Agent based value chain simulation model VAL-MAS

ODD Protocol: Technical documentation

Eva Winter, Version November 2021

Table of contents

1. Purpose

Question: What is the purpose of the model?

The VAL-MAS model is an ex-ante simulation model depicting the seed and breeding value chain as a multi-agent system. Agents have individual decision making behaviour based on mathematical programming and heuristics. It was developed to answer research questions revolving around changing policies in the seed and breeding sector of organic agriculture in a systematic way, such as the planned phasing out of derogations for the use of not chemically treated (NCT) seed. In the following, the reasons for the choice of this combination of model characteristics are laid out.

A value chain perspective was taken, because different actors and their interactions, i.e. breeders, seed producers and farmers, as well as the overarching political framework laid down in the EU organic regulation of the organic sector, contribute to the problems in the organic seed market. These actors are also all affected by current rules and regulations. Consequently, they need to be considered in a policy impact assessment. Mapping of value chains and subsequent benefit-cost or SWOT analyses with or without active stakeholder involvement have been repeatedly conducted in the past to analyse seed and other agricultural value chains (Bellù 2013; Mulugeta et al. 2010; Kumara et al. 2012). These methods can be complemented by more sophisticated assessment approaches that can give more in-depth insights into system changes under certain conditions. Rich et al. (2011) and Nang'ole et al. (2011) give an overview of existing agricultural value chain analysis frameworks highlighting that they are to a large extent qualitative. They recommend system dynamics and agent-based models so that quantitative ex ante policy assessments of value chains can be conducted in the future.

Ex ante policy assessment via simulation models is a useful means of testing policy instruments that could smooth the transition period as well as propose long term solutions to increase organic seed production and use. Many studies exist where agricultural policies and private sector interventions are tested through ex-ante assessment. These models mostly assess policy implications at farm or sector level (Janssen and van Ittersum 2007; Grovermann et al. 2017; Häring 2003; Bunte and Galen 2015; Schreinemachers and Berger 2011). Applications are often related to input choices under varying conditions (Schreinemachers and Berger 2011; Grovermann et al. 2017; Berger et al. 2017). In the following, the specific choices when implementing simulation modelling with the VAL-MAS model are discussed.

2. Entities, state variables, scales, process overview and scheduling

Questions: What kinds of entities are in the model? By what state variables, or attributes, are these entities characterized? What are the temporal and spatial resolutions and extents of the model?

Questions: Who (i.e., what entity) does what, and in what order? When are state variables updated? How is time modeled, as discrete steps or as a continuum over which both continuous processes and discrete events can occur? Except for very simple schedules, one should use pseudo-code to describe the schedule in every detail, so that the model can be re-implemented from this code. Ideally, the pseudo-code corresponds fully to the actual code used in the program implementing the ABM.

2.1 Entities in the VAL-MAS model: A multi agent system

In the VAL-MAS-model, agent-based modelling was used in combination with mathematical programming and heuristics. Agent based models are a valuable tool when modelling the behaviour of different agents in a heterogeneous population, where each entity takes individual decisions, but also reacts on decisions of other entities (Gjerdrum et al. 2010; Berger and Schreinemachers 2006). Since not only the actors along the seed value chain, but additionally the actors within one level of the seed value chain are very heterogeneous with respect to their decision-making behaviour, a multi-agent system is well suited when modelling the vertical and horizontal complexity of the seed value chain. Therefore, we chose an approach representing individual decision making over an aggregate modelling approach. In the VAL-MAS model, an entire agent population with individual decision making per agent is considered at the farm level. At multiplication and breeding levels, decision-making agents represent typical seed supply actors. Individual agents are thus:

- Breeding agents
- Seed multiplication agents
- Farm agents.

The number of agents and their representativity in the real world will be described in Section 4., input data.

2.2 Temporal resolution

As at the aim was to simulate several processes with different time horizons in one model, a dynamic model approach is essential in order to understand developments emerging in the course of different model scenarios. Moreover, a feed-back loop was needed between the stages of the value chain after activities in each year have been decided upon. As a consequence, the start values of a certain period need to be the end values of the previous period. We thus deemed embedding in a multi-agent system a positive recursive-dynamic model with decision-making through a combination of mathematical programming and heuristics as the most suited approach. Interactions between the value chain levels are based on information and material exchange regarding seed sales, seed amounts, prices, including a feedback loop on demand and supply of seed types (organic seed from typically used cultivars, NCT seed from typically used cultivars, seed from organic cultivars). Figure 1 shows the flow and interactions between agents in the model in a simplified way, illustrating the decision-making sequence of the agents. The food industry and policy framework are depicted in the square box as they are exogenous factors with no endogenous decision making within the model. Their influence can be seen if scenarios change, such as higher end product prices for organic seed use and policy schemes, such as the phasing out of derogations. The figure displays model processes in one year including the feedback loop ("update after sales") at the end of the year.

The model runs over 10 modelling periods (simulating years in the real world). Eight modelling years are being taken into consideration for results analysis. The first year is the spin off of the model (more info: Section 5). The last year is not considered, because the pay-offs of investments in e.g. planting biennial crops (such as carrot seed production) cannot be calculated satisfactorily. In each year, all agents optimise over the current year and the remaining years. They thus predict the future and can take longer-term decisions (e.g. planting biennial crops).

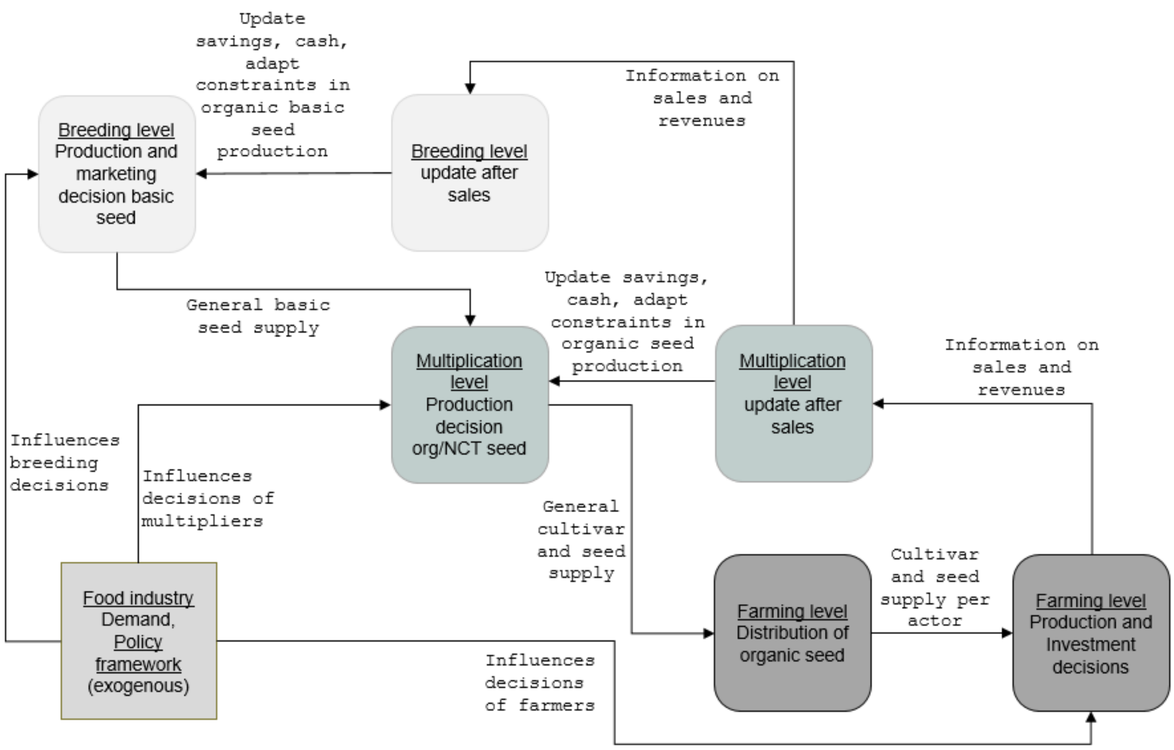

Figure 1: **Conceptual framework of the VAL-MAS model**

In Figure 2, the information flow between between the agents in one modelling period is depicted in a more detailed fashion. Loops over agent groups are indicated in Figure 2 by naming the respective group (e.g. Breeders), followed by the activity that all breeders conduct (e.g. optimisation). All solves, algorithms, and parameter and variable updates are indicated in framed boxes (These activities will be further explained in the following subsection). The arrows in Figure 2 indicate the direction in which thes information is passed on between activities. Unframed boxes always overlap with an arrow. The text in the unframed boxes indicates the information that is transmitted between the connected framed boxes.

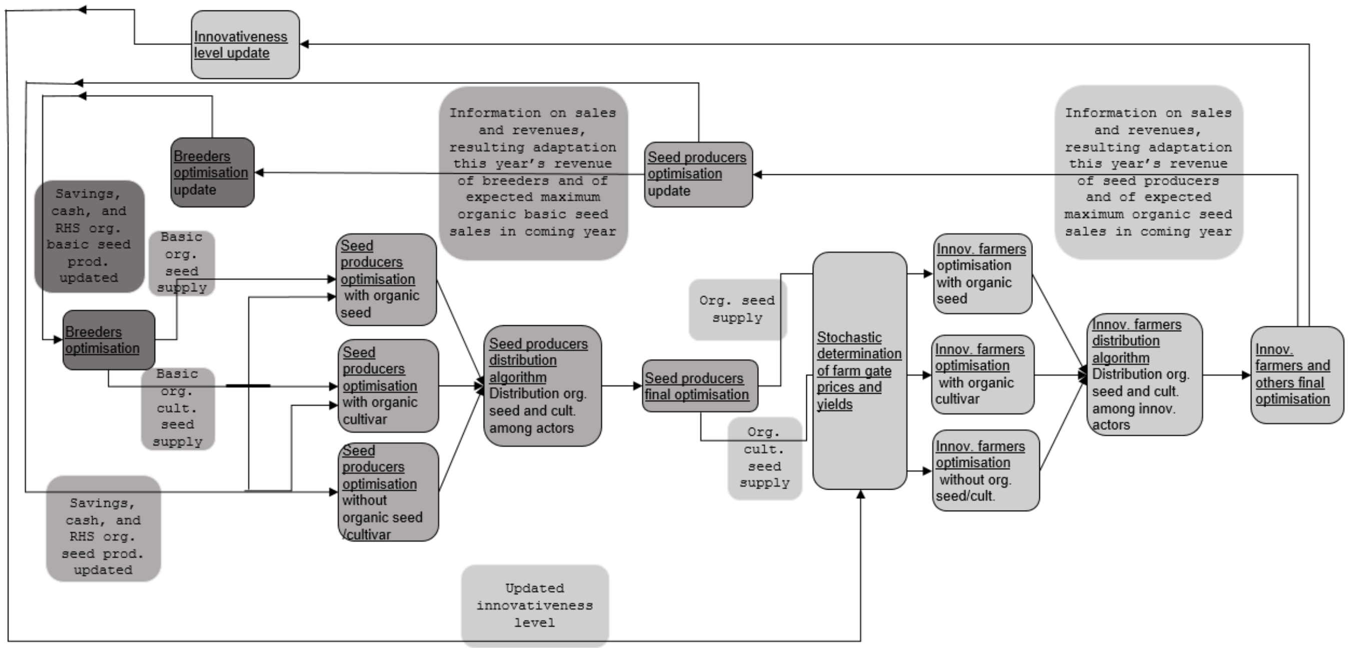

Figure 2: Information flow chart between solves, algorithms, and parameter and variable updates by agent groups in the GAMS Val-Mas Model within one modelling period (one year)

2.3 Scheduling of agent activities

The VAL-MAS model consists of three main sections: Firstly, all sets, parameters, variables, equations, and models are declared and assigned. Further, all baseline data and scenario data are included (lines A2.12-600). Secondly, the three models representing the three value chain stages breeding, seed production, and farming are being solved consecutively in a nested loop structure (lines A2.602-1144). Thirdly, model results are being exported to gdx and xslx files for further processing. This is a standard procedure, and is thus excluded here. The full GAMS code can be found in the Appendix A2.

The first part consists of four subsections. The first three subsections are structured in the same way:

- Firstly, sets indicating the modelling years, number of agents, activities, and constraints are declared and assigned (Breeding agents: Lines A2.14-48; Seed multiplier agents: Lines: A2.122-155 ; Farmer agents: Lines A2.257-285).
- Secondly, the parameters over these sets are declared and assigned. Parameters comprise right-hand-side values, technical coefficients, gross margins per crop, and parameters defining interactions between value chain levels, e.g. the calculation of

adaptive expectations (Breeding agents: Lines A2.50-84; Seed multiplier agents: Lines: A2.157-224; Farmer agents: Lines A2.287-331).

- Thirdly, variables, equations and the optimisation model are declared and assigned (Breeding agents: Lines A2.86-117; Seed multiplier agents: Lines: A2.226-255; Farmer agents: Lines A2.569-598).

In the fourth and last subsection of Section 1, additional data is loaded or generated:

- Parameters defining the distribution of farm gate prices and yields at farm level (lines A2.333-383)
- Starting values for the working capital at farm level (lines A2.385-413)
- Sets and parameters needed for the element of innovation diffusion at farm level (lines A2.421-447)
- Parameters for the scaling factors (lines A2.449-499)
- Calibration and implementation of excess willingness to pay at farm level (lines A2.502-550)
- Parameters necessary for the calculation of different scenarios (lines A2.553-566)

In the following, Section 2 (nested loop structure to solve submodels) of the GAMS model is described in detail. The first loop controls the set representing the model years (opening in line A2.601, closing in line A2.1141). In this application, ten years were modelled. Within one modelling year, five nested loops representing the value chains were included: Breeders are looped over once (opening in line A2.620, closing in line A2.634), while seed producers (First loop opening in line A2.647, closing in line A2.697, second loop opening in line A2.724, closing in line A2.735) and farmers are looped over twice (First loop opening in line A2.752, closing in line A2.866, second loop opening in line A2.889, closing in line A2.926). Figure 3 represents the flowchart of the model. Loops over agent groups are indicated in Figure 3 by naming the respective group (e.g. Breeders), followed by the activity that all breeders conduct (e.g. optimisation). The schematic flowchart follows the standard syntax according to DIN 66001.

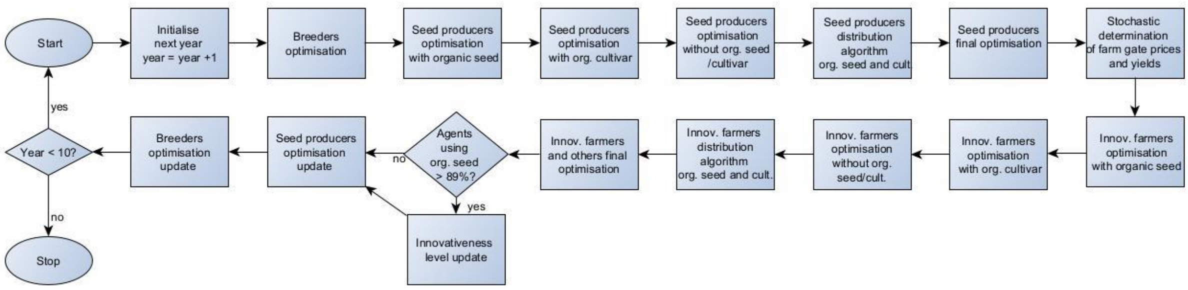

Figure 3: Flowchart of VAL-MAS model

The five loops are controlled respectively by the sets for the three different agent types. For each of these agent types, there is the set ag_* containing all agents of this type. There is further the alias of this set, aga_* and lastly, the subset currentag_*(ag_*), both containing the same set element as ag_* in the beginning. The alias and the subset are needed to be able to solve one agent's equations (one set element) at a time and keep the others fixed. The agent whose equations are solved is assigned to currentag_*(aga_*). All other agents, whose solution variables etc. are kept fixed at this time, are not assigned to currentag_*(ag_*) (e.g. for breeding agents, this can be found in lines A2.622-630). To technically implement this, firstly, all agents in set ag_* are taken out of the subset currentag_* (e.g. currentag_br(ag_br)=no;). After this, only one agent is assigned to the subset (e.g. currentag_br(aga_br)=yes;). The loop over aga_* controls which agent's turn it is. This is a technical solution to separate the objective functions of the agents. Solving each agent at a time allows us to access individual data of each agent and prevents agents from acting in a way that contributes to the overall welfare of all agents, even though it may disadvantage them. However, if the agents are assigned individual objective functions, the order of agents to e.g. get access to an innovation becomes of importance and needs to be determined. Examples are a randomization before each modelling period or a comparison of shadow prices.

In the first of these five loops over the agent type groups, the breeder model is solved once for each breeding agent, determining the amount of basic seed for ct, nct, organic seed and the organic cultivar that is being produced and offered for sale to the seed producers (line A2.632). In the second loop (the loop over the seed producers), the seed producer model is solved three times for each agent. In the first solve, each seed producer can access the entire amount of produced organic basic seed (line A2.666). The amount of basic organic seed bought and used by each seed producer is then stored in a parameter. In the second solve, the same is calculated for organic cultivars (line A2.681). The third solve calculates the optimal gross margin per agent without organic seed or cultivars (line A2.694). This was implemented to calculate a distribution scheme for organic seed based on the potential demand of each agent.

Then, the access to basic organic seed of the individual agents is calculated by an algorithm based on the results of the preceding three solves. The distribution algorithm functions as follows: The quantity of organic (basic) seed that the respective agent used in the first solve is divided by the amount of organic seed used by all agents and multiplied by the total amount of organic seed available (lines A2.699-719). Finally, in the third loop, all seed producer agents' gross margins are maximised subject to the assigned access to basic organic seed (line A2.734).

The fourth loop controls the innovative farm agents. The structure including the dynamic subset currentag_fa is still used in order to solve agents individually, however, following this, there is a nested loop structure over the dynamic subset innov_ag_fa(aga_fa) that gives only innovative farmers access to organic seed (opening in line A2.767, closing in line A2.864). Here, the model is solved again three times. The same distribution algorithm as explained above is applied (for organic seed on the farmers' level). Finally, in the fifth loop, all farmers agents' gross margins are maximised subject to the assigned access to organic seed.

