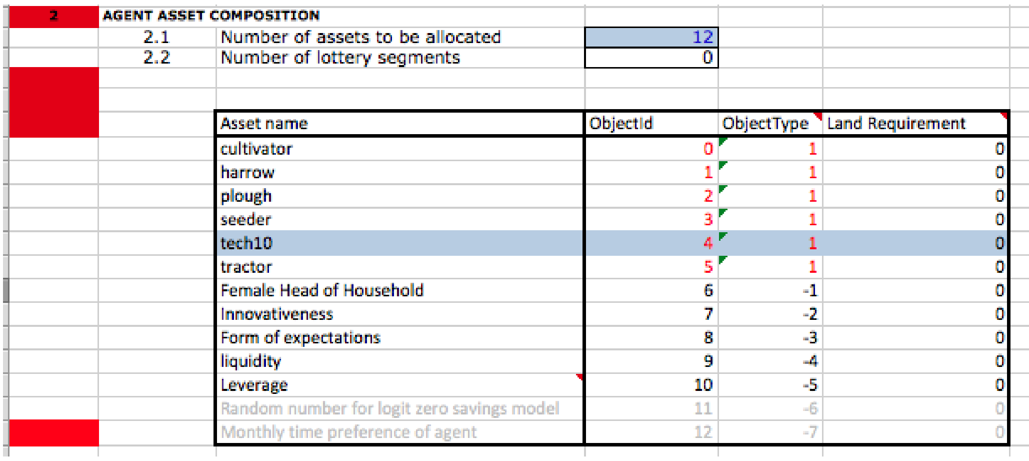
Number of investment goods: 6

Description of investment object	Object ID	Innovation ID	Object type	Divisibility	Acquisition cost	Lifetime	Suitability to soil type	Minimum investment	Activity index in MILP	Constraint index in MILP	Permanent crop yield constraint	Right-hand-side value	Innovativeness	Share own capital	Interest rat on borrowe capita
cultivator	0	1	1	0	900	8	-1	0	12	14	-1	100	4	0.25	0.080
harrow	1	1	1	0	760	8	-1	0	14	15	-1	100	4	0.25	0.080
plough	2	1	1	0	1000	6	-1	0	16	16	-1	50	4	0.25	0.080
seeder	3	1	1	0	3000	12	-1	0	18	17	-1	50	4	0.25	0.080
tech10	4	1	1	0	8000	8	-1	0	19	18	-1	400	4	0.25	0.100
tractor	5	1	1	0	10000	10	-1	0	21	19	-1	500	4	0.25	0.080
... add investment objects here															

Population.xls: In the last file we included the investment in the network diffusion model. Now we want to include the new technology within the assets available to farm agents and say which agents start the simulation with the technology available to them.

Worksheet: Clu0_0: Here we will include the new technology as an available asset

- Insert one row for agents' assets in the table "*AGENT ASSET COMPOSITION*." Link the "*Object ID*" cell to the "*Object ID*" cell in *Network.xls*. For "*ObjectType*" and "*Land Requirement*," copy the values and formulas from other investments.



2 AGENT ASSET COMPOSITION

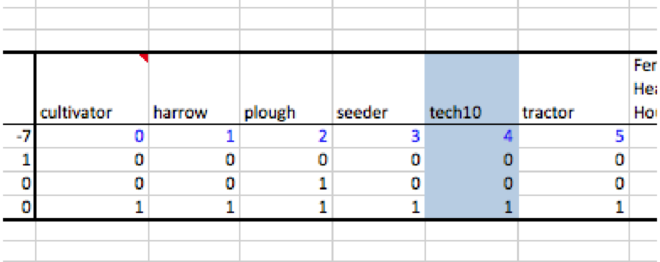
2.1 Number of assets to be allocated 12

2.2 Number of lottery segments 0

Asset name	ObjectId	ObjectType	Land Requirement
cultivator	0	1	0
harrow	1	1	0
plough	2	1	0
seeder	3	1	0
tech10	4	1	0
tractor	5	1	0
Female Head of Household	6	-1	0
Innovativeness	7	-2	0
Form of expectations	8	-3	0
liquidity	9	-4	0
Leverage	10	-5	0
Random number for logit zero savings model	11	-6	0
Monthly time preference of agent	12	-7	0

Worksheet: Assets0_0: And here we will state that only Agent 300 initially owns Tech10.

- Insert one column for Tech10. Link the "*Header*" cell to the "*ObjectID*" cell in the spreadsheet "*Clu0_0*." Define which agent has it available in the first period or not.



	cultivator	harrow	plough	seeder	tech10	tractor	Fer Hea Hou
-7	0	1	2	3	4	5	
1	0	0	0	0	0	0	
0	0	0	1	0	0	0	
0	1	1	1	1	1	1	

Save the changes and run the model.

2.4 Add “symbolic” objects

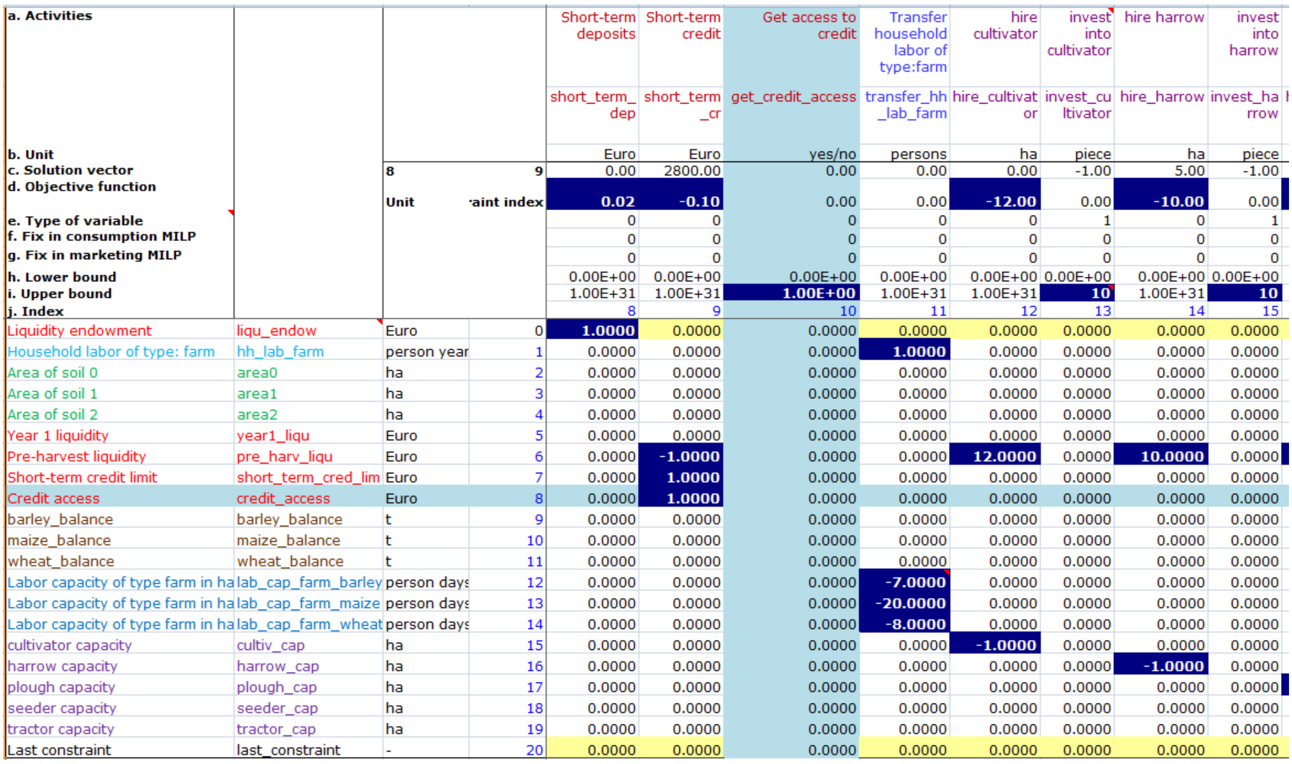
A very handy feature in MPMAS that we call “symbolic” objects can be used to grant or deny individual agents the access to certain LP activities. Suppose some but not all agents have access to preferential credit (or specific market channels, or any other activity that is linked to them personally). In this case, the access to these activities has similarities with “normal” investments, only they are without purchase price and, once they have been acquired, these “symbolic” objects have very long if not indefinite lifetimes. MPMAS implements this feature as a special activity in the LP that will be switched on for some agents (by inserting the value -1 in the respective constraint) and switched off for all others (by leaving all coefficients set to zero). Start with the basic model; the solutions can be found in *./TUT*_symbolic_object.*

In our example, only agent 15 should now be able to take short-term credit, but not agent 67 and agent 300 (they are excluded from the financial market). With over words, only agent 300 owns the symbolic object “*Get access to credit*” from the start on.

Files to modify: Matrix.xls; Network.xls; Population.xls

Matrix.xls: We first add the symbolic object activity to the agent decision matrix.

