MasFIRA: multi-agent-simulation on family dynamics and intra-family resource allocation
Neuwirth, Norbert

Veröffentlichungsversion / Published Version
Arbeitspapier / working paper

Zur Verfügung gestellt in Kooperation mit / provided in cooperation with:
SSG Sozialwissenschaften, USB Köln

gesis
Leibniz-Institut
für Sozialwissenschaften

Mitglied der
Leibniz-Gemeinschaft

Nr. 44 | 2004

Norbert Neuwirth

MasFIRA

Multi-Agent-Simulation on Family dynamics and Intra-family Resource Allocation

PAPERS

KONTAKT: norbert.neuwirth@oif.ac.at | +43-1-535 14 54-22

Abstract (E):

This is a documentation of the agent-based-simulation environment "masFIRA" that has been developed to demonstrate implications of game theoretical models of the partner matching process, bargaining approaches on intrafamily resource allocation, and fertility behaviour. As development still proceeds, this documentation is rather a status report on models and modules implemented by December 2004.

Abstract (D):

Dies ist die Dokumentation der agenten-basierten Simulationsplattform „masFIRA", die entwickelt wurde, um Implikationen von spieltheoretischen Modellen betreffend Partnersuchprozesse, intra-familiärer Ressourcenallokation und Fertilitätsverhaltens zu demonstrieren. Die Plattform ist nach wie vor in Weiterentwicklung, deshalb ist diese Dokumentation auch als Status Report der bis Ende 2004 implementierten Module zu betrachten.

4

List of figures:

1   Introduction

Structural family dynamics – partners' matching processes, stability of families, occurrence of divorces, and fertility – are handled quite differently within social sciences. While sociologists and psychologists primarily tend to investigate these topics with depth-grounded methods, economic theory and empirical approaches rely on comparably straightforward models of rational behaviour. Individuals (''agents'') differ by endowments – talents, human capital stocks, property, integration in social networks, … – and by their preferences. Most approaches abstract from these differences as economic models of household behaviour discuss the issues subjected by parameteriseable utility functions subject to endowment constraints held as general as possible. Therefore, economic models can come to unambiguous results of the optimisation problem of a prototype agent, but agents' behavioural heterogeneity is systematically neglected.

To meet observed or even potential heterogeneity, several techniques have been developed. One group of these techniques are agent based simulations in social sciences. Each agent can – and has to – adjust his behaviour to his individual environment and experience. Therefore, the agents' endowments and social environments but also the objectives change over time. The development of individual values and social norms will be embedded in a process of bounded rational behaviour of agents with limited knowledge, limited memory, and reduced rational expectations. But these values and norms are – together with emotions – the most important substitutes for ''deficiency'' in pure rationality. This logical circle can hardly be modelled with conventional economic models. Approaches of bounded rationality– from the seminal work of Herbert Simon (1966) to current developments – seem more appropriate. Another seminal work – Axelrod's work (1992) on cooperative behaviour and his extensions of cooperative game theory seem also promising for reducing the deficiencies of the homo oeconomicus-approach in respect to observed social behaviour. Last, not least recent developments in behavioural economics support these approaches, see e.g. Kahneman/Tversky (ed. 2000) for a comprehensive depiction of this economic school of thought.

Applications of agent based simulation to social sciences in general and family-related issues in concrete have been broadly published over the last years. Billari/Fuernkranz-Prskawez (ed. 2003) have collected valuable contributions of agent-based models to demographic sciences, Ballot/Weisbuch (ed. 2000) edited contributions of ABS-applications in the field of sociology, geography, urban planning, politics and history, and strategic market behaviour, just to name two of these publications. Permanent updates of the latest publications can be found in the electronic ''Journal of Artifical Societies and Social Simulation'' (JASSS)[1], latest developments in agent based computational economics (ACE) at ACE-website[2].

This is the documentation of a demo-version of an agent-based simulation environment, where well developed miroeconomic models of family dynamics and fertility behaviour, intra-household decisions and resource allocations, human capital investment, time allocation and labour supply are condensed to a – still compact – dynamic model. In section 2 the underlying economic models will be presented, followed by a general description of ABS programs (section 3). I concept of masFIRA will presented in section 5, followed by the description of some simulation runs in chapter 6. The paper concludes in an outlook (7). The program code is appended after the literature index.

---

[1] http://jasss.soc.surrey.ac.uk/JASSS.html
[2] http://www.econ.iastate.edu/tesfatsi/ace.htm

2 At a glance – the Family from economists' view

As intra-family decisions are the most original topic of economics[3], libraries have been written on that topic. Recently – say within the last two decades – the scope has been shifted from household studies to the analysis of firms and markets. In the 60ies the internal processes of the household sector were rediscovered for economic analysis. Since then, the household was not just consumer and supplier of labour force, the intra-household production process became evident to economists. More than that, the theoretical work on this issue first gave some (possible) insight to the "black box" of intra-household decisions.[4]

2.1 Economic reasons for marriage – and divorce

The first "black box" in economic thought was the question: "Why do (fully) rational individuals share a household, why do they marry?". Of course a set of answers regarding economic motives can be given ad hoc:

- o   Cost sharing for intra-household public (non-rival) goods,
- o   risk pooling among family members,
- o   division of labour to exploit comparative advantages and increasing returns of scale,
- o   extending possible credit volumes within imperfect credit markets and coordination of investment activities, and, most important,
- o   to raise children, as most individuals obviously have some preference for having children .

As these points show up (in different extent everlasting) benefits from marriage, they have traded off to some costs. Interestingly, this point has been emphasized rarely in the literature. Divorce occurs, when the benefits cited begin to fall under the costs of partnership.

2.2 The matching process

Having stated economic reasons for marrying, the question "Who marries – rationally – whom?" becomes evident. Therefore some seminal game theoretical models were developed, rationalizing the empirical fact of "perfect matching". Following these approaches individuals strongly prefer a partner with – at least – comparable social status like themselves. As both potential partners think that way, the only solution to this optimisation problem is a – society-wide – perfect partner matching. The only possibility for mating of members of different social classes arises, when genders are distributed differently over classes.

Of course also less restrictive approaches have established. Nevertheless, determinant of mate matching are also a central issue within respective simulation approaches.

---

[3] oikos (gr.) the household ➔ science of all facilities relevant to the private household; with the development of markets and the increasing specialisation of the labour force and the separation of workplace and place of residence, economics began to lift off sociology and started to become a science for its own.

[4] As economic modelling relies on rationality assumptions, the discussion within this chapter is reflected following this paradigma. As antithesis, as we will see later, agents based modelling and artificial intelligence approaches are following the paradigma of "bounded rationality".

2.3 Intra-family decisions on resource allocation

This is the issue at the hart of economic analysis. Some approaches have emphasized this issue as the central question. The other issues – mating, having children, and separating again – are just a form, respectively a consequence of resource allocation. Every resource allocated has to be considered in these models. Wealth, goods and services, labour, all items the utility functions respectively the budget constraints consist of. Finally, "time" can be seen as fundamental category all these items dissolve in[5].

2.3.6   Unitarian approach

Within this approach[6] decisions a settled jointly with no disagreement. Models are designed with a unitary utility function that contains all needs of all household members. Thinking on a "benevolent dictator" this model can be applied to households organized in strict hierarchy. The head of the household decides in favour of all. Agents do not have an individual utility function, they all contribute to the common objective function instead.

$$max\, U(\,\mathbf{x_1}, \mathbf{x_2}, \mathbf{X}, \mathbf{t_1}, \mathbf{t_2}\,)$$
$$\text{s.t.}\quad \sum_k \sum_i t_k^i = nT \qquad\qquad (1)$$
$$\mathbf{p'(x_1 + x_2)} + \mathbf{P'X} \leq I = \sum_k w_k^N t_k^N + V$$

The family's utility is gained my maximizing the – common – utility function. This function consist of private goods (vector $\mathbf{x_k}$; k is an index for household members), intra-household public goods ($\mathbf{X}$) and the individual agents' process benefits from vector of activities ($\mathbf{t_k}$), subject to total amount of time and wealth.

2.3.7   Collective approach

Contrary to this model alternatives have been developed. Within the so called collective models[7] or bargaining approach, partners negotiate over the resource allocation (time and goods). Therefore a Nash-bargaining model is used for analysis:

$$max\, N\{[\,U_1(\mathbf{x_1}, \mathbf{X}, \mathbf{t_1})-D_1] \;*\; [\,U_2(\mathbf{x_2}, \mathbf{X}, \mathbf{t_2})-D_2]\,\}$$
$$\text{s.t.}\quad \sum_i t^i = T \qquad\qquad (2)$$
$$\mathbf{p'(x_1 + x_2)} + \mathbf{P'X} \leq I = \sum_k w_k^N t_k^N + V$$

Within this approach both partners try to maximize their own, individual utility, but mind the other's situation in some   respect: When the utility function of either partner falls below her/his reservation level, this partner could – and within this cooperative game approach, would immediately – leave the

---

[5] This fundamental insight can be followed up by quotations like "Ökonomie der Zeit, darin löst sich schließlich alle Ökonomie auf" ("Economy of time, to this all economy ultimately reduces itself") (Karl Marx, 1857) or "Different constraints are decisive for different situations, but the most fundamental constraint is limited time" (Gary S. Becker, 1993)

[6] see Becker (1993) and Willis (1973) for details

[7] see Ott (1992), Bourguignon/Chiappori (1992),

partnership, as her/his outside options (the reservation level D_i) are beyond the utility level obtained within the partnership. Within this approach, agents are "restricted" to Pareto-efficient resource allocations and partnership constellations.

Within this simulation environment the underlying general assumption of fully rational behaviour is relaxed in some respect, nevertheless the (partly competing) models of this simulation are near to the idea of collectively rational egoistic agents, Chiappori (1988) developed.

3 What is Agent-Based-Simulation?

Agent based simulation (ABS) models have been developed quite far today. To describe the properties and power of ABS in the field of social science simulation, it is advisable to begin with its predecessor, the so called cellular automata (CA). A CA consists of a number of cells arranged in a regular grid. Every "agent" is a cell that is situated within other cells in a multidimensional array. In a two-dimensional setting this cell is located on a (seemingly) rectangular grid. In most CA models the border-cells are connected, so that a cell on the left "border" is connected to the cell on the "right bordered" cell in corresponding height of the grid. Analogously a cell on "top" of the grid is the neighbour of the corresponding cell on the "bottom". So each cell has the same number of neighbouring cells and the seemingly rectangular grid represents the surface of a torus. In social simulations cells may represent individuals or collective actors like households, firms, communities or even countries. Each cell can be in one of a few states, for example, `on' or `off', or `alive' or `dead', represent attitudes (e.g. supporting one of several political parties), individual characteristics (e.g. ethnical origin), or actions (e.g. co-operating or not co-operating with others). Time advances through the simulation in discrete steps. After each time step, the state of each cell may change. The state of a cell at any time step is determined by a set of rules which specify how that state depends on the previous state of that cell and the states of the cell's immediate neighbours. The same rules are used to update the state of every cell in the grid. The model is therefore homogeneous with respect to the rules. Because the rules only make reference to the states of other cells in a cell's neighbourhood, cellular automata are best used to model situations where the interactions are local. CAs have been used as models in many areas of mathematics, biology and physical science, as well as social science.

In contrast to CA simulation ABS models[8] are somewhat more complex in their internal processing and consequently in their behaviour. Such automata are conventionally called 'agents'. While there is no generally agreed definition of what an `agent' is, the term is usually applied to describe self-contained programs, which can control their own actions based on their perceptions of their operating environment. Agent based programming is rapidly becoming important outside the field of social simulation. For example, agents have been built to watch out for interesting information as it becomes available over the Internet, informing the user if it finds relevant sources. The agent is instructed about the topics thought to be interesting and it then continuously monitors known sources for items fitting this profile. Other agents have been built to guide people to use software more effectively (like the annoying wizard agents within Microsoft Office package).