At the end of the first modelling year, the solution variables of breeding and seed producing agents were updated through a final solve according to their sales (for breeding agents: Lines A2.1007-1024; for seed multiplication agents: Lines A2.1129-1139). This concluded the first modelling year and provided the starting values for the second modelling year. This was repeated seven times.

2.4 Scales

To ensure compatibility between the size of the farmers' buying market and the seed multipliers' selling market, scaling factors were implemented. If this was omitted, economies of scale would not be realised. The cultivated area in the case study would not be sufficient as a seed market for the larger of the two typical seed producers (more information in the next subsection). As the present agent-based value chain model is the first to fully simulate the seed value chain, there is no precedent for this procedure. Nevertheless, scaling factors are commonly used to ensure compatibility between agents or activities in multi-agent and integrated farm system modelling (Troost and Berger 2015b; Gibbons and Ramsden 2008). In this study, scaling factors are used to connect the three value chain levels in order to match supply and demand. For example, one of the interviewed carrot seed producers has a market size of 1,505 hectare organic carrot production, while according to German statistical data, the organic carrot producers in Germany cover 2,100 ha (Destatis 2018), of which around 1,200 to 1,400 ha are covered by carrots for main production and storage (German organic carrot expert estimation). Consequently, the scaling factor from multiplier to farmers is 0.8 to 0.93 and from farmers to multiplier 1.25 to 1.1.

The implementation of the scaling factors in the GAMS model can be found in Appendix A2. The declaration, assignment and calculation of the necessary parameters are found in lines A2.449-499. The scaling itself is conducted in lines A2.738-741 (scaling from seed multiplication level to farm level), A2.1026-1031 (scaling of variety 2 (organic seed) from farming to seed multiplication level) and A2.1075-1080 (scaling of variety 3 (organic cultivar) from farming to seed multiplication level).

3. Design concepts

Questions: There are eleven design concepts. Most of these were discussed extensively by Railsback (2001) and Grimm and Railsback (2005; Chapter. 5), and are summarized here via the following questions:

Basic principles. Which general concepts, theories, hypotheses, or modeling approaches are underlying the model's design? Explain the relationship between these basic principles, the complexity expanded in this model, and the purpose of the study. How were they taken into account? Are they used at the level of submodels (e.g., decisions on land use, or foraging theory), or is their scope the system level (e.g., intermediate disturbance hypotheses)? Will the model provide insights about the basic principles themselves, i.e. their scope, their usefulness in real-world scenarios, validation, or modification (Grimm, 1999)? Does the model use new, or previously developed, theory for agent traits from which system dynamics emerge (e.g., 'individual-based theory' as described by Grimm and Railsback [2005; Grimm et al., 2005])?

Emergence. What key results or outputs of the model are modeled as emerging from the adaptive traits, or behaviors, of individuals? In other words, what model results are expected to vary in complex and perhaps unpredictable ways when particular characteristics of individuals or their environment change? Are there other results that are more tightly imposed by model rules and hence less dependent on what individuals do, and hence 'built in' rather than emergent results?

Adaptation. What adaptive traits do the individuals have? What rules do they have for making decisions or changing behavior in response to changes in themselves or their environment? Do these traits explicitly seek to increase some measure of individual success regarding its objectives (e.g., "move to the cell providing fastest growth rate", where growth is assumed to be an indicator of success; see the next concept)? Or do they instead simply cause individuals to reproduce observed behaviors (e.g., "go uphill 70% of the time") that are implicitly assumed to indirectly convey success or fitness?

Objectives. If adaptive traits explicitly act to increase some measure of the individual's success at meeting some objective, what exactly is that objective and how is it measured? When individuals make decisions by ranking alternatives, what criteria do they use? Some synonyms for 'objectives' are 'fitness' for organisms assumed to have adaptive traits evolved to provide reproductive success, 'utility' for economic reward in social models or simply 'success criteria'. (Note that the objective of such agents as members of a team, social insects, organs—e.g., leaves—of an organism, or cells in a tissue, may not refer to themselves but to the team, colony or organism of which they are a part.)

Learning. Many individuals or agents (but also organizations and institutions) change their adaptive traits over time as a consequence of their experience? If so, how?

Prediction. Prediction is fundamental to successful decision-making; if an agent's adaptive traits or learning procedures are based on estimating future consequences of decisions, how do agents predict the future conditions (either environmental or internal) they will experience? If appropriate, what internal models are agents assumed to use to estimate future conditions or consequences of their decisions? What tacit or hidden predictions are implied in these internal model assumptions?

Sensing. What internal and environmental state variables are individuals assumed to sense and consider in their decisions? What state variables of which other individuals and entities can an individual perceive; for example, signals that another individual may intentionally or unintentionally send? Sensing is often assumed to be local, but can happen through networks or can even be assumed to be global (e.g., a forager on one site sensing the resource levels of all other sites it could move to). If agents sense each other through social networks, is the structure of the network imposed or emergent? Are the mechanisms by which agents obtain information modeled explicitly, or are individuals simply assumed to know these variables?

Interaction. What kinds of interactions among agents are assumed? Are there direct interactions in which individuals encounter and affect others, or are interactions indirect, e.g., via competition for a mediating resource? If the interactions involve communication, how are such communications represented?

Stochasticity. What processes are modeled by assuming they are random or partly random? Is stochasticity used, for example, to reproduce variability in processes for which it is unimportant to model the actual causes of the variability? Is it used to cause model events or behaviors to occur with a specified frequency?

Collectives. Do the individuals form or belong to aggregations that affect, and are affected by, the individuals? Such collectives can be an important intermediate level of organization in an ABM; examples include social groups, fish schools and bird flocks, and human networks and *organizations. How are collectives represented? Is a particular collective an emergent property of the individuals, such as a flock of birds that assembles as a result of individual behaviors, or is the collective simply a definition by the modeler, such as the set of individuals with certain properties, defined as a separate kind of entity with its own state variables and traits?*

Observation. What data are collected from the ABM for testing, understanding, and analyzing it, and how and when are they collected? Are all output data freely used, or are only certain data sampled and used, to imitate what can be observed in an empirical study ("Virtual Ecologist" approach; Zurell et al., 2010)?

Explanation: The 'Design concepts' element of the ODD protocol does not describe the model per se; i.e., it is not needed to replicate a model. However, these design concepts tend to be characteristic of ABMs, though certainly not exclusively. They may also be crucial to interpreting the output of a model, and they are not described well via traditional model description techniques such as equations and flow charts. Therefore, they are included in ODD as a kind of checklist to make sure that important model design decisions are made consciously and that readers are aware of these decisions (Railsback, 2001; Grimm and Railsback, 2005). For example, almost all ABMs include some kinds of adaptive traits, but if these traits do not use an explicit objective measure the 'Objectives' and perhaps 'Prediction' concepts are not relevant (though many ABMs include hidden or implicit predictions). Also, many ABMs do not include learning or collectives. Unused concepts can be omitted in the ODD description.

3.1 Decision making of the agents: Individual objective functions (Concepts covered: Basic principles, emergence, objectives, collectives)

Among the farm agents, the gross margin per farm enterprise agent is maximised. We defined the farm enterprise agent as the collection of crops in the organic carrot crop rotation. Among the seed producing agents, the gross margin of organic and NCT carrot seed production per seed multiplication agent is maximised. Processing, packaging, and marketing costs are largely the same for conventional untreated and organic seed, thus these costs were disregarded at multiplication level. Lastly, among the breeding agents, we maximised the overall yearly breeding budget, i.e., we implemented a revenue maximisation of the focus crop (wash/storage carrot) section of the breeding agent, including non-organic seed (Chemically treated (CT) and NCT). The breeding budget is represented by 10 to 30% of the seed sales revenue, depending on the typical actor. A revenue maximisation for the breeding agents was chosen, because the breeding costs were treated as constant over time, and thus the revenue maximisation can be seen as a proxy of profit maximisation. This was confirmed by the breeding actors to emphasise that constantly providing the breeding budget is of crucial concern: Both of the typical breeding actors we identified do not consider the gross margin at breeding level, but require a constant breeding budget as part of research and development (Kuin 2018; Syngenta 2015). Simplified linear programming matrices of an exemplary agent for the first two modelling years of each value chain stage can be found in Tables A3.1-3 in the Appendix.

3.2 Adaptive Expectations of seed producers and breeders (Concepts covered: Basic principles, emergence, adaption, prediction, interaction)

As it is likely that seed producers will not immediately react to changes in demand for organic seed, we implemented an adaptive expectations mechanism at multiplication level to smooth the organic supply quantity. The theory of adaptive expectations is based on the assumption that a behaviour, such as the production of organic seed, is determined by experiences of past sales (Galbács 2015). We defined the upper limit R of the amount that can be produced in a year as the average of the sold amount s of the last two years multiplied with a growth expectation factor. This growth expectation factor G indicates the trend in demand and is the difference in percentage between the sum of the current and last year and the sum of the last year and the year before, times the production reserve factor p. The production reserve factor indicates how much more than the estimated amount is produced for reserve in the case of unexpected higher demand. Lower and upper bounds of the growth expectation factor were defined as 0.5 and 2, unless differently specified in certain scenarios. Given the technical difficulties in organic hybrid carrot seed production, we assume that any increase above a doubling of seed production from one year to another is highly unlikely.

Furthermore, in order to account for uncertainties due to technical difficulties, the production reserve factor under a growth scenario where an increasing organic seed demand is expected is always indicated between 1.2 and 1.5. This range reflects the uncertainty based on difficulties to find organic carrot seed producers, suitable areas, and technical difficulties in production. These difficulties are substantial in the chosen case. However, the bounds of the growth expectation factor and the values of the production reserve factor are expert-based estimations. The values need thus be interpreted with caution. The formulae for calculating the Growth expectation factor G and of the upper limit R of the amount of seed that can be produced in a year are listed below.

$$G = \min(\max(((s_t + s_{t-1})/(s_{t-1} + s_{t-2}))p, G_{lo}), G_{up}) \quad (1)$$

$$R = \max(((s_t + s_{t-1})/2)G, R_{lo}) \quad (2)$$

The implementation of the growth expectation factors in the GAMS model can be found in Appendix A1, lines 50-84 (Declaration and Assignment of parameters) and A2.959-1011 (Calculation each modelling year) for the breeding agents and in lines A2.157-224 (Declaration and Assignment of parameters) and A2.1026-1127 (Calculation each modelling year) for the seed multiplication agents.

3.3 Theory of the diffusion of innovations and application in the VAL-MAS model (Concepts covered: Basic principles, emergence, objectives, collectives, interaction, adaptation)

Ann agent population at farm level based on German statistical agricultural census data and the procedure proposed by Berger and Schreinemachers (2006) and Troost and Berger (2015a) was created (see Section 4.2-3, Input data, for more information). The model includes a feature that represents the diffusion of an innovation according to the network threshold theory of Rogers (2003). According to this theory, farm agents were categorised into five segments:

- Innovators (2.5% of the population),
- early adopters (13.5%),
- early majority (34%),
- late majority (34%),

– and laggards (16%).

This reflects learning in a social network and incomplete information as can often be observed in reality. The mechanism behind this is that only if the first group has adopted the innovation, the second group is able to adopt and so on. The agents in the model were assigned to the network segments based on a statistical estimation of propensity scores. In order to establish the innovativeness scores in the agent population, influential characteristics were regressed on organic seed use using recent survey data from organic farmers (Orsini et al. 2020). A subsample of a European survey of 128 observations comprising those farms that produce vegetable in Germany, Netherlands, Switzerland, and Austria were analysed with a linear regression model. We took these additional countries into account to obtain a sample size that could be analysed in a meaningful way. As these countries are all in Central Europe, we expect the observations to be rather homogeneous and valid for the countries in the analysis. Independent variables in the analysis were education, farm size, country (for correction, as we include other Central European countries to obtain the sample size), and farm type (arable or vegetable). The dependent variable was organic seed use per hectare per farm enterprise in % of the overall number of hectares. The coefficients of the regression are given in Table 1. Based on the estimated coefficients of the three selected characteristics in the agent population, the predicted organic seed use per agent was computed. The error term of the regression as randomisation element was included in this computation. The agents were then ranked in a cumulative distribution according to their predicted organic seed use. On that basis, the assignment to innovation segments was determined. For example, if the predicted organic seed use of an agent was ranked among the highest 2.5%, the agent was assigned to the group of innovators.

The element of diffusion is implemented in the VAL-MAS model in the following way: If 90% of the innovators have adopted at least 25% organic seed per agent, the early adopter group gets access to the organic seed and so on. The implementation of this in the GAMS model can be found in Appendix A2. Set and parameter declarations and assignments are shown in lines A2.421-447. Agents having access to the innovation are included into the dynamic subset of all farming agents innov_ag_fa. The assessment of the actual adoption per agent and the subsequent re-allocation of agents into this dynamic subset if the conditions are met can be found in the modelling code in lines A2.928-954.

Table 1: Regression coefficients and of different farm characteristics on *organic seed use per farm enterprise*

Variable	Coefficient	Significance
Farm size in ha	-0.066	***
Farm type (arable=0, vegetable=1)	-0.134	*
Education (no education 0, education in farming 1)	0.21	**
Constant	0.91	

4. Input data

Question: Does the model use input from external sources such as data files or other models to represent processes that change over time?

4.1 Input data at breeding and seed multiplication level

Typical breeding and seed production entities were defined, as a representation of the entire population was impossible due to limited willingness of actors to share economic data. We defined a typical entity as a kind of company or initiative with a large market share in organic seed production and/or organic breeding. Two types of seed multiplication actors were identified. One type is an internationally active commercial seed and breeding company that sells NCT hybrid seed and organic hybrid seed to organic carrot producers in Germany. The second type is a small company or initiative dedicated to breeding and/or locally selling open-pollinated vegetable seed produced under fully organic conditions.

Two breeding types are also represented in the simulation model. Type I is defined as the breeding department of the internationally active company that also produces seed. No breeding programmes specifically or uniquely for organic carrot production are conducted, nevertheless, organic cultivar trials to choose the best suited cultivars for organic conditions are carried out. Hybrids are developed. To re-finance their breeding programmes, 13,545 ha carrot production planted with the company's seed are needed. Their organic area share is 1,505 ha, 11% of the total area. Their yearly fresh market carrot breeding budget is estimated to be around 30% of the revenue. Type II is specialised on breeding under organic conditions and exclusively develops open pollinated cultivars. Their financing strategy is pre-financing through voluntary contributions from seed multipliers, but mostly from donations and sponsorships. We assume that around 10% of the total seed sales from their cultivars is voluntarily given back to them to re-finance their breeding activities.

The input files are excel files that include right-hand-side (RHS), Technical Coefficients (A), and Gross margin data. Simplified decision tableaux showing the equation structure can be found in Appendix A3, Tables A3.2-3. The data is read into the GAMS model in the first Section of the model code (more info: Section 6, Submodels).

4.2 Input data and creation of an agent population at farming level

To determine the number of agents at farm level, we used information about the number of organic carrot producers in Germany that is available on request from the Statistical Offices of the Federal States (Destatis 2018). For the analysis, the statistical programme STATA15 was used. To narrow the number of organic carrot producers down to those who produce mainly wash/storage carrots, we consulted organic carrot experts during interviews. We identified around 325 organic wash/storage carrot producers in Germany. We model 100 organic wash/storage carrot producers explicitly. To justifiy this reduction of agents we rely on the law of large numbers. Furthermore, this reduction ensures parsimony of the model. In order to generate the farm agent population, we relied on the farm accountancy data ("Agrarstrukturerhebung") of the year 2016, provided by the national statistical office in Germany (RDC 2016). A copula approach to estimate a joint distribution between selected farm characteristics was used following the procedure proposed by Berger and Schreinemachers (2006). For the joint distribution, farm characteristics variables were divided into quintiles (a higher resolution was not possible due to privacy restrictions) and then matrices were created from the combinations of quintiles along the farm characteristics of each observation in the dataset. This is illustrated by Figure 4.

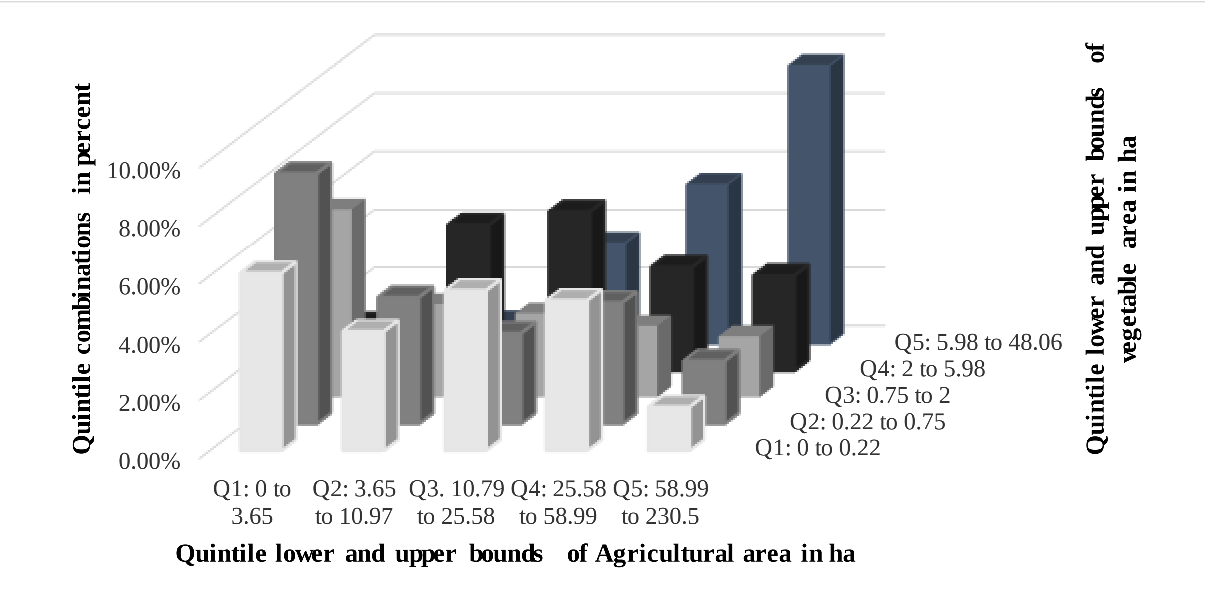

Figure 4: Copula of arable area and vegetable area based on own calculations using the data from the RDC (2016)

The observed frequencies within the multidimensional categories served as empirical copula from which the agents for the agent population were drawn. We created one copula including the farm characteristics total farm area, total agricultural area, organic vegetable area in rotation with other vegetable and with arable crops, and available labour per farm. We then created copulae including only two characteristics. In each of these smaller copulae, the total agricultural area was included in combination with one other relevant crop area or the farm manager's education. This approach was adopted in order to avoid the barring of values in the copulae due to privacy restrictions. Two types of farms could be identified. One type produces mostly vegetable in rotation with other vegetable and the other type produces vegetable in rotation with arable crops. The ratio of these farms could be obtained from the structural farm data (RDC 2016). For these two types, we defined different typical crop rotations. The vegetable type grows carrots in combination with leak, cabbage, salad, and clover in a horticultural system. The arable type grows carrots in combination with winter wheat, winter rye, beans, onions, and clover in an arable system.

