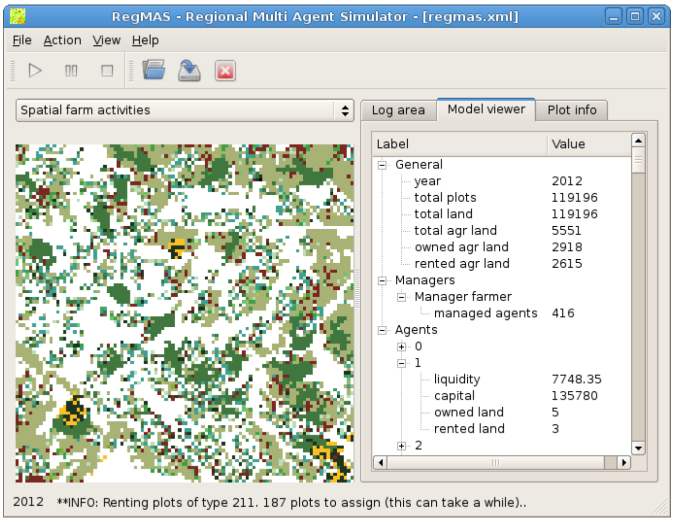
RegMAS User Manual

Version 1.3.0

Antonello Lobianco *†

September 30, 2008



Abstract

RegMAS (Regional Multi Agent Simulator) is a spatially explicit multi-agent model framework, developed in C++ language and designed for long-term simulations of government policies effects over agricultural systems (e.g. farm sizes, incomes, land use).

RegMAS uses a profit-maximisation algorithm to derive farmers behaviours. Using RegMAS researchers can write regional-specific models, adapting the pool of activities and resources to their specific region.

RegMAS is *Free Software*, that is, its source code is fully available and modifiable, and it runs on several operating systems.

This paper is designed to introduce RegMAS and to assist researchers in using it for their own simulations.

*Department of Economics, Marche Polytechnic University.

†I want to thanks Roberto Esposti for his precious directions, Franco Sotte and the Department Of Economics at UNIVPM for funding this project and Alfons Ballmann, Kathrin Happe and the whole IAMO team (Konrad, Christoph, Martin..) for their AgriPoliS model from which RegMAS has borrowed many concepts and their support in my studies of agent-based modelling.

Contents

1 Introduction

While many and heterogeneous models exist nowadays to assess the impact of government policies or market rules over agricultural systems, very few (if any) address the basic concerns of the academic community of transparency and reproducibility of experiments. Instead, most times the reader must be satisfied with validation experiments conducted by the researchers themselves.

To address this needs, we chose to develop our own model as *Free Software*, that is, its source code is publicly available and can be freely used and modified.

RegMAS (Regional Multi Agent Simulator) is a spatially explicit multi-agent model framework, developed in C++ language designed for long-term simulations of effects of government policies on agricultural systems (farm sizes, incomes, land use..).

More specifically, RegMAS conceives rural social systems (and in particular agricultural ones) as complex evolving systems, made of an heterogeneous set of individual acting “agents” (that is, farmers).

The researcher typically set the initial conditions where the farmers operate and “observe” their behaviours in such environment. Initial conditions can include pre-defined shocks along the temporal dimension, typically used to model changes in the policy or macro-economic variables (e.g. decoupling options, activities gross margins..).

This document will help you discover RegMAS and hopefully will make your life easier while using it for your own simulation exercises.

It is structured as follow.

Section 2 briefly describes the history and the methodology behind RegMAS, referring for details on several papers that discuss multi-agent models applied on agricultural systems.

Section 3 will guides users from obtaining the program with associated sample data trough having it working on their pc. It covers a typical MS Windows installation, while the compilation of RegMAS directly from the source code is covered in a corresponding section on its web-site wiki.

Once the program is installed, sections 4 to 7 will gradually introduces in the usage of RegMAS. Section 4 presents the program user's interface and the input files, allowing to run a complete RegMAS experiment using the predefined scenarios over the sample data. Users interested in creating their own scenario should consult section 5. A further point that most RegMAS users will likely be interested is how to apply the model over their own case-study region. This requires obtaining farm-specific and regional-level data (including spatial one) and it is covered on section 6.

Finally some researcher may want to personalise RegMAS, maybe changing the farmers behaviours or adding new details to the output. As previously noted, source code of RegMAS is fully available and users are encouraged to modify it for their own modelling needs. Section 7 gives a brief tour of the RegMAS Application Programming Interface (API), presents the tools used to develop RegMAS and point on the specific place where to get more help.

Once a user has became familiar on creating new scenarios or models, it will be time to analyse the results. Sections 8 and 9 respectively describe the kind of results that are saved by RegMAS and how to export them into a GIS for further analysis (the latter task is optional but recommended).

This manual ends with an Appendix covering the sources where further help in using RegMAS may be available and the license terms.

Have fun with RegMAS !

2 The model

2.1 Overview

Spatial explicit Agent-Based Models (ABM) within the specific agricultural context was pioneered by Balmann (1997) with the Agricultural Policy Simulator (AgriPoliS) model.

ABM have the benefit of catching the fundamental behaviour at the micro-level of the individuals farms, without the need of aggregating them in "representative" agents. Maybe even more important, ABM can catch the iterations of the heterogeneous farms when they deal with competition over common finite resources, e.g. land.

Boero (2006) and Parker (2003) have review several other ABM involving land use changes in various scientific areas, including agricultural economics, natural resource management, archaeology and urban simulation, but this section will briefly describes AgriPoliS as RegMAS borrows many concepts from it, *in primis* the utilisation of a profit-maximisation algorithm to derive farmers behaviours.

AgriPoliS allows to model heterogeneous farms behaviours under various external situations (typically, under different policy scenarios) and observe regional results by aggregating these micro-level behaviours.

In AgriPoliS agents are mainly farmers[1]. They have their own goals; in AgriPoliS, the farmer's objective is the maximisation of household income. To achieve this objective, farmers solve a Mixed Integer Programming (MIP) problem that, in some aspects, is specific to each farmer. Outside the linear programming problem, they can also decide to rent other agricultural plots or to release rented land.

Using a mixed integer linear programming approach to simulate each agent behaviour on one hand is very flexible, as it can cover the whole range of farm activities, from growing specific crops to investing in new machinery or hiring new labour units. Furthermore, it is simple to add new regional-specific activities.

On the other hand, however, linear programming techniques require a long calibration phase to assure a balanced choice of farm activities, avoiding unrealistic outcomes [2].

[1]Other agents in the model perform some specific tasks, e.g. managing land or coordinating product markets.

[2]RegMAS introduces a sub-region mode to help researchers to roughly calibrate their model before running a real (and slower) simulation.

Any farmer in the model is a real farmer whose data are taken from farm-level datasets (in Europe, FADN) and explicitly associated to a spatial location. Due to privacy-protection regulations, however, we don't have access to the real farm localisation. Therefore, we have to distribute farms randomly in the virtual region. Space (i.e. location) is important in the model because it influences transport costs and indirectly makes the farmers interact each other, e.g. by competing for the same land plots.