---

[8] or synonymously: multi agent systems (MAS)

Typically an ABS-model consists of the grid, similar to a CA environment, and agents inhabiting this grid. In general cells cannot move while agents can. The cells or patches are endowed with certain properties – like colour or height – that also can influence the neighbouring cells' characteristics. These properties can be – and in most cases actually are – 'inherited' by the actual patch in the next period. Agents situated on or moving towards a cell communicate with that particular cell: following a set of predefined rules the agent's properties and the cell's characteristics have to match. Unless such a feasible combination accomplishes, the agent has to move onwards (given it can regarding the rules for mobility), or the patch has to adjust its properties (assimilate to the agent's needs following the appropriate rules), or the agent has to alter its characteristics (assimilate to the cell's endowments following the relevant rules). The agent also simply can die, technically spoken it simply changes its status from 'alive' to 'dead' and vanishes from the population in the next period.

Following Wooldridge and Jennings (1995) agents have typically the following properties[9]:

❖ autonomy: agents operate without others having direct control of their actions and internal state;

❖ social ability: agents interact with other agents through some kind of `language' (a computer language, rather than natural language);

❖ reactivity: agents are able to perceive their environment (which may be the physical world, a virtual world of electronic networks, or a simulated world including other agents) and respond to it;

❖ pro-activity: as well as reacting to their environment, agents are also able to take the initiative, engaging in goal-directed behaviour.

In addition, agents often seem to have some degree of intentionality. That is, their behaviour is interpreted in terms of a metaphorical vocabulary of belief, desires, motives, and even emotions, concepts which are more usually applied to people rather than to computer programs. For the programmer's purposes, it is only necessary to view the ascription of intentionality as a matter of modelling: a computer agent does not have intentionality, but is constructed to simulate some (much simplified) aspects of human intentions. These general properties generate a set of (more concrete) attributes agents are endowed with:

Knowledge and beliefs: Agents will need to base their actions on what they think to know about their environment – the regions of the grid, the characteristics of particular patches within the region and the agents 'around'. Some of the information they have may be incorrect, as a result of faulty perception, faulty inference, or incomplete knowledge. This possibly erroneous information – the agent's beliefs – is distinguished from its true knowledge. From an observer's point of view this distinction can be evaluated, the agent of course can not figure out.

Inference: Given a set of beliefs, agents may be able to infer further information from it. For example, believing that agent B has recently 'eaten' some 'food', agent A could infer that the place to find food is near where agent B was located. Of course, this inference may be wrong (perhaps agent B consumed all the food).

---

[9] cited from: Gilbert/Troitzsch (1999)

3.1   Social models

Some agents may be capable of learning about the interrelationships between other agents in their world, for example, that agent A has recently interacted with agent B. On the basis of such snippets of data, an agent may be able to put together a picture of the social relationships in their environment, that is a `social model'. Agents may also have models of other aspects of their world, for example, they may develop a model of the `geography' of their environment. Note that these agents' models are quite different from the simulation model which the researcher builds; agent models are built by the agents themselves while the simulation runs.

3.2   Goals

Since agents are built to be autonomous and purposive, if they are to engage in action they need to be driven by a need to satisfy some internal goal such as survival. Surviving may in turn require the satisfaction of subsidiary goals, such as acquiring energy and avoiding lethal dangers.

3.3   Knowledge representation

In order to construct its models, an agent needs some way to represent its beliefs. Techniques for doing this have been studied by researchers on artificial intelligence. One generally useful approach is to use predicate logic to store declarative statements, such as "There is food at location X", and formulae such as "If an agent is eating at location X, there is food at location X", where X is a variable to be filled in depending on what the agent 'sees' around it.

3.4   Planning

An agent needs to have some way of determining what behaviour is likely to lead to the satisfaction of its goals, so the agent needs to plan its activities. Planning involves working backwards from a desired goal state, inferring what action would lead to that goal, what state would be required before that action can be carried out, what action is needed to arrive at that state, and so on, until one gets back to the current situation of the agent.

3.5   Language

All multi-agent models include some form of interaction between agents, or at a minimum, between individual agents and the environment in which they exist. The interaction might involve the passing of information from one agent to another, the (bilateral) negotiation of contracts – like in the masFIRA case, or unilateral signalling to evoke certain behaviour of the other agents around.

3.6   Modelling the environment

In all multi-agent simulations, the agents are located in an environment. What constitutes an environment depends on what is being modelled, but if the agents are individual people, rather than organisations, one of the main functions of the environment will be to provide a spatial context. Each agent will be located in a simulated space, in much the same way as actors are located within a cellular automata grid. In many such models, the agents will be able to move around the environment. Although such a spatial world is the most common environment, others are possible. For instance, the agents may move through a network of nodes and links (this might be useful if, for instance, the simulation was concerned with financial markets or state development issues, with the network modelling trading links).

Once agents are positioned within an environment, they will need 'sensors' to perceive their local neighbourhood and some means with which to affect the environment. Usually, communication between agents is routed through the environment, which forwards messages on to the appropriate recipient. In this case, agents will also need to be able to 'hear' messages coming from the environment and to send messages to the environment for onward transmission.

The designer will also need to decide about the order in which the agents in the simulation are given computing time. Ideally, all agents ought to operate in parallel. However, since most simulations run on sequential, rather than parallel computers, the desired parallel operation must itself be simulated, usually by running the program code for each agent within one loop, when all agents have done their activities for that round, the first will stand the next round of actions. Unfortunately, the order in which agents are run can have a major effect on the course of the simulation unless suitable precautions are taken. For example, if agent A sends a message to agent B, but B is run before A, agent B will not get the message from A until the next round, by which time the message may be no longer relevant.

3.7 Is it science?

Indeed, while agent based simulation has become a valuable and venerated "tool" within other disciplines – in biology or demography it has become quite important already – social sciences and especially economics have had viewed ABS quite sceptically. Empirical instruments have been developed quite far in econometrics, computer experiments are still generally associated with changing values of an exogenous variable and – ceteris paribus – watching and analyzing the shifts in endogenous variable(s) within a system of econometric equations. Even system dynamics simulations[10] have had a hard standing within economic sciences. Anyway, some economists have already specialized in ABS and founded a new research area, the so called agent-based computational economics (ACE). Despite of the still widespreaded scepticism, ABS – or ACE – are expected to become one of the most vital new methodological fields in economic and demographic analysis in the next years.

---

[10] system dynamics simulation procedures are systems of structured differential equations. Most prominent is the SD-simulation „World-3" written by Jay Forrester in the simulation environment "DYNAMO". Later these simulations were adopted by Dennis Meadows (within the improved simulation package "STELLA"), whose book (Meadows ea. 1972) – where he and his research team described the ecological and economical simulated developments of the earth – became very famous in the 70-80ies.

4 NetLogo – a programming environment for ABS

The simulation platform masFIRA is developed in NetLogo, a programming language especially designed for agent based programming[11]. NetLogo is a successor of the well known programming language LOGO, which had been frequently used from the 80ies, primarily in fields of education. The semantic of LOGO was, and the semantic of NetLogo still is quite near to spoken English, so the code is not too compact, but can be understood quite instantly, even by a reader, who has no experience within this programming language.

Operating system: NetLogo itself is programmed in Java, so it can be developed and run on different operating systems. masFIRA, for instance, has been primarily developed on a Windows 2000 resp. Windows XP computer, but some modules were designed and improved within LINUX operating system (SuSE 9.0, with latest Linux – (alpa)Kernel). For Windows platforms, the compiler comes as an executable file, so additional software is needed. Within LINUX, the (latest) Java-RE is needed to start the programming environment and run the applications. In general, NetLogo is designed as stand alone application.

Web-based applications: Individual models can be run as Java applets inside a web browser. Like the stand alone applications, a simulation run within a web browser takes all computational resources available (within Windows systems), so other programs run simultaneously will slow down considerably.

Turtles', patches', and globals' monitors: The status, personal endowments, and characteristics of each agent (so called "turtles" in NetLogo) can be viewed separately (and even simultaneously) at any stage of simulation via the so called "turtles monitor". A similar monitor exists for all cells of the grid ("patches" in NetLogo). Global variables of course have one unique monitor that displays the values of all global variables.

Behaviour space: A valuable element of NetLogo is the so called "behaviour space", where simulations at different parameter values can be run automatically. The researcher gets the results in structured tables and graphs. With this feature NetLogo has become a convincing tool for scientific research. Without this feature, to run comparable simulations (program driven), sophisticated meta-object would have to be defined with each project.

Hub Net control: For interactive classroom demonstrations or experimental simulations with interaction to a couple of "real" agents – humans that participate in scientific experiments – it is possible to program a so called "Hub Net". Stimuli and responses of the real agents are sent from their workstations or notepads to the hub-net server, are processed there and the reactions of the virtual agents come back to the humans, and so on. This is the only feature, masFIRA does not make use off yet.

License for NetLogo: The NetLogo software, models and documentation are distributed free of charge for education and research[12].

---

[11] See Wilensky (1999)
[12] COPYRIGHT STATEMENT for NetLogo:

4.1 A NetLogo-example of agent based simulations in social sciences.

To give some insight how ABS works, a small program is described[13]. Agents are constantly situated on their cells. They show up different grades of altruism. Agents with high grade or altruism prefer to make their living in cooperation with their neighbours – given that these neighbours also show a high grade of altruism – at some costs and at some (expected and realized) benefit from cooperation. Given an agent just has selfish neighbours, altruism does not pay any longer, so this agent will probably reduce his willingness to cooperate. A full description as well a depiction of a simulation run is in the Appendix (page 25ff).

Simulation starts with randomly distributed values for the degree of altruism. About the half of the population is attributed either to altruists or selfish agents. The other 50% of the population does not belong to either group yet (Figure 6).

First, the indifferent agents decide whether to have a high of low propensity to cooperate. In this short phase the share of altruists as well as the population of selfish agents rises. This takes about 10 periods in the simulation. Then, within the next 100 periods, the "breed" of altruists diminished, but after control action (1)[14] by the user this movement decelerated. Nevertheless, population of altruists still decreased. After control action (2)[15] altruists' population share started to rise, while the population of the selfish agents diminished (Figure 7).

[13] This is a simulation of Wilensky (1998)

[14] increasing (global) "harshness of living conditions" parameter (from value 0.00 ➔ to value 0.73)

[15] increasing "benefits of altruism" (0.48 ➔0.75)

5 The concept of masFIRA

5.1 General remarks

Generally, the masFIRA simulation platform is designed as open as possible. As the project will not be accomplished within one continuous study, rather a couple of additional modules will be programmed and implemented with each step. At the end of each step the simulation platform has to work with the elder procedures alone as well as together with the new modules. For example: with the first step the environment and the agents-matching procedure was developed. The following step – the bargaining procedures over the critical event "giving birth to another child" – has to be programmed that way it can be switched off easily. That way there is a guaranty that procedures can easily be replaced after a while or – more interesting – alternative procedures and competing models of human behaviour can be tested and compared sequentially (changing the behavioural model for all agents) or simultaneously (e.g. changing the behavioural model for half of the "male" agents by assigning every male agent with an even-numbered ID an alternative model).

Milestones in the development of masFIRA are:
- design of the fundamental environment (see 5.2)
- design of (new) output
- implementation of (competing) models of human behaviour at the core (see 5.3)
- stratification and alignment procedures ➔ link to demographic and econometric research

Note that these milestones are not in sequential order. Every module can have elements of all four milestones. Example: The fundamental environment changes with addition of the agents' characteristics that in turn have to be extended with implementation of additional behavioural elements. So, for instance, when agents just mate according to the value of a scalar concerning "attractiveness", agents just have to be endowed with this item. When – designed and implemented in another module – agents select their partners according to a group of values (say, education, age span, inclusion in certain social class, region, religion, and wealth) all values of the concerning vector have to be compared. In addition, agents have their individual valuing and weighting function for these characteristics, so the parameters of the valuing and weighting function also have to be implemented.