Further input data for the farm population were obtained from different sources. Whole-sale price data for washed carrots for the fresh market were available as a time series for ten years from AMI and detrended to correct for trend-related changes (Baum 2006). The ranges of the prices were implemented in the model as triangular distributions for sensitivity analysis. From the German national statistics on vegetable yields (time series data comprising five years), yields of crops in the crop rotation were estimated and the ranges were implemented in the model as triangular distributions for sensitivity analysis. Technical coefficients and variable costs for the crops in the crop rotation were taken from Kuratorium für Technik und Bauwesen in der Landwirtschaft e.V. (KTBL 2016), the German national data base on agricultural figures, and were matched with the agent population. The main matching variable was the vegetable area. With the help of a small survey among German organic carrot producers, we gathered and narrowed down information relating to the range of plot sizes, degrees of mechanisation, typical crop rotations, distance between farm and field, and the type of production system (bed or bank cropping). We conducted a survey among German organic carrot producers partly with the online tool Lime survey and partly via phone in the period of January to July 2019. The online survey was distributed through organic vegetable extension services and organic marketing organisations. We obtained 20 responses from organic carrot producers willing to take the survey. The results are not representative, however, a variety of distribution channels for sharing the survey were used, such as consultancies and a wide variety of marketing organisations. There is no obvious bias as regards organic seed or cultivar use (e.g. much more organic seed is used by the interviewed farmers than is likely to be observed in reality). The questions in the survey focused on organic carrot production in Germany, i.e. on crop rotations, used cultivars, and differences in production costs between organic carrots produced with organic seed, NCT seed, and organic cultivars, among other items. Furthermore, we included specific questions about the willingness to pay for organic seed and cultivars.

4.3 Agent population generation

Firstly, the copulae in the form of cumulative distributions are imported from Excel into the statistical programme (in this case STATA) (A1.lines 14 - 75). Secondly, the individual vectors per agent including the quintile combinations are drawn according to the copulae. Here, the characteristics in the main copula are drawn first (A1.lines 80 - 126). Then, the dummy indicating if an agent is a vegetable producer or an arable farmer is randomly drawn (A1.lines 129 – 134). As a third step, the smaller copulae are matched using the matching variable agricultural area and random values are drawn. The example of the characteristic winter wheat area is shown in the Appendix A1 (lines 140 - 276). The values drawn serve as Right-hand-side values (total area, available labour) and to determine the innovativeness score of the generated agents. The analysis of a farmer survey is described in Section 3.4. The application in STATA can be seen in lines A1.289 – 334. Furthermore, the 4 combinations of mechanisation and plot size available through KTBL data are randomly distributed within the bounds of the assigned vegetable plot size (lines A1.338 – 355). From line A1.372 on, all remaining data points (e.g. variable costs per ha and crop, labour requirements per ha and crop, etc.) per agent are assigned to the agents according to their characteristics (e.g. their mechanisation and type of agent (vegetable or arable producer) and saved in a matrix. This information is now ready to be read into Excel-Input files per agent. The creation of Excel-Input files per agent (RHS- and A-input files), and the merge into gdx-input files through a bash script are not included in the presented STATA code, as they are mere technical implementations.

5. Initialisation

Questions: What is the initial state of the model world, i.e., at time t = 0 of a simulation run? In detail, how many entities of what type are there initially, and what are the exact values of their state variables (or how were they set stochastically)? Is initialization always the same, or is it allowed to vary among simulations? Are the initial values chosen arbitrarily or based on data? References to those data should be provided.

The number of agents does not change over time in the current version of the model. There are 2 breeding agents, 2 seed multiplication agents and 100 farm agents. The parameters characterising the agents are listed and described in Section 2 and Appendix A2, first section, first two subsections.

- Stochastic parameters:
    - At $t = 0$, then fixed:
        - Farm agent characteristics: farm type, total available vegetable and arable land area in ha (and as a result, total farm area in the model for scaling, see Section 2.3), total available labour in personhours, mechanisation (further info see Section 4.2-4.3)
        - Farm agent characteristic excess willingness to pay for organic seed in € (further info see Section 7)
    - In the beginning of each modelling year: Yields at farm level in tons and farm gate prices in € (further info see Section 4.2)
- Adaptive/re-calculated parameters for each modelling year:
    - Growth expectation factor (see Section 3.3)
- Fixed parameters:

- All other model elements.

In the scenarios simulating voluntary measures, the model runs for 10 periods (modelling years). However, the first year are is taken into consideration as model results. In this year, organic seed diffuses through the innovator group to the early adopter group, reaching around 10% of organic seed use that can be observed in reality.

It would however be possible to include more agents or exclude agents over time to simulate structural change. In this case, agents would be generated beforehand and included over a dynamic subset.

6. Submodels

Questions: What, in detail, are the submodels that represent the processes listed in 'Process overview and scheduling'? What are the model parameters, their dimensions, and reference values? How were submodels designed or chosen, and how were they parameterized and then tested?

Explanation: The submodels are presented in detail and completely. The factual description of the submodel, i.e., equation(s) and algorithms, should come first and be clearly separated from additional information. From what previous model this submodel was taken or whether a new submodel was formulated, and why, can be explained. If parameterization is not discussed outside the ODD description, it can be included here. The parameter definitions, units, and values used (if relevant) should be presented in tables.

Any description of an ABM and its submodels will seem ad hoc and lack credibility if there is no justification for why and how formulations were chosen or how new formulations were designed and tested. Because agent-based modeling is new and lacks a firm foundation of theory and established methods, we expect ODD descriptions to include appropriate levels of explanation and justification for the design decisions they illustrate, though this should not interfere with the primary aim of giving a concise and readable account of the model. Justification can be very brief in the Overview and Design concepts sections, but the complete description of submodels is likely to include references to relevant literature, as well as independent implementation, testing, calibration, and analysis of submodels.

ODD-based model descriptions consist of the seven elements described above; however, in most cases it will be necessary to have a simulation experiments or model analysis section following the model description (see Discussion).

Currently, VAL-MAS does not consist of submodels.

7. Model analysis and simulation experiments: Verification, calibration, and validation of the VAL-MAS model

7.1 Verification

During verification, the generated agent population is examined as to how well it represents the characteristics of the observed data. The agent population in our case was verified by cross-checking summary statistics of generated variables and correlations between generated variables with the original farm accountancy data set.

7.2 Calibration

Calibration of a model is the process to adjust certain parameters so that the model produces results in the baseline that are as similar as possible to real world conditions (Howitt 1995; Troost and Berger 2015b). Calibration of the simulation model was conducted by calibrating the amount of organic seed used in the model to the real world observation of 10% seed use for German organic storage carrot, and of 1% seed used from organically bred cultivars (Herstatt 2017). This was achieved by assuming an excess willingness to pay for organic seed, depending on the innovativeness segment of the farm agent. This excess willingness to pay can be assumed as there is no subsidy for organic seed use and no evidence for a higher farm gate price rewarding organic seed use (Herstatt 2017; AMI 2020). Yet, we observed in reality a certain share of organic seed use among carrot growers. The overall distribution of the excess willingness to pay for organic seed in comparison to the NCT price was derived from a small survey among organic carrot producers in Germany that was described in the section above. We integrated an excess WTP into the simulation model. The upper bound of the overall distribution of the excess WTP across the farm population was assumed to represent the upper bound of the innovator group. Similarly, the lower bound of the overall distribution represented the lower bound of the laggards group. The overall distribution was then divided into five overlapping symmetric triangular sub-distributions with the same width on the x-axis indicating the WTP. The modes, minima, and maxima can be found in Table 2. In the calibration process, the modes, minima, and maxima of the sub-distributions were simultaneously shifted in steps of 50 € to adapt the WTP distribution so that the model results in an organic seed use share equal to the observation level of 10%. To verify if around 10% organic seed is used, the actual values per farm agent were then randomly drawn from the triangular sub-distribution representing the farm agent's innovativeness class and the baseline model was solved. Deviations of 1% in both ways were accepted, because the benchmark of 10 % organic seed use can vary to this extent. This procedure is inspired by the calibration process of PMP models, where usually cost functions are calibrated according to observed farm areas (Heckelei et al. 2012; Howitt 1995).

Table 2: Initial distribution values of symmetric triangular distributions for WTP calibration

	Minimum	Mode	Maximum
Innovators	350	400	450
Early adopters	300	350	400
Early majority	250	300	350
Late majority	200	250	300
Laggards	150	200	250

In the GAMS code, this calibration process can be seen in lines 502-550. Calibration was conducted for each seed value using the baseline model and then applied for scenario calculation for the respective seed value.

7.3 Validation

This was conducted by comparing the model outcomes with general statistics about areas, yields, and gross margins at the farm level, as well as aggregate model results such as overall area, production amounts, and number of agents (Table 3). In most cases, the model baseline result and the observation are close. Only in the case of the seed multiplication type II, the difference in gross margins is rather large. However, as all other values seem valid, this deviation is accepted and the relative changes calculated in the scenarios seen as valuable information.

Table 3: Overview of validation indicators, real-world observations and model results

Validation indicator	Observation	Model baseline result (Av. of ten agent populations as part of sensitivity analyses)
Farm level		
Total organic carrot production in tons and hectares	2,102.5 ha, 102,418.3 tons (Destatis 2018) On 50 – 60 % of this area, carrots for the fresh market and storage are produced (own data collection)	Carrots for the fresh market and storage: 1,300 ha, 51,023.3 tons
Organic carrot seed use in Mio seed	10% organic seed use and less than 1% organic seed use from organically bred cultivars (Herstatt 2017)	9 % organic seed use, 0.3 % seed use from organically bred cultivars
Farm enterprise gross margins in €/farm enterprise	Estimated gross margin at farm enterprise level is 7,503.8 € for a crop rotation comprising mostly arable crops and 14,954.79 € for a crop rotation comprising mostly vegetable crops (KTBL 2016; AMI 2020; Destatis 2018)	The average yearly gross margin at farm enterprise level over all farm agents is 6,457.82 € with a crop rotation comprising mostly arable crops and 11,589.45 € with a crop rotation comprising mostly vegetable crops
Seed multiplication and breeding level		
Gross margin at organic carrot multiplication level in € (excluding costs for processing and packaging).	Type I: 848,025 € Type II: 5,975.2 € (own data collection)	Type I: 717,065 € Type II: 1,365 €
Breeding budget for carrots in €	Type I: 5,180,480 € Type II: 300 – 1,500 €, as only less than 5 % acquired through re-financing and 30,000 € of yearly carrot breeding budget mostly acquired through donations (own data collection)	Type I: 5,121,817 € Type II: 314 € if 10 % of sales revenue goes back into breeding. However, the breeding budget is mostly financed through alternative sources. This assumption of 10% seed sales going back into organic breeding results in a coverage of around 1% of the current yearly breeding budget in the baseline scenario.

Furthermore, to validate aspects of the model where there is a lack of real-world observations to compare them with, a structural validation can be useful. During a structural validation, stakeholders involved in the investigated problem are consulted to validate these assumptions and model results (Qudrat-Ullah 2005). Aspects such as the assumption that the excess WTP stays constant across scenarios and different scenario outcomes (e.g. the occurrence of an organic seed shortage in a derogation scenario) underwent a structural validation through expert interviews.

8. Literature

AMI. 2020. Time series data on washed fresh organic carrot whole sale prices and marketing channels, ed. Agrarmarkt Information GmbH.

Baum, Christopher F. 2006. *An introduction to modern econometrics using Stata*. College Station, U.S.: Stata Press. ISBN: 1597180130.

Bellù, Lorenzo Giovanni. 2013. Value Chain Analysis for Policy Making Methodological Guidelines and country cases for a Quantitative Approach. *EASYPol-Resources for policy making*:1-178. http://www.fao.org/3/a-at511e.pdf

Berger, Thomas, and Pepijn Schreinemachers. 2006. Creating agents and landscapes for multiagent systems from random samples. *Ecology and Society* 11 (2).

Berger, Thomas, Christian Troost, Tesfamicheal Wossen, Evgeny Latynskiy, Kindie Tesfaye, and Sika Gbegbelegbe. 2017. Can smallholder farmers adapt to climate variability, and how effective are policy interventions? . *Agricultural Economics* 48 (6):693-706. doi: https://doi.org/10.1111/agec.12367.

Bunte, F., and M. A. van Galen. 2015. Auswirkungen der Erhöhung des Mehrwertsteuersatzes für Zierpflanzenerzeugnisse.1-28. http://edepot.wur.nl/368696

Destatis. 2018. Gemüseerhebung - Anbau und Ernte von Gemüse und Erdbeeren

Galbács, Peter. 2015. *The Theory of New Classical Macroeconomics*. Contributions to Economics. Heidelberg, Germany: Springer. ISBN: 978-3-319-17577-5.

Gibbons, JM, and SJ Ramsden. 2008. Integrated modelling of farm adaptation to climate change in East Anglia, UK: Scaling and farmer decision making. *Agriculture, Ecosystems & Environment* 127 (1-2):126-134. doi: 10.1016/j.agee.2008.03.010.

Gjerdrum, Jonatan, Nilay Shah, and Lazaros G. Papageorgiou. 2010. A combined optimization and agent-based approach to supply chain modelling and performance assessment. *Production Planning & Control* 12 (1):81-88. doi: 10.1080/09537280150204013.

Grovermann, Christian, Pepijn Schreinemachers, Suthathip Riwthong, and Thomas Berger. 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. doi: 10.1016/j.ecolecon.2016.09.031.

Häring, Anna Maria. 2003. *An interactive approach to policy impact assessment for organic farms in Europe*. Hohenheim, Germany: University of Hohenheim. ISBN: 3-933403-09-X.

Heckelei, Thomas, Wolfgang Britz, and Yinan Zhang. 2012. Positive mathematical programming approaches–recent developments in literature and applied modelling. *Bio-based and Applied Economics Journal* 1 (1):109-124. doi: 10.22004/ag.econ.125722.

Herstatt, M. 2017. Economic analysis of different seed value chains for organic carrot production. In *Master Thesis*. Leuven, Belgium: KU Leuven.

Howitt, Richard E. 1995. A calibration method for agricultural economic production models. *Journal of agricultural economics* 46 (2):147-159. doi: https://doi.org/10.1111/j.1477-9552.1995.tb00762.x.

Janssen, Sander, and Martin K. van Ittersum. 2007. Assessing farm innovations and responses to policies: A review of bio-economic farm models. *Agricultural Systems* 94 (3):622-636. doi: 10.1016/j.agsy.2007.03.001.

KTBL. 2016. Webanwendung Leistungs-Kostenrechnung Pflanzenbau. https://www.ktbl.de/webanwendungen/leistungs-kostenrechnung-pflanzenbau. Accessed 01.04.2020.

Kuin, Bart. 2018. Breeding for Organic or Organic Breeding, Bejo Zaden. Paper presented at the Biofach, Nuremberg.

Kumara, Ranjit, Khurshid Alama, Vijesh V. Krishnab, and K. Srinivasa. 2012. Value Chain Analysis of Maize Seed Delivery System in Public and Private Sectors in Bihar. *Agricultural Economics Research Review* 25:387-398. http://ageconsearch.umn.edu/record/136392/files/4-Ranjit-Kumar.pdf

Mulugeta, Fikre, Jemberu Eshetu, and Olani Nikus. 2010. Seed Value Chain Analysis as a means for Sustainable Seed System: A case of farmers based seed production and marketing in Arsi Zone, Oromia Region.1-48. https://coin.fao.org/coin-static/cms/media/7/13029383782720/seed_vca.pdf

Nang'ole, Eddah, Dagmar Mithöfer, and Steven Franzel. 2011. *Review of guidelines and manuals for VCA for agricultural and forest products*. The World Agroforestry Centre. ISBN: 978-92-9059-301-0. https://dlc.dlib.indiana.edu/dlc/bitstream/handle/10535/7718/OP11160.pdf

Orsini, Stefano, Ambrogio Costanzo, Francesco Solfanelli, Raffaele Zanoli, Susanne Padel, Monika M Messmer, Eva Winter, and Freya Schaefer. 2020. Factors Affecting the Use of Organic Seed by Organic Farmers in Europe. *Sustainability* 12 (20):8540. doi: https://doi.org/10.3390/su12208540.

Qudrat-Ullah, Hassan. 2005. Structural validation of system dynamics and agent-based simulation models. *19th European Conference on Modelling and Simulation*, Riga, Latvia.

RDC. 2016. Research Data Centers of the Federal Statistical Office and Statistical Offices of the Federal States, Agrarstrukturerhebung, own calculations.

Rich, Karl M., R. Brent Ross, A. Derek Baker, and Asfaw Negassa. 2011. Quantifying value chain analysis in the context of livestock systems in developing countries. *Food Policy* 36 (2):214-222. doi: 10.1016/j.foodpol.2010.11.018.