AgriPolis, as it takes into account many aspects of a real farm, is a very complex model, with lot of code dedicated to cover specific aspects (e.g. quotas, generational changes, multi-years investments). A detailed description of AgriPolis can be found in Happe et al. (2004) or in Kellermann et al. (2007). While Happe et al. (2004) focus is on the methodological advantage of using ABM in agriculture as compared with other instruments as partial and general equilibrium models on one side and individual farm-level models on the other, Kellermann et al. (2007) details the latest implementation of AgriPoliS (2.0). In addiction to this two papers, Sahrbacher et al. (2005) describes AgriPoliS implementation over several case-study regions and Lobianco (2007) presents an adaptation of AgriPoliS for the Mediterranean regions, further adding some general background on agent-based modelling and to its motivations.

As AgriPoliS, RegMAS is spatially explicit, a characteristic that can not be neglected when modelling the agricultural sector. For example the spatial heterogeneity allows the model to associate on each plot a different rental price and investigate possible land abandonment phenomenas even when the land is *on average* profitable.

Differently from AgriPoliS, the spatial dimension is initialized from real land-use data, using satellite information, and plots are explicitly modeled in the decision matrix as individual resources.

As a further distinction, RegMAS has been designed from the ground-up to explicitly consider farmers as one type of several possible type of agents. In RegMAS farmers have sensitivity of the overall environment, including extra-agricultural variables. On a technical point, "farmer" agents in RegMAS derive from a more general type of "spatial" agents that in turn derive from a "base" type. Each agent type has its own "manager" agent that dialogue with a "Super Agent Manager". The formers are a sort of interface "agent side" while the latter implements the same interface on the program core side. In this way the model core doesn't need to know anything about agents internal logic. While this approach allows for rapid development of different agent types (only specific characteristics need to be modelled) at current RegMAS development stage only farmer agents are fully implemented.

2.2 Farmer agents behaviour

Farmers autonomously make their decisions solving a Mixed Integer Programming (MIP) problem similar to those shown in Figure 1. This happens any time farmers bid for renting a land plot in order to calculate its shadow price, or plan for new investments, or finally produce using the given assets. Symbol  in Figure 2 denotes a step in the model when one or more MIP problems have to be computed at the farm level.

From FADN data we can establish the initial farm's endowment: financial assets, availability of land, machinery, animals and so on. From a linear programming point of view, these data represent the right terms of the constrain equations (A in the cited figure). Any farmer choose from a list of activity options. They can be divided in two categories: activities that can be run entirely within one year (B) and activities that generate results over multiple years (investments, C). Investments are bounded to be integer and the same investment type is available in different size-options, allowing scale-effects to emerge in the model. Section 2.2.2 details the farmers finance modelling, including how investments enter the MIP together with yearly activities.

Individual activities to include in the model are left to the RegMAS users. Currently we implemented models where, very synthetically, all aspects of running a farm are considered, including financial and labour ones. To solve this problem the farmers chose the quantities (D) of the various activities that maximise the objective function (E). In our case, this is the maximisation of household income, and the gross margins of the various activities are the parameters of the objective functions(F).

Before entering the MIP, individual activies are "filtered" by the farmers. Currently, this is implemented in order to change the gross margins to include transport costs from the farmstead to the plot and to take into account the plot's altitude (under the hypothesis that mountain plots are less productive than plain ones). Finally, The matrix of the constraint coefficients (G) links the available activities (B+C) with their technical requirements (H).

RegMAS can take into account changes of resource endowment and activity gross margins, generated either endogenously to the MIP core, in case these changes occur as a consequence of the solving procedure (e.g., an investment improves the number of available activities) or exogenously to it, in case these changes occur in other parts of the model (e.g., renting/releasing land, or as a consequence of market prices changes).

Paris (1991); Arfini (2000) present respectively an in-deep analytical description and a literature review of linear programming techniques applied to farm problems.

2.2.1 The space dimension in the optimisation problem

As result of the spatial explicitness the farmer maximisation problem is composed of the following activities:

- i spacial explicit activities * p_i suitable plots;
- j spacial explicit investments * p_j suitable plots;
- q other spacial implicit activities;

Figure 1: Agent's Mixed Integer Programming Problem (excerpt)