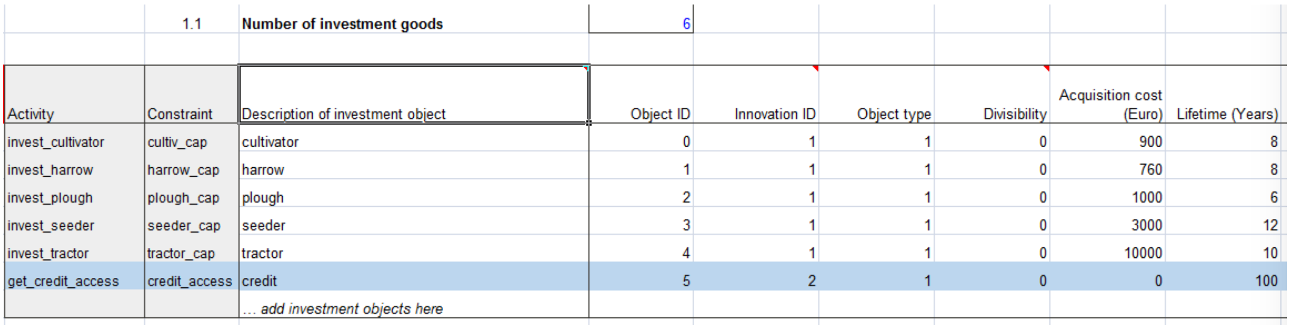
- Insert one new column/activity with the activity name “*Get access to credit*”
- Give the index cell a name using Excel's 'Name Manager' (e.g. “*get_credit_access*”)
- The values in the rows “*c. Solution vector*”, “*d. Objective function*”, “*e. Type of variable*”, “*f. Fix in consumption*”, “*g. Fix in market*”, “*h. Lower bounds*” and “*j. Upper bounds*” differ from “normal” investment activities. All coefficients are zero except the upper bound which is 1, because this activity is either switched on or off (yes or no).
- Insert one new row/constraint. Give the index cell a name (e.g. “*credit_access*”). Be sure that the cells for “*1 Type*,” “*2 Range*,” “*3 LHS*,” “*4 Sign*” and “*5 RHS*” have the same values and formulas as the other constraint cells. Insert a value of 1 in the short-term credit column, so that this activity can only be selected if the agent owns the symbolic object (in which case MPMAS will insert during run-time a value of -1 in the neighboring access to credit column)
- Do not forget to save the file



a. Activities				Short-term deposits	Short-term credit	Get access to credit	Transfer household labor of type:farm	hire cultivator	invest into cultivator	hire harrow	invest into harrow
				short_term_dep	short_term_cr	get_credit_access	transfer_hh_lab_farm	hire_cultivator	invest_cultivator	hire_harrow	invest_harrow
b. Unit				Euro	Euro	yes/no	persons	ha	piece	ha	piece
c. Solution vector		8	9	0.00	2800.00	0.00	0.00	0.00	-1.00	5.00	-1.00
d. Objective function		Unit	'aint index	0.02	-0.10	0.00	0.00	-12.00	0.00	-10.00	0.00
e. Type of variable				0	0	0	0	0	1	0	1
f. Fix in consumption MILP				0	0	0	0	0	0	0	0
g. Fix in marketing MILP				0	0	0	0	0	0	0	0
h. Lower bound				0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
i. Upper bound				1.00E+31	1.00E+31	1.00E+00	1.00E+31	1.00E+31	10	1.00E+31	10
j. Index				8	9	10	11	12	13	14	15
Liquidity endowment	liqu_endow	Euro	0	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Household labor of type: farm	hh_lab_farm	person year	1	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
Area of soil 0	area0	ha	2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Area of soil 1	area1	ha	3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Area of soil 2	area2	ha	4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Year 1 liquidity	year1_liqu	Euro	5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Pre-harvest liquidity	pre_harv_liqu	Euro	6	0.0000	-1.0000	0.0000	0.0000	12.0000	0.0000	10.0000	0.0000
Short-term credit limit	short_term_cred_lim	Euro	7	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Credit access	credit_access	Euro	8	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
barley_balance	barley_balance	t	9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
maize_balance	maize_balance	t	10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
wheat_balance	wheat_balance	t	11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Labor capacity of type farm in ha	lab_cap_farm_barley	person days	12	0.0000	0.0000	0.0000	-7.0000	0.0000	0.0000	0.0000	0.0000
Labor capacity of type farm in ha	lab_cap_farm_maize	person days	13	0.0000	0.0000	0.0000	-20.0000	0.0000	0.0000	0.0000	0.0000
Labor capacity of type farm in ha	lab_cap_farm_wheat	person days	14	0.0000	0.0000	0.0000	-8.0000	0.0000	0.0000	0.0000	0.0000
cultivator capacity	cultiv_cap	ha	15	0.0000	0.0000	0.0000	0.0000	-1.0000	0.0000	0.0000	0.0000
harrow capacity	harrow_cap	ha	16	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-1.0000	0.0000
plough capacity	plough_cap	ha	17	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
seeder capacity	seeder_cap	ha	18	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
tractor capacity	tractor_cap	ha	19	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Last constraint	last_constraint	-	20	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Network.xls: Now we include the symbolic object in the sheet ***Worksheet: Network0***

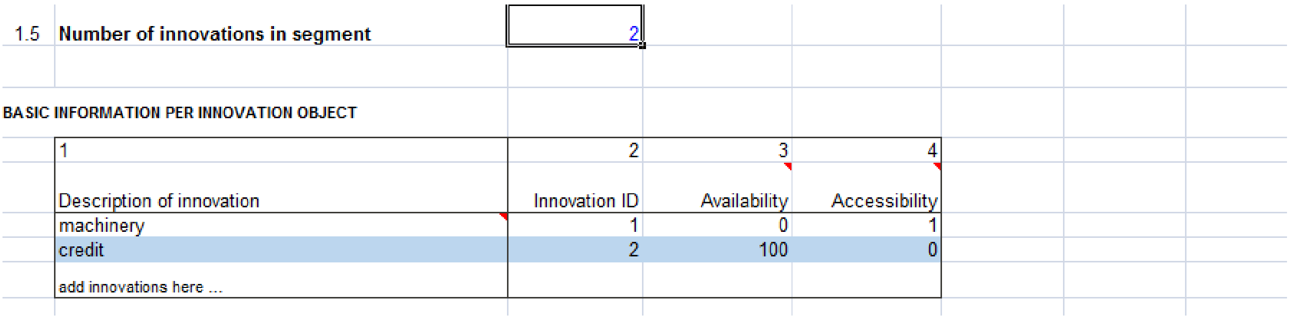
- Insert a row in the table "*Number of investment goods*".
- Add activity and constraint names as used in Matrix.xls
- Define the "*Object ID*" number (5) and "*Innovation ID*" (2).
- Insert values for "*Acquisition cost,*" (0) "*Lifetime*" (10) and "*Right-hand-side value*" (10000).
- Link the cells: *"Activity index in MILP,"* and "*Constraint index in MILP,*" to the index cells in Matrix.xls (activity index and constraint index respectively). Create the "*Object name*" with no more than 10 characters.
- Be sure that the cell "*Number of Investment Goods*" is correctly updated



	1.1	Number of investment goods	6					
Activity	Constraint	Description of investment object	Object ID	Innovation ID	Object type	Divisibility	Acquisition cost (Euro)	Lifetime (Years)
invest_cultivator	cultiv_cap	cultivator	0	1	1	0	900	8
invest_harrow	harrow_cap	harrow	1	1	1	0	760	8
invest_plough	plough_cap	plough	2	1	1	0	1000	6
invest_seeder	seeder_cap	seeder	3	1	1	0	3000	12
invest_tractor	tractor_cap	tractor	4	1	1	0	10000	10
get_credit_access	credit_access	credit	5	2	1	0	0	100
		... add investment objects here						

Network.xls: ***Worksheet: Segment0_0*** We need to add some additional information on the type of innovation (detailed explanation concerning the innovation diffusion model is omitted here, please consult the technical documentation in case of further questions).

- Insert a row in table “*Basic information per innovation object*”
- Insert “*Innovation ID*” (2)
- Set “*Availability*” to high number (e.g., 100), which means it will never become available to other agents after initialization
- Insert value for “*Accessibility*” (0)
- Ensure that value in cell “*Number of innovations in segment*” is correctly updated
- Save the file



1.5 **Number of innovations in segment** 2

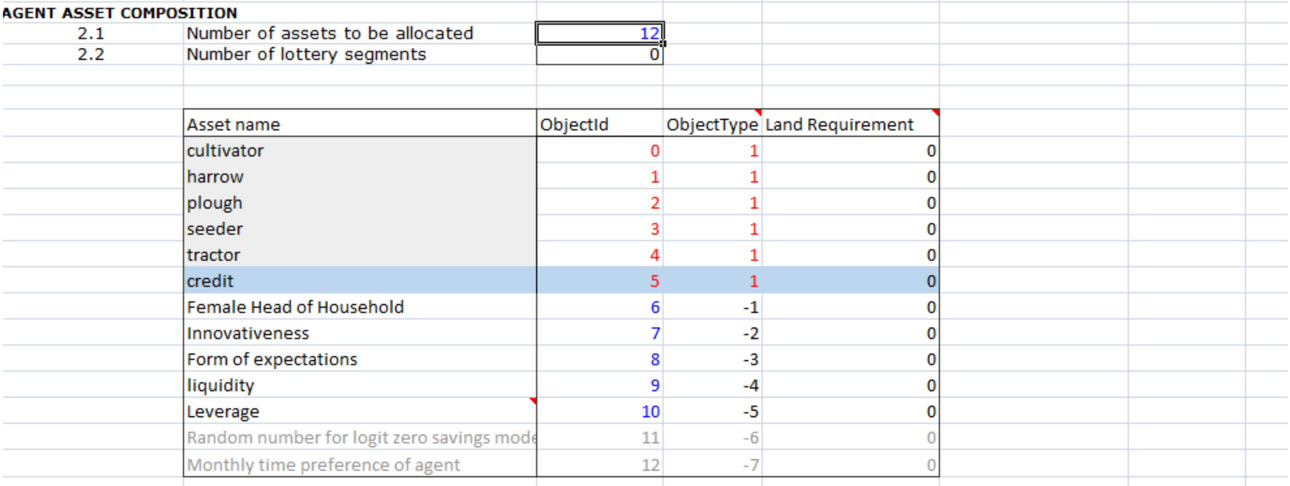
BASIC INFORMATION PER INNOVATION OBJECT

1	2	3	4
Description of innovation	Innovation ID	Availability	Accessibility
machinery	1	0	1
credit	2	100	0
add innovations here ...			