Rogers, E.M. 2003. *Diffusion of Innovations*. 5 ed. Aufl. New York, U.S.: Simon and Schuster ISBN: 0743222091. https://www.academia.edu/download/28804464/rogers1985.pdf

Schreinemachers, Pepijn, and Thomas Berger. 2011. An agent-based simulation model of human–environment interactions in agricultural systems. *Environmental Modelling & Software* 26 (7):845-859. doi: 10.1016/j.envsoft.2011.02.004.

Syngenta. 2015. Financial Business Case for Breeding Programs. In *Educational Material*.

Troost, Christian, and Thomas Berger. 2015a. Dealing with Uncertainty in Agent-based Simulation: Farm-level Modeling of Adaptation to Climate Change in Southwest Germany. *American Journal of Agricultural Economics* 97 (3): 833-854. doi: 10.1093/ajae/aau076.

Troost, Christian, and Thomas Berger. 2015b. Process-based simulation of regional agricultural supply functions in Southwestern Germany using farm-level and agent-based models. Paper presented at the ICAE, Milan, Italy.

Appendix

A1. Stata code agent generation

```
**************************************************************************
* Do file creating 100 farm agents based on RDC of the Federal Statistical
* Office and Statistical Offices of the Federal States. 2016.
* Agrarstrukturerhebung, own calculations.
* Author: Eva Winter Year: 2021 Contact: eva.winter26@gmx.de
* This code may be used for scientific purposes when cited accordingly.
*Commercial use is prohibited.
**************************************************************************
clear

//Define input data path

global datenpfad   "xx"

//Import data

import excel "$datenpfad\final_copula_svy.xls", sheet("Sheet1") firstrow


local N = _N
set matsize `N'

local nq = 5

//Transform data to matrix format

mkmat svy_cumul_copula svy_copula      svy_C0210_5  svy_C0240_5  svy_biogemuese_5
///
svy_Net_AKh_fix_5   svy_C0210_LB svy_C0240_LB svy_biogemuese_LB ///
svy_Net_AKh_fix_LB  svy_C0210_UB svy_C0240_UB svy_biogemuese_UB ///
svy_Net_AKh_fix_UB , matrix(PERC)

matrix list PERC


save "$datenpfad\cumulative_copula.dta", replace


clear

//set number of observations (_N)

local _NAgents =100
set obs `_NAgents'

//set seed

local seednumber = 53467
set seed `seednumber'

//Define number of quantiles

local nq = 5

//Generate uniform distributional variable with which random agents
//according to the cumulative distribution function can be drawn

generate u1 = runiform()
```

```
//Generate variables again to fill in with values from matrix

local varlist = "C0210 C0240 biogemuese Net_AKh_fix"

tokenize `varlist'

foreach n of numlist 1(1)4 {

gen svy_``n''_`nq'= 0
gen svy_``n''_LB= 0
gen svy_``n''_UB= 0

}


save "$datenpfad\randomnumbers.dta", replace

//Drawing the agents out of the main copula according to the cumulative
//frequencies

local n = `_NAgents'
local mrows = rowsof(PERC) /*total rows in matrix */

forvalues ag = 1/`n' {
      local mr = `mrows' /*pointer of row for looping */

            *loop through rows in matrix to match random percentile with
            *percentiles of cumulative distribution function
            while ((PERC[`mr', 1] > u1[`ag'] | PERC[`mr', 1] == 0) & `mr' > 0){
                  local mr = `mr' -1 /*shift to next row*/
                  }
      local mr = `mr' + 1 /*avoid that local is 0*/
      replace u1 = PERC[`mr', 2]  in `ag'

      local vars   = "svy_C0210_5      svy_C0240_5  svy_biogemuese_5 ///
      svy_Net_AKh_fix_5   svy_C0210_LB svy_C0240_LB svy_biogemuese_LB ///
      svy_Net_AKh_fix_LB  svy_C0210_UB svy_C0240_UB svy_biogemuese_UB ///
      svy_Net_AKh_fix_UB"

tokenize `vars'
      foreach n of numlist 1(1)12{

       /*replace data entry with value corresponding to matching percentile*/
      replace ``n'' = PERC[`mr', `n'+2]  in `ag'
      }
      }

//Create random values from uniform distributions within the lower and
//upper bounds


local varlist = "svy_C0210 svy_C0240 svy_biogemuese svy_Net_AKh_fix"

tokenize `varlist'

foreach n of numlist 1(1)4 {

generate ``n''_uni = runiform(``n''_LB,``n''_UB)
//replace missing values with 0 (there are missing values created when
//both bounds are 0, which is quite often the case with the vars for
//wheat and triticale)

recode ``n''_uni (.=0)
}
```

```
//Generation dummy indicating if the agent is a vegetable or arable farm type

gen farmtypeint = runiform()


recode farmtypeint (0/0.85=1 "acker") (0.85/1=2 "gemuese"), gen(farmtype)

*********************************************************************
*********************************************************************


//Matching small copulae with main copula. Matching variable: Arable land area

//EXAMPLE Winter wheat (same procedure for beans, rye, green manure, education):

save "$datenpfad\until_agent_creation_full_copula.dta", replace

clear

import excel "$datenpfad\simple_copula_C0101.xls", sheet("Sheet1") firstrow

keep simplecopula_C0101_5  cumul_simplecopula_C0101_5 C0210_5      C0101_5


//Extract highest value of cumul to use as upper and lower bounds for
//generating random values to assign labour quintiles to land within in
//the land quintiles (5 labour quintiles per land quintile)
//this is the upper bound

egen cumul_scopula_C0101_max =max(cumul_simplecopula_C0101_5), by (C0210_5)

//create a variable that has always the one lower value, so that you
//can match the upper bound of the lower class as lower bound of the
//higher class

gen cumul_scopula_C0101_min = .


recode C0210_5 (1=.)(2=1) (3=2) (4=3) (5=4), gen(C0101_rank)

// create a matrix out of these to read in the values, for this,
//save the original data

save "$datenpfad\until_cumul_simple_copula.dta", replace


//drop all unnecessary lines

duplicates drop cumul_scopula_C0101_max, force


mkmat  C0210_5 cumul_scopula_C0101_max , matrix(C0101_max)


use "$datenpfad\until_cumul_simple_copula.dta", replace


local n = _N
local mrows = rowsof(C0101_max) /*total rows in matrix */

//matching lower bounds (upper bounds of lower classes)

forvalues ag = 1/`n' {
```

```
	local mr = `mrows' /*pointer of row for looping */

while ((C0101_max[`mr', 1] >= C0101_rank[`ag'] | C0101_max[`mr', 1] == 0)///
 & `mr' > 0){
			local mr = `mr' -1 /*shift to next row*/
			}
	local mr = `mr' + 1 /*avoid that local is 0*/
	replace cumul_scopula_C0101_min = C0101_max[`mr', 2]  in `ag'
}

replace cumul_scopula_C0101_min =0 if cumul_scopula_C0101_min==.

duplicates drop C0210_5,force

mkmat C0210_5 cumul_scopula_C0101_min  cumul_scopula_C0101_max,///
 matrix(C0101_bounds)

use "$datenpfad\until_agent_creation_full_copula.dta", replace

gen C0101_scopula_LB =.
gen C0101_scopula_UB =.

local n = _N
local mrows = rowsof(C0101_bounds) /*total rows in matrix */

//Matching lower and upper bound for the respective quintile

forvalues ag = 1/`n' {
	local mr = `mrows' /*pointer of row for looping */

while ((C0101_bounds[`mr', 1] >= svy_C0210_5[`ag'] | ///
C0101_bounds[`mr', 1] == 0) & `mr' > 0){
			local mr = `mr' -1 /*shift to next row*/
			}
	local mr = `mr' + 1 /*avoid that local is 0*/
	replace C0101_scopula_LB = C0101_bounds[`mr', 2]  in `ag'
	replace C0101_scopula_UB = C0101_bounds[`mr', 3]  in `ag'

}

//Draw random value

gen C0101_scopula = runiform(C0101_scopula_LB,C0101_scopula_UB) if farmtype==1
gen C0101_LB =.
gen C0101_UB =.

//Compare random value against winter wheat copula to determine quintile

save "$datenpfad\until_cumulative_distr_value.dta", replace

clear

import excel "$datenpfad\simple_copula_C0101.xls", sheet("Sheet1") firstrow

mkmat cumul_simplecopula_C0101_5 C0101_5 C0101_LB    C0101_UB, ///
matrix(C0101_quintiles)

use "$datenpfad\until_cumulative_distr_value.dta", replace

local n = _N
local mrows = rowsof(C0101_quintiles) /*total rows in matrix */

//match winter wheat quintile lower and upper bound

forvalues ag = 1/`n' {
```

```
       local mr = `mrows' /*pointer of row for looping */

while ((C0101_quintiles[`mr', 1] >= C0101_scopula[`ag'] | ///
C0101_quintiles[`mr', 1] == 0) & `mr' > 0){
                    local mr = `mr' -1 /*shift to next row*/
                    }
       local mr = `mr' + 1 /*avoid that local is 0*/
       replace C0101_scopula = C0101_quintiles[`mr', 2]  in `ag'
       replace C0101_LB = C0101_quintiles[`mr', 3]  in `ag'
       replace C0101_UB = C0101_quintiles[`mr', 4]  in `ag'

}


//draw individual winter wheat variable values


gen C0101_uni = runiform(C0101_LB,C0101_UB) if farmtype==1

**************************************************************


//After verification: Generate RHS Variables
**********************************************************************

gen biogemuese_g_total = svy_biogemuese_uni if farmtype==2

egen biogemuese_a_total = rowtotal(svy_biogemuese_uni C0101_uni ///
C0104_uni    C0123_uni    C0132_uni) if farmtype==1

**********************************************************************

//Create Innovation score


//Linear regression results from LIVESEED farmer survey

gen farmsize_coeff =  -.0664367
gen farmtype_coeff = -.1336177
gen edu_coeff =  .2084144
gen CONSTANT = .9093093
gen ERROR_TERM = rnormal(0,.2746588)

//Recoding the two variables edu_uni and farmtype to dummies ///
//and take the ln of farm size


recode farmtype (1=0 "acker") (2=1 "gemuese"), gen(farmtype_dummy)
recode edu_uni (1=0 "none") (2=1 "education") (3=1 "education"), gen(edu_dummy)
gen ln_svy_C0240_uni = ln(svy_C0240_uni)

//Sumproduct of coefficients and values in the agent population

gen predicted_org_seed_use = ln_svy_C0240_uni*farmsize_coeff ///
+farmtype_dummy*farmtype_coeff+ edu_dummy*edu_coeff + CONSTANT + ERROR_TERM


//Create inverse of dependent variable

gen einsminuspredict_org_seed_use = 1- predicted_org_seed_use

sort einsminuspredict_org_seed_use

cumul einsminuspredict_org_seed_use , generate(cum_pred_ord_seed_use) equal
```

```
//Create innovativeness score

gen innov_score = .
replace innov_score = 1 if cum_pred_ord_seed_use  <= 0.025
replace innov_score = 2 if cum_pred_ord_seed_use  > 0.025 & ///
 cum_pred_ord_seed_use <= 0.135
replace innov_score = 3 if cum_pred_ord_seed_use  > 0.135 & ///
 cum_pred_ord_seed_use <= 0.345
replace innov_score = 4 if cum_pred_ord_seed_use  > 0.345 & ///
 cum_pred_ord_seed_use <= 0.68
replace innov_score = 5 if cum_pred_ord_seed_use  > 0.68



//Randomly distribute the 4 combinations of mechanisation and plot size
// according to the assigned vegetable plot size
generate u2 = runiform(0, 0.181818182) if biogemuese_a  > 0 & ///
 biogemuese_a <= 0.5
replace u2 = runiform(0, 0.454545455) if biogemuese_a > 0 & ///
 biogemuese_a > 0.5 & biogemuese_a <= 1
replace u2 = runiform(0, 0.727272727) if biogemuese_a > 0 & ///
 biogemuese_a > 1 & biogemuese_a <= 2
replace u2 = runiform(0, 1) if biogemuese_a > 0 &  biogemuese_a > 2

generate u3 = runiform(0, 0.181818182) if biogemuese_g  > 0 & ///
 biogemuese_g <= 0.5
replace u3 = runiform(0, 0.454545455) if biogemuese_g > 0 & ///
 biogemuese_g > 0.5 & biogemuese_g <= 1
replace u3 = runiform(0, 0.727272727) if biogemuese_g > 0 & ///
 biogemuese_g > 1 & biogemuese_g <= 2
replace u3 = runiform(0, 1) if biogemuese_g > 0 & ///
 biogemuese_g > 2


//Create a matrix with the 4 combinations of mechanisation and plot size,
//as well as all variable costs, labour need, seed prices and seed needs,
//and yields



save "C:\Users\eva.winter\Dropbox\PhD\liveseed\Simulationsmodell ///
\creation farm population\randomagents.dta", replace


clear



//Matching technical coefficents etc. for arable crop rotation and
//vegetable crop rotation, Example arable crop rotation:

import excel "C:\Users\eva.winter\Dropbox\PhD\liveseed\Simulationsmodell ///
\creation farm population\inputsKTBLandother.xlsx", sheet("Acker") ///
cellrange(A1:AM12) firstrow

mkmat Plotsizeindicator_a  Mech_Grad_a  Plotsize_a   Varcostsperha_wcar_a ///
      VarAkhperha_wcar_a  FixAkhperha_wcar_a  SeedorgMioperha_wcar_a ///
      SeedpriceMioorg_wcar_a    SeedconvMioperha_wcar_a   SeedpriceMioconv_wcar_a
///
      Price_wcar_a Yield_wcar_a Varcostsperha_bean  VarAkhperha_bean ///
      FixAkhperha_bean Price_bean      Yield_bean   Varcostsperha_wheat ///
      VarAkhperha_wheat FixAkhperha_wheat Varcosts_Festmist_ha_wheat    ///
      Festmistt_ha_wheat  Price_wheat   Yield_wheat   Varcostsperha_rye ///
```

```
       VarAkhperha_rye     FixAkhperha_rye     Varcosts_Festmist_ha_rye ///
       Festmistt_ha_rye    Price_rye    Yield_rye    Varcostsperha_clover2 ///
       VarAkhperha_clover2 FixAkhperha_clover2 Varcostsperha_onion ///
       VarAkhperha_onion   FixAkhperha_onion   Price_onion   Yield_onion, ///
       matrix(plotsize_a)

matrix list plotsize_a



//Match them with u2, your randomly generated number that will
//choose the plot size for the agents

use "C:\Users\eva.winter\Dropbox\PhD\liveseed\Simulationsmodell ///
\creation farm population\randomagents.dta", replace

//create empty variables to be filled with matrix contents

local vars = "Plotsizeindicator_a      Mech_Grad_a  Plotsize_a   ///
Varcostsperha_wcar_a      VarAkhperha_wcar_a  FixAkhperha_wcar_a  ///
SeedorgMioperha_wcar_a    SeedpriceMioorg_wcar_a    SeedconvMioperha_wcar_a   ///
SeedpriceMioconv_wcar_a   Price_wcar_a Yield_wcar_a ///
Varcostsperha_bean  VarAkhperha_bean    FixAkhperha_bean ///
Price_bean   Yield_bean   Varcostsperha_wheat VarAkhperha_wheat ///
FixAkhperha_wheat   Varcosts_Festmist_ha_wheat Festmistt_ha_wheat ///
Price_wheat  Yield_wheat  Varcostsperha_rye   VarAkhperha_rye     ///
FixAkhperha_rye     Varcosts_Festmist_ha_rye  Festmistt_ha_rye ///
Price_rye    Yield_rye    Varcostsperha_clover2     VarAkhperha_clover2 ///
FixAkhperha_clover2 Varcostsperha_onion VarAkhperha_onion ///
FixAkhperha_onion   Price_onion  Yield_onion"

tokenize `vars'
       foreach n of numlist 1(1)39{


       gen ``n'' = .
       }


//grow through matching and replacing procedure for u2 ak GEMUESE MIT ACKER

local n = `_NAgents'
local mrows = rowsof(plotsize_a)

forvalues ag = 1/`n' {
local mr = `mrows'
while ((plotsize_a[`mr', 1] > u2[`ag'] | plotsize_a[`mr', 1] == 0) & `mr' > 0){
                    local mr = `mr' -1
                    }
       local mr = `mr' + 1 /*avoid that local is 0*/
       replace u2 = plotsize_a[`mr', 1]  in `ag'

local vars = "Plotsizeindicator_a      Mech_Grad_a  Plotsize_a   ///
Varcostsperha_wcar_a      VarAkhperha_wcar_a  FixAkhperha_wcar_a ///
SeedorgMioperha_wcar_a    SeedpriceMioorg_wcar_a    SeedconvMioperha_wcar_a ///
SeedpriceMioconv_wcar_a   Price_wcar_a Yield_wcar_a ///
Varcostsperha_bean  VarAkhperha_bean FixAkhperha_bean Price_bean ///
Yield_bean Varcostsperha_wheat   VarAkhperha_wheat FixAkhperha_wheat ///
Varcosts_Festmist_ha_wheat Festmistt_ha_wheat Price_wheat ///
Yield_wheat  Varcostsperha_rye VarAkhperha_rye      FixAkhperha_rye     ///
Varcosts_Festmist_ha_rye Festmistt_ha_rye Price_rye  Yield_rye ///
Varcostsperha_clover2     VarAkhperha_clover2 FixAkhperha_clover2 ///
Varcostsperha_onion VarAkhperha_onion FixAkhperha_onion     Price_onion
       Yield_onion"
tokenize `vars'
```

```
        foreach n of numlist 1(1)39{

         /*replace data entry with value corresponding to matching percentile*/
        replace ``n'' = plotsize_a[`mr', `n']  in `ag'
        }
        }



save "C:\Users\eva.winter\Dropbox\PhD\liveseed\Simulationsmodell ///
\creation farm population\randomagents.dta", replace
```