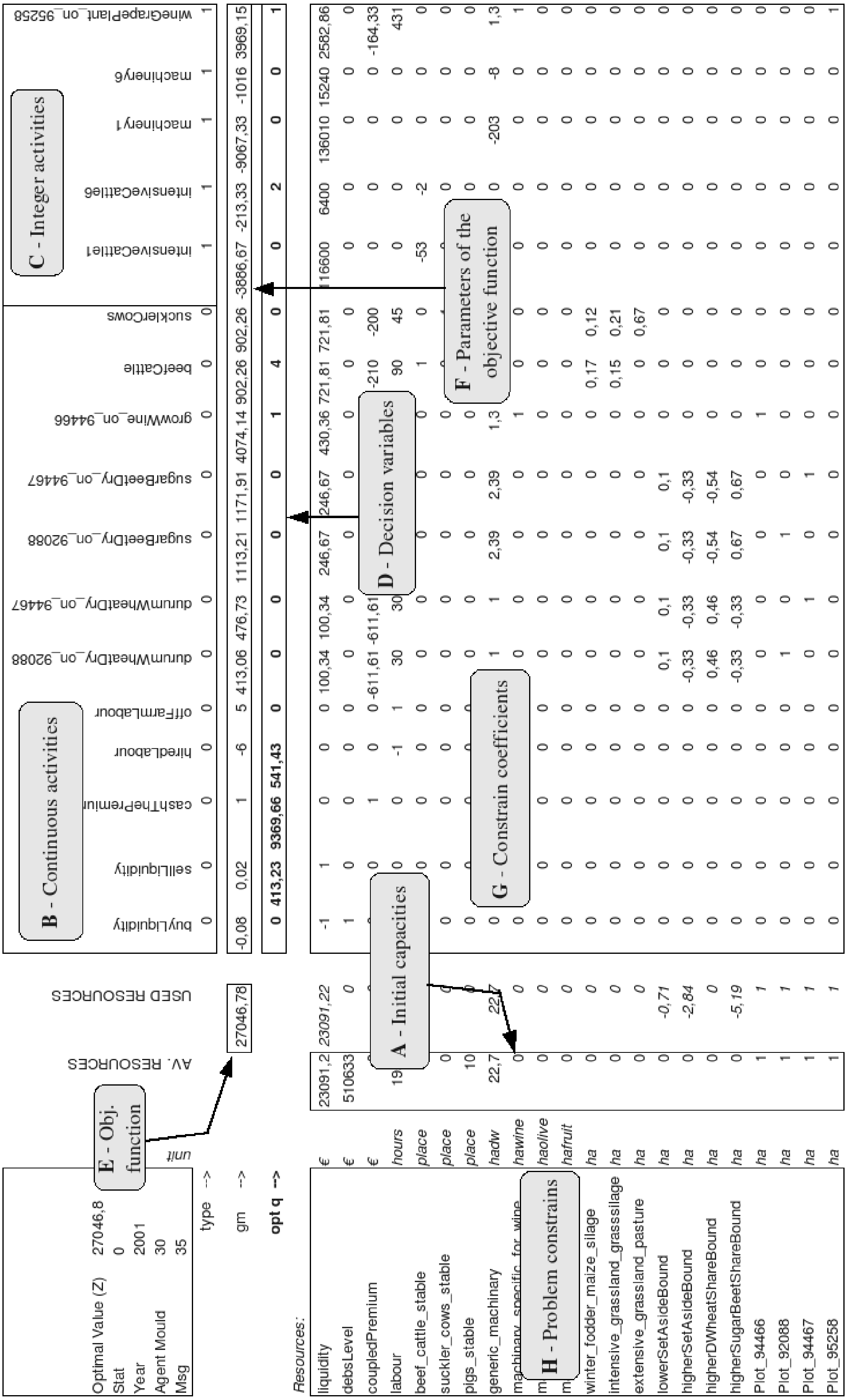
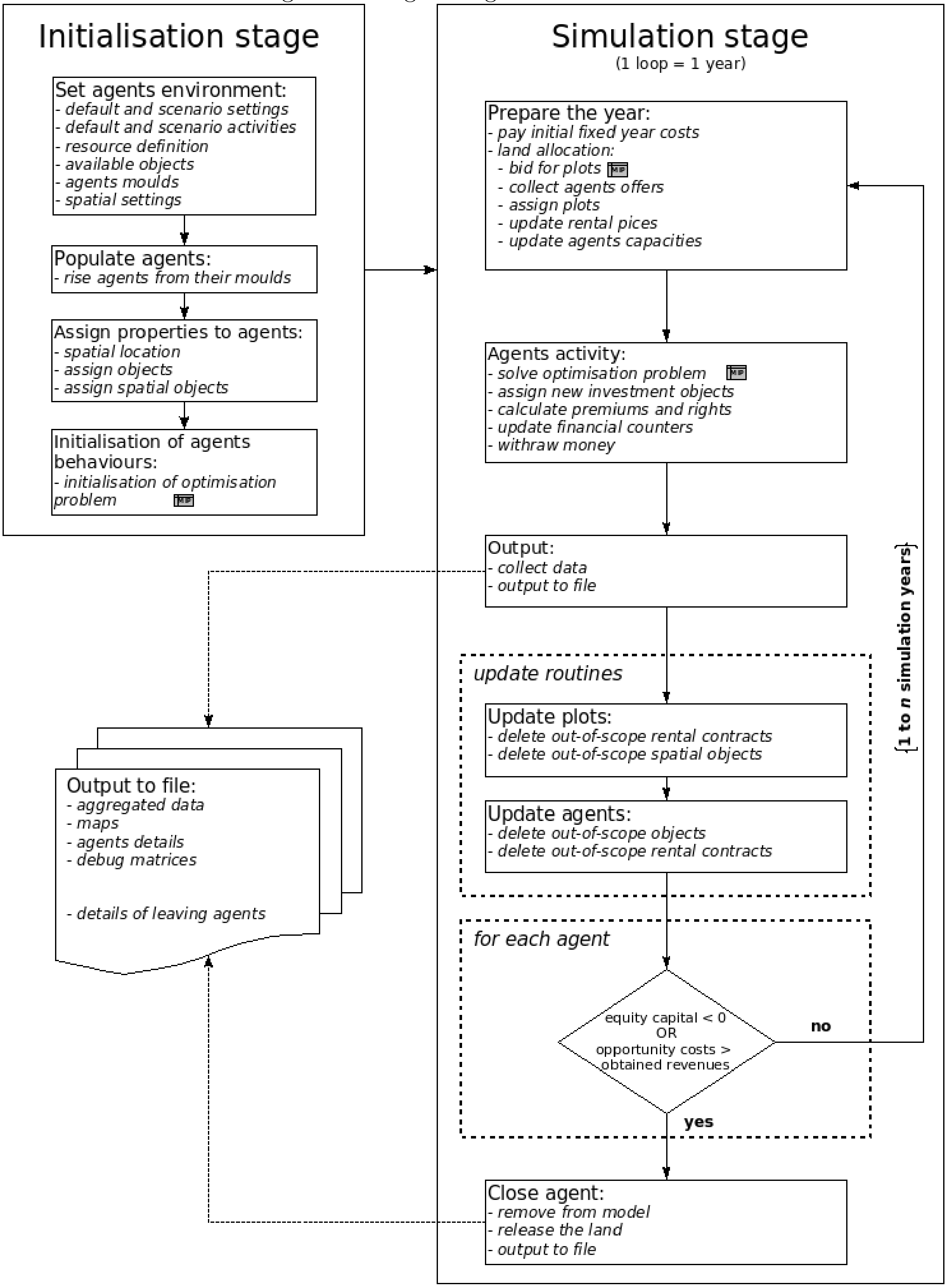


Figure 2: RegMAS general Flowchart



- z other spacial implicit investments.

Is the researcher task to select which characteristics need to have a plot to be considered for each spatial explicit activity. Within the input data, he/she can select which land types and even which spatial objects need to be already present for each activity. To make an example, growing wine requires land type **221** (vineyards) and the presence of a **wineGrapePlant** object. All *activity_x_plot* combination that miss this requirements are automatically discarded even before entering the MIP, reducing the MIP dimension and therefore the required computational power [3].

Even with this trick, the spacial explicitness has the disadvantage of a much larger matrix, especially in case of farmers with many plots, but it has also many advantages:

- transport costs are associated directly with individual pixels and already included in the Gross Margin;
- pixel heterogeneity (e.g. different slopes) can be included straightforward, just changing the gross margins (e.g. through the "filter" function cited in the previous section);
- an explicit map of the region can be draw to show the different productions;
- binds on individual plots can be easily inserted on the MIP problem.

2.2.2 Financial aspects

In RegMAS models investments require liquidity that can be obtained using open-credit line that can grow up to a model-fixed share of the capital (e.g. 80

Each year the farmer optimize the quantity of money to ask to the bank. There isn't an end-term for farmers to give back the money as the "bound" is rather as the debit share of the capital.

Differently from AgriPoliS there is no difference from short-term loans and long-terms one. This is because loans are completely decoupled from investments. On the other side, there is no needs to assume a constant share of investment covered by the loan: the individual farmers are free to implicitly optimize the share of the investments covered by loans depending on their financial situation.