Population.xls: Now we want to include the symbolic object within the assets available to farm agents and say which agents start the simulation with access to credit.

Worksheet: Clu0_0: Here we will include the symbolic object as an available asset

- Insert one row for agents’ assets in the table “*AGENT ASSET COMPOSITION*.” Link the “*Object ID*” cell to the “*Object ID*” cell in *Network.xls*. For “*ObjectType*” and “*Land Requirement*,” copy the values and formulas from other investments.



AGENT ASSET COMPOSITION

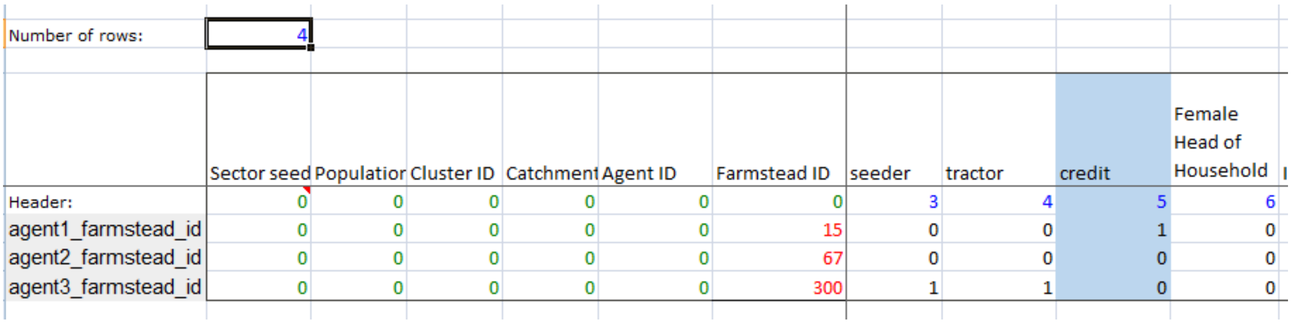
2.1 Number of assets to be allocated 12

2.2 Number of lottery segments 0

Asset name	ObjectId	ObjectType	Land Requirement
cultivator	0	1	0
harrow	1	1	0
plough	2	1	0
seeder	3	1	0
tractor	4	1	0
credit	5	1	0
Female Head of Household	6	-1	0
Innovativeness	7	-2	0
Form of expectations	8	-3	0
liquidity	9	-4	0
Leverage	10	-5	0
Random number for logit zero savings mode	11	-6	0
Monthly time preference of agent	12	-7	0

Worksheet: Assets0_0: And here we will state that only Agent 15 initially owns the symbolic object.

- Insert one column for “credit”. Link the “*Header*” cell to the “*ObjectID*” cell in the spreadsheet “*Clu0_0*.” Define agent 15 as the only agent has it available in the first period or not.



Number of rows:	4									
	Sector seed	Populatior	Cluster ID	Catchment	Agent ID	Farmstead ID	seeder	tractor	credit	Female Head of Household
Header:	0	0	0	0	0	0	3	4	5	6
agent1_farmstead_id	0	0	0	0	0	15	0	0	1	0
agent2_farmstead_id	0	0	0	0	0	67	0	0	0	0
agent3_farmstead_id	0	0	0	0	0	300	1	1	0	0

Save the changes and run the model.

2.5 Add more soil types

In our original model, we had three soil types (Soil 0, Soil 1, and Soil 2). You can remember this from when we added our “*Grow Potato*” activities for each soil type. Now we will return to the base model and add one additional soil type. For this, we are going to need to change a few more files than we have in the past two exercises. The solution files can be found in *./TUT*_soils*.

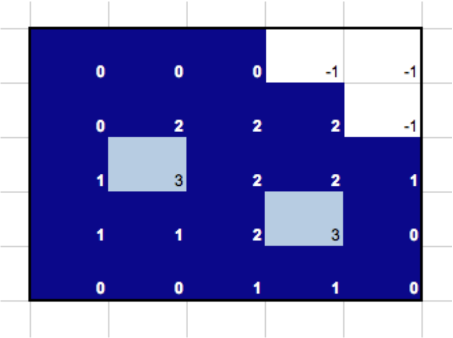
Files to modify: BasicData.xls; Map.xls; Matrix.xls; Network.xls; Xresults.xls

Basicdata.xls: The input file *BasicData.xls* collects parameter values that are not exclusive to a single input file and stay the same over simulation periods. Since the number of soil types will impact a number of different files, we make sure to include that information in the *Basicdata.xls* file.

- Under “*3.03 number of soil types*,” change the number from three to four to reflect the new total number of soil types.

Map.xls: The map file is used to define the cellular landscape component of MPMAS. Using the input from the map file, agents will be assigned different positions in a spatial area—including definitions for what soil types they have. For this reason, any change to soil types must be updated in the map file.

- In the sheet “*CatchMap00Soil*” you have to add one more soil type. As the study area is already covered with all three soil types (0,1,2) you have to add the fourth soil type by arbitrarily taking away a plot from the other soil types and fill it with the new soil type 3.



0	0	0	-1	-1
0	2	2	2	-1
1	3	2	2	1
1	1	2	3	0
0	0	1	1	0

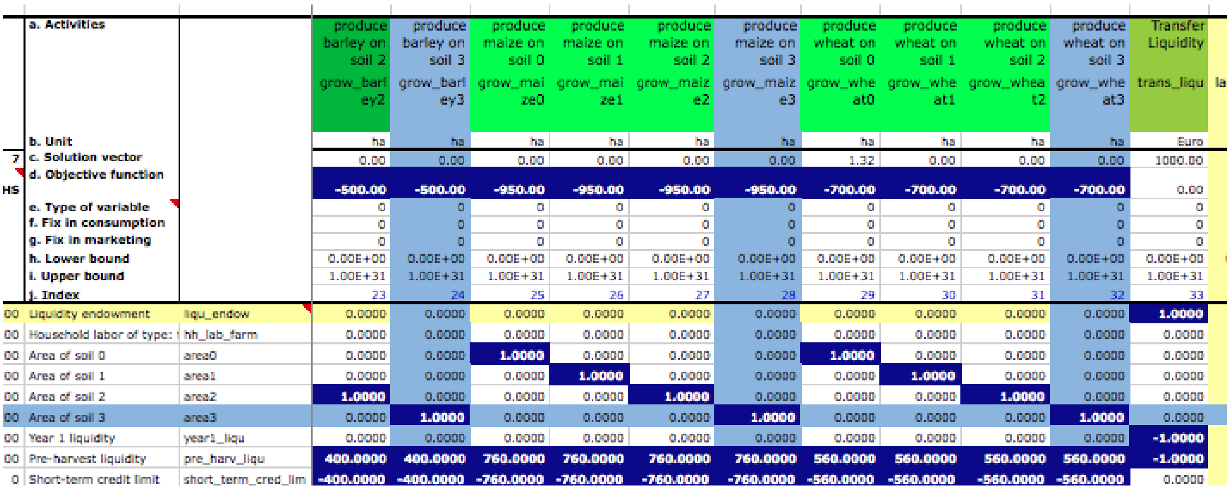
Matrix.xls: Now we need to include new activities for producing each crop on the new soil type and we need to define a new constraint reflecting the total available area of our new soil type.

- As we assume that you can grow each crop on each soil you have to add a growing activity on Soil 3 for each crop (barley, maize and wheat). We chose all coefficients to be the same as with the other soil types except yield; we chose a number a little bit higher than was used for Soil 2

- Define a cell name for each activity index (e.g. "a_produce_wheat_s3").

- Take care that in the upper part of the matrix file, under "*1.2 Other information about the matrix,"* the column "*Number nonzero activities*" changes correspondingly (You must change the reference to the index name of your inserted column so that it will be counted as a nonzero activity)

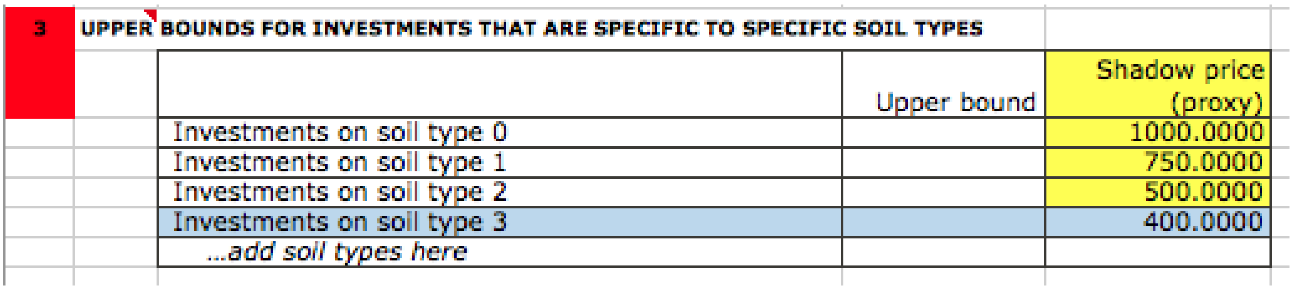
- Add a new row for the constraint "Area of Soil 3."