A2. GAMS Model code

```
**************************************************************************
* 3 stages of the value chain, linked by production amounts,
* handed down the stages, feed-back loop after each modelling
* period (yearly steps)
* Author: Eva Winter Year: 2021 Contact: eva.winter26@gmx.de
* This code may be used for scientific purposes when cited accordingly.
*Commercial use is prohibited.
**************************************************************************

*Value chain stage breeding
****************************

*Declaring sets

Sets
con_br                    Constraints at breeding level
act_br                    Activities at breeding level
yr                        Years
ag_br                     Breeding agents
currentag_br(ag_br)       Dynamic subset for individual agents
;

*Assigning sets

Set yr  /yr1*yr10/;
alias (yry,yr);
alias (yr, run) ;

*loading sets

$call gdxxrw.exe Sets_mult_ag_equloop_breeder_inclagents_create_\\\
agents4_br_mu_carrots_twotypes_FDZ.xlsx Set=con_br Rng=Tabelle1!B1\\\
Cdim=1 Set=act_br Rng=Tabelle1!B2 Cdim=1  Set=yr Rng=Tabelle1!B3\\\
Cdim=1 Set=ag_br Rng=Tabelle1!B4 Cdim=1
$gdxin Sets_mult_ag_equloop_breeder_inclagents_create_agents4_br_mu_\\\
carrots_twotypes_FDZ.gdx
$load con_br
$load act_br
$load ag_br

alias (ag_br, aga_br);

*Subset for reporting


Set  seedrests_org_wcar_br(con_br) /  ExpmaxsaleOrg_br, ExpmaxsaleOrg_cult_br/
;

*Declaring Parameters

Parameters
RHS_ag_br(con_br,yr,ag_br)                Right hand side values
a_ag_br(con_br,yry,act_br,yr,ag_br)       Technical coefficients
GM_ag_br(act_br,yr,ag_br)                 Gross margins
Boughtvar2_br_s(ag_br,run)                s=supply  (Basic organic seed (var2)\\\
                                          amount sold to seed multiplier)
Expmaxsalevar2coef_br                     Growth expectation factor (var2)
Expmaxsalevar2Min_br                      Minimum amount basic seed production
Boughtvar3_br_s(ag_br,run)                s=supply (Basic organic cultivar (var3)\\\
                                          seed amount sold to seed multiplier)
Expmaxsalevar3coef_br                     Growth expectation factor (var3)
Expmaxsalevar3Min_br                      Minimum amount basic seed production
```

```
Adaptive_Expectations_coeff              Production reserve factor
Upper_bound_Expmaxsalevarcoef            Upper bound of growth expectation\\\
                                         Factor;

*Assigning Parameters

GM_ag_br("TGM_br","yr10",ag_br)= 1;
Expmaxsalevar2coef_br = 2.5;
Expmaxsalevar2Min_br = 40;
Expmaxsalevar3coef_br = 2.5;
Expmaxsalevar3Min_br = 40;
Adaptive_Expectations_coeff=1.5;
Upper_bound_Expmaxsalevarcoef=2;

$call gdxxrw.exe Parameters_mult_ag_equloop_breeder_inclagents_create_\\\
agents4_br_mu_carrots_twotypes_FDZ.xlsx Par=RHS_ag_br Rng=Tabelle1!B1\\\
rdim=3 Par=a_ag_br Rng=Tabelle2!A1 rdim=2 cdim=3
$gdxin Parameters_mult_ag_equloop_breeder_inclagents_create_agents4_\\\
br_mu_carrots_twotypes_FDZ.gdx
$load RHS_ag_br
$load a_ag_br

*Declaring variables

Positive Variable x_br(act_br,yr,ag_br) Solution variables;

Variable
zw_br(yr,ag_br) intermediate objective
obj_br objective value;

*Declaring equations

Equation ziel_br objective function over all years and agents;
Equation limits_br(con_br,yr, ag_br) constraint equations;
Equation zwziel_br(yr,ag_br) intermediate objective function per year and agent;

*Assigning equations

ziel_br .. sum((yr,currentag_br), zw_br(yr,currentag_br)) =E= obj_br;

zwziel_br(yr,ag_br)      .. SUM(act_br, GM_ag_br(act_br,yr,ag_br) *\\\
x_br(act_br,yr,ag_br)) =E= zw_br(yr,ag_br);

limits_br(con_br,yry,ag_br) .. SUM((act_br,yr),\\\
a_ag_br(con_br,yry,act_br,yr,ag_br) * x_br(act_br,yr,ag_br)) =L=\\\
RHS_ag_br(con_br,yry,ag_br);

*Model definition

Model LPmodel_br /ziel_br, zwziel_br, limits_br/;

*Assigning starting values of solution variables

x_br.fx(act_br,yr,ag_br)=0;

**Value chain stage seed multiplication
*****************************************

*Declaring sets

Sets
con_mu                      Constraints at multiplication level
act_mu                      Activities at multiplication level
ag_mu                       Seed multiplication agents
currentag_mu(ag_mu)         Dynamic subset for individual agents
```

```
;

*Loading and assigning sets

$call gdxxrw.exe Sets_mult_ag_equloop_multiplier_inclagents_create_\\\
agents4_mu_carrots_twotypes_FDZ.xlsx Set=con_mu Rng=Tabelle1!B1 Cdim=1\\\
Set=act_mu Rng=Tabelle1!B2 Cdim=1  Set=yr Rng=Tabelle1!B3 Cdim=1\\\
Set=ag_mu Rng=Tabelle1!B4 Cdim=1
$gdxin Sets_mult_ag_equloop_multiplier_inclagents_create_agents4_\\\
mu_carrots_twotypes_FDZ.gdx
$load con_mu
$load act_mu
$load ag_mu

alias (ag_mu, aga_mu);

*Subsets for reporting

Sets
accounts_mu(con_mu)/accountopen_mu/
con_yield_mu(con_mu)/yield_conv_wcar_mu,yield_org_wcar_mu,\\\
yield_org_cult_wcar_mu/
grow_mu(act_mu)/multiply_conv_wcar_mu,multiply_org_wcar_mu,\\\
multiply_org_cult_wcar_mu/
seedrests_org_wcar_mu(con_mu)/Expmaxsaleorg_wcar_mu,\\\
Expmaxsaleorg_cult_wcar_mu/
;

*Declaring Parameters

Parameters
RHS_ag_br(con_br,yr,ag_br)                Right hand side values
a_ag_br(con_br,yry,act_br,yr,ag_br)       Technical coefficients
GM_ag_br(act_br,yr,ag_br)                 Gross margins
Boughtvar2_br_s(ag_br,run)                s=supply  (Basic organic seed (var2)\\\
                                          amount sold to seed multiplier)
Expmaxsalevar2coef_br                     Growth expectation factor (var2)
Expmaxsalevar2Min_br                      Minimum amount basic seed production
Boughtvar3_br_s(ag_br,run)                s=supply (Basic organic cultivar (var3)\\\
                                          seed amount sold to seed multiplier)
Expmaxsalevar3coef_br                     Growth expectation factor (var3)
Expmaxsalevar3Min_br                      Minimum amount basic seed production
Adaptive_Expectations_coeff               Production reserve factor
Upper_bound_Expmaxsalevarcoef             Upper bound of growth expectation\\\
                                          factor
RHS_ag_mu(con_mu,yr,ag_mu)                Right hand side values
a_ag_mu(con_mu,yry,act_mu,yr,ag_mu)       Technical coefficients
GM_ag_mu(act_mu,yr,ag_mu)                 Gross margins
Quantityvar2_mu(run)                      Par storing Produced basic seed var2
Obj_inclvar2_mu(ag_mu)                    Par storing Objective value including\\\
                                          option var2
Quantityvar3_mu(run)                      Par storing Produced basic seed var3
Obj_inclvar3_mu(ag_mu)                    Par storing Objective value including\\\
                                          option var3
Obj_exclvar2_mu(ag_mu)                    Par storing Objective value excluding\\\
                                          option var2 and var3
x_inclvar2_mu(aga_mu)                     Par storing Used basic seed var2 by\\\
                                          seed multiplier agent
x_inclvar3_mu(aga_mu)                     Par storing Used basic seed var3 by\\\
                                          seed multiplier agent
x_inclvar2_total_mu                       Par storing Total used basic seed var2\\\
                                          by seed multipliers
x_inclvar3_total_mu                       Par storing Total used basic seed var3\\\
                                          by seed multipliers
Boughtvar2_mu_d(run)                      d:demanded amount (from breeder)
```

```
Boughtvar3_mu_d(run)
Boughtvar2_mu_s(ag_mu,run)                s: sold amount (effectively sold to\\\
                                          farmer)
Boughtvar3_mu_s(ag_mu,run)
Expected_Boughtvar2_mu_s(ag_mu,run)       Average bought basic seed var2 current\\\
                                          and last year
Expected_Boughtvar3_mu_s(ag_mu,run)       Average bought basic seed var3 current\\\
                                          and last year
Expmaxsalevar2coef_mu                     Growth expectation factor (var2)
Expmaxsalevar3coef_mu                     Growth expectation factor (var3)
Expmaxsalevar2Min_mu                      Minimum amount seed production (var2)
Expmaxsalevar3Min_mu                      Minimum amount seed production (var3)

*Assigning Parameters

GM_ag_mu("TGM_mu","yr10",ag_mu)= 1;
Boughtvar2_br_s(ag_br,run)=0;
Boughtvar3_br_s(ag_br,run)=0;

Expmaxsalevar2coef_mu =2.5;
Expmaxsalevar3coef_mu =2.5;
Expmaxsalevar2Min_mu = 50 ;
Expmaxsalevar3Min_mu = 50 ;

$call gdxxrw.exe Parameters_mult_ag_equloop_multiplier_inclagents_create_\\\
agents4_mu_carrots_twotypes_FDZ.xlsx Par=RHS_ag_mu Rng=Tabelle1!B1 rdim=3\\\
Par=a_ag_mu Rng=Tabelle2!A1 rdim=2 cdim=3
$gdxin Parameters_mult_ag_equloop_multiplier_inclagents_create_agents4\\\
_mu_carrots_twotypes_FDZ.gdx
$load RHS_ag_mu
$load a_ag_mu

*Declaring variables

Positive Variable x_mu(act_mu,yr,ag_mu) Solution variables;

Variable
zw_mu(yr,ag_mu) intermediate objective
obj_mu objective value;

*Declaring equations

Equation ziel_mu objective function over all years and agents;
Equation limits_mu(con_mu,yr, ag_mu) constraint equations;
Equation zwziel_mu(yr,ag_mu)intermediate objective function per year and agent;

*Assigning equations

ziel_mu .. sum((yr,currentag_mu), zw_mu(yr,currentag_mu)) =E= obj_mu;
zwziel_mu(yr,ag_mu)       .. SUM(act_mu, GM_ag_mu(act_mu,yr,ag_mu)\\\
 * x_mu(act_mu,yr,ag_mu)) =E= zw_mu(yr,ag_mu);
limits_mu(con_mu,yry,ag_mu) .. SUM((act_mu,yr),\\\
a_ag_mu(con_mu,yry,act_mu,yr,ag_mu) * x_mu(act_mu,yr,ag_mu))\\\
=L= RHS_ag_mu(con_mu,yry,ag_mu);

*Model definition

Model LPmodel_mu /ziel_mu,zwziel_mu ,limits_mu/;

*Assigning starting values of solution variables

x_mu.fx(act_mu,yr,ag_mu)=0;

**Value chain stage farmers
*****************************************
```

```
*transform excel input files containing sets and R and A parameters of
*farm agents into gdx-files

$call bash agtogdx.sh
$call gdxxrw.exe ..\GAMS_Input\Sets_fa.xlsx o=..\GAMS_Input\Sets_fa.gdx\\\
DSet=ag_fa Rng=A1 Cdim=1 Rdim=0  DSet=con_fa Rng=A2 Cdim=1 Rdim=0\\\
DSet=act_fa Rng=A3 Cdim=1 Rdim=0     DSet=yr Rng=A4 Cdim=1 Rdim=0

*Declaring sets

set ag_fa Farm agents;
set con_fa Constraints at farm level;
set act_fa Activities at farm level;

*Loading and assigning sets

$gdxin ..\GAMS_Input\Sets_fa.gdx
$load ag_fa
$load con_fa
$load act_fa
$gdxin

*Declaring Parameters

Parameter RHS_ag_fa_load(ag_fa,con_fa,yr) Right hand side values\\\
(intermediate for import);
Parameter RHS_ag_fa(con_fa,yr,ag_fa)Right hand side values;

Parameter a_ag_fa_load(ag_fa,con_fa,yry,act_fa,yr) Technical coefficients\\\
(intermediate for import) ;
Parameter a_ag_fa(con_fa,yry,act_fa,yr,ag_fa) Technical coefficients;

*Loading and assigning parameters

$gdxin ..\GAMS_Input\Data_RHS_ag_fa.gdx
$load RHS_ag_fa_load
$gdxin

$gdxin ..\GAMS_Input\Data_A_ag_fa.gdx
$load a_ag_fa_load
$gdxin

RHS_ag_fa(con_fa,yr,ag_fa) = RHS_ag_fa_load(ag_fa,con_fa,yr);
a_ag_fa(con_fa,yry,act_fa,yr,ag_fa) = a_ag_fa_load(ag_fa,con_fa,yry,act_fa,yr);

Sets

currentag_fa(ag_fa)           Dynamic subset for individual agents
;

alias (ag_fa, aga_fa);

Parameters
GM_ag_fa(act_fa,yr,ag_fa)    Gross margins

Obj_inclvar2_fa(ag_fa)       Par storing Objective value including option var2
Obj_inclvar3_fa(aga_fa)      Par storing Objective value including option var3
Obj_exclvar2_fa(ag_fa)       Par storing Objective value excluding option var2\\\
                             and var3
Quantityvar2_fa(run)         Par storing Produced seed var2
Quantityvar3_fa(run)         Par storing Produced seed var3
x_inclvar2_fa(aga_fa)        Par storing Used seed var2 by farmer agent
x_inclvar3_fa(aga_fa)        Par storing Used seed var3 by farmer agent
x_inclvar2_total_fa          Par storing Total used seed var2 by farmers
```

```
x_inclvar3_total_fa          Par storing Total used seed var3 by farmers
Boughtvar2_fa_d(run)         d:demanded amount (from seed multiplicators)
Boughtvar3_fa_d(run)
;

*Assigning Parameters

Boughtvar2_mu_s(ag_mu,run)=0;
Boughtvar3_mu_s(ag_mu,run)=0;
GM_ag_fa("TGM_fa","yr10",ag_fa)= 1 ;

*Sets and Parameters for farm gate prices:
***************************************************
Sets market_fa(act_fa)/market_conv_wcar_fa,market_salad_fa,market_leak_fa,\\\
market_cab_fa,market_beans_fa,market_wheat_fa,market_rye_fa, market_onion_fa/
market_org_wcar_fa(act_fa)/market_org_wcar_fa,market_org_cult_wcar_fa/;
Parameter
price_a_ag_fa(market_fa,yry)      Par storing randomly generated farm gate\\\
                                  prices within triangular distributions
mean_price_fa(market_fa)          Par storing mean of farm gate prices
lb_price_fa(market_fa)            Par storing lower bound of farm gate prices
ub_price_fa(market_fa)            Par storing upper bound of farm gate prices
;
$call gdxxrw.exe ..\GAMS_Input\prices_fa_FDZ.xlsx o=..\GAMS_\\\
Input\prices_fa_FDZ.gdx Par=mean_price_fa Rng=Tabelle1!A2 cdim=1\\\
Par=lb_price_fa Rng=Tabelle2!A2 cdim=1  Par=ub_price_fa Rng=Tabelle3!A2  cdim=1
$gdxin ..\GAMS_Input\prices_fa_FDZ.gdx
$load mean_price_fa
$load lb_price_fa
$load ub_price_fa

*Sets and Parameters for yields at farm level across crop rotations:
*******************************************************************
Sets
grow_fa(act_fa)/grow_conv_wcar_v_fa,grow_conv_wcar_a_fa,\\\
grow_org_cult_wcar_v_fa,grow_org_cult_wcar_a_fa,grow_salad_fa,grow_leak_fa,\\\
grow_cab_fa,grow_bean_fa,grow_wheat_fa,grow_rye_fa,grow_onion_fa/
grow_org_wcar_fa(act_fa)/grow_org_wcar_v_fa,grow_org_wcar_a_fa/
con_yield_fa(con_fa)/yield_conv_wcar_fa,yield_org_cult_wcar_fa,\\\
yield_salad_fa,yield_leak_fa,yield_cab_fa,yield_bean_fa,yield_wheat_fa,\\\
yield_rye_fa , yield_onion_fa/
;

Parameter

yield_a_ag_fa(ag_fa,con_yield_fa,grow_fa,yry)    Par storing randomly generated\\\
                                                 crop yields within triangular\\\
                                                 distributions
lb_yield_fa(con_yield_fa,grow_fa)                Par storing mean of crop yields
mean_yield_fa(con_yield_fa,grow_fa)              Par storing lower bound of\\\
                                                 crop yields
ub_yield_fa(con_yield_fa,grow_fa)                Par storing upper bound of\\\
                                                 crop yields
$call gdxxrw.exe ..\GAMS_Input\yields_fa_FDZ.xlsx o=..\GAMS_Input\yields_\\\
fa_FDZ.gdx Par=mean_yield_fa Rng=Tabelle1!A2 rdim= 1 cdim=1\\\
Par=lb_yield_fa Rng=Tabelle2!A2 rdim= 1 cdim=1  Par=ub_yield_fa\\\
Rng=Tabelle3!A2 rdim= 1 cdim=1
$gdxin ..\GAMS_Input\yields_fa_FDZ.gdx
$load mean_yield_fa
$load lb_yield_fa
$load ub_yield_fa
;

*Setting starting values working capital per agent according to land area (SOURCE:
XXX)
```

```
****************************************************************************
* calibration factor:
Parameter
calibrate_starting_capital
;
calibrate_starting_capital=0;
loop(ag_fa,
  if(RHS_ag_fa("land_v_fa","yr1",ag_fa)+RHS_ag_fa("land_a_fa","yr1",ag_fa)<21,
  RHS_ag_fa("accountopen_fa","yr1",ag_fa)= calibrate_starting_capital\\\
  +1436 *(RHS_ag_fa("land_v_fa","yr1",ag_fa)+RHS_ag_fa("land_a_fa","yr1",ag_fa))

  elseif RHS_ag_fa("land_v_fa","yr1",ag_fa)\\\
  +RHS_ag_fa("land_a_fa","yr1",ag_fa)>20 and\\\
  RHS_ag_fa("land_v_fa","yr1",ag_fa)+RHS_ag_fa("land_a_fa","yr1",ag_fa)<31,
  RHS_ag_fa("accountopen_fa","yr1",ag_fa)= calibrate_starting_capital\\\
  +897 *(RHS_ag_fa("land_v_fa","yr1",ag_fa)+RHS_ag_fa("land_a_fa","yr1",ag_fa))

  elseif RHS_ag_fa("land_v_fa","yr1",ag_fa)\\\
  +RHS_ag_fa("land_a_fa","yr1",ag_fa)>30 and\\\
  RHS_ag_fa("land_v_fa","yr1",ag_fa)+RHS_ag_fa("land_a_fa","yr1",ag_fa)<51,
  RHS_ag_fa("accountopen_fa","yr1",ag_fa)= calibrate_starting_capital+534*\\\
  (RHS_ag_fa("land_v_fa","yr1",ag_fa)+RHS_ag_fa("land_a_fa","yr1",ag_fa))

  elseif RHS_ag_fa("land_v_fa","yr1",ag_fa)+RHS_ag_fa("land_a_fa","yr1",ag_fa)>50,
  RHS_ag_fa("accountopen_fa","yr1",ag_fa)= calibrate_starting_capital\\\
  +571 *(RHS_ag_fa("land_v_fa","yr1",ag_fa)+RHS_ag_fa("land_a_fa","yr1",ag_fa));
  );
);

*Additional subsets for report parameter
Sets
accounts_fa(con_fa)/accountopen_fa/
seedrests_org_wcar_fa(con_fa)/seedrest_org_wcar_fa,seedrest_org_cult_wcar_fa/
;

*Sets and Parameters for Innovation diffusion:
************************************************

Set
innov_ag_fa(ag_fa)            Dynamic set containing innovative agents
;

Parameter
Diff_innov_score(ag_fa)    Parameter containing innovation score per agent
Innovation_diffusion       Par storing the current percentage of agents having\\\
                           adopted the innovation
Innov_level                Par storing the last unlocked innovation segment\\\
                           of which agents can adopt the innovation
Org_seed_user_counter      Par storing the absolute number of agents having\\\
                           adopted the innovation in the last unlocked segment
Org_seed_group_counter     Par storing the absolute number of agents that can\\\
                           adopt the innovation in the last unlocked segment
;

$call gdxxrw.exe ..\GAMS_Input\Innovation_score.xlsx o=..\GAMS_Input\\\
\Innovation_score.gdx Par=Diff_innov_score Rng=A1 rdim= 1
$gdxin ..\GAMS_Input\Innovation_score.gdx
$load Diff_innov_score

*Initiation value of unlocked innovation segments

Innov_level=1;

*Declaring Parameters for scaling factors