Liquidity is calculated as follow:

$$\begin{aligned} liquidity_t &= liquidity_{t-1} + profits_{t-1} \\ &\quad -withdraws_{t-1} - \sum_{n=0}^{N} invCosts_{t-1,n} - sunkCosts_t \qquad (1) \end{aligned}$$

To calulate the liquidity available to farmers on a specific year we sum to the liquidity available on the previous year all the revenues (profits) and costs

[3] The Reference Manual has a pseudo-code that details the steps the model does to add activities to the MIP problem, available at `http://regmas.org/doc/referenceManual/html/classOpt.html`.

(withdraws and new investments) occurred in the previous year and we detract the sunk costs farmers need to pay before producing (these are costs generated from prevoius choices, like multi-year rental costs or investment maintenance costs).

Withdraws are the money required by the farmers to support their own private life. They are calculated as a fixed portion of the profits plus a minimum level that depends from the size of the farm (misured in family annual work units):

$$\begin{aligned} withdraws &= perCapMinWithdrawal * AWU \\ &\quad + max(0, profits * withdrawalProfitShare) \end{aligned} \tag{2}$$

If the liquidy can be tought as a buffer, the debit level shoud be considered as a threshold, expressed as the share of the whole capital that farmers can't overpass.

The whole capital is in turn calculated as the sum of the liquidity, the investments current value and the land capital:

$$capital_t = liquidity_t + \sum_{i=0}^{I} investmentCurrentValue_{i,t} + landCapital \tag{3}$$

with I is the number of owned investments. The current value of investment objects depends from the kind of investment itself: e.g. in perennial crops the value first grows up and then decreases, in stables instead it linearly decreases. *landCapital* is, at least now, fixed and read from the farmers' data file.

2.2.3 MIP solver

In RegMAS MIP problems have to be computed for each individual farm and in several steps during each simulated period, resulting in levels of thousands computations for period. It follows that the speed of the solving algorithm become a critical factor. Due the fact that in RegMAS plots enter the problem individually, matrices can become quite large, however they are very sparse allowing specialised software to solve the problems in terms of fractions of second.

In fact, RegMAS use external libraries to solve this problems. RegMAS class **Opt()** is responsible to establish the direction of the objective function (in our case, a maximisation), the set of bounds, objective coefficients and constrain coefficients. At this point the problem "object" is solved calling an external Dynamically Linked Library (DLL).

RegMAS uses the open-source GNU Linear Programming Kit (GLPK) (Makhorin, 2007) that employs a two-phase revised Simplex method (that is guaranteed to find the optimal solution, if one exist) to retrieve continuous solutions, and then apply a Branch & Bound method in case of an integer optimisation.

GLPK recently added an interior-point algorithm, but we found it to be still too unstable at this time.

2.3 Land allocation and land market

An obvious problem when dealing with spatially explicit agent based models is the localisation of the agents and of their spatial objects. The problem is complicated by the fact that there is already an informative layer, consisting of the land use, and we have to keep this layer consistent with the model, applying the farm allocation over it.

Firstly, farms are assigned a random location picking up a plot compatible with their assets, starting from the less common. The idea is that rare land use types have the precedence over common land use types to minimize distances from them to the farmsteads. So if a farm has both fruit land and arable land, the farmstead will be placed within a fruit land type.

Subsequently, plots are assigned to the closest farm that has still an uncompleted capacity for that specific soil type, giving precedence to owner plots in comparison to rented ones.

This is not a optimization algorithm as plots are not assigned to farms in a way that minimize the total plots-farmsteads distance. But on the other hand also the real world situation is far from an optimal land (de)fragmentation, as physical bounds and hereditary rules often split the farmer land in various disconnected plots. Figure 3 shows the result of the initial land allocation, with the plots assigned to their owner farm (each colour represent a different farm).

Figure 3: Owned plots after initial land allocation.

During the simulation farmers can bid to rent new plots. Currently (as in AgriPoliS) RegMAS doesn't allow for land transfer nor for direct *farmer-to-farmer* renting contracts. Instead, farmers can only rent land owned by an anonymous agent that collect the land arising from farms leaving the model and from the initial pool of rented plots to make it available in a bid to the farmer offering the highest price.

Farmers asked to bid offer a share of their shadow price for such plot, to take into account of fixed and variable negotiation costs and overheads. The shadow price for the new plot is calculated simply performing two MIP problem optimisation, with and without the plot, and calculating the difference.

While AgriPoliS can use some optimisation techniques as land is homogeneous within the same soil type, the full heterogeneity of plots in RegMAS prevent using such tricks, making this process very computationally intensive. Therefore RegMAS offer the option to limit the bidding process to farmers within a certain range from the plot. The natural trade-off when increasing this option is on one side that it increases the likelyhood to get the highest bid, but on the other side that it dramatically increases the computational time.

Once the plot is assigned to a farm a rental contract is established for a random period (within user defined limits) and the plot enter the farmer's MIP problem as a new resource.

3 Installing RegMAS

3.1 System requirements

While RegMAS doesn't have many software requirements (it currently run on any PC with MS Windows or Linux [4]), its hardware requirements are of much more constrain.

It will make a severe trial of a pc's CPU and RAM. Exactly how much RAM it needs will depends on the size of the region. For very tiny regions (let's say 50x50 pixels) 256 or 384 MB may be enough, but for a larger region (e.g. 300x300 pixels) we recommend at least 2 GB of RAM. Even in such case a 15 years simulation will last a few hours.

3.2 Getting the software

RegMAS can be download in two forms: as a Windows installer or as source-code that requires compilation before it can be used.

The latter form is suggested for users that want to adapt the model to their own needs, or that use Linux (our development environment, for which an installer is not available) or yet that need the very latest version. In such cases the source code can be downloaded directly from our Concurrent Version System (CVS) and compiled, as directed from an apposite page on the documentation wiki.

In all other cases the suggested way is to download the installer[5] and keep reading this section.

A local installation of OpenOffice[6], a free and multi platform MS Office clone suit, is required to edit the input data. In fact, while OpenOffice is not strictly needed lo load the data, from RegMAS version 1.2.0 onward input data is read directly from an OpenDocument Spreadsheet file that can be handled with OpenOffice Calc.

[4] As RegMAS is written in standard C++ and its dependent libraries are know to work with Macintosh, a Mac port should be easy, ideally just a compilation. However we don't have any experience on such platform and therefore we can not offer assistance on such platform.