5.2 Environment and the fundamental procedures

In contrast to the compact altruism model discussed in 4.1, the idea of masFIRA is much wider. As the model is designed open to additional models, the environment (agents' and cells' characteristics, interaction rules) have to be defined flexible to meet future changes and extensions. Within this simulation environment, agents act actively while cells are entirely passive. The just change properties when the agents situated on them have called for this change.

First, behavioural rules for of agents, and the properties of cells and globals will be listed. After that, the most important underlying behavioural models – the core of this ABS – will be described.

5.2.1 The agents

Agents are embedded in their family network. Each agent therefore knows his parents and his children. Further information on family networks can be queried by each agent from the relatives s/he knows

(e.g. to have a list of first grade nephews and nieces the agent first asks his/her mother for the ID of his/her siblings, then s/he queries the IDs of the children of these agents). In fact agents forget about their former partners, but – given they have common children – they can ask them who this partner was.

The search process for partners – it's rules, and the communication tasks within – is quite straightforward: Agents are separated in gender classes and have a certain age. Females are located persistently on their grid[16]. When having become adult[17], males start to search partners. Males are moving around the grid, with differing speed dependent on their propensity to find a partner. Depending on the speed, density of the population[18], and the age-specific gender distribution the male will come – sooner or later – to a cell inhabited by a female. Now two scenarios can arise: Either the male's age lies within the accepted partner's age span[19] of this female, then he can stay for a moment, or not – he will be thrown out. Typically the male moves on one cell in either direction, but, when after a number of trials he does not come to a female-empty cell, he has to jump farer distances[20]. In case that the male's age is convenient, the male now considers the age of the female. Given this match in successful, the potential partners have a closer look on the other, described in 5.3.1

Agents are also equipped with negotiation power – simply a scalar with limits [0,1] – that supports them in intra-family bargaining situations. This scalar is fixed yet, but will become a fuzzy function[21] of past experience and human capital. Past experience will be individually weighted by actuality.

Agents are equipped with a scalar that values the satisfaction within current partnership [0,1], where single agents have [0]. (The case of "formerly experienced spouses, now convinced singles" is also recognized, but not depicted in this parameter).

Agents differ in their propensity of having children – their fecundity. Males' fecundity lies in the interval of [0,3.5] while females' fecundity can come up to 4 children. Note that fecundity is real, not integer! For every full interger a child is wanted for sure, for the rest, the decimal values remaining a more or less strong position is taken in the bargaining process whether to have another child or not.

The agents also "remember" certain dates, like the own day of birth or the day of mating the current partner – gender makes no difference in this respect[22]. Further agents are endowed with certain stocks of human capital that depreciates over time. Agents have to decide whether to reinvest at some costs (both, in terms of money and time). Agents receive income – wages, capital income, and transfers[23]. They assign their time to market labour, home production, child care, leisure and educational activities. Agents own private goods, that are purchased on markets at globally known prices. Families own – in

---

[16] In fact females only move once their life – when having become adult and leaving the parent's home (cell)
[17] all agents become adult at age 20, but this can easily be heterogenized
[18] number of (adult) agents per number of cells on the grid
[19] Males and females generally have different age spans they accept for their partners. The age spans can be fixed or be a function of the quantity of potential partners globally "offered" within a year. As described below, the time span also differs individually.
[20] Still, it is possible that one male meets the same female more than once, but more unlikely the shorter the time interval.
[21] the fuzziness will be controlled globally
[22] but of course – in order to meet reality – this can be implemented. Just kidding.
[23] Only private transfers are considered yet. Neither public households nor finance sectors have been considered yet. Capital income is gained due to a unique and exogenously fixed interest rate on property. This rate does not vary.

addition – public goods[24], also associated to globally known prices. These goods cannot be consumed directly by the agents. They have rather to be transformed by the home production process to consumable "commodities".

Males, in addition, have a certain degree of finding a partner [0,1] that determines their speed, women have show certain propensity to take a partner. Within these two parameters the historical experience, the difference of individual fecundity and fertility, and the possibility to improve the current (economic) situation are reflected.

5.2.2 The grid

Currently, a cell on the simulation grid just can have two states. Either, a household has been build on it (cell is marked yellow), or not (default colour). No mate-searching male will try to get on a marked cell[25].

Although depicted as rectangle, the grid has – like in many ABS platforms – the geometric form of the surface of a torus. Hence, like described in beginning of chapter 3, an agent "leaving" the grid on the left will "return" on the right. No agent will drop out.

5.2.3 Global values

As described above, prices – and hence wages – are globally known and exogenous to the individual agents[26]. In addition

- Time (measured in days and years; per day one transformation in state can be executed perr agent),
- Demographic variables like
  - Pregnancies (percent on women in partnership),
  - Share of fertile partnerships (just age criterion; does not control for zero-fecundity),
  - Completed fertility (share of "realized" children in respect to initial fecundity[27]),
  - Unrealized fecundity (rest-fertility),
  - Age distribution by gender, and
- Variables needed for technical reasons

are globals to the simulation procedure.

---

[24] For simplicity, these goods are associated with the cell the household is located on. In future versions, public goods can be associated to agents – at least temporarily. In case of a divorce, partners have to negotiate on the shares of public goods each adult agent gets. Up to now females stay with the public goods, but do not act strategically in this respect. Anyway, in case of divorce, males have to build up public again. As the property of public goods is not a criterion for the attractivity parameter, this plays no mayor role.

[25] Perhaps later this assumption will be relaxed to improve to reality

[26] in fact it will be the most fruitful challenge to endogenize the price vector. In this stage just one private and one public good exists. While agents can differ in their productivity regarding home production, they do not in respect of market labour. This reflects widely the state of the discussion within the theoretical papers, this simulation is grounded on.

[27] As – implemented in a recent module – fecundity varies with experience and social norms, the definition of this global has to be changed to [# children / (# children + present fecundity)], as the agents do not remember their initial values of fecundity. Anyway, this information is more valuable to the experimenter.

5.3   The core – variation in agents' behaviour

According to underlying game-theoretical models, following procedures are went though by every simulation loop.

5.3.1   Partner matching

As described in 5.2.1, currently unmated males search for females by wandering a random walk over the grid. Once they have found a matchable female (according to age), partnership negotiation begins: As every agent (males like females) is endowed with a certain degree of attractiveness[28], both compare the attractiveness level of the potential spouse to their "attractiveness reservation level". This reservation level is a function of

- primarily, the own attractiveness; additionally
- social status, indicated by
  - human capital level, and
  - wealth.

When attractivity meets the demand bilaterally, a partnership is initiated. Theoretical models refer to the fact, that prevailing differences in attracivity between the partners contribute to differences in bargaining positions, as outside options differ. In this simulation model, this contribution to strategic settings is left disregarded, as the agents have no information about outside options in general. Of course, in later versions this also can be implemented by defining respective globals.

A special issue on agents who haven't found a partner "for years" has been added. Comparable to contact announcements in newspapers, a global "blackboard" has been implemented. Agents with respectively high propensity to find/take a partner that haven't succeeded up to a certain "panic age" that can be set globally by the user – as the age a female/male "has to be mated" at refers rather to a social norm than to individual values[29] - can put their "names" (agent IDs) on this global blackboard. Whenever a possibly matchable female (again, the male agent just can identify the gender and the age of the annotated female) can be identified on this blackboard, the male possibly crosses the whole grid to get to her[30].

Finally, as will be seen in the depictions of some simulations, enough matches can be made to hold population at least constant. But as many events are depending on well-defined probabilities, this can not been guaranteed[31].

---

[28] Attractiveness in it's general meaning. Far more than physical attractiveness.

[29] A valuable contribution on this issue has been programmed and evaluated by Billari/Prskawetz/Fürnkranz (2003), represented as – reprogrammed – NetLogo application at the conference *on "ABCD – Agent-Based Computational Demography*", organized by the Vienna Institute for Demography – Austrian Academy of Sciences (Vienna, January 2004)

[30] up to now no costs are associated to searching and moving

[31] Up to the present stage, agents haven't become "intelligent", they can not adapt rules to meet changes in their environment. So – negatively formulated – at this stage, it is still a question of time a populations dies out, either slowly because of insufficient fertility, or fast, due to an overcrowding of the grid. In this situation the male agents, who are searching for mates, can not continue to move. Single females stay unmated, grow old, no new births occur. The elder stay in their cells (homes) 40+ up to (possibly) age 99, "wasting" the space needed by the younger, still fertile agents.

5.3.2   Resource allocation

As partners have matched, the daily struggles have begun. This simulation focuses on the distribution of leisure, private and public goods as well as on future chances. Agents' preferences are described by a quasi-utility function, which represents the time allocation, public and private goods, and an expected value of human capital level[32]. With every loop (every simulated year) partners renegotiate their realizable levels of these items. Given the negotiation outcome falls short in case of one item in respect of its reservation level, the agent considers to leave the partner. Satisfaction with current partnership (agent's parameter) declines. As reservation levels change with outside options, the negotiation contends different arguments each year.

5.3.3   Decision for/against offsprings

Following Becker (1993, p 135ff), parents consider children like "long-term (intra-household-public) consumption goods" they gain additional utility from. These "goods" are associated with costs (time, wealth). Up to this stage of the project, female agents in partnership[33] give birth to children according to the minimum of fecundity of both partners. Expample: male's fecundity (2.2) & female's fecundity (3.1) ➔ the female agent will become pregnant – controlled by the hazard rate of becoming pregnant within the recent month – and give birth to the child within the appropriate time span (38 - 40 weeks). When the partnership ends, female agents of course stay pregnant, but can not become pregnant unless a new partner is found. Given the partnership holds for longer period – and fecundity does not change too much - a second pregnancy will come (as both agents have fecundity parameter > 2.0). After birth of the second child negotiation on a third child occurs. Depending on the agents' bargaining power and respective values of "rest-fertility" (m: 0.2; f:1.1) the probability of having another child is calculated. Based on this probability, a Bernoulli-experiment on a third pregnancy is executed. Within this approach, unfulfilled fecundity is not an argument for partnership stability[34], so when once the decision was set against a(nnother) pregnancy, this negotiation will not be repeated within this partnership, but it is no argument for slitting up. Nevertheless, when new partnerships are formed, the agent's rest-fertilities are again subject for negotiation.

As agents negotiate over children immediately after a partnership has begun and renegotiate after a quite short interval after birth, timing and spacing of births is very frequent. No life-course interdependency has been introduced yet.

5.3.4   Partnership stability

As mentioned above, partnership holds as long reservation levels are kept exceeded. So, the most likely case for separation occurs, when agents have recently founded a partnership, they have accepted the other as reasonable (age) and sufficiently attractive (regarding general attractiveness, wealth and human capital – items that can not be adjusted within short time spans), and then start to negotiate over time investments, distribution of private goods, and levels of public goods. When reservation level for an item is not reached, this partnership splits up immediately[35].

---

[32] In fact reservation levels of (uncompensated) demand function are assigned to the agents
[33] single-motherhood is excluded from simulation yet – that's another issue for further modules
[34] this is a great challenge to an utility-based negotiation procedure to be developed
[35] as these steps all occur within one simulated day, this refers – in some sense – to a one night stand. Contrary to one night stands observed in reality, agents ever honestly try to build up a long term partnership.

Given a partnership has lasted longer[36], partners do not split up immediately when reservation level of a single item hasn't been exceeded. In this case, the parameter for "satisfaction in current partnership" is reduced[37]. Having started a partnership the value is set to [1]. After renegotiation the value is replaced by

$$s = 1 - \sum_i x_i^u q_i \Big/ \sum_i x_i q_i \qquad (3),$$

where x_i^u denotes an occurrence of shortfall, x_i the kind of item investigated, q_i the agent's preference for this item . The probability of ending the partnership is simply set to $1 - s$. The decision of one partner for splitting is sufficient. The partnership – possibly – ends within 365 days (time of splitting up is uniformly distributed).