	a. Activities		produce barley on soil 2 grow_barley2	produce barley on soil 3 grow_barley3	produce maize on soil 0 grow_maize0	produce maize on soil 1 grow_maize1	produce maize on soil 2 grow_maize2	produce maize on soil 3 grow_maize3	produce wheat on soil 0 grow_wheat0	produce wheat on soil 1 grow_wheat1	produce wheat on soil 2 grow_wheat2	produce wheat on soil 3 grow_wheat3	Transfer Liquidity trans_liqu	la
	b. Unit		ha	ha	ha	ha	ha	ha	ha	ha	ha	ha	Euro	
7	c. Solution vector		0.00	0.00	0.00	0.00	0.00	0.00	1.32	0.00	0.00	0.00	1000.00	
HS	d. Objective function		-500.00	-500.00	-950.00	-950.00	-950.00	-950.00	-700.00	-700.00	-700.00	-700.00	0.00	
	e. Type of variable		0	0	0	0	0	0	0	0	0	0	0	
	f. Fix in consumption		0	0	0	0	0	0	0	0	0	0	0	
	g. Fix in marketing		0	0	0	0	0	0	0	0	0	0	0	
	h. Lower bound		0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	
	i. Upper bound		1.00E+31	1.00E+31	1.00E+31	1.00E+31	1.00E+31	1.00E+31	1.00E+31	1.00E+31	1.00E+31	1.00E+31	1.00E+31	
	j. Index		23	24	25	26	27	28	29	30	31	32	33	
00	Liquidity endowment	liqu_endow	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	
00	Household labor of type: (	hh_lab_farm	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
00	Area of soil 0	area0	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	
00	Area of soil 1	area1	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	
00	Area of soil 2	area2	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	
00	Area of soil 3	area3	0.0000	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000	
00	Year 1 liquidity	year1_liqu	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-1.0000	
00	Pre-harvest liquidity	pre_harv_liqu	400.0000	400.0000	760.0000	760.0000	760.0000	760.0000	560.0000	560.0000	560.0000	560.0000	-1.0000	
0	Short-term credit limit	short_term_cred_lim	-400.0000	-400.0000	-760.0000	-760.0000	-760.0000	-760.0000	-560.0000	-560.0000	-560.0000	-560.0000	0.0000	

Network.xls: ***Worksheet: Segment0_0:*** The network file contains information on innovations, which are sometimes specific to certain soil-types. In order to calculate information for these investments, it is necessary to include information on the shadow price of each soil type. In this example, there are no soil-specific investments, but it is still necessary to include this information as MPMAS will not run if the number of soil types included in *Network.xls* doesn't match the number of soil types in the model.

- In Table “*3 UPPERBOUNDS FOR INVESTMENTS THAT ARE SPECIFIC TO SOIL TYPES*,” add a row for soil type 3
- Include any non-negative value under the column “*Shadow Price (Proxy)*.” Because the value is not used in any calculations in this example, it is only necessary to have a non-negative value rather than a specific one



3 UPPER BOUNDS FOR INVESTMENTS THAT ARE SPECIFIC TO SPECIFIC SOIL TYPES

	Upper bound	Shadow price (proxy)
Investments on soil type 0		1000.0000
Investments on soil type 1		750.0000
Investments on soil type 2		500.0000
Investments on soil type 3		400.0000
...add soil types here		

Xresults.xls: Now we want to make sure that our results get fed into the appropriate columns of our output files.

- Update headers of activities and constraints in this file (copy and past from *Matrix.xls*)

Run the model

2.6 Add more agents

Our original model included three farm household agents. Now we want to add one additional agent to our simulation. Because we link our model to a cellular landscape component, we have to think about where our agents are going to fit in on our current maps. Here, we will add the agent’s farmstead and plots on cells that are not already occupied by other agents (given in the file *Map.xls*). The solution files can be found in */TUT*_agent*.

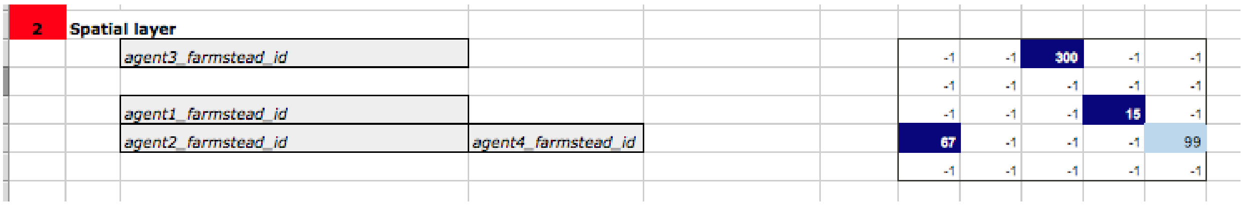
Files to modify: Map.xls; Population.xls

Map.xls:

Worksheet: CatchMap00Farm: The first thing we want to do is define the new agent's ID within the agent population

- Write the agent ID in a grid cell to specify the location of the agent’s farmstead. Notice that the location of the new farmstead has to be within the limits of the Catchment (see the matrix on the worksheet “*CatchMap00Catchment*,” which defines the cellular plots of land) and cannot overlap with any of the farm plots owned by other agents (see the corresponding matrix on worksheet “*CatchMap00Prop”*). In other words, the agent can be included on any cell within the catchment matrix not currently occupied by other agents.

- Assign a name to the cell (e.g. “*agent4_farmstead_id*”)

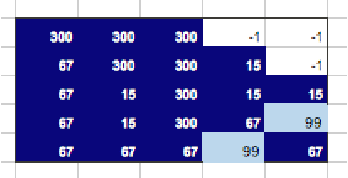


2	Spatial layer		
	agent3_farmstead_id		
	agent1_farmstead_id		
	agent2_farmstead_id	agent4_farmstead_id	

-1	-1	300	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	15	-1
67	-1	-1	-1	99
-1	-1	-1	-1	-1

Worksheet: CatchMap00Prop: Now we want to identify which plots of current land are used by the new farmstead. We will use the agent ID we assigned in the previous worksheet.

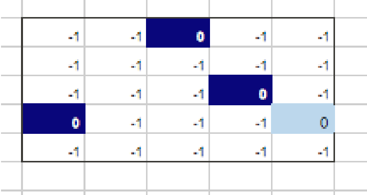
- Assign farm plots to the new agent and specify its location by writing the agent's ID in the cells. As before, the location of farm plots has to be within the limits of the catchment and cannot overlap with any of the farm plots owned by other agents.



300	300	300	-1	-1
67	300	300	15	-1
67	15	300	15	15
67	15	300	67	99
67	67	67	99	67

Worksheet: CatchMap00Pop: Now that we have defined an ID and a location for our new farmstead, we want to include it within the agent population of the model.

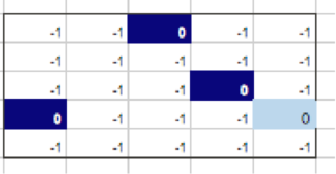
- Assign the agent to a population. In this case all agents belong to the population "0"



-1	-1	0	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	0	-1
0	-1	-1	-1	0
-1	-1	-1	-1	-1

Worksheet: CatchMap00Clu: MPMAS can separate the agent population into different clusters. In this model, we have only one population cluster, '0'. We have to assign our new agent to this cluster.

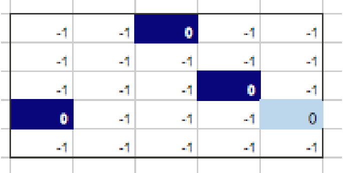
- Assign the agent to a population cluster. In this case all agents belong to the cluster "0"



-1	-1	0	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	0	-1
0	-1	-1	-1	0
-1	-1	-1	-1	-1

Worksheet: CatchMap00Netw: Networks are defined within the population and used to model innovation diffusions. When adding a new agent, be sure to include them in a network. In this model, we have only one network, '0'.

- Assign the agent to a network. In this case all agents belong to the network "0"



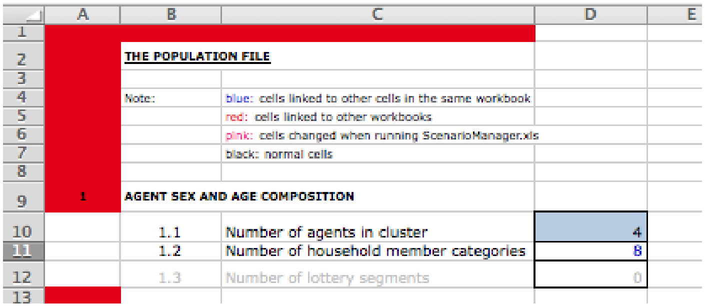
-1	-1	0	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	0	-1
0	-1	-1	-1	0
-1	-1	-1	-1	-1

Save changes

Population.xls: The changes to *Map.xls* helped include our new agent in the cellular landscape component of MPMAS. Now we want to include make sure our new agent is assigned demographics and assets for use in the decision-making process of MPMAS. This is done through the *Population.xls*.

Worksheet: Clu0_0: MPMAS assigns demographics and assets to agents. In order for the new agent to receive these attributes, they must be included within the population file under the cluster they belong to. Remember, in our model there is only one population cluster, '0'.

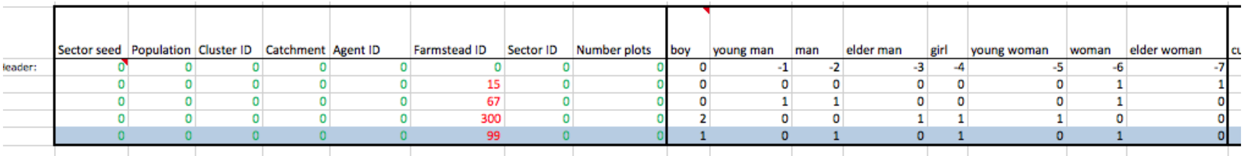
- Change *"1.1 Number of agents in cluster*" to fit the new amount of agents in each cluster. In this case, there is only one cluster (Clu0_0).



	A	B	C	D	E
1					
2		THE POPULATION FILE			
3					
4		Note:	blue: cells linked to other cells in the same workbook		
5			red: cells linked to other workbooks		
6			pink: cells changed when running ScenarioManager.xls		
7			black: normal cells		
8					
9	1	AGENT SEX AND AGE COMPOSITION			
10		1.1	Number of agents in cluster	4	
11		1.2	Number of household member categories	8	
12		1.3	Number of lottery segments	0	
13					

Worksheet: Assets0_0: Now we will assign our new agent their initial demographics and resources, using the farmstead ID defined in *Map.xls*.