[5] `http://www.regmas.org/download`

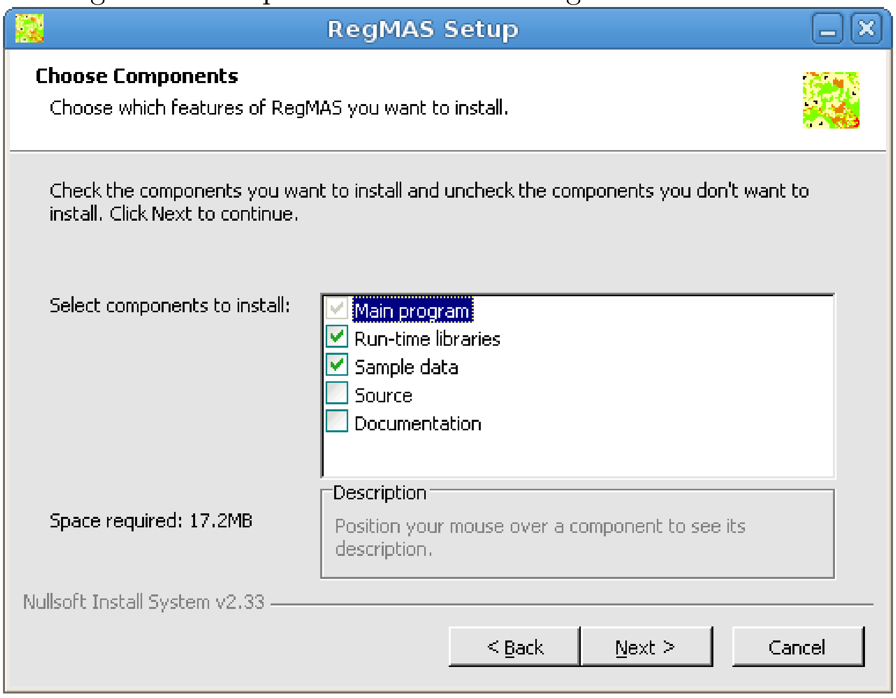
[6] `http://www.openoffice.org`

Users can refer to the OpenOffice web-site for directions on how to download and install OpenOffice.

3.3 Installing the software

The Windows installation wizard let users decide witch components of the software should be installed (Fig. 4). Aside to the main program, also the run-time libraries should be installed, as they are required to run RegMAS. Sample data

Figure 4: Component selection during installation wizard



are suggested in order to test the software before building a different model.

Finally, users may want to include the source code and a local copy of the documentation. While unnecessary to run the model, the model code is very useful in order to deeply understand the model behaviour.

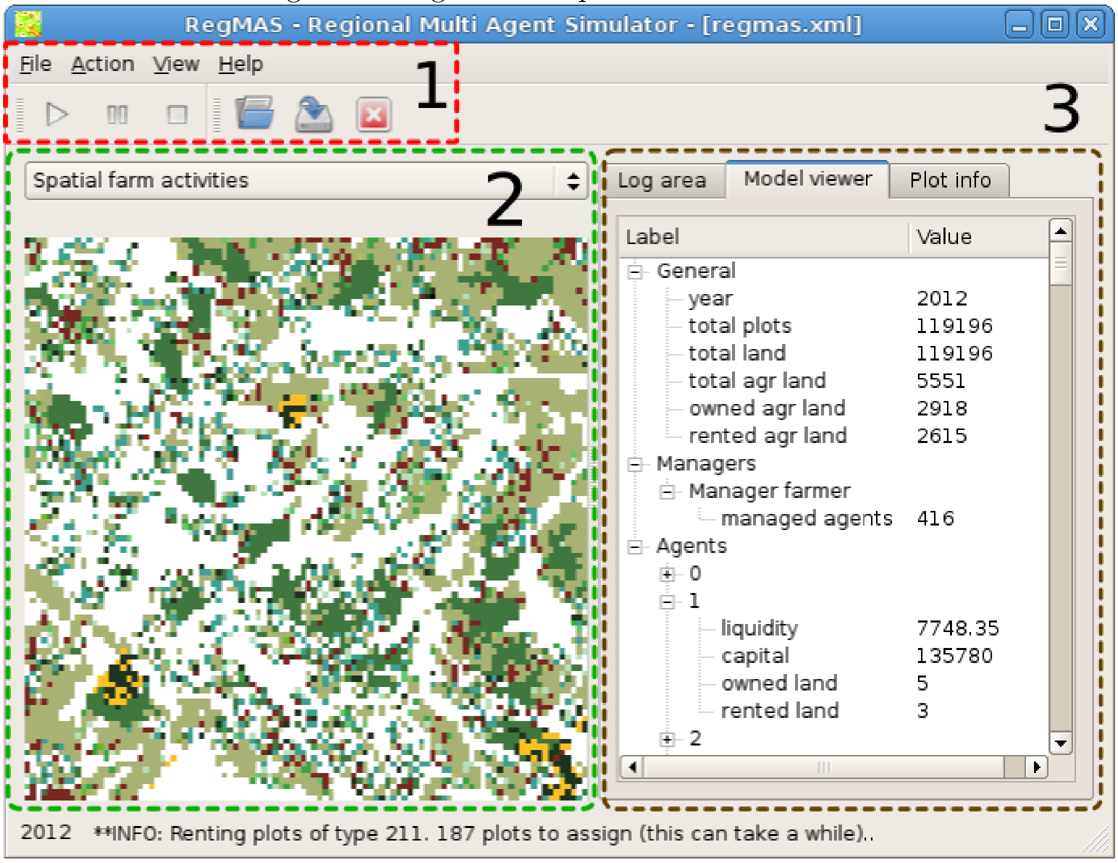
4 Running default scenarios

Once the installation is completed RegMAS can be started from its shortcut under *StartMenu* → *Programs* → *RegMAS* → *Version* or clicking directly on regmas.exe using Explorer.

4.1 The Graphical User Interface

Figure 5 shows RegMAS Graphical User Interface (GUI). It is logically divided into 3 areas.

Figure 5: RegMAS Graphical User Interface



The first area includes the COMMANDS, divided into the *action* toolbar and the *file* toolbar. From left to right, they allow to *run* the model, to *pause* it, to permanently *stop* the model, to *load* a new model, to *save* a model log and finally to *exit* the program.

Even if we checked the code very accurately, it is not advisable to run several simulations consecutively. This because memory leaks are likely to arise and in the worst scenario to interfere with the model. Instead, users can close the program and re-open it at each newer simulation.

The second area displays the MAPS (spatial layers) that are loaded or drawn by the model. They are upgraded at least once for each simulation year, and are the quickest way to see what it is happening on the model during run-time. Layers can be switched using the drop-down menu at their top and can be zoomed or scrolled using the mouse. Maps can always set back to fit the window using the *Fit map in Window* command from the *View* menu.

The third area is a multi-page QUANTITATIVE VIEWER. At present, three pages (tabs) are available. The first page displays the MODEL LOG, and its content can be saved as a text file during simulation or at its end. The central tab acts as a MODEL STATUS VIEWER. Again, it helps to check the status of the model while still running, this time more quantitatively. It provides information on the general region, on each agents-manager (the individual agent that ”manage“ all agents of its type) and, if the region is small, on each individual agents[7]. Finally, the third tab displays a cross-layer PIXEL INFO AREA. It is activated right-clicking over a plot and it displays its value across all the layers, the current one being highlighted.