6   Simulation runs

Before showing results, here some additional information on the process sequences.

6.1   Setup procedure and starting population

In this procedural first step agents are created. The experimenter can define the number of male and female adults. In addition the experimenter sets the initial net reproduction rate (NRR) of this artificial population. With this information, the simulation platform creates the number of adults plus – according to the NRR – the number of children a female agent already has. The probability of having (multiple) children is determined by the fecundity parameter and the age of the female – as it is more likely for a female aged 35 to have 3 children, as it is for an agent aged 20.

The experimenter can control for the age distribution of the artificial starting population by varying the maximum age. Following a uniform probability distribution for age[38], the adults are aged [20 – maximum age]. As number of agents is comparably low – at least for experiments that should process comparably fast – the uniformity of age distribution can hardly be reconstructed. At the end of the setup procedure the age distribution is calculated and displayed (Figure 1).
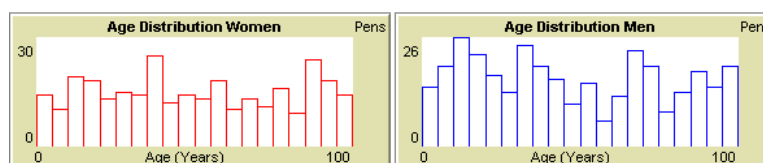
Figure 1: Age distribution of population

All parameters of the adult agents are initialized within this setup procedure. In contrast, child agents haven't initialized all values. They rather "inherit" some parameters from their parents. So, the driving determinant of fecundity of an adolescent child is the number of (siblings + 1) plus a globally defined

---

[36] At minimum two days; but as bargaining over resource allocation happens once a year, renegotiation on the next day is very unlikely

[38] to be exact, for a certain day of birth; as the starting day of the simulation is labelled "0", the day of birth of members of the starting population has to be negative

propensity. Therefore individual values as social norms influence the starting value of this individual parameter.

6.2   The simulation

After setup has been completed[39], the simulation is ready to start. With the GO command, males start to move and search for mates, matched partners negotiate resource allocations and fertility decisions, some females become pregnant, after while new agents are born, divorces occur, child agents become adults, and elder agents die.

In this section some computational experiments will be shown. Although the behavioural model behind this simulation has been designed and grown in manifold dimensions, little emphasis has been placed on the output yet. This has several reasons: first, the average outcome of individual negotiations should be depicted on individual basis, second, aggregated levels of, say, satisfaction with current partnership, will not show its distribution. Further modules will provide distribution statistics of these parameters and controls on global social norms[40].
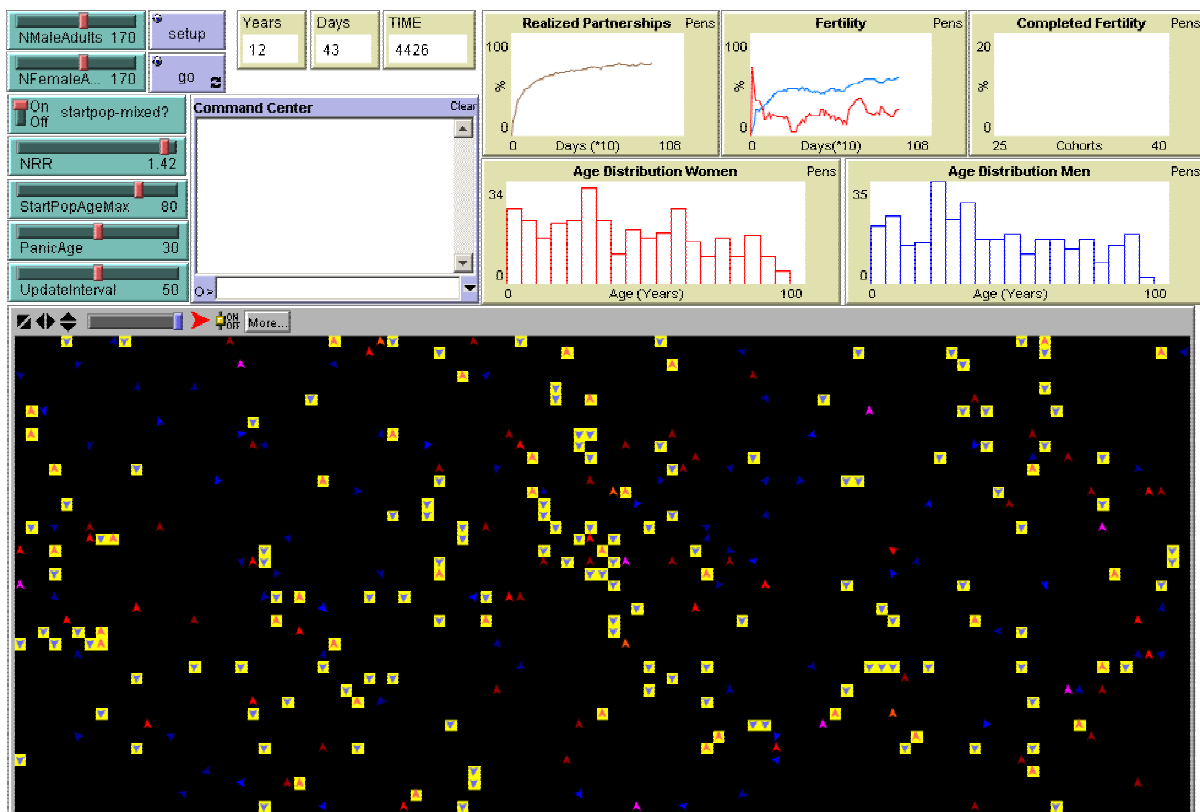
Figure 2: masFIRA simulation run (standard setting)

The simulation depicted in Figure 2 shows typical results: The starting population contained 170 male and female adult agents each plus a number of children (living at the mother agent's cell. At startup, the

---

[39] can take several minutes for larger populations!

[40]   for instance a controller for splitting the partnership, when satisfaction level is below one; (0;1]. With this controller the probability of divorce can be decreased globally

agents were aged up to age 80, were situated in a mixed population setting[41] . After 12 years and 42 days (4426 simulation runs) about 60% of adult agents are living in partnerships. The others are actively searching (males) or passively waiting to be found by an attractive partner(females)[42].

The simulation grid (black area) depicts the simulation grid. Males (blue) are searching for female mates in every direction possible with individual speed. Females (red) in turn wait[43] and see. Agents that have founded a partnership move together (both on one cell). This cell therefore turn yellow. Child agents (males: purple; females: orange) also live on that cell (share the household with their parents)

Figure 3 shows results of other simulations, where first the probability for divorce had been set globally near zero (a). Some oscillation of aggregated values of realized partnerships occurred, but that was due to the fact that some child agents became adult, shifting the total number of adults up (at same time no deaths occurred), so the slope of the share of adults in partnerships sometimes became negative.
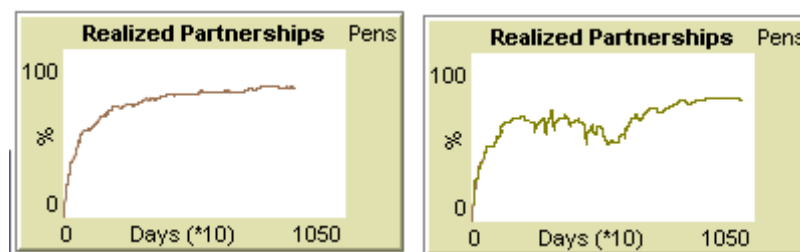
Figure 3: Realized partnerships (a) no divorces   (b) some divorces from sim-period 2000 – 5000

The same parameter settings were used in the beginning of the second simulation in (b). Around simulated period 2000 (5 years 175 days) the experimenter raised probability of divorce by enabling the possibility that parameter ''satisfaction in current partnership'' becomes less that ''1'', so divorces rose, the share of realized partnerships started to oscillate, even with a negative trend. Around period 5000 another change occurred: The experimenter changed divorce behaviour by changing weights of selected reservation levels. Although is was still possible to divorce, as de-facto just one utility-gaining activity stayed relevant for reducing the satisfaction within current partnership (leisure) all agents that would have separated due to inconveniences regarding other activities or goods now had no (valuable) reason for separating. As agents have grown older, mostly with the experience, that they could not agree on resource allocation with sufficiently attractive agents[44], the part that has passed ''panic age'' – the social norm for being in a long-term partnership – now search more efficiently by watching the global blackboard for contact announcements. For this reason the slope of the share of adults in partnership rises more strongly than at periods near under 2000.

---

[41] for some simulations a gender-segregated setup can be selected
[42] of course this setting could be designed in vice versa. The matching probability decreases sharply when both genders search actively.
[43] All females point their head to in one direction, but sometimes an adolescent female moves from the mother's place to a free cell, so this particular agent points her head temporarily to another direction
[44] The ''one night stands''
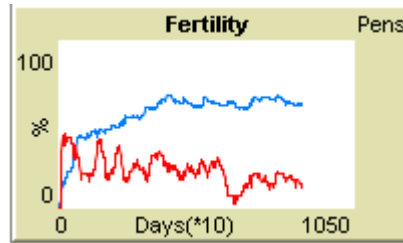
Figure 4: Fertile agents in partnerships and fertility. Simulated from setup procedure on.

The information on the share of adult agents in partnerships in quite valuable, more interesting is the information on adult fertile agents in partnerships. The graph depicted in Figure 4 shows this share (blue line) for a simulation with no divorce. The oscillation is due to the fact that agents join this group (adolescents), others leave the group (females > age 40; this share depicts in fact just females). A certain percentage has decided to become pregnant, most are pregnant in fact (red line). As pregnancy occurs quite immediately after forming a partnership, pregnancies typically rise up sharply in early stages of the simulation, returning to adequate levels after about 2 years. For this reason often pre-simulated starting population are used, where population has been - at least partly – generated by the simulation process than by the setup procedure (Figure 5).
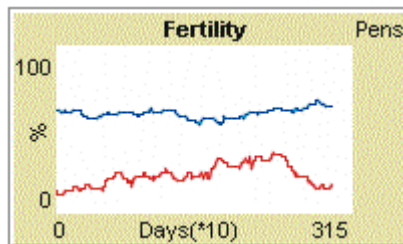
Figure 5: Fertile agents in partnerships and fertility. Population pre-simulated.

7 Summary and Outlook

Within this documentation the fundamentals of bargaining models concerning

- family formation,
- fertility,
- resource distribution,
- and partnership dissolution

were reflected. A short introduction on agent based modelling shed some light on this rather new technique for economic analysis. In a compact section the agent-based-simulation developing environment NetLogo was introduced. A description of the most important processes and optional modules within masFIRA followed. Finally some simulation outcomes were discussed.

As masFIRA will be continually developed and updated, this documentation should rather be seen as status report on modules that were implemented by end of 2004. In 2005 further challenging modules will be developed. The implementation of a comprehensive bargaining model, where all relevant decisions are negotiated within, and where cardinal utilities functions can replace the quasi-Marshallian demand functions used now, is planned. Further, special alignment processes will be implemented. For instance – to describe the most simplest invention – the risk of dying will not continue to be arbitrary, but depend on observed hazard rates, implemented by life tables.

The simulation platform masFIRA is expected to become a valuable demonstration tool for demo-economic models, as well a valid quasi-empirical simulation tool that helps analyse relevant observations.