- Add an extra row on the table and assign manually the characteristics and assets of the new agent. Link the cell corresponding to your agent on the "*Farmstead ID*" column with the cell corresponding to the Farmstead ID on the file *Map.xls* (worksheet "*CatchMap00Farm*")



	Sector seed	Population	Cluster ID	Catchment	Agent ID	Farmstead ID	Sector ID	Number plots	boy	young man	man	elder man	girl	young woman	woman	elder woman	cu
leader:	0	0	0	0	0	0	0	0	0	-1	-2	-3	-4	-5	-6	-7	
	0	0	0	0	0	15	0	0	0	0	0	0	0	0	1	1	
	0	0	0	0	0	67	0	0	0	1	1	0	0	0	1	0	
	0	0	0	0	0	300	0	0	2	0	0	1	1	1	0	0	
	0	0	0	0	0	99	0	0	1	0	1	0	1	0	1	0	

Save the changes and run the model

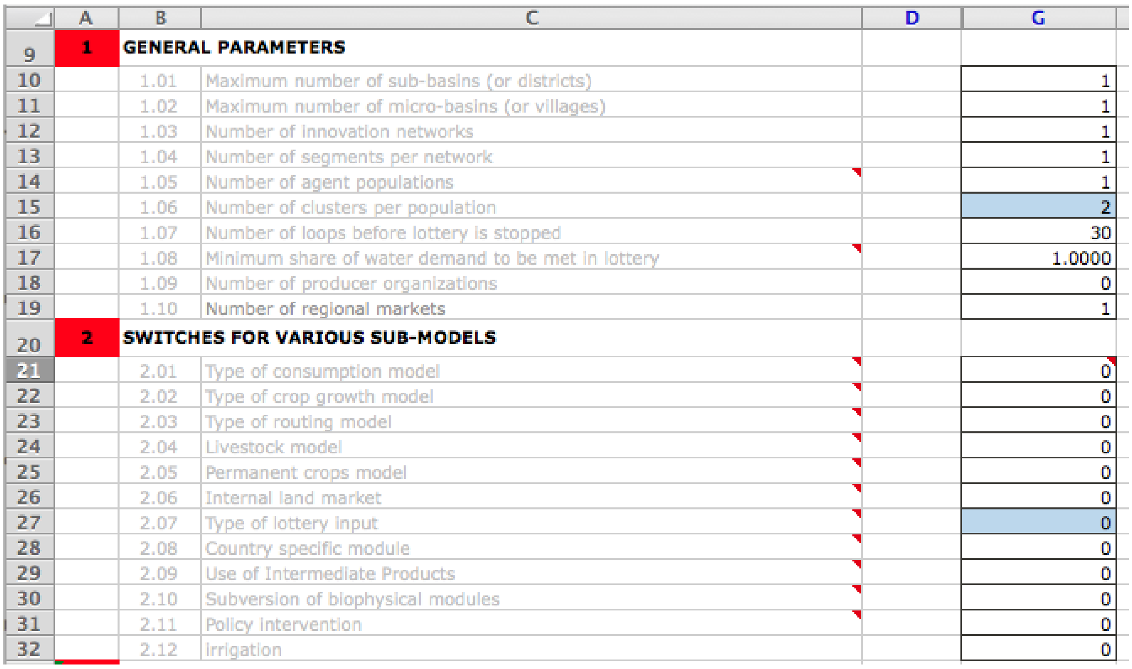
2.7 Assignment of assets by lottery

Now we are going to extend the model significantly, by adding more agents and land, a new population cluster and assigning assets to agents by means of a lottery. Adding new agents and plots is familiar from above, but this time it will be done by defining a new population cluster into the model. The solutions to this exercise can be found in *./TUT*_lottery*.

Files to modify: BasicData.xls; Population.xls; Map.xls; ScenarioManager.xls

BasicData.xls: ***Worksheet: Bdata:*** Remember that *BasicData.xls* contains important information shared across input files and simulation scenarios. Here we are expanding our agent population by adding a new clusters and introducing a lottery input.

- Change "*1.06 Number of clusters per population*" to "2" (because now we are creating a new cluster)
- Change "*2.07 Type of lottery inpu*t" to "0." By doing this, the information for the lottery is taken from *Population.xls* (worksheets "*Clu0_0*," "*Clu0_1*," etc.), when creating the input Files.



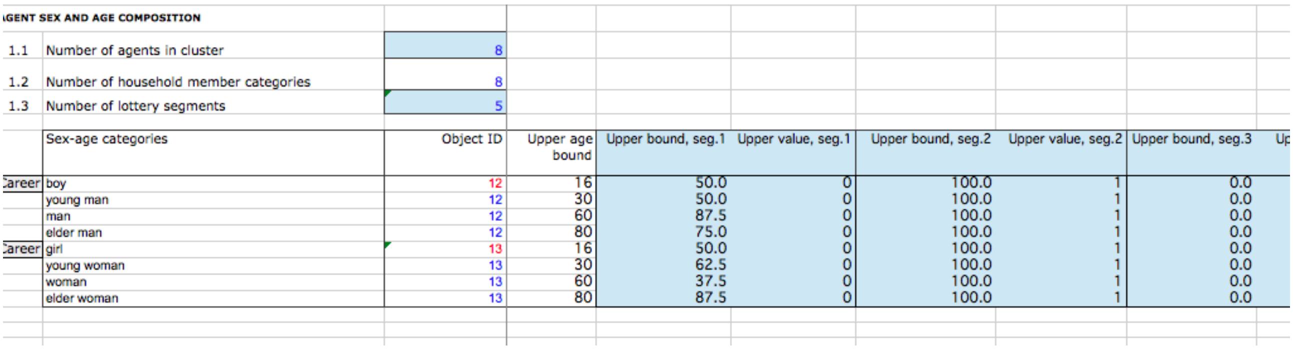
	A	B	C	D	G
9	1	GENERAL PARAMETERS			
10		1.01	Maximum number of sub-basins (or districts)		1
11		1.02	Maximum number of micro-basins (or villages)		1
12		1.03	Number of innovation networks		1
13		1.04	Number of segments per network		1
14		1.05	Number of agent populations		1
15		1.06	Number of clusters per population		2
16		1.07	Number of loops before lottery is stopped		30
17		1.08	Minimum share of water demand to be met in lottery		1.0000
18		1.09	Number of producer organizations		0
19		1.10	Number of regional markets		1
20	2	SWITCHES FOR VARIOUS SUB-MODELS			
21		2.01	Type of consumption model		0
22		2.02	Type of crop growth model		0
23		2.03	Type of routing model		0
24		2.04	Livestock model		0
25		2.05	Permanent crops model		0
26		2.06	Internal land market		0
27		2.07	Type of lottery input		0
28		2.08	Country specific module		0
29		2.09	Use of Intermediate Products		0
30		2.10	Subversion of biophysical modules		0
31		2.11	Policy intervention		0
32		2.12	Irrigation		0

Population.xls: Now we want to introduce lottery segments into our existing population and add our new population in a new cluster. We do this using *Population.xls*.

Worksheet: Clu0_0 In the original model we did not assign agent assets by lottery (compare *Popluation.xls* found in *./TUT*/xlsinput*). In addition to adding agents, we need to introduce a lottery for assigning demographics and assets to our agents. We do this first for our original cluster.

- Include the lottery segments and the inverse cumulative distributions as explained in Remark 2 below. The generation of our distribution frequencies can be seen by looking at the Worksheet “*Data*” in *Population.xls* found in *./TUT*_lottery/xlsinput*.

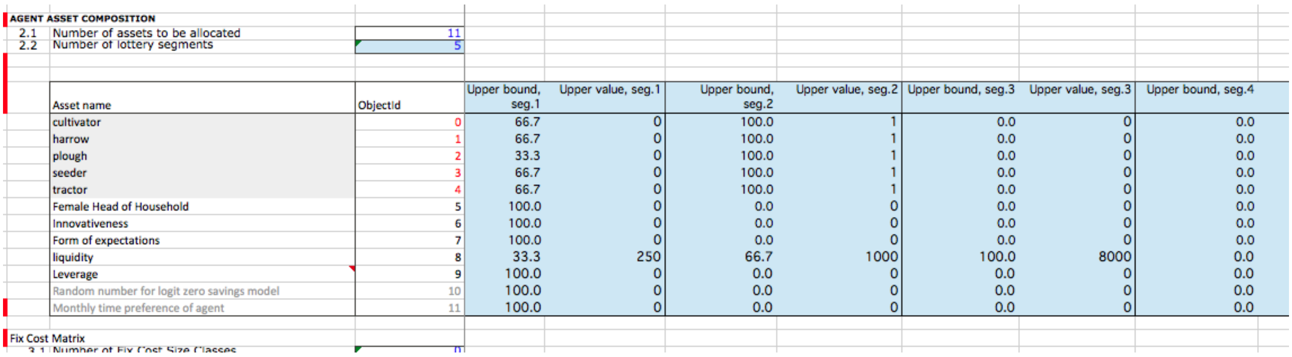
- Change manually “*1.1 Number of agents*” to indicate the amount of agents in Cluster 0 (in this case, it is eight)
- Update 1.3 and 2.2 “*Number of lottery segments*” manually (In this case, it is five).