```

```
Parameter

Agents_Geographical_Area_fa      Par containing number of observed farms in\\\
                                 chosen geographical region
Agents_Model_fa                  Par containing number of agents in model
Market_size_br                   Par containing market size breeder type 1 in\\\
                                 ha of the agricultural crop that can be\\\
                                 planted with the produced seed
Market_size2_br                  Par containing market size breeder type 2 in\\\
                                 ha of the agricultural crop that can be\\\
                                 planted with the produced seed
Market_size_fa                   Par containing market size of observed farms\\\
                                 in geographical area in ha
Scaling_Agents_br                Coefficient of Agents_Geographical_Area_fa\\\
                                 and Agents_Model_fa
Scaling_Agents_fa                Coefficient of Agents_Model_fa and\\\
                                 Agents_Geographical_Area_fa
Scaling_market_size_br           Coefficient of Market_size_br and\\\
                                 Market_size_fa
Scaling_market_size_fa           Coefficient of Market_size_fa and\\\
                                 Market_size_br
Scaling_factor_br                Scaling from farmer to breeder1
Scaling_factor_fa                Scaling from breeder1 to farmer
Scaling_factor2_br               Scaling from farmer to breeder2
Scaling_factor2_fa               Scaling from breeder2 to farmer
;

*Assigning Parameters for scaling factors

Agents_Geographical_Area_fa=325;
Agents_Model_fa=100;
Market_size_br=1505;
Market_size2_br=1000;
Market_size_fa=1260;

Scaling_Agents_br=  Agents_Geographical_Area_fa/Agents_Model_fa;
Scaling_Agents_fa=  Agents_Model_fa/Agents_Geographical_Area_fa;

Scaling_market_size_br=   Market_size_br/Market_size_fa;
Scaling_market_size_fa=   Market_size_fa/Market_size_br;

Scaling_factor_br = Agents_Geographical_Area_fa/Agents_Model_fa *\\\
Market_size_br/Market_size_fa   ;
Scaling_factor_fa = Agents_Model_fa/Agents_Geographical_Area_fa *\\\
Market_size_fa/Market_size_br    ;
Scaling_factor2_br = Agents_Geographical_Area_fa/Agents_Model_fa *\\\
Market_size2_br/Market_size_fa   ;
Scaling_factor2_fa = Agents_Model_fa/Agents_Geographical_Area_fa *\\\
Market_size_fa/Market_size2_br    ;

***************************************************************
*************************Excess willingness to pay: *********
***************************************************************
*Declaring parameter for calibration of Excess WTP

Parameter
Motivation_premium_organic_seed_use(ag_fa)
addition_premium
;

loop(ag_fa,
  if(Diff_innov_score(ag_fa)=1,
  Motivation_premium_organic_seed_use(ag_fa) = randTriangle(350,400,450)

  elseif Diff_innov_score(ag_fa)=2,
```

```
    Motivation_premium_organic_seed_use(ag_fa) = randTriangle(300,350,400)

    elseif Diff_innov_score(ag_fa)=3,
    Motivation_premium_organic_seed_use(ag_fa) = randTriangle(250,300,350)

    elseif Diff_innov_score(ag_fa)=4,
    Motivation_premium_organic_seed_use(ag_fa) = randTriangle(200,250,300)

    elseif Diff_innov_score(ag_fa)=5,
    Motivation_premium_organic_seed_use(ag_fa) = randTriangle (150,200,250);
    );
  );

  *Defining additional Excess WTP (Calibration process)

  *Kalibration seed 1xxx:
  addition_premium=55;

  *Calculation Excess WTP

  Motivation_premium_organic_seed_use(ag_fa)    =\\\
  Motivation_premium_organic_seed_use(ag_fa) + addition_premium;

  *Substracting Excess WTP from organic seed prices

  loop(ag_fa,

    a_ag_fa("accountopen_fa", yry, "buyseed_org_wcar_fa",yry, ag_fa) =\\\
    max(a_ag_fa("accountopen_fa", yry, "buyseed_org_wcar_fa",yry, ag_fa) - \\\
    Motivation_premium_organic_seed_use(ag_fa),50);

    a_ag_fa("accountopen_fa", yry, "buyseed_org_cult_wcar_fa",yry, ag_fa) =\\\
    max(a_ag_fa("accountopen_fa", yry, "buyseed_org_cult_wcar_fa",yry, ag_fa)\\\
    - Motivation_premium_organic_seed_use(ag_fa),50);
  );

  ****************************************************
  *Scenario Subsidy or premium price on organic seed use *
  ****************************************************
  *Declaring Scenario Parameters:

  Parameter

  Subsidy_fa       Hectare bound subsidy for organic seed use
  Price_premium    Price premium for organic seed use
  ;

  *Defining Scenario parameters:

  Subsidy_fa=150;
  Price_premium=5;

  *Declaring variables

  Positive Variable x_fa(act_fa,yr,ag_fa) Solution variables;

  Variable
  zw_fa(yr,ag_fa)  intermediate objective
  obj_fa           objective value;

  Equation ziel_fa                         objective function over all years\\\
                                           and agents;
  Equation limits_fa(con_fa,yr, ag_fa)     constraint equations;
  Equation zwziel_fa(yr,ag_fa)             intermediate objective function per\\\
                                           year and agent;
```

```
583
584 *Assigning equations
585
586 ziel_fa .. sum((yr,currentag_fa), zw_fa(yr,currentag_fa)) =E= obj_fa;
587 zwziel_fa(yr,ag_fa)      .. SUM(act_fa, GM_ag_fa(act_fa,yr,ag_fa) *\\\
588 x_fa(act_fa,yr,ag_fa))=E= zw_fa(yr,ag_fa);
589 limits_fa(con_fa,yry,ag_fa) .. SUM((act_fa,yr),\\\
590 a_ag_fa(con_fa,yry,act_fa,yr,ag_fa) * x_fa(act_fa,yr,ag_fa))  =L=\\\
591 RHS_ag_fa(con_fa,yry,ag_fa);
592
593 *Model definition
594
595 Model LPmodel_fa/ziel_fa,zwziel_fa, limits_fa/;
596
597 *Assigning starting values of solution variables
598
599 x_fa.fx(act_fa,yr,ag_fa)=0;
600
601 *Beginning of yearly loop ("run") over all actors
602
603 Loop(run,
604 *Randomly drawing yearly farm gate prices for farm agents. Assignment to
605 *farm agents will take place within the farm agent loops
606 loop(yry,
607   price_a_ag_fa(market_fa,yry)   =   randTriangle\\\
608   (lb_price_fa(market_fa), mean_price_fa(market_fa),ub_price_fa(market_fa));
609 );
610
611 *Randomly drawing yearly crop yields for farm agents. Assignment to farm agents
612 *will take place within the farm agent loops
613
614 loop((ag_fa,yry),
615   yield_a_ag_fa(ag_fa,con_yield_fa,grow_fa,yry)= randTriangle\\\
616   (lb_yield_fa(con_yield_fa,grow_fa), mean_yield_fa(con_yield_fa,grow_fa),\\\
617   ub_yield_fa(con_yield_fa,grow_fa));
618 );
619
620 *Actor Breeder, loop over agents
621
622 loop (aga_br,
623 *the dynamic set currentag_br ensures that the agents optimise individually
624 currentag_br(ag_br)=no;
625 currentag_br(aga_br)=yes;
626 *all solution variables of the agent that is not currently optimising are fixed
627 x_br.fx(act_br,yr,ag_br)$((not currentag_br(ag_br)) and\\\
628 ord(yr)>=ord(run))=x_br.l(act_br,yr,ag_br);
629 *all solution variables of the agent that is currently optimising for the
630 *current and future years are released
631 x_br.lo(act_br,yr,currentag_br)$ (ord(yr)>=ord(run))=0;
632 x_br.up(act_br,yr,currentag_br)$ (ord(yr)>=ord(run))=+inf;
633
634 Solve LPmodel_br using mip maximizing obj_br;
635 *End loop agents_br starting in line 623
636 );
637
638 *Saving of RHS of basic seed available for seed multipliers
639
640 Quantityvar2_mu(run) = sum(ag_br, x_br.l("market_org_br", run, ag_br)) ;
641 Quantityvar3_mu(run) = sum(ag_br, x_br.l("market_org_cult_br", run, ag_br)) ;
642
643 *Fixing solution variables so that they are not being overridden in the next run
644
645 x_br.fx(act_br,run,ag_br)=x_br.l(act_br,run,ag_br);
646
647 *Actor Multiplier, loop over agents
```

```
loop (aga_mu,
*the dynamic set currentag_mu ensures that the agents optimise individually
currentag_mu(ag_mu)=no;
currentag_mu(aga_mu)=yes;
*all solution variables of the agent that is not currently optimising are fixed
x_mu.fx(act_mu,yr,ag_mu)$((not currentag_mu(ag_mu)) and\\\
ord(yr)>=ord(run))=x_mu.l(act_mu,yr,ag_mu);
*all solution variables of the agent that is currently optimising for
*the current and future years are released
x_mu.lo(act_mu,yr,currentag_mu)$ (ord(yr)>=ord(run))=0;
x_mu.up(act_mu,yr,currentag_mu)$ (ord(yr)>=ord(run))=+inf;

*Assigning RHS of basic organic seed (var2) available for seed multipliers

RHS_ag_mu("basic_seed_rest_org_wcar_mu",run,aga_mu)= Quantityvar2_mu(run);

*First solve at seed multiplication level:  Testing the uptake of var2 per
*agent if total produced basic seed amount is available

Solve LPmodel_mu using mip maximizing obj_mu;

*Saving gross margins in Parameter
Obj_inclvar2_mu(aga_mu) = obj_mu.l;
*Saving used amount of var2 in Parameter
x_inclvar2_mu(aga_mu) = x_mu.l("buy_basic_seed_org_mu",run,aga_mu);

*Assigning RHS of basic organic seed (var3) available for seed multipliers

RHS_ag_mu("basic_seed_rest_org_wcar_mu",run,aga_mu)=0;
RHS_ag_mu("basic_seed_rest_org_cult_wcar_mu",run,aga_mu)= Quantityvar3_mu(run);

*Second solve at seed multiplication level:  Testing the uptake of var3 per
*agent if total produced basic seed amount is available

Solve LPmodel_mu using mip maximizing obj_mu;

*Saving gross margins in Parameter
Obj_inclvar3_mu(aga_mu) = obj_mu.l;
*Saving used amount of var3 in Par
x_inclvar3_mu(aga_mu) = x_mu.l("buy_basic_seed_org_cult_mu",run,aga_mu);

*Third solve at seed multiplication level: Calculating the gross margin
*per seed multiplication agent without var2 and 3

RHS_ag_mu("basic_seed_rest_org_wcar_mu",run,aga_mu)= 0       ;
RHS_ag_mu("basic_seed_rest_org_cult_wcar_mu",run,aga_mu)= 0  ;

Solve LPmodel_mu using mip maximizing obj_mu;

Obj_exclvar2_mu(aga_mu)  = obj_mu.l;
*End loop agents_mu starting in line 650
);
*Calculating total amount of basic seed of VAR2 bought by seed multipliers
*in previous test run
x_inclvar2_total_mu= sum(aga_mu,(x_inclvar2_mu(aga_mu) ));

*Distributing basic seed amount VAR2 relative to purchase in test run
if (x_inclvar2_total_mu >0 ,
RHS_ag_mu("basic_seed_rest_org_wcar_mu",run,aga_mu) = \\\
Quantityvar2_mu(run)*  (x_inclvar2_mu(aga_mu)/x_inclvar2_total_mu )
else RHS_ag_mu("basic_seed_rest_org_wcar_mu",run,aga_mu)=0;
);

*Calculating total amount of basic seed of VAR3 bought by seed multipliers in
```

```
*previous test run
x_inclvar3_total_mu= sum(aga_mu,(x_inclvar3_mu(aga_mu) ));

*Distributing basic seed amount VAR3 relative to purchase in test run
if (x_inclvar3_total_mu >0 ,
RHS_ag_mu("basic_seed_rest_org_cult_wcar_mu",run,aga_mu) = \\\
Quantityvar3_mu(run)*  (x_inclvar3_mu(aga_mu)/x_inclvar3_total_mu )
else RHS_ag_mu("basic_seed_rest_org_cult_wcar_mu",run,aga_mu)=0;
);

*Fourth solve at seed multiplication level: Making the actual distribution of
*basic seed of both VARs

loop (aga_mu,
  currentag_mu(ag_mu)=no;
  currentag_mu(aga_mu)=yes;

  x_mu.fx(act_mu,yr,ag_mu)$((not currentag_mu(ag_mu)) and\\\
  ord(yr)>=ord(run))=x_mu.l(act_mu,yr,ag_mu);

  x_mu.lo(act_mu,yr,currentag_mu)$ (ord(yr)>=ord(run))=0;
  x_mu.up(act_mu,yr,currentag_mu)$ (ord(yr)>=ord(run))=+inf;

  Solve LPmodel_mu using mip maximizing obj_mu;
);

*Saving of RHS of scaled amount of seed available for farmers
Quantityvar2_fa(run)=sum(ag_mu,x_mu.l("market_org_wcar_mu",run,ag_mu))\\\
 *Scaling_factor_fa;
Quantityvar3_fa(run)=sum(ag_mu,x_mu.l("market_org_cult_wcar_mu",run,ag_mu))\\\
 *Scaling_factor2_fa;

*Fixing solution variables so they are not being overridden in the next run
x_mu.fx(act_mu,run,ag_mu)=x_mu.l(act_mu,run,ag_mu);

*Actor Farmer, loop over agents
*************************************************************************

*Assigning innovational farmers
innov_ag_fa(ag_fa) $(Diff_innov_score(ag_fa)=Innov_level)= yes;

loop (aga_fa,
*the dynamic set currentag_fa ensures that the agents optimise individually
currentag_fa(ag_fa)=no;
currentag_fa(aga_fa)=yes;

*all solution variables of the agent that is not currently optimising are fixed
x_fa.fx(act_fa,yr,ag_fa)$((not currentag_fa(ag_fa)) and \\\
ord(yr)>=ord(run))=x_fa.l(act_fa,yr,ag_fa);
*all solution variables of the agent that is currently optimising for
*the current and future years are released
x_fa.lo(act_fa,yr,currentag_fa)$ (ord(yr)>=ord(run))=0;
x_fa.up(act_fa,yr,currentag_fa)$ (ord(yr)>=ord(run))=+inf;

*loop over innovative agents (var2 and 3 are only accessible to innovative
*farm agents)
if(innov_ag_fa(aga_fa),

*Assigning farm gate prices to individual innovative farm agents
loop(yry,
  a_ag_fa("accountopen_fa",yry+1,market_fa,yry,currentag_fa) $\\\
 (ord(yry)>=ord(run)and ord(yry)< card(yry)) = price_a_ag_fa(market_fa,yry);
  a_ag_fa("Accountclose_fa",yry,market_fa,yry,currentag_fa) $\\\
 (ord(yry) = card(yry)) = price_a_ag_fa(market_fa,yry);
  loop(market_org_wcar_fa,
```

```
      a_ag_fa("accountopen_fa",yry+1,market_org_wcar_fa,yry,currentag_fa) $\\\
     (ord(yry)>=ord(run)and ord(yry)<card(yry))=price_a_ag_fa\\\
     ("market_conv_wcar_fa",yry);
     a_ag_fa("Accountclose_fa",yry,market_org_wcar_fa,yry,currentag_fa)$\\\
     (ord(yry) = card(yry)) = price_a_ag_fa("market_conv_wcar_fa",yry);
  );
);

*Assigning crop yields to individual innovative farm agents
loop(yry,
  a_ag_fa(con_yield_fa,yry,grow_fa,yry,currentag_fa) $\\\
  (ord(yry)>=ord(run)) =yield_a_ag_fa(aga_fa,con_yield_fa,grow_fa,yry);
  a_ag_fa("yield_org_wcar_fa",yry,"grow_org_wcar_v_fa",yry,currentag_fa) $\\\
  (ord(yry)>=ord(run))=a_ag_fa("yield_conv_wcar_fa",yry,\\\
  "grow_conv_wcar_v_fa",yry,currentag_fa);
  a_ag_fa("yield_org_wcar_fa",yry,"grow_org_wcar_a_fa",yry,currentag_fa) $\\\
  (ord(yry)>=ord(run)) = a_ag_fa("yield_conv_wcar_fa",yry\\\
  "grow_conv_wcar_a_fa",yry,currentag_fa);
  ;
);

*Scenarios:

*Flat discount on organic seed per hectare:
a_ag_fa("accountopen_fa", yry, "grow_org_wcar_a_fa",yry, currentag_fa)$\\\
(ord(yry)=ord(run))=a_ag_fa("accountopen_fa", yry, "grow_org_wcar_a_fa",\\\
yry, currentag_fa) - Subsidy_fa;
a_ag_fa("accountopen_fa", yry, "grow_org_cult_wcar_a_fa",yry, currentag_fa)\\\
 $(ord(yry)=ord(run))  = a_ag_fa("accountopen_fa", yry,\\\
"grow_org_cult_wcar_a_fa",yry, currentag_fa) - Subsidy_fa;
a_ag_fa("accountopen_fa", yry, "grow_org_wcar_v_fa",yry, currentag_fa)$\\\
(ord(yry)=ord(run))=a_ag_fa("accountopen_fa",yry,"grow_org_wcar_v_fa",yry,\\\
currentag_fa) - Subsidy_fa;
a_ag_fa("accountopen_fa",yry,"grow_org_cult_wcar_v_fa",yry,currentag_fa)$\\\
(ord(yry)=ord(run))=a_ag_fa("accountopen_fa",yry,"grow_org_cult_wcar_v_fa",\\\