Literature

Apps, P. and R. Rees (1996) *"Labour supply, household production and intra-family welfare distribution"*; Journal of Public Economics 60-2; pp 199 - 219

Axelrod, R. (1992) *"The evolution of cooperation"*; Basic Books

Ballot, G. and G. H. Weisbuch (2000) *"Applications of Simulation to Social Siences"*; Hermes

Becker, G. S. (1993) *"A Treatise on the Family"*; Havard Univ. Press

Billari, F. C. and A. H. Fürnkranz-Prskawetz (2003) *"Agent-Based computational Demography. Using Simulation to Improve Our Understanding of Demographic Behaviour"*; Physica

Bourguignon, F. and P.-A. Chiappori (1992) *"Collective Models of Household Behavior"*; European Economic Review 36-2-3; pp 355 - 364

Chiappori, P.-A. (1988) *"Rational household labour supply"*; Econometrica 56-1; pp 63-89

Chiappori, P.-A. (1997) *"Introducing household production in collective models of labour supply"*; The Journal of Political Economy 105-1; pp 191 - 209

Gilbert, N. and K. G. Troitzsch (1999) *"Simulation for the Social Scientist"*; Open University Press

Kahneman, D. and A. Tversky (2000) *"Coices, Values and Frames"*; Cambridge Univ.Press

Konrad, K. A. and K. E. Lommerud (2000) *"The bargaining family revisited"*; The Canadian Journal of Economics 33-2; pp 471 - 488

Meadows, D., et al. (1972) *"Grenzen des Wachstums, Die"*; Deutsche Verlags-Anstalt DVA

Ott, N. (1992) *"Intrafamily bargaining and household decisions"*; Springer

Rosenzweig, M. R. and O. Stark (1997) *"Handbook of Population and Family Economics"*; Elsevier

Simon, H. (1966) *"Models of man"*; Wiley

Simon, H., et al. (1992) *"Economics, Bounded Rationality, and the Cognitive Revolution"*; Edward Elgar

Weiss, Y. (1997) *"Formation and dissolution of families: Why marry? Who marries whom? And what happens upon Divorce"*; in: M. R. Rosenzweig and O. Stark: "Handbook of Population and Family Economics"; North-Holland (Elsevier);pp 82 - 123

Wilensky, U. (1998); Altruism http://ccl.northwestern.edu/netlogo/models/Altruism

Wilensky, U. (1999) *"NetLogo";* *http://ccl.northwestern.edu/netlogo*

Willis, R. J. (1973) *"New Approach to the Economic Theory of Fertility Behavior"*; Journal of Political Economy 81-2; pp 14 - 64

Wooldridge, M. and N. R. Jennings (1995) *"Intelligent agents: theory and practice"*; Knowledge Engineering Review 10

Appendix

An example for ABS-simulation in NetLogo: "ALTRUISM"

To see how NetLogo works, a simple model of human behaviour is presented here[45].

This model (Cooperation and Divide the Cake) are part of the EACH curriculum: "Evolution of Altruistic and Cooperative Habits: Learning About Complexity in Evolution".

This is an evolutionary biology model. It models population genetics with respect to the fitness of traits that are affected by social and environmental conditions. The model has two types of patch agents: altruistic agents and selfish agents.

The basic premise of the model is that the selfish agents and the altruistic agents are competing for each spot in the world by entering into a genetic lottery. You can imagine these agents as plants who "seed" for a spot, and the dominant seed generally wins. The details of the lottery are explained below in HOW IT WORKS.

Under normal (non-interfering) environmental conditions, the selfish agents win, and the altruistic population is driven to extinction. However, as outlined in 'CONTROLS ', when the environmental conditions are made more harsh, the altruistic population is able to survive, and even dominate the selfish population.

How it works:
1. Patches live in five-cell, plus-sign-shaped neighborhoods. Whenever a patch is calculating something about its fitness, it is the center of the neighborhood. For another patch, when that patch is calculating, it becomes merely one of the neighbors.

2. Each patch is an agent that has a fitness. Each patch is also the location of a lottery for its space. The patch and the four surrounding patches put in "seeds" to try to get the patch turned to their type of patch, altruist or selfish. Being successful in the lottery is getting patches to turn to your type. We are assuming here that the type (altruistic or selfish) is the important genetic trait.

3. Each patch calculates its own fitness using equation:
if it is A (altruist): 1 - cost + (Number Altruists in Neighborhood / 5 * benefit from Altruists)
if it is S (selfish): 1 + (Number Altruists in Neighborhood / 5 * benefit from Altruists)

Thus, the fitness of the S patch will be higher than the fitness of the A's. If the cost is 0.2 and benefit is 0.5, for an A surrounded by two S's and two A's, then the fitness of this spot is 1 - 0.2 + (3/5 * 0.5) = 1.1.

---

[45] See http://ccl.northwestern.edu/cm/EACH/ for more information. Following text is taken from the program documentation
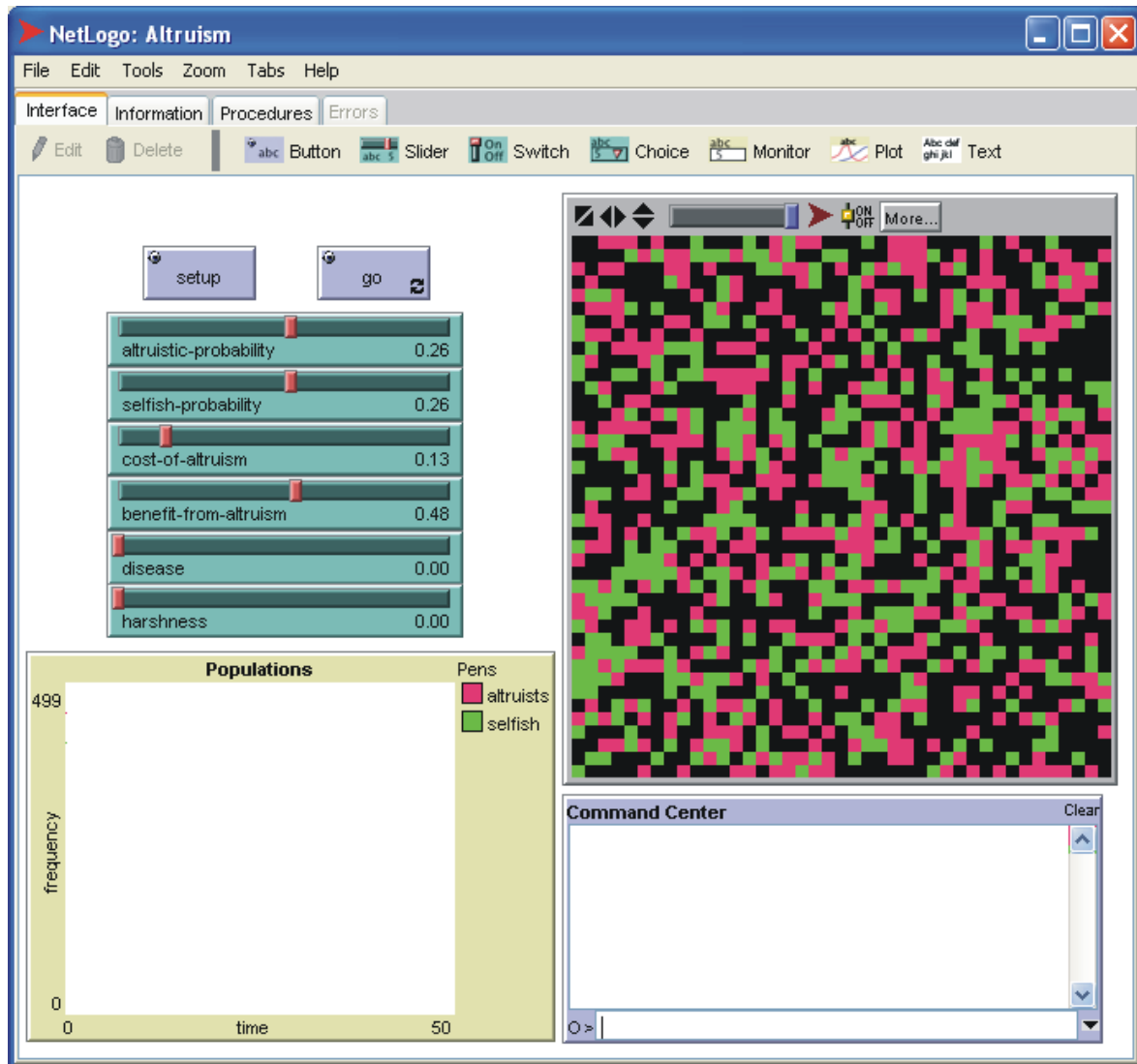
Figure 6: Simulation „ALTRUISM". Setup values

4. After each patch has calculated its fitness, it looks to its four neighbors. Each of the five patches, including itself, puts a weighted seed into a genetic lottery for this center spot. So, for example, if the neighborhood is ASASA, each of the three A's register their fitness value, and each of the two S's put in their fitness. The A's are added, and the S's are added. Let us assume that the A's add up to 3.2 (this includes the A in the center spot), and the S's add up to 2.6. These two numbers are the altruist weight and selfish weight respectively, in the lottery for the center spot. Now, the larger number, whichever it is, is called the Major seed; it is divided by the sum of all the fitnesses.

Thus, 3.2/(3.2 + 2.6) = .552
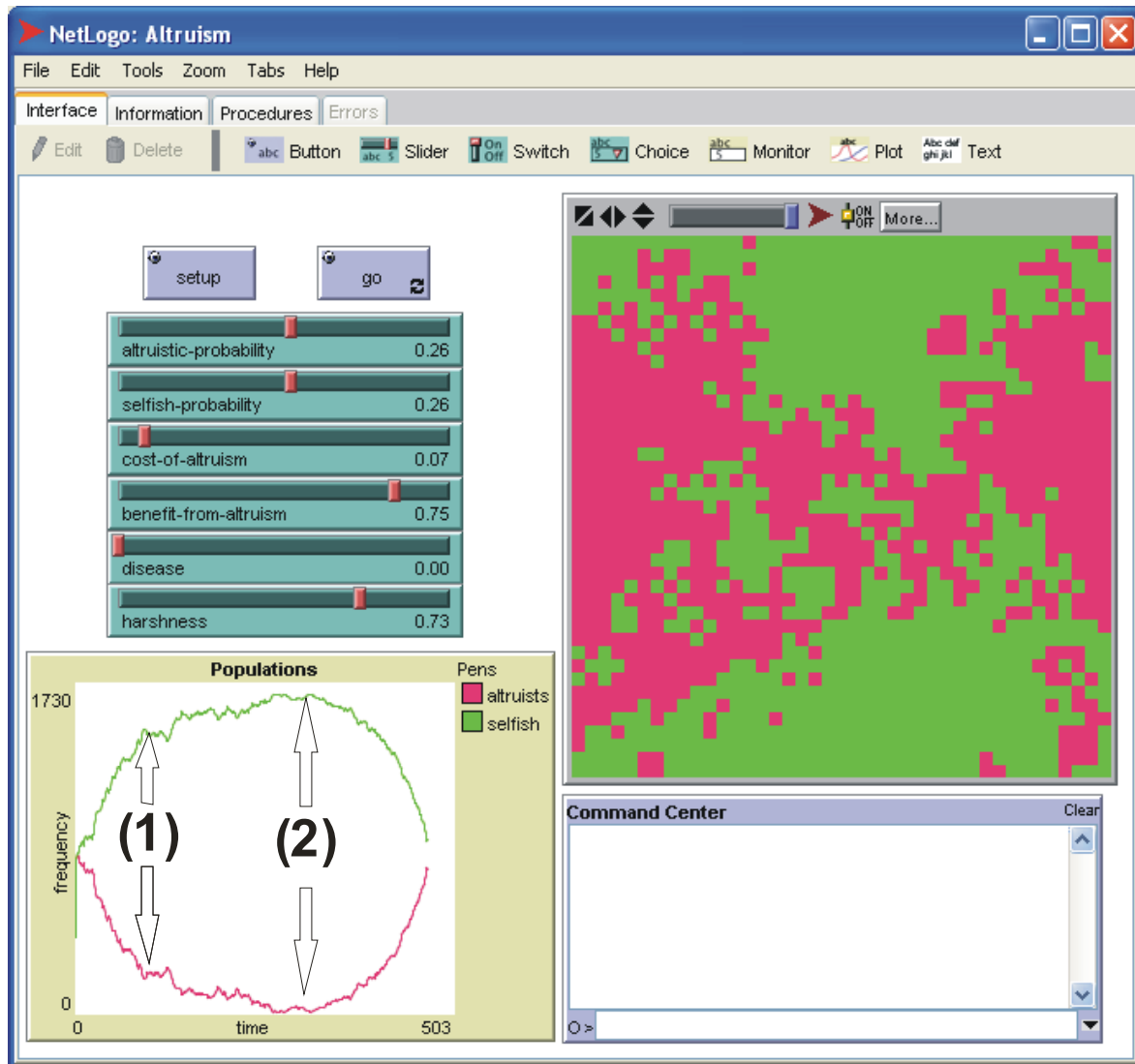
Figure 7: Simulation „ALTRUISM", run for about 450 periods[46]

This number is the Altruism seed in the lottery. The minor seed is 2.6/(3.2 + 2.6) = .448. (Notice that the Altruism seed of the parent is 3/5 = .600, while the child's is .552. Even though altruism is dominating, it is losing ground.)