ιGENT SEX AND AGE COMPOSITION

1.1	Number of agents in cluster	8
1.2	Number of household member categories	8
1.3	Number of lottery segments	5

	Sex-age categories	Object ID	Upper age bound	Upper bound, seg.1	Upper value, seg.1	Upper bound, seg.2	Upper value, seg.2	Upper bound, seg.3	Up
Career	boy	12	16	50.0	0	100.0	1	0.0	
	young man	12	30	50.0	0	100.0	1	0.0	
	man	12	60	87.5	0	100.0	1	0.0	
	elder man	12	80	75.0	0	100.0	1	0.0	
Career	girl	13	16	50.0	0	100.0	1	0.0	
	young woman	13	30	62.5	0	100.0	1	0.0	
	woman	13	60	37.5	0	100.0	1	0.0	
	elder woman	13	80	87.5	0	100.0	1	0.0	



AGENT ASSET COMPOSITION

2.1	Number of assets to be allocated	11
2.2	Number of lottery segments	5

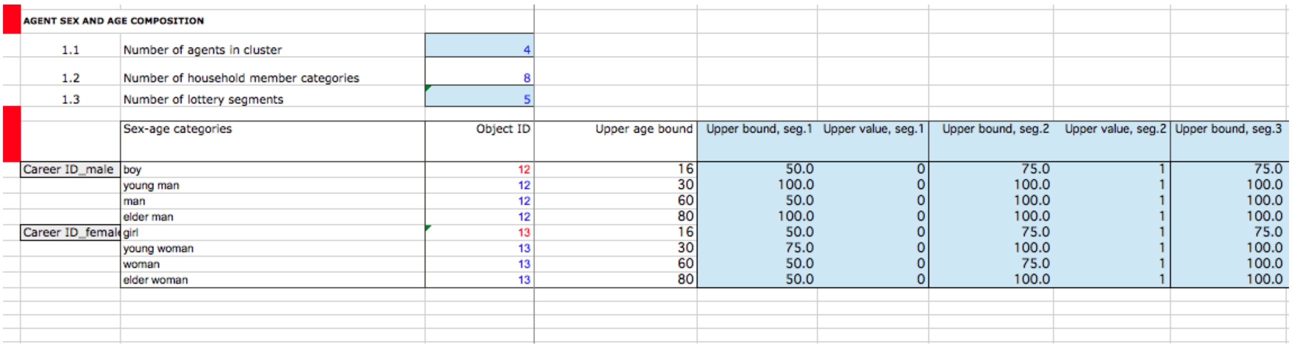
Asset name	ObjectId	Upper bound, seg.1	Upper value, seg.1	Upper bound, seg.2	Upper value, seg.2	Upper bound, seg.3	Upper value, seg.3	Upper bound, seg.4
cultivator	0	66.7	0	100.0	1	0.0	0	0.0
harrow	1	66.7	0	100.0	1	0.0	0	0.0
plough	2	33.3	0	100.0	1	0.0	0	0.0
seeder	3	66.7	0	100.0	1	0.0	0	0.0
tractor	4	66.7	0	100.0	1	0.0	0	0.0
Female Head of Household	5	100.0	0	0.0	0	0.0	0	0.0
Innovativeness	6	100.0	0	0.0	0	0.0	0	0.0
Form of expectations	7	100.0	0	0.0	0	0.0	0	0.0
liquidity	8	33.3	250	66.7	1000	100.0	8000	0.0
Leverage	9	100.0	0	0.0	0	0.0	0	0.0
Random number for logit zero savings model	10	100.0	0	0.0	0	0.0	0	0.0
Monthly time preference of agent	11	100.0	0	0.0	0	0.0	0	0.0

Fix Cost Matrix

3.1 Number of Fix Cost Size Classes 0

Worksheet: Clu0_1: After introducing a lottery component to the original cluster, now we want to introduce a new cluster. We do this by creating an entirely new worksheet directly after our “*Clu0_0*” worksheet. The basic layout of the worksheet is the same.

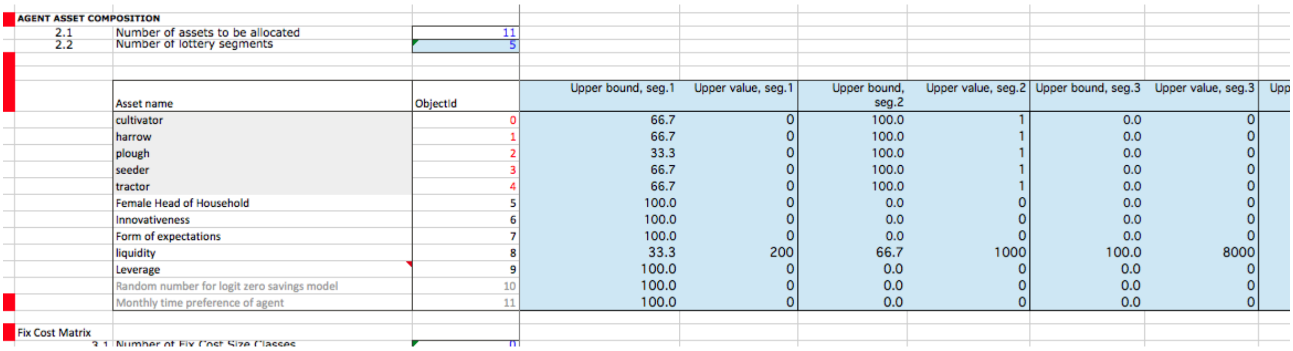
- Create a new worksheet “*Clu0_1*” following the same structure as in the worksheet “*Clu0_0.*” Insert this new worksheet right after worksheet “*Clu0_0.*”
- Similar to before, include the lottery segments and the inverse cumulative distributions. Notice that in the pictures below the distribution frequencies are different for Cluster 1 than they were for Cluster 0. Again, consult Remark 2 below for more information on this process.
- Change manually “*1.1 Number of agents*” to indicate the amount of agents in Cluster 1 (in this case, it is four)
- Update 1.3 and 2.2 “*Number of lottery segments*” manually (In this case, it is five).



AGENT SEX AND AGE COMPOSITION

1.1	Number of agents in cluster	4
1.2	Number of household member categories	8
1.3	Number of lottery segments	5

	Sex-age categories	Object ID	Upper age bound	Upper bound, seg.1	Upper value, seg.1	Upper bound, seg.2	Upper value, seg.2	Upper bound, seg.3
Career ID_male	boy	12	16	50.0	0	75.0	1	75.0
	young man	12	30	100.0	0	100.0	1	100.0
	man	12	60	50.0	0	100.0	1	100.0
	elder man	12	80	100.0	0	100.0	1	100.0
Career ID_femal	girl	13	16	50.0	0	75.0	1	75.0
	young woman	13	30	75.0	0	100.0	1	100.0
	woman	13	60	50.0	0	75.0	1	100.0
	elder woman	13	80	50.0	0	100.0	1	100.0



AGENT ASSET COMPOSITION

2.1	Number of assets to be allocated	11
2.2	Number of lottery segments	5

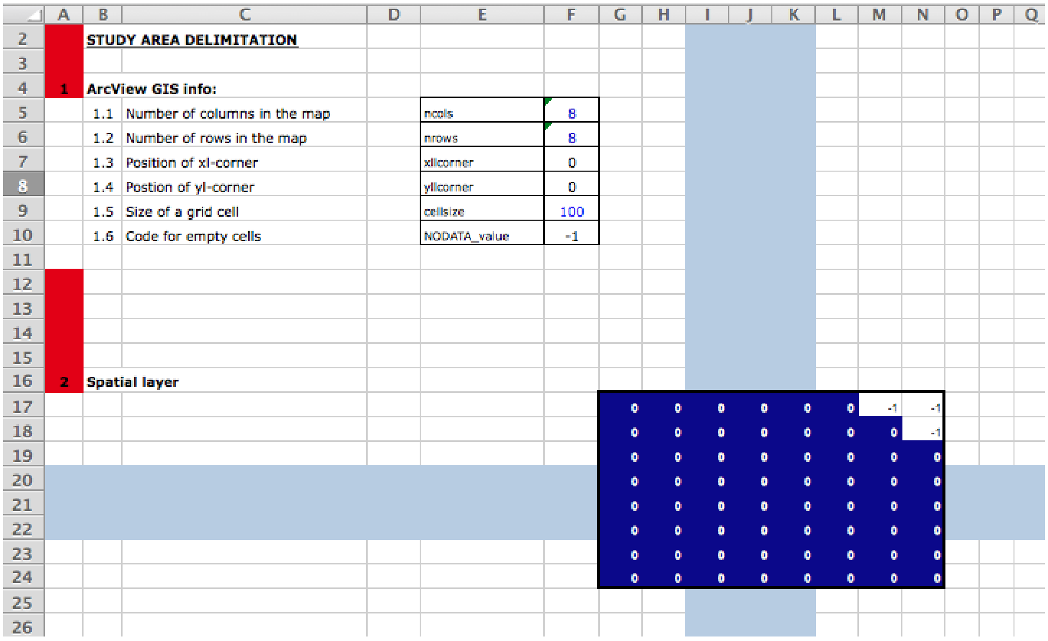
Asset name	ObjectId	Upper bound, seg.1	Upper value, seg.1	Upper bound, seg.2	Upper value, seg.2	Upper bound, seg.3	Upper value, seg.3	Upp
cultivator	0	66.7	0	100.0	1	0.0	0	
harrow	1	66.7	0	100.0	1	0.0	0	
plough	2	33.3	0	100.0	1	0.0	0	
seeder	3	66.7	0	100.0	1	0.0	0	
tractor	4	66.7	0	100.0	1	0.0	0	
Female Head of Household	5	100.0	0	0.0	0	0.0	0	
Innovativeness	6	100.0	0	0.0	0	0.0	0	
Form of expectations	7	100.0	0	0.0	0	0.0	0	
liquidity	8	33.3	200	66.7	1000	100.0	8000	
Leverage	9	100.0	0	0.0	0	0.0	0	
Random number for logit zero savings model	10	100.0	0	0.0	0	0.0	0	
Monthly time preference of agent	11	100.0	0	0.0	0	0.0	0	

Fix Cost Matrix

3.1 Number of Fix Cost Size Classes 0

Map.xls: In the last step, we introduced our new agent cluster into the agent-side of the model. Now we want to include our new population cluster into the cellular landscape component of our model. Doing so is very similar to how we added new agents with new land and soils types above.