4.2 Model input files

Each model is composed of a OpenDocument Spreadsheet file containing several tables and auxiliary files placed by default under the DATA directory where RegMAS has been installed:

activities Includes the activities that each agent may activates;

resources Has details over the available resources;

settings Provides regional-wide settings;

***scenarioName*_activities** Modify the default activities according to the specific scenario, e.g. including policy variables or changing their gross margin;

***scenarioName*_settings** Overrides the default setting to change them according to the relevant scenario;

agenttype Lists the agents *moulds* with their details;

gis\gis.xml *(file)* Lists the layers that need to be loaded on start-up with their legend;

gis*regionNamelayerName*.grd** *(file)* Includes the data for a specific layer, in ArcInfo or GRASS ASCII grid format.

4.3 Loading and running the model

To run a specific model users need to press the *load* button and browse for its main file. Once a model is loaded the three buttons *run*, *pause* and *stop* can be used to control its execution.

As a ”normal“ simulation requires hours, a special mode (option **subRegionMode** in the setting sheet) allows to run the simulation only on a smaller subregion. In this way it is possible to make rough calibrations of the model much more quickly.

When *run* is pressed a pop-up appeares to allow for the selection of the scenario.

From version 1.1.0, RegMAS can also run in console mode (that is, without any Graphical User Interface), e.g. under a remote cluster, using the option `-c` (use `-h` for a list of other options available in console mode).

[7]As upgrading each agent to the GUI is very time-consuming, we chose to display agent data only if the region is small. However models can easily override this setting.

5 Creating new scenarios

RegMAS allows for the creation of scenarios that differs from the main model for a limited set of inputs. Variables that are not explicitly redefined in the scenario remains at their defualt values.

To create a scenario, firstly its name and a brief description should be defined under sheet **scenarios** (we recommend to not use spaces nor unusual characters for its name). To specify the characteristics of the new scenario two new sheets should be inserted in the file, namely ***{scenarioName}*_settings** and ***{scenarioName}*_activities**.

***{scenarioName}*_settings** overrides variables in sheet settings, but it doesn't allow to introduce new variables. Most variables are just scalar while some are vectors (you can define up to 100 values each). For some of these latter ones (e.g. *forceFarmingForCashPremium*) the vector is interpreted in the model as a temporal dimension, e.g. the second value is the value of the second year from when the simulation started. As a convenience, on these variables that are clearly labeled as such in the datafile, users can input only the first year as a short way to write the same value on all simulation years.

***{scenarioName}*_activities** lets change the characteristics of each individual activity in the model from the default scenario, specifying on which year the change should be applied. There are three parts of an activity that could be changed: the gross margin, the decoupling rate and the matrix coefficients.

Changing the gross margin it is possible to simulate changes in product prices or costs.

The decoupling rates lets specify for each activity/year if the corresponding coupled premium should be "recorded" for future right assignment calculation ("-1"), if nothing should be done about decoupling ("0") of if previously recorded rights should be gave back to farmers as decoupled premium and in which percentage. RegMAS is able to store the rights of each individual farmer, allowing a very precise modelling of decoupling mechanism.

Results in the output are clearly labeled with the running scenario, allowing easy confrontation between them.

6 Applying RegMAS to a new Region

While RegMAS ships with a default region that can be used for simulation purposes, most likely users want to run simulations over their own case-study regions.

In order to be modelled with RegMAS a region must be available with the following set of data:

Quantitative regional data Aggregated data of the region, normally available from the Census.

Individual farmer detailed data Individual farmers are used in the model as "bricks" to build a simulation region and the crucial information here become the individual farms production factors (capitals, humans, land). In order to obtain satisfactory congruence between the real and the simulated region, a basket in the magnitude of tens of farmers data is often necessary[8].

Technical coefficients and prices Technical coefficients are needed to link the activities pool with the resource pool, while prices are needed for both products and factors

Land use map As RegMAS is fully spatial explicit, it requires a detailed map of land uses (In Europe this is available from the Corine Land Cover project[9]).

The basic idea is to reproduce in the model a simulation region composed only of "typical" farms, but of which aggregate values are as close as possible to the real one.

To each farm in the region of which detailed data is available (e.g. because the farm is member of the FADN network) is associated a coefficient. A 0-coefficient means that the farm is not selected, while a non-0 coefficient implies that the farm becomes one of the typical farms of our simulation region. The key point is to find these scaling coefficients that minimise the difference between the simulation region and the real one (Eq. 4).

$$\min \sum_{k=1}^{K} \left(\frac{\sum_{n=1}^{N} (FADN_{n,k} * UC_n)}{REGIO_k} - 1 \right)^2 \quad sub \quad UC_n \geqslant 0 \quad \forall n \tag{4}$$

Where:

Indeces:	Variables:
$n = \{1...N\}$ Individual farms	$FADN_{n,k}$ FADN data
$k = \{1...K\}$ Characteristics	$REGIO_k$ Regional aggregated data
	UC_n "upscaling" coefficient

This procedure is called "upscaling" and it is well documented in Kellermann et al. (2007), while a practical implementation is discussed in Sahrbacher et al. (2005)[10].

The upscaling can be conveniently obtained using the quadratic solver in Excel, like shown on Fig 6 [11]

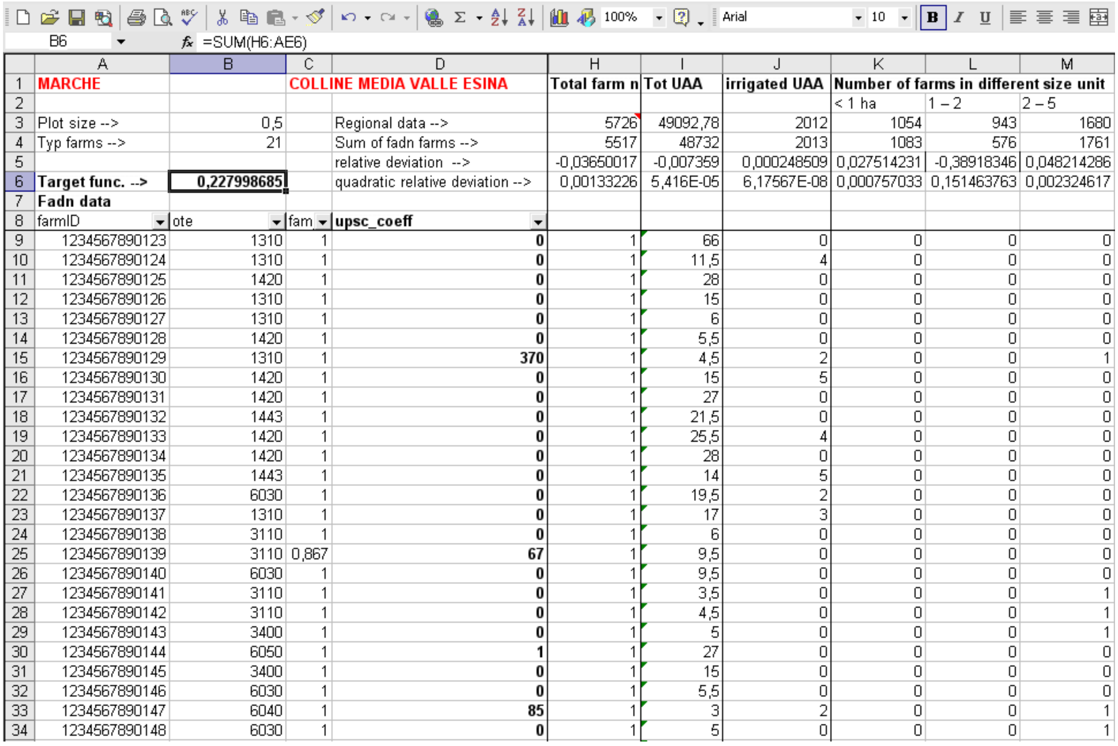
[8] The exact number depends on three parameters: *(1)* the number of elements that should be compared between the real and the simulated regions, *(2)* how good the typical farms reflect the total of the farms in that region and *(3)* the statistical discrepancy that the user is willing to accept.