yry, currentag_fa) - Subsidy_fa;

$ontext
*Another possibility turned off here with ontext: Price_premium on farm gate price:

a_ag_fa("accountopen_fa",yry+1,"market_org_wcar_fa",yry,currentag_fa)$\\\
(ord(yry)=ord(run))=a_ag_fa("accountopen_fa",yry+1,"market_org_wcar_fa",\\\
yry, currentag_fa) - Price_premium;
a_ag_fa("accountopen_fa", yry+1, "market_org_cult_wcar_fa",yry,\\\
currentag_fa)$(ord(yry)=ord(run))=a_ag_fa("accountopen_fa",yry+1,\\\
"market_org_cult_wcar_fa",yry, currentag_fa) - Price_premium;
a_ag_fa("Accountclose_fa",yry,"market_org_wcar_fa",yry,currentag_fa) $\\\
(ord(yry)=ord(run))= a_ag_fa("Accountclose_fa",yry,"market_org_wcar_fa",\\\
yry,currentag_fa)- Price_premium;
a_ag_fa("Accountclose_fa",yry,"market_org_cult_wcar_fa",yry,currentag_fa)$\\\
(ord(yry)=ord(run))=a_ag_fa("Accountclose_fa",yry,"market_org_cult_wcar_fa",\\\
yry,currentag_fa)- Price_premium;
$offtext

*Assigning RHS of organic seed (var2) available for farmers
RHS_ag_fa("seedrest_org_wcar_fa",run,aga_fa)= Quantityvar2_fa(run);

*First solve at farming level:  Testing the uptake of var2 per agent if total
*produced seed amount is available
Solve LPmodel_fa using mip maximizing obj_fa;

*Saving gross margins in Parameter
Obj_inclvar2_fa(aga_fa) = obj_fa.l;
*Saving used amount of var2 in Parameter
x_inclvar2_fa(aga_fa) = x_fa.l("buyseed_org_wcar_fa",run,aga_fa);
```

```

*Assigning RHS of basic organic seed (var3) available for seed multipliers
RHS_ag_fa("seedrest_org_wcar_fa",run,aga_fa)= 0;
RHS_ag_fa("seedrest_org_cult_wcar_fa",run,aga_fa)= Quantityvar3_fa(run);

*Second solve at farming level:  Testing the uptake of var3 per agent if total
*produced seed amount is available
Solve LPmodel_fa using mip maximizing obj_fa;

*Saving gross margins in Parameter
Obj_inclvar3_fa(aga_fa) = obj_fa.l;
*Saving used amount of var3 in Par
x_inclvar3_fa(aga_fa) = x_fa.l("buyseed_org_cult_wcar_fa",run,aga_fa);

*Third solve at farming level: Calculating the gross margin per farming agent
*without var2 and 3
RHS_ag_fa("seedrest_org_wcar_fa",run,aga_fa)= 0;
RHS_ag_fa("seedrest_org_cult_wcar_fa",run,aga_fa) = 0;

Solve LPmodel_fa using mip maximizing obj_fa;

Obj_exclvar2_fa(aga_fa)  = obj_fa.l;
*End if-loop including only innovative Agents starting in line 770:
);
*End loop agents_fa starting in line 755:
);

*Calculating total amount of seed of VAR2 bought by farmers in previous test run
x_inclvar2_total_fa= sum(innov_ag_fa,(x_inclvar2_fa(innov_ag_fa)));

*Distributing seed amount VAR2 relative to purchase in test run
if(x_inclvar2_total_fa >0,RHS_ag_fa("seedrest_org_wcar_fa",run,innov_ag_fa)\\\
= Quantityvar2_fa(run)*(x_inclvar2_fa(innov_ag_fa)/x_inclvar2_total_fa )
else RHS_ag_fa("seedrest_org_wcar_fa",run,innov_ag_fa)=0;
);

*Calculating total amount of seed of VAR3 bought by farmers in previous test run
x_inclvar3_total_fa= sum(innov_ag_fa,(x_inclvar3_fa(innov_ag_fa)));

*Distributing seed amount VAR3 relative to purchase in test run
if(x_inclvar3_total_fa >0 , RHS_ag_fa("seedrest_org_cult_wcar_fa",\\\
  run,innov_ag_fa)=Quantityvar3_fa(run)*(x_inclvar3_fa(innov_ag_fa)/\\\
  x_inclvar3_total_fa)
  else RHS_ag_fa("seedrest_org_cult_wcar_fa",run,innov_ag_fa)=0;
);

*Fourth solve at farming level: Making the actual distribution of seed of both
*VARs
loop (aga_fa,
currentag_fa(ag_fa)=no;
currentag_fa(aga_fa)=yes;

x_fa.fx(act_fa,yr,ag_fa)$((not currentag_fa(ag_fa)) and \\\
ord(yr)>=ord(run))=x_fa.l(act_fa,yr,ag_fa);
x_fa.lo(act_fa,yr,currentag_fa)$ (ord(yr)>=ord(run))=0;
x_fa.up(act_fa,yr,currentag_fa)$ (ord(yr)>=ord(run))=+inf;

*Assigning farm gate prices to individual non-innovative farm agents
if(not innov_ag_fa(aga_fa),
  loop(yry,a_ag_fa("accountopen_fa",yry+1,market_fa,yry,currentag_fa) $\\\
    (ord(yry)>=ord(run)and ord(yry)< card(yry)) = price_a_ag_fa(market_fa,yry);
    a_ag_fa("Accountclose_fa",yry,market_fa,yry,currentag_fa) $\\\
    (ord(yry) = card(yry)) = price_a_ag_fa(market_fa,yry);
    loop(market_org_wcar_fa,a_ag_fa("accountopen_fa",yry+1,market_org_wcar_fa,\\\
      yry,currentag_fa)$(ord(yry)>=ord(run)and ord(yry)< card(yry)) =\\\
```

```
       price_a_ag_fa("market_conv_wcar_fa",yry) ;
       a_ag_fa("Accountclose_fa",yry,market_org_wcar_fa,yry,currentag_fa) $\\\
       (ord(yry) = card(yry)) = price_a_ag_fa("market_conv_wcar_fa",yry);
       );
     );
 );

*Assigning crop yields to individual non-innovative farm agents
loop(yry, a_ag_fa(con_yield_fa,yry,grow_fa,yry,currentag_fa) $\\\
  (ord(yry)>=ord(run)) = yield_a_ag_fa(aga_fa,con_yield_fa,grow_fa,yry);
  a_ag_fa("yield_org_wcar_fa",yry,"grow_org_wcar_v_fa",yry,currentag_fa) $\\\
  (ord(yry)>=ord(run))=a_ag_fa("yield_conv_wcar_fa",yry,\\\
  "grow_conv_wcar_v_fa",yry,currentag_fa);
  a_ag_fa("yield_org_wcar_fa",yry,"grow_org_wcar_a_fa",yry,currentag_fa) $\\\
  (ord(yry)>=ord(run))=a_ag_fa("yield_conv_wcar_fa",yry,\\\
  "grow_conv_wcar_a_fa",yry,currentag_fa);
  ;
);
Solve LPmodel_fa using mip maximizing obj_fa;
*End loop agents_fa starting in line 892:
);

*Calculation Innovation acceptance farmers: How many farmers actually used
*organic seed or an organic cultivar?
******************************
*Set counter to zero before each new calculation:
Org_seed_user_counter =0;
Org_seed_group_counter=0;

loop(innov_ag_fa$(Diff_innov_score(innov_ag_fa)=Innov_level),
  Org_seed_group_counter = Org_seed_group_counter+1;
  if((x_fa.l("buyseed_org_wcar_fa",run,innov_ag_fa)+x_fa.l\\\
  ("buyseed_org_cult_wcar_fa",run,innov_ag_fa)+x_fa.l\\\
  ("buyseed_conv_wcar_fa",run,innov_ag_fa))>0,
    if((x_fa.l("buyseed_org_wcar_fa",run,innov_ag_fa)+x_fa.l\\\
    ("buyseed_org_cult_wcar_fa",run,innov_ag_fa))/(x_fa.l\\\
    ("buyseed_org_wcar_fa",run,innov_ag_fa)+x_fa.l("buyseed_org_cult_wcar_fa",\\\
    run,innov_ag_fa)+x_fa.l("buyseed_conv_wcar_fa",run,innov_ag_fa))>0.25,
    Org_seed_user_counter=Org_seed_user_counter+1;
    );
  );
);

Innovation_diffusion= Org_seed_user_counter /Org_seed_group_counter;

*If Innovation_diffusion=> 0.9, then include subsequent subset into innov_ag_fa
if((Innovation_diffusion>0.9 and Innov_level<5),
 Innov_level   = Innov_level+1;
);

*Fixing solution variables so that they are not being overridden in the next run
x_fa.fx(act_fa,run,ag_fa)=x_fa.l(act_fa,run,ag_fa);

*Sum basic seed purchases seed multipliers (VAR2):
Boughtvar2_mu_d(run) = sum(ag_mu,x_mu.l("buy_basic_seed_org_mu",run,ag_mu));
if (Quantityvar2_mu(run)=0, Boughtvar2_br_s(ag_br,run)=0  else \\\
Boughtvar2_br_s(ag_br,run) = x_br.l("market_org_br",run,ag_br)/\\\
Quantityvar2_mu(run) * Boughtvar2_mu_d(run));

*Calculation of Growth expectation factor (VAR2)
loop(ag_br,
  if((Boughtvar2_br_s(ag_br,run)+ Boughtvar2_br_s(ag_br,run-1))>0,
    if (ord(run)=3,
    Expmaxsalevar2coef_br =min(max((Boughtvar2_br_s(ag_br,run)/Boughtvar2_br_s\\\
   (ag_br,run-1))*Adaptive_Expectations_coeff, 0.5),Upper_bound_Expmaxsalevarcoef);
```

```
      elseif ord(run)>3,
      Expmaxsalevar2coef_br =min(max(((Boughtvar2_br_s(ag_br,run)+\\\
      Boughtvar2_br_s(ag_br,run-1))/(Boughtvar2_br_s(ag_br,run-1)+\\\
      Boughtvar2_br_s(ag_br,run-2)))*Adaptive_Expectations_coeff, 0.5),\\\
      Upper_bound_Expmaxsalevarcoef);
      );
  );
);

*Setting back sales amounts to actually sold ones and assigning RHS for
*next year's production
x_br.fx("market_org_br",run,ag_br) = Boughtvar2_br_s(ag_br,run);
RHS_ag_br("ExpmaxsaleOrg_br", yr,ag_br)$ (ord(yr)>ord(run))= max\\\
(Boughtvar2_br_s(ag_br,run)* Expmaxsalevar2coef_br,Expmaxsalevar2Min_br);

*Sum basic seed purchases seed multipliers (VAR3):
Boughtvar3_mu_d(run)=sum(ag_mu,x_mu.l("buy_basic_seed_org_cult_mu",run,ag_mu));
if(Quantityvar3_mu(run)=0,Boughtvar3_br_s(ag_br,run)=0 else\\\
Boughtvar3_br_s(ag_br,run) = x_br.l("market_org_cult_br",run,ag_br)/\\\
Quantityvar3_mu(run) * Boughtvar3_mu_d(run));

*Calculation of Growth expectation factor (VAR3)
loop(ag_br,
 if((Boughtvar3_br_s(ag_br,run)+ Boughtvar3_br_s(ag_br,run-1))>0,
   if (ord(run)=3,
   Expmaxsalevar3coef_br =min(max((Boughtvar3_br_s(ag_br,run)/Boughtvar3_br_s\\\
   (ag_br,run-1))*Adaptive_Expectations_coeff, 0.5),Upper_bound_Expmaxsalevarcoef);
   elseif ord(run)>3,
   Expmaxsalevar3coef_br =min(max(((Boughtvar3_br_s(ag_br,run)+\\\
   Boughtvar3_br_s(ag_br,run-1))/(Boughtvar3_br_s(ag_br,run-1)+\\\
   Boughtvar3_br_s(ag_br,run-2)))*Adaptive_Expectations_coeff, 0.5),\\\
   Upper_bound_Expmaxsalevarcoef);
    );
  );
);

*Setting back sales amounts to actually sold ones and assigning RHS for
*next year's production
x_br.fx("market_org_cult_br",run,ag_br) = Boughtvar3_br_s(ag_br,run);
RHS_ag_br("ExpmaxsaleOrg_cult_br", yr,ag_br)$ (ord(yr)>ord(run))= max\\\
(Boughtvar3_br_s(ag_br,run)* Expmaxsalevar3coef_br,Expmaxsalevar3Min_br);

*release of all solution variables necessary to re-calculate the gross
*margin with actual sales information
x_br.lo(act_br,yr,ag_br)$ (ord(yr)>ord(run))=0;
x_br.up(act_br,yr,ag_br)$ (ord(yr)>ord(run))=+inf;
x_br.lo("TGM_br",yr,ag_br)$ (ord(yr)=ord(run))=0;
x_br.up("TGM_br",yr,ag_br)$ (ord(yr)=ord(run))=+inf;

*Include all breeding agents again into dynamic set
currentag_br(ag_br)=yes;

*Solve to re-calculate the gross margin with actual sales information
Solve LPmodel_br using mip maximizing obj_br;

*Scaled sum seed purchases farmers (VAR2):
Boughtvar2_fa_d(run)=sum(ag_fa,x_fa.l("buyseed_org_wcar_fa",run,ag_fa))*\\\
Scaling_factor_br;
if (Quantityvar2_fa(run)=0, Boughtvar2_mu_s(ag_mu,run)=0 else\\\
Boughtvar2_mu_s(ag_mu,run) = x_mu.l("market_org_wcar_mu",run,ag_mu)/\\\
(Quantityvar2_fa(run)*Scaling_factor_br) * Boughtvar2_fa_d(run));

*Setting back sales amounts to actually sold ones
x_mu.fx("market_org_wcar_mu",run,ag_mu) = Boughtvar2_mu_s(ag_mu,run);
```

```
*Calculation of Growth expectation factor (VAR2)
loop(ag_mu,
  if((Boughtvar2_mu_s(ag_mu,run)+Boughtvar2_mu_s(ag_mu,run-1))>0,
    if(ord(run)>1,
    Expected_Boughtvar2_mu_s(ag_mu,run)=(Boughtvar2_mu_s(ag_mu,run)+\\\
    Boughtvar2_mu_s(ag_mu,run-1))/2;
    );
  );
);

loop(ag_mu,
  if((Boughtvar2_mu_s(ag_mu,run)+ Boughtvar2_mu_s(ag_mu,run-1))>0,
    If (ord(run)= 3,
    Expmaxsalevar2coef_mu =min(max((Boughtvar2_mu_s(ag_mu,run)/Boughtvar2_mu_s\\\
   (ag_mu,run-1))*Adaptive_Expectations_coeff, 0.5),Upper_bound_Expmaxsalevarcoef);
    elseif ord(run)>3,
    Expmaxsalevar2coef_mu =min(max(((Boughtvar2_mu_s(ag_mu,run)+\\\
    Boughtvar2_mu_s(ag_mu,run-1))/(Boughtvar2_mu_s(ag_mu,run-1)+\\\
    Boughtvar2_mu_s(ag_mu,run-2)))*Adaptive_Expectations_coeff, 0.5),\\\
    Upper_bound_Expmaxsalevarcoef);
    );
  );
);

*Assigning RHS for next year's production
if(ord(run)=1,
  RHS_ag_mu("Expmaxsaleorg_wcar_mu", yr, ag_mu)$ (ord(yr)> ord(run))= 301
  elseif ord(run)=2,
  RHS_ag_mu("Expmaxsaleorg_wcar_mu", yr, ag_mu)$ (ord(yr)> ord(run))=\\\
  max(Boughtvar2_mu_s(ag_mu,run)* Expmaxsalevar2coef_mu, Expmaxsalevar2Min_mu);
  elseif ord(run)>2,
  RHS_ag_mu("Expmaxsaleorg_wcar_mu", yr, ag_mu)$ (ord(yr)> ord(run))=\\\
  max(Expected_Boughtvar2_mu_s(ag_mu,run)* Expmaxsalevar2coef_mu,\\\
  Expmaxsalevar2Min_mu);
  elseif ord(run)=card(run),
  RHS_ag_mu("Expmaxsaleorg_wcar_mu", yr, ag_mu)$ (ord(yr)>ord(run))=\\\
  max(Boughtvar2_mu_s(ag_mu,run-1)* Expmaxsalevar2coef_mu, Expmaxsalevar2Min_mu);
);

*Scaled sum seed purchases farmers (VAR3):
Boughtvar3_fa_d(run)=sum(ag_fa,x_fa.l("buyseed_org_cult_wcar_fa",run,ag_fa))\\\
*Scaling_factor2_br;
if(Quantityvar3_fa(run)=0, Boughtvar3_mu_s(ag_mu,run)=0 else\\\
Boughtvar3_mu_s(ag_mu,run) = x_mu.l("market_org_cult_wcar_mu",run,ag_mu)/\\\
(Quantityvar3_fa(run)*Scaling_factor2_br) * Boughtvar3_fa_d(run));

*Setting back sales amounts to actually sold ones
x_mu.fx("market_org_cult_wcar_mu",run,ag_mu) = Boughtvar3_mu_s(ag_mu,run);

*Calculation of Growth expectation factor (VAR3)
loop(ag_mu,
  if((Boughtvar3_mu_s(ag_mu,run)+Boughtvar3_mu_s(ag_mu,run-1))>0,
    if(ord(run)>1,
    Expected_Boughtvar3_mu_s(ag_mu,run) =(Boughtvar3_mu_s(ag_mu,run)+\\\
    Boughtvar3_mu_s(ag_mu,run-1))/2;
    );
  );
);

*Calculation of Growth expectation factor (VAR2)
loop(ag_mu,
  if((Boughtvar3_mu_s(ag_mu,run)+ Boughtvar3_mu_s(ag_mu,run-1))>0,
  if(ord(run)= 3 and Boughtvar3_mu_s(ag_mu,run-1)>0,
    Expmaxsalevar3coef_mu =min(max((Boughtvar3_mu_s(ag_mu,run)/Boughtvar3_mu_s\\\
   (ag_mu,run-1))*Adaptive_Expectations_coeff, 0.5),Upper_bound_Expmaxsalevarcoef);
```

```
      elseif ord(run)= 3 and Boughtvar3_mu_s(ag_mu,run-1)=0,
      Expmaxsalevar3coef_mu =min(max((Boughtvar3_mu_s(ag_mu,run)/0.1)*\\\
      Adaptive_Expectations_coeff, 0.5),Upper_bound_Expmaxsalevarcoef);
      elseif ord(run)>3 and (Boughtvar3_mu_s(ag_mu,run-1)+\\\
      Boughtvar3_mu_s(ag_mu,run-2))> 0,
      Expmaxsalevar3coef_mu =min(max(((Boughtvar3_mu_s(ag_mu,run)\\\
      Boughtvar3_mu_s(ag_mu,run-1))/(Boughtvar3_mu_s(ag_mu,run-1)+\\\
      Boughtvar3_mu_s(ag_mu,run-2)))*Adaptive_Expectations_coeff, 0.5),\\\
      Upper_bound_Expmaxsalevarcoef);
      );
  );
);

*Assigning RHS for next year's production
if(ord(run)=1,
  RHS_ag_mu("Expmaxsaleorg_cult_wcar_mu", yr, ag_mu)$ (ord(yr)> ord(run))= 15;
elseif ord(run)=2,
  RHS_ag_mu("Expmaxsaleorg_cult_wcar_mu", yr, ag_mu)$ (ord(yr)> ord(run))=\\\
  max(Boughtvar3_mu_s(ag_mu,run)* Expmaxsalevar3coef_mu, Expmaxsalevar3Min_mu);
  elseif ord(run)>2,
  RHS_ag_mu("Expmaxsaleorg_cult_wcar_mu", yr, ag_mu)$ (ord(yr)> ord(run))=\\\
  max(Expected_Boughtvar3_mu_s(ag_mu,run)*Expmaxsalevar3coef_mu,\\\
  Expmaxsalevar3Min_mu);
  elseif ord(run)=card(run),
  RHS_ag_mu("Expmaxsaleorg_cult_wcar_mu", yr, ag_mu)$ (ord(yr)>ord(run))=\\\
  max(Boughtvar3_mu_s(ag_mu,run-1)* Expmaxsalevar3coef_mu, Expmaxsalevar3Min_mu);
);

*release of all solution variables necessary to re-calculate the gross margin
*with actual sales information
x_mu.lo(act_mu,yr,ag_mu)$ (ord(yr)>ord(run))=0;
x_mu.up(act_mu,yr,ag_mu)$ (ord(yr)>ord(run))=+inf;
x_mu.lo("TGM_mu",yr,ag_mu)$ (ord(yr)=ord(run))=0;
x_mu.up("TGM_mu",yr,ag_mu)$ (ord(yr)=ord(run))=+inf;

*Include all seed multiplication agents again into dynamic set
currentag_mu(ag_mu)=yes;
*Solve to re-calculate the gross margin with actual sales information
Solve LPmodel_mu using mip maximizing obj_mu;
*End loop over years ("run") starting in line 604:
);
```