- Create extra plots that agents can farm on. In this case, we need to increase each of the catchment matrices on all worksheets as such (using the worksheet "*CatchMap00Catchment*" as an example):



STUDY AREA DELIMITATION

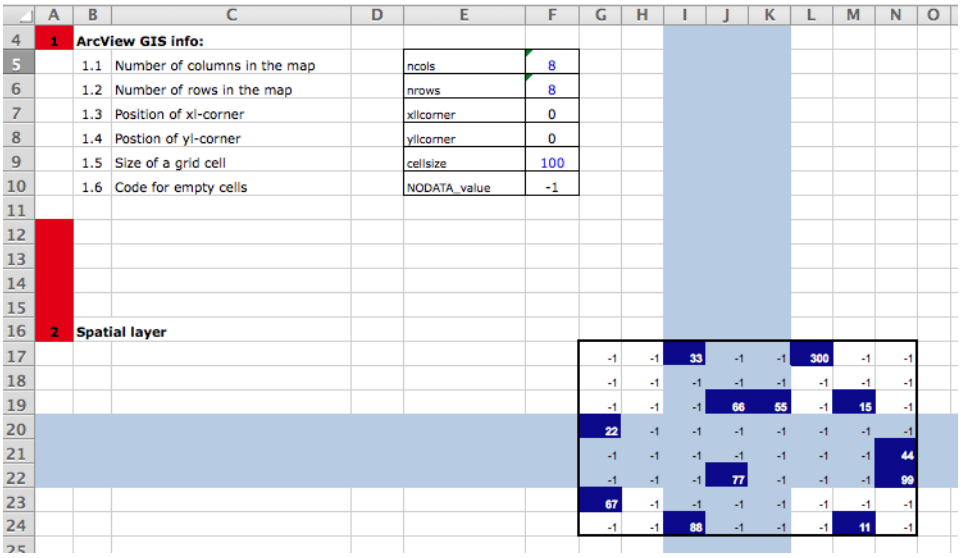
1 ArcView GIS info:

1.1	Number of columns in the map	ncols	8
1.2	Number of rows in the map	nrows	8
1.3	Position of xl-corner	xllcorner	0
1.4	Postion of yl-corner	yllcorner	0
1.5	Size of a grid cell	cellsize	100
1.6	Code for empty cells	NODATA_value	-1

2 Spatial layer

	G	H	I	J	K	L	M	N
17	0	0	0	0	0	0	-1	-1
18	0	0	0	0	0	0	0	-1
19	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0

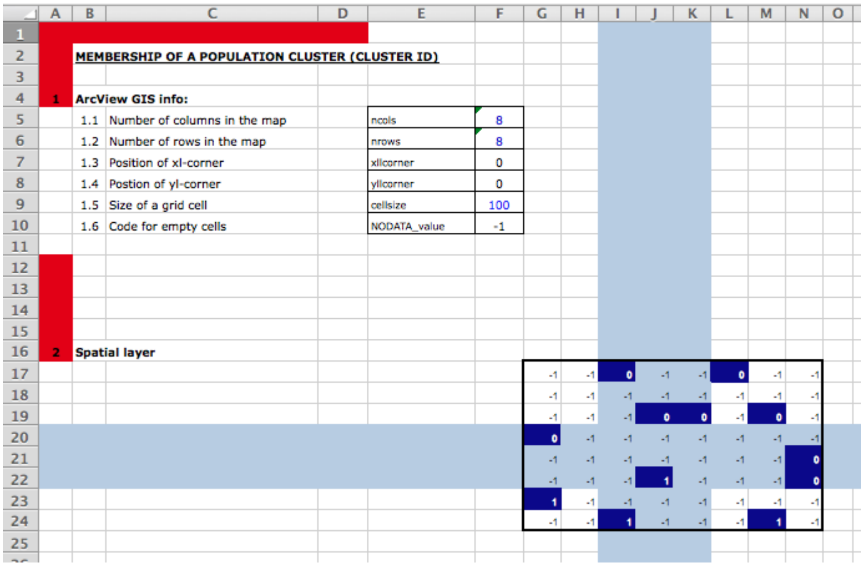
- Now be sure to create our new agents, again as explained above. For example, the worksheet "*CatchMap00Farm*" should look as such:



	A	B	C	E	F
4	1	ArcView GIS info:			
5		1.1	Number of columns in the map	ncols	8
6		1.2	Number of rows in the map	nrows	8
7		1.3	Position of xl-corner	xllcorner	0
8		1.4	Postion of yl-corner	yllcorner	0
9		1.5	Size of a grid cell	cellsize	100
10		1.6	Code for empty cells	NODATA_value	-1
16	2	Spatial layer			

	G	H	I	J	K	L	M	N
17	-1	-1	33	-1	-1	300	-1	-1
18	-1	-1	-1	-1	-1	-1	-1	-1
19	-1	-1	-1	66	55	-1	15	-1
20	22	-1	-1	-1	-1	-1	-1	-1
21	-1	-1	-1	-1	-1	-1	-1	44
22	-1	-1	-1	77	-1	-1	-1	99
23	67	-1	-1	-1	-1	-1	-1	-1
24	-1	-1	88	-1	-1	-1	11	-1

- ***Worksheet: CatchMap00Clu:*** Once you have updated all maps containing farm plots and agents, you now need to make sure that each farm agent is properly assigned to the correct cluster. Assign each of the agents to a population cluster, so that it matches the number of agents per cluster that we entered in *Population.xls*. In this case, we used "0" and "1" to define the different clusters (remember, we defined our population clusters in *Population.xls* with the worksheets "*Clu0_0*" and "*Clu0_1*").



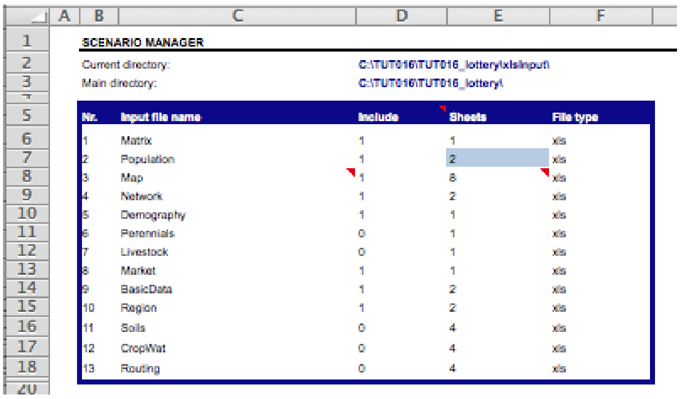
	A	B	C	E	F
2		MEMBERSHIP OF A POPULATION CLUSTER (CLUSTER ID)			
4	1	ArcView GIS info:			
5		1.1	Number of columns in the map	ncols	8
6		1.2	Number of rows in the map	nrows	8
7		1.3	Position of xl-corner	xllcorner	0
8		1.4	Postion of yl-corner	yllcorner	0
9		1.5	Size of a grid cell	cellsize	100
10		1.6	Code for empty cells	NODATA_value	-1
16	2	Spatial layer			

	G	H	I	J	K	L	M	N
17	-1	-1	0	-1	-1	0	-1	-1
18	-1	-1	-1	-1	-1	-1	-1	-1
19	-1	-1	-1	0	0	-1	0	-1
20	0	-1	-1	-1	-1	-1	-1	-1
21	-1	-1	-1	-1	-1	-1	-1	0
22	-1	-1	-1	1	-1	-1	-1	0
23	1	-1	-1	-1	-1	-1	-1	-1
24	-1	-1	1	-1	-1	-1	1	-1

ScenarioManager.xls:

Worksheet: Manager

- The file *ScenarioManager.xls* requires information about the various workbooks and files that MPMAS will use to run the simulations to know what information it needs to convert to ASCII input file. This includes information on the number of worksheets each file has. As we have added a new worksheet in the *Population.xls*, we need to make sure that the number of worksheets for this file is correct in *ScenarioManager.xls* (it should be "2"). If we did not update this, when MPMAS completes its asset assignment by lottery it would only include one sheet.



SCENARIO MANAGER

Current directory: C:\TUT016\TUT016_lottery\xls\input\

Main directory: C:\TUT016\TUT016_lottery\

Nr.	Input file name	Include	Sheets	File type
1	Matrix	1	1	xls
2	Population	1	2	xls
3	Map	1	8	xls
4	Network	1	2	xls
5	Demography	1	1	xls
6	Perennials	0	1	xls
7	Livestock	0	1	xls
8	Market	1	1	xls
9	BasicData	1	2	xls
10	Region	1	2	xls
11	Soils	0	4	xls
12	CropWat	0	4	xls
13	Routing	0	4	xls

Run the model

Remark 1: To initialize the random distribution of household composition and assets, a seed number is required. This number should be a negative integer, and can be specified in the *Region.xls* file (worksheet: "*Region*"). In the original model file, a negative number is already included. When this number is not changed across simulation runs, the allocation of agent household composition and assets will be the same every time MPMAS starts running.

Remark 2: Generating Inverse Cumulative Distribution Functions

Very often there is no information available regarding the actual composition of the entire population of real-world farm households (hh). Therefore, it is necessary to generate the model's population of agents by extrapolating from data available on a sample of the real-world population. (For more information on methods for generating agents, refer to Berger and Schreinemachers (2006) plus the technical documentation available at http://mp-mas.uni-hohenheim.de/documentation).