5. There are a number of ways of doing the lottery itself. Currently, we choose a random number between 0 and 1. Now, if the Number is below the Minor seed, the minor weight gets the spot, and if it is above the major seed, the major seed gets the spot. So, in the example, if the random number is anywhere from .449 to 1, then the Major seed gets it. If it is between 0 and .448, the minor seed gets it.

---

[46] . Two changes of parameters by the user: (1) increased "harshness" of living conditions (0 ➔ 0.73) ; (2) increased "benefits of altruism" (0.48 ➔ 0.75)

<u>Controls for the user:</u>

SETUP button -- sets up the model by creating the agents.

GO button -- runs the model

ALTRUISTIC-PROBABILITY slider -- lets you determine the initial proportion of altruists

SELFISH-PROBABILITY slider -- determines the initial proportion of selfish agents.

ALTRUISM-COST slider -- determines the value of cost in the above fitness equations.

BENEFIT-FROM-ALTRUISM slider -- determines the value of benefit in the above fitness equations.

There are two sliders for controlling environmental variables:

HARSHNESS slider -- sets the value for the resistance of empty patch spots to being populated by agents. The value for this slider determines a corresponding value in the lottery for each empty (black) spot on the grid; the higher this value, the more difficult it is to populate.

DISEASE slider -- sets the value for the possibility that the agents in occupied spots will die. The value for this slider is factored into the genetic lottery, and determines the percentage chance that each agent will die out from their spot.