[9] `http://dataservice.eea.europa.eu/dataservice/metadetails.asp?id=950`

[10] Both paper refer to the preparation of a simulation region for AgriPoliS, but the methodology can be equally applied to RegMAS

[11] A template/example file can be downloaded from the documentation wiki, at `http://www.regmas.org/doc/doku.php?id=model:other:upscaling`

Figure 6: Regional upscaling using Excel



B6 =SUM(H6:AE6)

	A	B	C	D	H	I	J	K	L	M
1	MARCHE		COLLINE MEDIA VALLE ESINA		Total farm n	Tot UAA	irrigated UAA	Number of farms in different size unit		
2								< 1 ha	1 – 2	2 – 5
3	Plot size -->	0,5		Regional data -->	5726	49092,78	2012	1054	943	1680
4	Typ farms -->	21		Sum of fadn farms -->	5517	48732	2013	1083	576	1761
5				relative deviation -->	-0,03650017	-0,007359	0,000248509	0,027514231	-0,38918346	0,048214286
6	**Target func. -->**	**0,227998685**		quadratic relative deviation -->	0,00133226	5,416E-05	6,17567E-08	0,000757033	0,151463763	0,002324617
7	**Fadn data**									
8	farmID	ote	fam	**upsc_coeff**						
9	1234567890123	1310	1	**0**	1	66	0	0	0	0
10	1234567890124	1310	1	**0**	1	11,5	4	0	0	0
11	1234567890125	1420	1	**0**	1	28	0	0	0	0
12	1234567890126	1310	1	**0**	1	15	0	0	0	0
13	1234567890127	1310	1	**0**	1	6	0	0	0	0
14	1234567890128	1420	1	**0**	1	5,5	0	0	0	0
15	1234567890129	1310	1	**370**	1	4,5	2	0	0	1
16	1234567890130	1420	1	**0**	1	15	5	0	0	0
17	1234567890131	1420	1	**0**	1	27	0	0	0	0
18	1234567890132	1443	1	**0**	1	21,5	0	0	0	0
19	1234567890133	1420	1	**0**	1	25,5	4	0	0	0
20	1234567890134	1420	1	**0**	1	28	0	0	0	0
21	1234567890135	1443	1	**0**	1	14	5	0	0	0
22	1234567890136	6030	1	**0**	1	19,5	2	0	0	0
23	1234567890137	1310	1	**0**	1	17	3	0	0	0
24	1234567890138	3110	1	**0**	1	6	0	0	0	0
25	1234567890139	3110	0,867	**67**	1	9,5	0	0	0	0
26	1234567890140	6030	1	**0**	1	9,5	0	0	0	0
27	1234567890141	3110	1	**0**	1	3,5	0	0	0	1
28	1234567890142	3110	1	**0**	1	4,5	0	0	0	1
29	1234567890143	3400	1	**0**	1	5	0	0	0	1
30	1234567890144	6050	1	**1**	1	27	0	0	0	0
31	1234567890145	3400	1	**0**	1	15	0	0	0	0
32	1234567890146	6030	1	**0**	1	5,5	0	0	0	0
33	1234567890147	6040	1	**85**	1	3	2	0	0	1
34	1234567890148	6030	1	**0**	1	5	0	0	0	1

7 Advanced: changing RegMAS structure

If agents should behave in a different way that the default one (profit maximisation), or if the model should work with more agent types, RegMAS needs to be modified to achieve such tasks.

It doesn't yet provide a plug-in architecture, so changes have to be done directly in its source tree. However the design is made in a way to separate as much as possible the core model logic from those of the agents. The logic of the agents is described by two main classes: `Manager_base` and `Agent_base`. The former is the agent-side interface with the model, and manages the behaviour of the individual agents (that are instances of the latter class). As the name suggests, they are base classes that are derived in more specialised classes, like `Agent_space` and `Agent_farmer`.

The equivalent interface at the model-side is `SuperAgentManager` and there is only one instance of this class. It doesn't know anything of individual agents, as its relationship is only with the different managers.

The model run sequentially following two main functions: `Init::setInitLevel()` during initialisation stage and `Scheduler::run()` during simulation. The latter function performs a loop for each iteration that is due (see the Flowchart, Fig. 2)

Other important classes are those managing the regional data (`RegData`), the model objects (`ModelObject`) the regional level activities and resources (respectivelly `RegActivities` and `RegResources`) and finally the optimisation problem associated to each agent (`Opt`).

All classes are documented in the Reference Manual that is build directly from the source code using Doxygen.

RegMAS is know to compile using Gcc on both Linux and Windows. In the CVS a *project file* is provided for two user-friendly Integrated Development Environment (IDE) for both the OSes, KDevelop and Dev-C++. Using one of this two IDE is strongly recommended, as fully tested.

Source code is available under the GPL license, see the appendix for details.

8 Model output

The verbosity of the output can be set through the *outputLevel* parameter, from 0 (no output) to 7 (all possible outputs).

In the middle, option 3 will print yearly maps for each layer and aggregated regional data (including data for leaving farms) and option 5 adds individual agents data.

With option 7 it is also possible to print the matrices comprising the mixed-integer problem that agents need to solve in order to "produce" in the model.

Furthermore it is possible to decide if having a fully deterministic model, or - changing the seed of the random number generator with the *randomSimulation* option - allow for a stochastic simulation.

8.1 Spatial results

As RegMAS is spatially explicit any information in the model with a spatial attribute can be printed as a map, e.g. for further elaboration with a GIS (see next section).

Available maps are the farms location, their rented and owned plots, their spatial activities (e.g. durum wheat) and the overall agricultural usage. The last one allows users to have a quick view of abandoned land.

Maps are exported as ArcINFO or GRASS ASCII grids and from RegMAS version 1.0 RC2 they can be exported directly as images (.png files).

8.2 Quantitative results

On each simulation year, RegMAS can produces a very large amount of data, as shown on Table 1.