A3. Decision matrices

Table A3.1: Simplified matrix overview of the MP decision-making model at farm level

	Years	Unit	Crop rotation	Grow carrot NCT	Grow carrot organic	Grow carrot organic cultivar	Buy NCT carrot seed	Buy organic carrot seed	Buy organic carrot cultivar seed	Grow crop 1	Grow crop 2	Grow crop 3	Grow green manure	Market carrot NCT	Market carrot organic	Market carrot organic cultivar	Market crop 1	Market crop 2	Market crop 3	Cash transfer	Total gross margin	Crop rotation	Grow carrot NCT	Grow carrot organic	Grow carrot organic cultivar	Buy NCT carrot seed	Buy organic carrot seed	Buy organic carrot cultivar seed	Grow crop 1	Grow crop 2	Grow crop 3	Grow green manure	Market carrot NCT	Market carrot organic	Market carrot organic cultivar	Market crop 1	Market crop 2	Market crop 3	Cash transfer	Total gross margin	Relation	RHS
Years			yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr2	yr2	yr2	yr2	yr2	yr2	yr2	yr2	yr2	yr2	yr2	yr2	yr2	yr2	yr2	yr2	yr2	yr2	yr2		
Unit			ha	ha	ha	ha	Mio seed	Mio seed	Mio seed	ha	ha	ha	ha	ton	ton	ton	ton	ton	ton	€	€	ha	ha	ha	ha	Mio seed	Mio seed	Mio seed	ha	ha	ha	ha	ton	ton	ton	ton	ton	ton	€	€		
Obj. function			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		
Land constr.	yr1	ha	1																																						≤	**R**
Main crop rot. constr.	yr1	ha	-1	1	1	1				1	1	1	1																												≤	0
Specific crop rotation constr. 1	yr1	ha	-A	1	1	1																																			≤	0
Specific crop rotation constr. 2	yr1	ha	-A							1																															≤	0
Specific crop rotation constr. 3	yr1	ha	**-A**								1																														≤	0
Specific crop rotation constr. 4	yr1	ha	**-A**									1																													≤	0
Specific crop rotation constr. 5	yr1	ha	**A**										-1																												≤	0
Labour constr.	yr1	Per.-hours		**A**	**A**	**A**				**A**	**A**	**A**	**A**																												≤	**(R)**
Seed requ. NCT	yr1	Mio seed		**A**			-1																																		≤	0
NCT seed restr.	yr1	Mio seed																																							≤	0

Seed requ. organic	yr1	Mio seed		**A**			-1																										≤	0
Organic seed restr.	yr1	Mio seed					1																										≤	**(R)**
Seed requ. organic cultivar	yr1	Mio seed			**A**			-1																									≤	0
Organic cultivar seed restr.	yr1	Mio seed						1																									≤	**(R)**
Yield organic carrot NCT	yr1	ton	**(-Y)**										1																				≤	0
Yield organic carrot organic	yr1	ton		**(-Y)**										1																			≤	0
Yield organic carrot organic cultivar	yr1	ton			**(-Y)**										1																		<=	0
Yield organic crop 1	yr1	ton							**(-Y)**							1																	<=	0
Yield organic crop 2	yr1	ton								**(-Y)**							1																≤	0
Yield organic crop 3	yr1	ton									**(-Y)**							1															≤	0
Account open	yr1	€	**C**	**C**	**C**	**C**	**C**	**C**	**C**	**C**	**C**	**C**							1														≤	**R**
Account close	yr1	€																															≤	0
Land constr.	yr2	ha																		1													≤	**R**
Main crop rot. constr.	yr2	ha																		-1	1	1	1				1	1	1	1			≤	0
Specific crop rot. constr. 1	yr2	ha																		**-A**	1	1	1										≤	0
Specific crop rot. constr. 2	yr2	ha																		**-A**							1						≤	0
Specific crop rot. constr. 3	yr2	ha																		**-A**								1					≤	0
Specific crop rot. constr. 4	yr2	ha																		**-A**									1				≤	0
Specific crop rot. constr. 5	yr2	ha																		**A**										-1			≤	0
Labour constraint	yr2	Per.-hours																			**A**	**A**	**A**				**A**	**A**	**A**	**A**			≤	**(R)**
Seed requ. NCT	yr2	Mio seed																			**A**			-1									≤	0
NCT seed restr.	yr2	Mio seed																															≤	0
Seed requ. organic	yr2	Mio seed																				**A**			-1								≤	0
Organic seed restr.	yr2	Mio seed																							1								≤	**(R)**
Seed requ organic cultivar	yr2	Mio seed																					**A**			-1							≤	0
Organic cultivar seed restr.	yr2	Mio seed																								1							≤	**(R)**
Yield organic carrot NCT	yr2	ton																			**(-Y)**										1		≤	0
Yield organic	yr2	ton																				**(-Y)**										1	≤	0

carrot organic																													
Yield organic carrot organic cultivar	yr2	ton										(-Y)										1						≤	0
Yield organic crop 1	yr2	ton														(-Y)							1					≤	0
Yield organic crop 2	yr2	ton															(-Y)							1				≤	0
Yield organic crop 3	yr2	ton																(-Y)							1			≤	0
Account open	yr2	€	**(-C)**	**(-C)**	**(-C)**	**(-C)**	**(-C)**	**(-C)**	-1.03	**C**	**C**	**C**	**C**	**C**	**C**	**C**	**C**	**C**	**C**								1		≤	**R**
Account close	yr2	€																		**(-C)**	**(-C)**	**(-C)**	**(-C)**	**(-C)**	**(-C)**	-1.03	1	≤	0

Note: E = Expected values, C = price coefficients, Y = Crop yields, A = Technical coefficients, R = Available resources. Values in round brackets are adjusted inside the model. Bold values are agent-specific.

Table A3.2: Simplified matrix overview of the MP decision-making model at seed multiplication level

	Years	Unit	Multiply NCT seed	Multiply organic seed	Multiply organic cultivar seed	Buy basic NCT seed	Buy basic organic seed	Buy basic organic cultivar seed	Market NCT seed	Market organic seed	Market seed organic cultivar	Cash transfer	Total gross margin	Multiply NCT seed	Multiply organic seed	Multiply organic cultivar seed	Buy basic NCT seed	Buy basic organic seed	Buy basic organic cultivar seed	Market NCT seed	Market organic seed	Market seed organic cultivar	Cash transfer	Total gross margin	Relation	RHS
Years			yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr2	yr2	yr2	yr2	yr2	yr2	yr2	yr2	yr2	yr2	yr2		
Unit			1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	€	€	1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	€	€		
Obj. function			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		
Typical organic market size constr.	yr1	ha	**A**	**A**																					≤	**Typical organic market size**

Typical organic cultivar market size constr.	yr1	ha			**A**														≤	**Typical organic cultivar market size**
Basic seed requ. NCT	yr1	1 Mio seed	**A**			**-1**													≤	0
Basic seed requ. organic	yr1	1 Mio seed		**A**			**-1**												≤	0
Basic seed requ. organic cultivar	yr1	1 Mio seed			**A**			**-1**											≤	0
Basic organic seed constr.	yr1	1 Mio seed					**1**												≤	**(R)**
Basic organic cultivar seed constr.	yr1	1 Mio seed						**1**											≤	**(R)**
Seed yield NCT	yr1	1 Mio seed							**1**										≤	**($E_{NCT\ seed\ sales}$)**
Seed yield organic	yr1	1 Mio seed								**1**									≤	**($E_{organic\ seed\ sales}$)**
Seed yield organic cultivar	yr1	1 Mio seed									**1**								≤	**($E_{organic\ cultivar\ seed\ sales}$)**
Expected organic seed sales	yr1	1 Mio seed																	≤	0
Expected organic cultivar sales	yr1	1 Mio seed																	≤	0
Account open	yr1	€	**C**	**C**	**C**	**C**	**C**	**C**	**C**	**C**	**C**	1							≤	**R**
Account close	yr1	€																	≤	0
Typical organic market size constr.	yr2	ha											**A**	**A**					≤	**Typical organic market size**
Typical organic cultivar market size constr.	yr2	ha													**A**				≤	**Typical organic cultivar market size**
Basic seed requ. NCT	yr2	1 Mio seed											**A**			**-1**			≤	0
Basic seed requ. organic	yr2	1 Mio seed												**A**			**-1**		≤	0
Basic seed requ. organic cultivar	yr2	1 Mio seed													**A**			**-1**	≤	0

Basic organic seed constr.	yr2	1 Mio seed												**1**							≤	**(R)**
Basic organic cultivar seed constr.	yr2	1 Mio seed													**1**						≤	**(R)**
Seed yield NCT	yr2	1 Mio seed	**-Y**													**1**					≤	0
Seed yield organic	yr2	1 Mio seed		**-Y**													**1**				≤	0
Seed yield organic cultivar	yr2	1 Mio seed			**-Y**													**1**			≤	0
Expected organic seed sales	yr2	1 Mio seed															**1**				≤	**($E_{organic\ seed\ sales}$)**
Expected organic cultivar sales	yr2	1 Mio seed																**1**			≤	**($E_{organic\ cultivar\ seed\ sales}$)**
Account open	yr2	€	**C**	**C**	**C**	**-C**	**-C**	**-C**	-1.03	**C**	**C**	**C**	**C**	**C**	**C**	**-C**	**-C**		1		≤	0
Account close	yr2	€								**C**	**C**	**C**				**-C**	**-C**		-1.03	1	≤	0

Note: E = Expected values, C = price coefficients, Y = Crop seed yields, A = Technical coefficients, R = Available resources. Values in round brackets are adjusted inside the model. Bold values are agent-specific.

Table A3.3: Simplified matrix overview of the MP decision-making model at breeding level

	Years	Unit	Produce basic seed	Produce basic seed of organic cultivar	Market CT basic seed	Market NCT basic seed	fMarket organic basic seed	Market basic seed or organic cultivar	Cash transfer	Total gross margin	Produce basic seed	Produce basic seed of organic cultivar	Market CT basic seed	Market NCT basic seed	Market organic basic seed	Market basic seed of organic cultivar	Cash transfer	Total gross margin	Relation	RHS
Years			yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr1	yr2	yr2	yr2	yr2	yr2	yr2	yr2	yr2		

Unit																				
			1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	€	€	1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	1 Mio seed	€	€		
Obj. function			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		
Typical market size constr.	yr1	ha			**A**	**A**	**A**												≤	**Typical market size**
Typical organic market size constr.	yr1	ha				**A**	**A**												≤	**Typical organic market size**
Typical organic cultivar market size constr.	yr1	ha						**A**											≤	**Typical organic cultivar market size**
Basic seed yield	yr1	1 Mio seed	**-1**		**1**	**1**	**1**												≤	0
Basic organic cultivar seed yield	yr1	1 Mio seed		**-1**				**1**											≤	0
Expected organic basic seed sales	yr1	1 Mio seed					**1**												≤	**($E_{\text{organic basic seed sales}}$)**
Expected organic basic cultivar sales	yr1	1 Mio seed						**1**											≤	**($E_{\text{organic cultivar basic seed sales}}$)**
Account open	yr1	€							1										≤	**R**
Account close	yr1	€																	≤	0
Typical market size constr.	yr2	ha											**A**	**A**	**A**				≤	**Typical market size**
Typical organic market size constr.	yr2	ha												**A**	**A**				≤	**Typical organic market size**
Typical organic cultivar market size constr.	yr2	ha														**A**			≤	**Typical organic cultivar market size**
Basic seed yield	yr2	1 Mio seed									**-1**		**1**	**1**	**1**				≤	0
Basic organic cultivar seed yield	yr2	1 Mio seed										**-1**				**1**			≤	0
Expected organic basic seed sales	yr2	1 Mio seed													**1**				≤	**($E_{\text{organic basic seed sales}}$)**
Expected organic basic cultivar sales	yr2	1 Mio seed														**1**			≤	**($E_{\text{organic cultivar basic seed sales}}$)**
Account open	yr2	€			**-C**	**-C**	**-C**	**-C**	-1.03								1		≤	**R**
Account close	yr2	€											**-C**	**-C**	**-C**	**-C**	-1.03	1	≤	0

Note: E = Expected values, C = price coefficients, Y = Crop seed yields, A = Technical coefficients, R = Available resources. Values in round brackets are adjusted inside the model. Bold values are agent-specific.