The NetLogo Code of masFIRA (at 2004-12-27)

```
globals [
  time
  days
  years
  days-in-current-year
  partnerships-pct
  fertile-partnerships-pct
  pregnancies-pct
  any-break-condition-met?
  completed-fertility
  rest-fertility
  realized-fertility
  rest-fertility-avg
  realized-fertility-avg
  m-agelist
  f-agelist
  avg-female-desire-for-late-partnership
  avg-male-desire-for-late-partnership
  search-partner-blackboard
]

turtles-own [
  mother                                   ;; ID of mother
  father                                   ;; father's ID
  child                                    ;; all children ID
  agent-in-start-population?               ;; mark agents in start population
  adult?                                   ;; adulthood not just depending on age, rather a
functionof several factors (adults are prepared to search/accept a partner)
  age                                      ;; age in years
  agedays0                                      ;; age in days at setup (if agent in start
population)
  agedays                                  ;; age in days
  day-of-birth                              ;; since simulation started (if agent in start
population  => agedays0 * (-1))
  attractiveness                                ;; {(0,1)+delta} males are "less attractive" =>
have to compensate with social status variables
  negotiation-power                        ;; (0,1)
  satisfaction-in-current-partnership        ;; (0,1) - 0 at start
  propensity-for-having-children           ;; (0,4.5) - setup range varies by sex;
  day-found-recent-partner                 ;; no partner => day of birth - 1
;  human-capital                            ;; (0,1)
;  income                                   ;; (0,.)
;    wage                                   ;; (0,1000)
;    capital-income                         ;; (0,.)
;    transfers                              ;; (0,200)
;  property                                 ;; (0,.)
; activity-labour
; activity-home-production
; activity-child-care
; activity-leisure
; activity-education
; commodity-public
; commodity-private
; price-of-commodity-public
; price-of-commodity-private
]

breeds [male female]          ;; define types of agents

male-own  [
```

```
    propensity-to-search-for-partner              ;; (0,1)
    found-attractive-woman?
    partner                                        ;;  the turtle that is our partner, or "nobody"
if we don't have one
]

female-own  [
    propensity-to-take-a-partner                  ;; (0,1)
    considered-by-man?
    partner                                        ;;  the turtle that is our partner, or "nobody"
if we don't have one
    day-got-pregnant                               ;;  not pregnant => day of birth - 1
    pregnant?                                     ;;
]

to setup
  ca
  set-starting-values
  create-agents
;  giving-birth-to-startpop-children
  setup-plots
end

to go
  break-conditions
  if any-break-condition-met? [stop]
  search-mates
  intra-partnership-negotiations
  time-dependent-events                                ;; all time-dependent-events
end; go

;;~~~~~~~~~~~~~~~~ begin SETUP procedures  ~~~~~~~~~~~~~~~~
to set-starting-values
  set time 0
  set days 0
  set years 0
  set any-break-condition-met? false
;  ask patches [set pcolor green - 2]
end; set-starting-values

to create-agents                      ;; create types of agents
  create-adult-agents
  giving-birth-to-startpop-children
  calc-age-distribution
end ; create-agents

to create-adult-agents
  create-custom-male NMaleAdults [       ;; male adults
    set color blue
    set age 20 + random(StartPopAgeMax - 20)
    set-startpop-adults-unisex-properties
    set propensity-to-search-for-partner random-float (1)
    set propensity-for-having-children (random-float(2) + random-float(1) + random-float(0.5))
    set found-attractive-woman? false
  ]
  create-custom-female NFemaleAdults  [    ;; female adults
    set color red
    set age 20 + random(StartPopAgeMax - 20)
    set-startpop-adults-unisex-properties
    set propensity-to-take-a-partner random-float(1)
    set propensity-for-having-children (random-float(2) + random-float(1) + random-float(1) +
random-float(0.5))
    set considered-by-man? false
    set day-got-pregnant (day-of-birth - 1)
```

```
      set pregnant? false
  ]
  ask turtles [without-interruption[
    ifelse (breed = male)
    [ ifelse (startpop-mixed?)
      [ fd random (screen-edge-x * sqrt(2))
        if (count male-here > 1)
        [ go-to-male-empty-cell ]]
      [ fd random (screen-edge-x * 0.5) ]
    ]
    [ ifelse (startpop-mixed?)
      [fd random (screen-edge-x * sqrt(2))]
      [fd (screen-edge-x * 0.5) + random ((screen-edge-x * 0.5) * sqrt(2))]
    ]
  ]]
  ask female with [adult?] [without-interruption[
    if (count female-here > 1) [
      go-to-female-empty-cell
      fd random (screen-edge-x * sqrt(2))
    ]
    set heading 0
  ]]
end; create-adult-agents

to giving-birth-to-startpop-children
  locals [number-of-startpop-children x1]
  ask female with [adult? and age < 60] [without-interruption[
      ifelse (((age - 19) / 20 ) > 1) ; set weight parameter for completeness of fertility
period
        [ set x1  1 ]
        [
          set x1 ((age - 19) / 20 )
        ]
      set number-of-startpop-children  floor (propensity-for-having-children * x1 * (0.8 +
random-float 0.4) * NFR + 0.5)
      hatch number-of-startpop-children [set mother myself]
      ifelse (number-of-startpop-children < propensity-for-having-children)
      [ set  propensity-for-having-children  (propensity-for-having-children  -  number-of-
startpop-children) ]
      [ set propensity-for-having-children 0 ]
  ]]
  ask turtles with [mother != nobody][
    set adult? false
    set heading 0
    set age (age-of mother - 20 - random 21)
    ifelse (random-float 2 < 1)  ;set sex per coin-throwing    ;
    [ set breed female
      set-startpop-female-children-properties ]
    [ set breed male
      set-startpop-male-children-properties ]
  ]
  ask turtles with [age >= 20 and not adult?] [die]
  ask female with [adult?] [
    set child turtles-here with [mother = myself]
  ]
end; giving-birth-to-startpop-children

to set-startpop-female-children-properties
    set-startpop-children-unisex-properties
    set color orange
    set partner nobody
    set propensity-to-take-a-partner 0
    set considered-by-man? false
    set day-got-pregnant (day-of-birth - 1)
```

```
    set pregnant? false
end; set-startpop-female-children-properties


to set-startpop-male-children-properties
    set-startpop-children-unisex-properties
    set color magenta
    set propensity-to-search-for-partner 0
    set found-attractive-woman? false
end; set-startpop-male-children-properties


to set-startpop-adults-unisex-properties
;    set-all-new-agents-properties
    set negotiation-power random-float(1)
    set attractiveness random-float(1)
    set agedays0 (age * 365.25 + random(365))
    set day-of-birth (agedays0 * (-1))
    set satisfaction-in-current-partnership 0
    set agent-in-start-population? true
    set adult? true
    set mother nobody
    set child nobody
    set father nobody
    set partner nobody
    set day-found-recent-partner (day-of-birth - 1)
end; set-startpop-adults-unisex-properties


to set-startpop-children-unisex-properties
    set-all-new-agents-properties
    set agedays0 (age * 365.25 + random(365))
    set day-of-birth (agedays0 * (-1))
    set agent-in-start-population? true
end; set-startpop-children-unisex-properties


to set-all-new-agents-properties
    set day-of-birth days
    set adult? false
    set negotiation-power 0
    set attractiveness 0
    set propensity-for-having-children 0
    set satisfaction-in-current-partnership 0
    set agent-in-start-population? false
    set partner nobody
    set father nobody
    set child nobody
    set day-found-recent-partner (day-of-birth - 1)
end; set all-new-agents-properties


to set-new-female-agents-properties
  set-all-new-agents-properties
  set day-got-pregnant (day-of-birth - 1)
  set pregnant? false
  set age 0
end; set-new-female-agents-properties
to set-new-male-agents-properties
  set-all-new-agents-properties
  set age 0
end; set-new-male-agents-properties
;;~~~~~~~~~~~~~~~~~  end SETUP procedures  ~~~~~~~~~~~~~~~~~


;;~~~~~~~~~~~~~~~~ begin  GO   procedures  ~~~~~~~~~~~~~~~~~
to break-conditions
; progam stops when no male or female adults have suvived the last period
  if ((count male = 0) or (count female = 0)) [
    show "population died out!"
```

```
    show "survivors:"
    type "males    =>> "   show(count male)
    type "females =>> "   show(count female)
    set any-break-condition-met? true
  ]
end; break-conditions


to search-mates
  man-searchs-for-woman       ;; male moves around searching for a partner around his age
(accepted range of ages depending on female's attractiveness)
  initial-bargaining          ;; female accepts or rejects to enter negotiation due to males's
attractiveness
end; search-mates


to intra-partnership-negotiations
 dynamic-bargaining           ;; (day by day) renegotiations due to changes in environment =>
resource (time) reallocations
 crucial-events               ;; having children, entering/leaving labour force,
end; intra-partnership-negotiations


;; GO ----> begin: search-mates
;; GO ------> begin: man-searchs-for-woman
to man-searchs-for-woman                    ;; parts of code from "Partners Example" in NetLogo
Code Examples
  ask male with [partner = nobody and adult?] [without-interruption [
    do-dirmem-random-walk               ;; random walk with some memory of direction agent
had before
    if (any? male-here with [self != myself and adult?])
      [ go-to-male-empty-cell ]  ;; do not share a cell with another male
    if  ((any? female-here with [partner = nobody and adult?]) and
        (age-of random-one-of female-here > age-of self * (1 - (attractiveness-of random-one-
of female-here) / 2)) and
        (age-of random-one-of female-here < age-of self * (1 + (attractiveness-of random-one-
of female-here) / 3))
        )  ;; male accepts female that is down to half or up to 4/3 of his age depending on
females attractiveness
    [
      set partner random-one-of female-here with [partner = nobody and adult?]
      set color blue + 2
      set partner-of partner self
      set color-of partner red + 2
      set heading 180
    ]
  ]]
end; man-searchs-for-woman


to partner-search-via-public-blackboard
  locals [desire-for-late-partnership i j ]
  ;; females > slider(panic-age) & with probablity of (propensity-to-take-partner &
propensity-for-having-children & age) writes her "name" on pulic partnership-blackboard
  ;; males   > slider(panic-age) & with probablity of (propensity-to-search-for-partner &
propensity-for-having-children & age) select women from list (beginning from heading => FIFO-
algorithm
  ;; NAME: search-partner-blackboard
  ask female with [age >= PanicAge and partner = nobody] [
    set desire-for-late-partnership ((age - PanicAge + 1) / (100 - PanicAge) * propensity-to-
take-a-partner * propensity-for-having-children)
    if (desire-for-late-partnership >= avg-female-desire-for-late-partnership )
    [ set search-partner-blackboard lput who search-partner-blackboard ]
  ]
  ask male with [age >= PanicAge and partner = nobody] [
    set desire-for-late-partnership ((age - PanicAge + 1) / (100 - PanicAge) * propensity-to-
take-a-partner * propensity-for-having-children)
    if (desire-for-late-partnership >= avg-male-desire-for-late-partnership)[
```

```
      set i (first search-partner-blackboard)
      setxy (xcor-of turtle i) (ycor-of turtle i) - 1
      set heading 0
      set search-partner-blackboard remove i search-partner-blackboard
    ]
  ]
  update-partnership-blackboard

end; partner-search-via-public-blackboard


to update-partnership-blackboard
  locals [female-d4lp male-d4lp]
  ask turtles with [age >= PanicAge and partner = nobody][
    ifelse (breed = female)
    [ set female-d4lp lput ((age - PanicAge + 1) / (100 - PanicAge) * propensity-to-take-a-
partner * propensity-for-having-children) female-d4lp]
    [ set   male-d4lp lput ((age - PanicAge + 1) / (100 - PanicAge) * propensity-to-take-a-
partner * propensity-for-having-children) male-d4lp]
  ]
  ifelse (breed = female)
  [ set avg-female-desire-for-late-partnership (sum female-d4lp / length female-d4lp) ]
  [ set avg-male-desire-for-late-partnership (sum male-d4lp / length male-d4lp) ]

end; update-partnership-blackboard

;; GO ------> end: man-searchs-for-woman

;; GO ----> begin: initial-bargaining
to initial-bargaining     ;; woman accepts or rejects to enter partnership-negotiation due to
man's attractiveness
  ask female with [partner != nobody and adult?] [without-interruption [
    ifelse (attractiveness-of partner > attractiveness-of self * 0.7)[
        define-threat-points
        form-household
        set day-found-recent-partner days;; set date (day) partnership starts
        set day-found-recent-partner-of partner day-found-recent-partner
      ]
      [; male has to leave female (with considerable distance) - cancel all partner infos
      ask partner [without-interruption [
        set partner nobody
        set color blue - 2
        fd (random 3) + 2
        go-to-male-empty-cell
      ]]
      set partner-of partner nobody
      set partner nobody
      set color red - 2
      ]
  ]]
end; initial-bargaining

to define-threat-points
  ;; initial bargaining results, time allocation 'guideline', threat points
end; define-threat-points

to form-household  ;; setup joint household
  ask male with [partner != nobody] [without-interruption[
    if ((any? male-here != myself and adult?) or (any? female-here != partner-of myself and
adult?)) [
      ask male-here with [self != myself and adult?] [without-interruption[
        while [any? male-here with [adult?]] [
          fd random 2
          go-to-male-empty-cell
        ]
```

```
    ]]
    ask female-here with [adult? and self != partner-of myself] [without-interruption[
      while [any? female-here with [adult?]] [
        fd random 2
        go-to-female-empty-cell
      ]
      set heading 0
    ]]
  ]
]]
  set pcolor yellow
end; form-household
;; GO ----> end: initial-bargaining
;; GO --> end: search-mates

;; GO --> begin: intra-partnership-negotiations
to dynamic-bargaining
;; redefine threat points
end; dynamic-bargaining

to crucial-events ; crucial events within partnership
  ;; getting (an additional) child,
  ask female with [adult? and age < 40 and propensity-for-having-children > 0.1 and partner !=
nobody ][
    negotiate-for-another-child
    give-birth-to-another-child
  ]
  ;; entering/leaving labour force,
  ;; education,
end; crucial-events

;; GO ----> begin - crucial-events-procedures
to negotiate-for-another-child
  ; arguments: propensity for having children - both partners; weight by negotiation-power &
life phase & stability of partnership (history + satisfaction-in-current-partnership
  if ( propensity-for-having-children-of partner > 0.1 and not pregnant?
  and ((2 * (age - 19) / 20) * (propensity-for-having-children * negotiation-power +
propensity-for-having-children-of partner * negotiation-power-of partner) / (negotiation-power
+ negotiation-power-of partner)) > 1
  )
;  [ if ((days - day-found-recent-partner) + random 200 > 300 )
    [
      set day-got-pregnant (days + random 400)
      set pregnant? true
;    ]
  ]
end; negotiate-for-another-child

to give-birth-to-another-child
  if (pregnant? and days > day-got-pregnant + (39 * 7)) [
    hatch 1 [
      set mother myself
      set day-of-birth days
    ]
    ifelse (propensity-for-having-children > 1)
    [ set propensity-for-having-children (propensity-for-having-children - 1) ]
    [ set propensity-for-having-children 0 ]
    ask turtles-here with [mother = myself and day-of-birth = days][
      set adult? false
      set heading 0
      ifelse (random-float 2 < 1)  ;set sex per coin-throwing
      [ set breed female
        set-new-female-agents-properties ]
      [ set breed male
```

```
       set-new-male-agents-properties ]
     if (any? male-here with [adult?])
     [ set father random-one-of male-here with [adult?]]
    ]
    set day-got-pregnant (day-of-birth - 1)
    set pregnant? false
  ]
end; give-birth-to-another-child
;; GO ----> end - crucial-events-procedures
;; GO --> end: intra-partnership-negotiations


;; GO --> begin: time dependent events
to time-dependent-events
    set time time + 1
    set days floor (time)                           ;; days are some linear function of time
(depending on setup size, ram, os, etc.)
    if (years < floor (days / 365.25)) [
      set years floor (days / 365.25)
      calc-age-distribution
      update-plot-age-distribution-women
      update-plot-age-distribution-men
      calc-completed-fertility
      update-plot-completed-fertility
    ]
    set days-in-current-year floor (days - years * 365.25)
    if (UpdateInterval <= 0) [  set UpdateInterval  1  ]
    if ((days mod (UpdateInterval)) = 0)
    [
      calc-partnerships-percent
      calc-fertile-partnerships-percent
      calc-pregnangcies-percent
      update-plot-realized-partnerships
      update-plot-fertile-partnerships
      ask turtles with [age > 100] [ die ]
      transition-to-adulthood
  ;    partner-search-via-public-blackboard
    ]
    ask turtles [without-interruption[
      ifelse agent-in-start-population?
        [set agedays (agedays0 + days)]
        [set agedays (days - day-of-birth)]
      if (age  < floor (agedays / 365.25))
        [ set age floor (agedays / 365.25) ]
    ]]
end; time-dependent-events
;; GO --> end: time dependent events


;;~~~~~~~~~~~~~~~~~ end   GO   procedures  ~~~~~~~~~~~~~~~~

;;=*=*=*=*=*=*=*begin  PLOT  procedures  =*=*=*=*=*=*=*=*
to setup-plots
  setup-plot-age-distribution-women
  setup-plot-age-distribution-men
  setup-plot-completed-fertility
  setup-plot-realized-partnerships
  setup-plot-fertile-partnerships
  update-plot-age-distribution-women
  update-plot-age-distribution-men
  update-plot-realized-partnerships
  update-plot-fertile-partnerships
  calc-completed-fertility
  update-plot-completed-fertility
end; setup-plots
```

```
to setup-plot-age-distribution-women
  set-current-plot "Age Distribution Women"
  set-plot-y-range 0 2
  set-plot-x-range 0 100
end; setup-plot-age-distribution-women
to setup-plot-age-distribution-men
  set-current-plot "Age Distribution Men"
  set-plot-y-range 0 2
  set-plot-x-range 0 100
end; setup-plot-age-distribution-Men
to setup-plot-completed-fertility
  set-current-plot "Completed Fertility"
  set-plot-y-range 0 2
  set-plot-x-range 25 40
end; setup-plot-completed-fertility
to setup-plot-realized-partnerships
  set-current-plot "Realized Partnerships"
  set-plot-y-range 0 100
  set-plot-x-range 0 (UpdateInterval * .1 + 10)
end; setup-plot-realized-partnerships
to setup-plot-fertile-partnerships
  set-current-plot "Fertility"
  set-plot-y-range 0 100
  set-plot-x-range 0 (UpdateInterval * .1 + 10)
end; setup-plot-fertile-partnerships

to update-plot-age-distribution-women
  set-current-plot "Age Distribution Women"
  histogram-from (female) [age]
end; update-plot-age-distribution-women
to update-plot-age-distribution-men
  set-current-plot "Age Distribution Men"
  histogram-from (male) [age]
end; update-plot-age-distribution-men

to update-plot-realized-partnerships
  set-current-plot "Realized Partnerships"
  plot partnerships-pct
end; update-plot-realized-partnerships
to update-plot-fertile-partnerships
  set-current-plot "Fertility"
  set-current-plot-pen "FertilePartnerships"
  plot fertile-partnerships-pct
  set-current-plot-pen "RecentPregnancies"
  plot pregnancies-pct
end; update-plot-fertile-partnerships

to update-plot-completed-fertility
  set-current-plot "Completed Fertility"
  histogram-list completed-fertility
end;;update-plot-completed-fertility
;;=*=*=*=*=*=*=*  end  PLOT  procedures  =*=*=*=*=*=*=*

;;=*=*=*=*=*=*=*begin BASIC procedures  =*=*=*=*=*=*=*=
to calc-partnerships-percent
  ifelse (count male with [adult?] > count female with [adult?])
  [ set partnerships-pct (100 * (count female with [partner != nobody and adult?]) / count
female with [adult?]) ]
  [ set partnerships-pct (100 * (count male with [partner != nobody and adult?]) / count male
with [adult?]) ]
end; calc-partnerships-percent

to calc-fertile-partnerships-percent
  ifelse (count male with [adult? and age < 60] > count female with [adult? and age < 40])
```

```
  [ set fertile-partnerships-pct (100 * (count female with [partner != nobody and adult? and
age < 40]) / count female with [adult? and age < 40]) ]
  [ set fertile-partnerships-pct (100 * (count male with [partner != nobody and adult? and age
< 60]) / count male with [adult? and age < 60]) ]
end; calc-fertile-partnerships-percent

to calc-pregnangcies-percent
  ifelse (count female with [partner != nobody and adult? and age < 40] > 0)
  [ set pregnancies-pct (100 * count female with [pregnant?] / count female with [partner !=
nobody and adult? and age < 40]) ]
  [ set pregnancies-pct 0 ]
end; calc-pregnangcies-percent

to calc-completed-fertility
  locals [rf40 rf35 rf30 rf25 c40 c35 c30 c25 cf40 cf35 cf30 cf25 rf40-avg rf35-avg rf30-avg
rf25-avg c40-avg c35-avg c30-avg c25-avg cf40-avg] ;rest-fertility & number of  children at
agerange (X to X+5)
  ;; just females (as fertility period for males is quite fuzzy)
  ifelse (count female with [age >= 40 and age < 45] > 1)
  [ set rf40 sum values-from female with [age >= 40 and age < 45] [propensity-for-having-
children]
    set c40 sum values-from female with [age >= 40 and age < 45] [child]
    set rf40-avg (rf40 / count female with [age >= 40 and age < 45])
    set c40-avg (c40 / count female with [age >= 40 and age < 45])
    ifelse (c40 + rf40 > 0)
    [ set cf40 100 * (1 - (rf40 / (c40 + rf40))) ]
    [ set cf40 0 ]
  ]
  [ set cf40 0 ]
  ifelse (count female with [age >= 35 and age < 40] > 1)
  [ set rf35 sum values-from female with [age >= 35 and age < 40] [propensity-for-having-
children]
    set c35 sum values-from female with [age >= 35 and age < 40] [child]
    set rf35-avg (rf35 / count female with [age >= 35 and age < 40])
    set c35-avg (c35 / count female with [age >= 35 and age < 40])
    ifelse (c35 + rf35 > 0)
    [ set cf35 100 * (1 - (rf35 / (c35 + rf35))) ]
    [ set cf35 0 ]
  ]
  [ set cf35 0 ]
  ifelse (count female with [age >= 30 and age < 35] > 1)
  [ set rf30 sum values-from female with [age >= 30 and age < 35] [propensity-for-having-
children]
    set c30 sum values-from female with [age >= 30 and age < 35] [child]
    set rf30-avg (rf30 / count female with [age >= 30 and age < 55])
    set c30-avg (c30 / count female with [age >= 30 and age < 55])
    ifelse (c30 + rf30 > 0)
    [ set cf30 100 * (1 - (rf30 / (c30 + rf30))) ]
    [ set cf30 0 ]
  ]
  [ set cf30 0 ]
  ifelse (count female with [age >= 25 and age < 30] > 1)
  [ set rf25 sum values-from female with [age >= 25 and age < 30] [propensity-for-having-
children]
    set c25 sum values-from female with [age >= 25 and age < 30] [child]
    set rf25-avg (rf25 / count female with [age >= 25 and age < 30])
    set c25-avg (c25 / count female with [age >= 25 and age < 30])
    ifelse (c25 + rf25 > 0)
    [ set cf25 100 * (1 - (rf25 / (c25 + rf25))) ]
    [ set cf25 0 ]
  ]
  [ set cf25 0 ]
  set  completed-fertility (list cf25 cf30 cf35 cf40)
  set  rest-fertility (list rf25 rf30 rf35 rf40)
```

```
  set  realized-fertility (list c25 c30 c35 c40)
  set  rest-fertility-avg (list rf25-avg rf30-avg rf35-avg rf40-avg)
  set  realized-fertility-avg (list c25-avg c30-avg c35-avg c40-avg)
end; calc-completed-fertility

to do-dirmem-random-walk              ;; random walk with some memory of direction agent had
before
  rt (random 20 - random 20)
  fd (propensity-to-search-for-partner)
end;

to transition-to-adulthood
  ask turtles with [age > 19 and not adult?] [without-interruption[
    set negotiation-power random-float(1)
    set attractiveness random-float(1)
    set adult? true
    ifelse (breed = male)
    [
      set color blue
      set propensity-to-search-for-partner random-float (1)
      set  propensity-for-having-children  (random-float(2)  +  random-float(1)  +  random-
float(0.5))
      set found-attractive-woman? false
      go-to-male-empty-cell
    ]
    [
      set color red
      set propensity-to-take-a-partner random-float(1)
      set propensity-for-having-children (random-float(2) + random-float(1) + random-float(1)
+ random-float(0.5))
      set considered-by-man? false
      go-to-female-empty-cell
    ]
  ]]
end; transition-to-adulthood

to go-to-female-empty-cell
  rt random-float 360
  fd random-float 10
  if any? other-female-here
    [ go-to-female-empty-cell ]          ;; keep going until we find an unoccupied patch
  setxy pxcor pycor  ;; move to center of patch
end; go-to-female-empty-cell

to go-to-male-empty-cell
  rt random-float 360
  fd random-float 10
  if any? other-male-here
    [ go-to-male-empty-cell ]          ;; keep going until we find an unoccupied patch
  setxy pxcor pycor  ;; move to center of patch
end; go-to-male-empty-cell

;;=*=*=*=*=*=*=*  end BASIC procedures  =*=*=*=*=*=*=*=*=

;;=*=*=*=*=*=*=*begin CONTAINER =*=*=*=*=*=*=*=*=
to OLD_go-to-female-empy-cell
  locals [female-empty-patches target no-female-empty-patch?]
  set no-female-empty-patch? true
  while [no-female-empty-patch?] [without-interruption [
    set female-empty-patches neighbors with [not any? female-here with [adult?]]
    ifelse (any? female-empty-patches)
    [
      set target random-one-of female-empty-patches
      set heading towards target
```

```
      fd distance target
      set no-female-empty-patch? false
    ]
    [
      set heading random 360
      jump 3
      if (not any? female-here with [adult?]) [set no-female-empty-patch? false]
    ]
  ]]
end; OLD_go-to-female-empty-cell


to OLD_go-to-male-empty-cell
  locals [male-empty-patches target no-male-empty-patch?]
  set no-male-empty-patch? true
  while [no-male-empty-patch?] [without-interruption [
    set male-empty-patches neighbors with [not any? male-here with [adult?]]
    ifelse (any? male-empty-patches)
    [
      set target random-one-of male-empty-patches
      set heading towards target
      fd distance target
      set no-male-empty-patch? false
    ]
    [
      set heading random 360
      jump 3
      if (not any? male-here with [adult?]) [set no-male-empty-patch? false]
    ]
  ]]
end; OLD-go-to-male-empty-cell


to calc-age-distribution
;this procedure sums up the number of agents within specific age ranges  =>  2 vectors of age
cohorts per sex
  locals [ i ]
  set i 0
  set m-agelist (list)
  set f-agelist (list)
  while [i < 100] [
        set m-agelist lput (count male with [age = i]) m-agelist
        set f-agelist lput (count female with [age = i]) f-agelist
        set i (i + 1)
  ]
end; calc-age-distribution
;;=*=*=*=*=*=*=*=*  end CONTAINER  =*=*=*=*=*=*=*=*=*=
```