Furthermore similar results are available for farmers that leave the agricultural sector.

Most data is collected trough a "survey" that each agent is asked to fill, more or less how it happens with real statistics.

Table 1: Result categories

Aggregated results	**Individual agent results**
(Sum) of Agents by type	Agent type
	Agent ID
	Agent Mould ID
	Host plot ID & coordinates
(Avg) Total, land and object capitals	Total, land and object Capitals
(Avg) Profit	Production profit
(Avg) Coupled premiums	Coupled premiums
(Avg) Decoupled premium	Decoupled premium (SFP)
(Avg) Sunk costs	Sunk cost paid
(Sum) Rental land prices by type	
(Sum) Abandoned and idle plots by type	
(Sum) Owned plots by type	Owned plots by type
(Sum) Rented plots by type	Rented plots by type
	Owned objects by type
(Sum) Available objects by type	Available objects by type
(Sum) New investments by type	New investments by type
(Sum) Production activities by type	Production activities by type

Rental prices are updated from the previous year (or, on the first year, from initial renting prices) only if in the model there are at least 5 new contracts for that specific soil type.

Abandoned plots sum two different kind of plots: those owned or rented by an existing farm but not utilised at all, and those owned by a farm that left the model and that are srill unrented.

The difference in the detailed data between owned and available objects is due to the fact that agents may have rented plots including spatial objects that are not in their property basket but that thanks to the renting contract become available to their production activities: the formers are important in the capital formation, the latter to the production decisions.

On both the aggregated and the detailed results the "activities by type" values refer to the results arising form the solution of the agents profit-optimisation problem.

9 Exporting results to a Geographical Information System

Under the *output/map* folder a couple of scripts are provided to automatise the importing of RegMAS results into the GRASS GIS.

Although quite irksome to learn, Grass GIS is a very powerful open source multi platform desktop GIS that is particularly suited for raster spatial analysis and, even more important, is fully scriptable. Users can read more of it on its web-site[12] or it can be downloaded it in bundle qith QGis [13], a gis application that wrap Grass making it much more easy to be used.

To import spatial data inside a GIS program manually, users have to import the raw data, define the categories and pick-up the colour for each category. These are located respectively under the *asciiGrid*, *cats* and *colr* subfolders. Furthermore an empty "Region" in the Grass format is provided in the *grass/ColliEsini* subfolder, containing a "skeleton" structure (e.g. the geographic coordinated) for the sample region.

Under Linux or Windows/Cygwin environment, the *importInGrass.sh* script will does the import automatically (calling the second script *runWithinGrass.sh* to invoke Grass commands) otherwise individual maps can be imported from the Grass command *r.in.ascii* and then the associated category and colour description files should be copied in the working region.

GIS post-processing is useful when we want to place into relation several spatial data, e.g. in order to calculate the land abandonment, or the production intensity or even the farm density by elevation zone or by soil type.

We could also build a matrix to show which productions are switched to an other production type after a policy change.

Plans are to create a repository of scripts to produce such statistics from the RegMAS output and user submission of their own scripts is welcome.

[12] `http://grass.itc.it/`

[13] `http://www.qgis.org`

References

Arfini, F. (2000), 'I modelli di programmazione matematica per l'analisi della politica agricola comune'. INEA seminar *Valutare gli effetti della Politica Agricola Comune*, Rome, 24 October 2000. Available from: `http://web.archive.org/web/20041024164241/http://www.inea.it/opaue/pac/arfini.PDF`.

Balmann, A. (1997), 'Farm-based modelling of regional structural change: A cellular automata approach', *European Review of Agricultural Economics* **24**(1-2), 85–108. doi:10.1093/erae/24.1-2.85.

Boero, R. (2006), 'The spatial dimension and social simulations: A review of three books', *JASSS, Journal of Artificial Societies and Social Simulation*. Available from: `http://jasss.soc.surrey.ac.uk/9/4/reviews/boero.html`.

Happe, K., Balmann, A. & Kellermann, K. (2004), 'The agricultural policy simulator (agripolis) - an agent-based model to study structural change in agriculture', IAMO discussion paper 71. Available from: `http://www.iamo.de/dok/dp71.pdf`.

Kellermann, K., Happe, K., Sahrbacher, C. & Brady, M. (2007), 'Agripolis 2.0 – documentation of the extended model', IDEMA working paper 20. Available from: `http://www.sli.lu.se/IDEMA/WPs/IDEMA_deliverable_20.pdf`.

Lobianco, A. (2007), The effects of decoupling on two Italian regions. An agent-based model., PhD thesis, Università Della Tuscia. Available from: `http://associazionebartola.univpm.it/pubblicazioni/phdstudies/phdstudies2.pdf`.

Makhorin, A. (2007), 'Gnu linear programming kit. reference manual'. Available from: `http://www.gnu.org/software/glpk/`.

Paris, Q. (1991), *An economic interpretation of Linear Programming*, Iowa State University Press. Also available in Italian with title *Programmazione lineare. Un'interpretazione economica*.

Parker, D. C. (2003), 'Multi-agent systems for the simulation of land-use and land-cover change: A review', *Annals of the Association of American Geographers* **93**(2), 314–337. Available from: `http://www.blackwell-synergy.com/doi/abs/10.1111/1467-8306.9302004`, doi:10.1111/1467-8306.9302004.

Sahrbacher, C., Schnicke, H., Happe, K. & Graubner, M. (2005), 'Adaptation of agent-based model agripolis to 11 study regions in the enlarged european union', IDEMA working paper 10. Available from: `http://www.sli.lu.se/IDEMA/WPs/IDEMA_deliverable_10.pdf`.

A Getting help

RegMAS documentation is organised on three main levels.

- The **Official Documentation** includes the *User Manual* (this document) and the *Reference Manual*. The former introduce the program and present the tasks that users needs to accomplish to run their own simulation. The latter is instead addressed to advanced users with programming experience. It is automatically generated from the source code using the Doxygen tool and describes the classes that compose ReGMAS and how they are related.
- The **Contributed Documentation** is instead organised as a *wiki*, where users can find (and possibly, contribute) on specific topics. Contributed documentation does not claim to be well-organised and completed like the official documentation, but includes specific topics that are not documented in any other place (e.g. how the agents decision matrix is composed).
- Finally the **Community Project** (hosted at SourceForge) includes a forum and a mailing-list for discussions and a tracking system to report bugs, wish-list, to-do etc..

All documentation is available from the url `http://www.regmas.org/doc`.

B Licence

B.1 GPL

RegMAS is released under the GNU General Public License, Version 3 (GPL).

A copy of such licence is always enclosed with any release of the software and can also be obtained from `http://regmas.org/cgi-bin/viewvc.cgi/regmas/COPYING?view=co`.

In shorts, RegMAS can be copied and it is freely modified to produce derivative work, as long as the licence terms of the derivative work doesn't change and the source code (including those of the derivative work) remains publicly available.