In this example problem, we make (simplified) use of the Monte Carlo approach explained in Berger and Schreinemachers (2006). The Monte Carlo approach requires us to include information into MPMAS on the inverse cumulative distribution frequencies of the various characteristics and assets we want to assign to agents. This means that we take a sample of data from the real-world population and attempt to generate a similar population in our model using a probabilistic procedure for assigning agent demographics and assets. In order for this to work, there are a few steps that must be taken:

1. Generating the demographic composition of the population

    - A first step for generating the agents' population is to classify agents into clusters. This classification is specific for each study. In that sense, agents can be classified e.g. according to educational level, main crops they plant, level of technology used, etc.

- For this example, clusters were defined by household size, assuming that this is the variable most strongly correlated with all other variables. Two clusters were chosen, as this number captured all the different household sizes. Cluster0_0 groups all households composed by three to four members and Cluster0_1 for all household composed by five to six members.

- Within each cluster agents were divided into eight sex-age categories (four for women and four for men). The individual distributions were then calculated for each sex-age category as the inverse cumulative distribution functions (ICDFs).

- As the number of household members falling in a specific sex-age category is composed of discrete units, a piecewise linear segmentation was used to implement the ICDFs. Five segments were chosen, as this captured all the variability in the sample.

- To see a short example of the step by step implementation of the ICDF go to the Worksheet *"SurveyData"* in *Population.xls* for *TUT*_lottery*

2. Generating the assets composition of the population

- First the sample was divided into the same two clusters used for the generation of the demographic composition of the population.

- As most assets are expressed in discrete units, a piecewise linear segmentation was also used to implement the ICDFs. Five segments were used, as this captured most resource levels.

3 Further reading and references

In addition to those extensions covered here, a number of additional options are available within MPMAS, including agent-learning, consumption behavior, adaptation, land-use change, coupled hydrological models, alternative methods for generating agent populations, additional products and outputs, input use, and markets, among others (see the overview paper of Schreinemachers and Berger (2011)). More information can also be found on the MPMAS website (http://mp-mas.uni-hohenheim.de) and in publications by the MPMAS developer team.

In addition, a number of country-specific technical documentations is available for download:

Germany: Troost, C. 2014. MPMAS Central Swabian Jura (Version 3.1) – Model Documentation. Includes information on: Agent expectations and learning; biogas production; additional livestock output (manure); policy interventions; farm succession; land markets; agent populations

Uganda: Latynskiy, E. 2014. Agent-based simulation modeling for analysis and support of rural producers organizations in agriculture.
Includes information on: Producers' organizations and institutions; consumption preferences (food and non-food); remittances
Schreinemachers, P. 2006. The (Ir)relevance of the Crop Yield Gap Concept to Food Security in Developing Countries With an Application of Multi Agent Modeling to Farming Systems in Uganda.
Includes information on: Three-step decision process; three-step budgeting process; use of an Almost-Ideal Demand System

Ethiopia: Berger, T., S. Gbegbelegbe, E. Latynskiy, W. McClain, K. Tesfaye, C. Troost, and T. Wossen. 2014. Adaptation of farm-households to increasing climate variability in Ethiopia: Simulation with MPMAS.
Includes information on: Coping strategies; measuring food security; farmer adaptation strategies

Thailand: Schreinemachers, P., A. Sirjinda, C. Potchanasin, T. Berger and S. Praneetvatakul. 2009. An agent-based land use model of the Mae Sa watershed area, Thailand. Detailed model documentation. MPMAS Version 2.0.
Includes information on: Pesticide use, soil loss from erosion

Additional information, applications and technical issues can be found in publications by the MPMAS developer team:

Berger, T., Troost, C., Wossen, T., Latynskiy, E., Tesfaye, K., Gbegbelegbe, S. Can smallholder farmers adapt to climate variability, and how effective are policy interventions? Agent-based simulation results for Ethiopia. Agricultural Economics (online first), doi: 10.1111/agec.12367

Wossen, T., Berger, T., Haile, M., Troost, C. Impacts of Climate Variability and Food Price Volatility on Household Income and Food Security of Farm Households in East and West Africa. Agricultural Systems (online first), doi 10.1016/j.agsy.2017.02.006)

Latynskiy, E., Berger, T., 2017. Assessing the income effects of group certification for smallholder coffee farmers: Agent-based simulation in Uganda. Journal of Agricultural Economics (online first), doi: 10.1111/1477-9552.12212

Carauta, M., Latynskiy, E., Mössinger, J., Gil, J., Libera, A., Hampf, A., Monteiro, L., Siebold, M., Berger, T., 2017. Can preferential credit programs speed up the adoption of low-carbon agricultural systems in Mato Grosso, Brazil? Results from bioeconomic microsimulation. Regional Environmental Change (online first), doi: 10.1007/s10113-017-1104-x

Grovermann, C., Schreinemachers, P., Riwthong, S., Berger, T, 2017. ‘Smart’ policies to reduce pesticide use and avoid income trade-offs: An agent-based model applied to Thai agriculture. Ecological Economics 132, 91-103

Latynskiy, E., Berger, T., 2016. Networks of rural producer organizations in Uganda: What can be done to make them work better? World Development 78, 572-586

Bannwarth, M., Grovermann, C., Schreinemachers, P., Ingwersen, J., Lamers, M., Berger, T., Streck, T., 2016. Non-hazardous pesticide concentrations in surface waters: An integrated approach simulating application thresholds and resulting farm income effects. Journal of Environmental Management 165, 298-312

Troost, C., Berger, T., 2015. Dealing with Uncertainty in Agent-Based Simulation: Farm-Level Modeling of Adaptation to Climate Change in Southwest Germany. American Journal of Agricultural Economics. doi: 10.1093/ajae/aau076

Troost, C., Walter, T., Berger, T., 2015. Climate, energy and environmental policies in agriculture: Simulating likely farmer responses in Southwest Germany. Land Use Policy 46, 50-64

Arnold, T., Troost, C., Berger, T., 2015. Quantifying the economic importance of irrigation water reuse in a Chilean watershed using an integrated agent-based model. Water Resources Research 51, 648-668

Wossen, T., Berger, T., 2015. Climate Variability, Food Security and Poverty: Agent-Based Assessment of Policy Options for Farm Households in Northern Ghana. Environmental Science & Policy 47, 95-107

Quang, D.V., Schreinemachers, P., Berger, T., 2014. Ex-ante assessment of soil conservation methods in the uplands of Vietnam: An agent-based modeling approach. Agricultural Systems 123, 108–119.

Berger, T., Troost, C., 2014. Agent-based Modeling of Climate Adaptation and Mitigation Options in Agriculture. Journal of Agricultural Economics, 65, 323-348.

Wossen, T, Berger, T., Swamikannuh, N., Ramilan, T., 2014. Climate variability, consumption risk and poverty in semi-arid Northern Ghana: Adaptation options for poor farm households. Environmental Development 12, 2-15

Latynskiy, E., Berger, T., Troost, C., 2014. Assessment of Policies for Low-Carbon Agriculture by means of Multi-Agent Simulation. In: Ames, D.P., Quinn, N.W.T., Rizzoli, A.E. (Eds.), Proceedings of the 7th International Congress on Environmental Modeling and Software, June 15-19, San Diego, California, USA.

Marohn, C., Schreinemachers, P., Quang, D.V., Berger, T., Siripalangkanont, P., Nguyen, T.T., Cadisch, G., 2012. A software coupling approach to assess low-cost soil conservation strategies for highland agriculture in Vietnam. Environmental Modeling & Software 45, 116–128

Troost, C., Calberto, G., Berger, T., Ingwersen, J., Priesack, E., Warrach-Sagi, K., Walter, T., 2012. Agent-based modeling of agricultural adaptation to climate change in a mountainous area of South West Germany. In: Seppelt, R., Voinov, A., Lange, S., Bankamp, D. (Eds.), International Environmental Modeling and Software Society (iEMSs): 2012 International

Congress on Environmental Modeling and Software: Managing Resources of a Limited Planet, Sixth Biennial Meeting, Leipzig, Germany

Schreinemachers, P., Berger, T., 2011. An agent-based simulation model of human environment interactions in agricultural systems. Environmental Modeling & Software 26, 845-859.

Schreinemachers, P., Potchanasin, C., Berger, T., Roygrong, S., 2010. Agent-based modeling for ex ante assessment of tree crop innovations: litchis in northern Thailand. Agricultural Economics 41(6), 519–536.

Schreinemachers, P., Berger, T., Sirijinda, A., Praneetvatakul, S., 2009. The diffusion of green-house agriculture in northern Thailand: Combining econometrics and agent-based modeling. Canadian Journal of Agricultural Economics 57 (4), 513-536.

Schreinemachers, P., Berger, T., Aune, J.B., 2007. Simulating soil fertility and poverty dynamics in Uganda: A bio-economic multi-agent systems approach. Ecological Economics 64(2), 387-401.

Berger, T., Birner, R., Díaz, J., McCarthy, N., Wittmer, H., 2007. Capturing the Complexity of Water Uses and Water Users within a Multi-Agent Framework. Water Resources Management 21(1), 129-148.

Schreinemachers, P., Berger, T., 2006. Land-use decisions in developing countries and their representation in multi-agent systems. Journal of Land Use Science 1(1), 29-44.

Berger, T., Schreinemachers, P., 2006. Creating agents and landscapes for multi agent systems from random samples. Ecology and Society 11(2), Art.19.

Berger, T., Schreinemachers, P., Woelcke, J., 2006. Multi-agent simulation for the targeting of development policies in less-favored areas. Agricultural Systems 88, 28-43.

Berger, T., 2001. Agent-based models applied to agriculture: a simulation tool for technology diffusion, resource use changes and policy analysis. Agricultural Economics 25(2/3), 245-260.