Zuletzt erschienene *WORKING PAPERS*

- Martin Spielauer, Franz Schwarz, Kurt Schmid: **Education and the Importance of the First Educational Choice in the Context of the FAMSIM+ Family Microsimulation Model for Austria.** Nr. 15/2002

- Coomaren P. Vencatasawmy: **Modelling Fertility in a Life Course Context: Some Issues.** Nr. 16/2002

- Norbert Neuwirth: **Labor Supply of the Family – an Optimizing Behavior Approach to Microsimulation.** Nr. 17/2002

- Martin Spielauer: **The Potential of Dynamic Microsimulation in Family Studies: A Review and Some Lessons for FAMSIM+.** Nr. 18/2002

- Sabine Buchebner-Ferstl: **Die Partnerschaft als Ressource bei kritischen Lebensereignissen am Beispiel der Pensionierung.** Nr. 19/2002

- Sonja Dörfler, Karin Städtner: **European Family Policy Database – Draft Manual.** Nr. 20/2002

- Johannes Pflegerl: **Family and Migration. Research Developments in Europe: A General Overview.** Nr. 21/2002

- Sonja Dörfler: **Familienpolitische Maßnahmen zum Leistungsausgleich für Kinderbetreuung – ein Europavergleich.** Nr. 22/2002

- Franz Schwarz, Martin Spielauer, Karin Städtner: **Gender, Regional and Social Differences at the Transition from Lower to Upper Secondary Education.** An Analysis in the Context of the FAMSIM+ Family Microsimulation Model for Austria. Nr. 23/2002

- Veronika Pfeiffer-Gössweiner, Johannes Pflegerl: **Migration in the European Union: An Overview of EU Documents and Organisations Focusing on Migration.** Nr. 24/2002/E

- Karin Städtner: **Arbeitsmarktrelevante Konsequenzen der Inanspruchnahme von Elternkarenz.** Nr. 25/2002

- Franz Schwarz, Martin Spielauer: **The Composition of Couples According to Education and Age. An Analysis in the Context of the FAMSIM+ Family Microsimulation Model for Austria.** Nr. 26/2002

- Franz Schwarz, Martin Spielauer, Karin Städtner: **University Education. An Analysis in the Context of the FAMSIM+ Family Microsimulation Model for Austria.** Nr. 27/2002

- Sabine Buchebner-Ferstl: **Partnerverlust durch Tod. Eine Analyse der Situation nach der Verwitwung mit besonderer Berücksichtigung von Geschlechtsunterschieden.** Nr. 28/2002

- Karin Städtner, Martin Spielauer: **The Influence of Education on Quantum, Timing and Spacing of Births in Austria.** Nr. 29/2002

- Sonja Dörfler: **Familienpolitische Leistungen in ausgewählten europäischen Staaten außerhalb der Europäischen Union.** Nr. 30/2002

- Sonja Dörfler: **Nutzung und Auswirkungen von Arbeitsarrangements zur besseren Vereinbarkeit von Familie und Erwerb.** Nr. 31/2003

- Franz Schwarz: **Sozio-ökonomische Ungleichheiten im Gesundheitsverhalten in Österreich / Socioeconomic Inequalities in Health Behavior in Austria.** Nr. 32/2003

- Karin Städtner: **Female Employment Patterns around First Childbirth in Austria.** Nr. 33/2003

- Olaf Kapella: **Stahlhart – Männer und erektile Dysfunktion.** Nr. 34/2003

- Olaf Kapella, Christiane Rille-Pfeiffer: **Über den Wunsch, ein Kind zu bekommen. Kinderwunsch hetero- und homosexueller Paare.** Nr. 35/2004

- Sonja Dörfler: **Die Wirksamkeit von Arbeitsbedingungen zur besseren Vereinbarkeit von Familie und Erwerb. Teil 1: Arbeitszeit und Arbeitsort.** Nr. 36/2004

- Josef Meichenitsch: **Kinder + Studium = Gesundheitsvorsorge? Eine empirische Analyse des primären Gesundheitsverhaltens in Österreich.** Nr. 37/2004

- Sonja Dörfler, Josef Meichenitsch: **Das Recht auf Teilzeit für Eltern. Top oder Flop?** Nr. 38/2004

- Guido Heineck: **Religion, Attitudes towards Working Mothers and Wives' Full-time Employment. Evidence for Austria, Germany, Italy, the UK, and the USA.** Nr. 39/2004

- Guido Heineck, Astrid Haider und Norbert Neuwirth: **Determinanten abhängiger Selbstständigkeit in Österreich.** Nr. 40/2004

- Sabine Buchebner-Ferstl: **Das Paar beim Übergang in den Ruhestand. Ergebnisse einer qualitativen Studie**. Nr. 41/2004

- Astrid Haider, Guido Heineck und Norbert Neuwirth: **Zwischen Reproduktions- und Erwerbsarbeit: Der Zusammenhang von Kinderbetreuung, Pflege und Frauenerwerbstätigkeit**. Nr. 42/2004

- Sonja Dörfler: **Außerfamiliale Kinderbetreuung in Österreich - Status Quo und Bedarf**. Nr. 43/2004

Alle zu beziehen bei: Österreichisches Institut für Familienforschung (ÖIF)
A-1010 Wien, Gonzagagasse 19/8
Tel: +43-1-5351454-19, Fax: +43-1-535 14 55
E-Mail: edeltraud.puerk@oif.ac.at