Swidden Farming Version 2.0

By C. Michael Barton

MODEL OVERVIEW

Purpose

This model simulates some of the dynamics of house-hold level swidden agriculture (also called shifting cultivation or slash-and-burn agriculture). The model can run in controlled or adaptive mode. In controlled mode, the user sets values related to farming decisions. In adaptive mode, these values are set by the agents.

The model is designed to explore some of the factors affecting swidden (sometimes called slash-and-burn) agriculture. Agricultural households take control of the land around them and rotate agricultural fields within this area. Field fertility decreases if a patch is used, and the patch or patches with the highest potential net return is chosen to farm during each time step. The model explores the importance of soil fertility upon swidden strategies as well as issues of land ownership. In addition, the model also explores the effects of swidden agriculture on vegetation communities.

In controlled mode, the researcher sets all parameter values. In adaptive mode, the model explores the success (or failure) of strategies created randomly at the model's initialization, and during agent reproduction. The agricultural strategy of the agents results from the combination of six key values (move_dist, move_threshold, fission_energy, swidden_radius, min_fertility, & n_patches). These values are all initialized with random values from within specified ranges. If an agent reproduces, a copy of the agent is created; however, any of the six key values may randomly change. The resulting selection will produce a set of agents with decision rule values that out compete all other agents.

State Variables and Scales

The fundamental variable for this simulation is the Household agent. All households are initialized with the same values for each variable.

energy	The amount of energy each agent has. If it reaches 0 the agent is removed from the simulation. This is initialized as *init-energy* = 100
move-threshold	If a household's energy falls below this value, then the household will move. In controlled mode, the user sets *init-move-threshold* (0-100), which is converted into *move-rate* (which varies from 0-1). In adaptive mode, *move-rate* is a random float between 0 and 1 and can vary by household agent. The *move_threshold* = *move-rate* * *init-energy*.
net-return	The net amount of energy received by a household as a result of farming a patch after each step in the simulation. It is the harvest value, weighted by fertility, minus the farming cost, vegetation-clearing cost. *Net_return* = ((*harvest-rate* * *init-energy*) * *patch*

fertility / *divisor*) - (*farm-cost* * *init-energy*) - *veg_clear_cost* - ((*distance from patch to farmstead*) / 5).

fission-rate	The amount of energy a household needs in order to create a new household. Given as a proportion of the initial household energy. Thus, a value of *fission-rate* = 2.0 means that a new household will be created when the household's energy is 200% of the amount of energy the household was initialized with. In controlled mode, this is calculated as *fission*-energy (set by the user) / 100. In adaptive mode, *fission-rate* can vary randomly by household agent from 1-2.
farm-dist	The distance (in patches) that an agent will travel to farm and the radius of the area over which a farmstead will claim ownership, if the patches are not already owned by another farmstead. This is set by the user in controlled mode and can vary randomly by household agent from 1-20 in adaptive mode.
min-fertility	The minimum value that a patch must have for an agent to consider taking possession of a patch for future farming. This is fixed at 0.8 for controlled mode and can vary by household agent from 0-1 in adaptive mode.
movedist	The distance an agent will move if it must find a new farmstead location. This value may vary randomly from agent to agent. The specific value ranges from 1 - 5.

Patches are the other important agent type within this simulation. They represent patches of land on which the households live and farm.

vegetation	Numerical value (0-50) that represents the vegetation community on a patch. A value >= 40 represents forest, 40 >value >=20 represents shrub, 20>value>0 herbaceous, and value of 0 represents a bare patch.
fertility	This value (0.0-1.0) represents the agricultural productivity of a patch. It is lowered by *fertility-loss-rate* each time a patch is farmed. Households cannot farm when the fertility of a patch is 0. *fertility-loss-rate* is calculated from *fertility-loss* (set by the user between 0-100) / 100.
farmstead	Records the presence (1) or absence (0) of a household agent on a patch. A farmstead may not move to a patch already occupied by a farmstead. Also, the patch is owned by the agent but cannot be farmed.
state	Records whether a farmstead is "active" or "abandoned".
field	Records if a patch is under cultivation (1 or 0).

owner	Records which agent owns a given patch (=agent ID). Patches need not be under cultivation to be owned.
fallow	Tracks the number of steps a patch has not been cultivated.
veg-clear-cost	The cost of clearing vegetation from a patch for the purpose of farming. As long as the *vegetation* value and *clearing-rate* of a patch is > 0, the *veg-clear-cost* = ((*init-energy* * *clearing-rate*) / *vegetation*) It is updated for all patches at each step.
site	Whether or not a patch is an archaeological site ("True"/"False")

The global variables for this simulation are described below. Some of the respective values of these variables may be user controlled.

init-households	The initial number of households at the start of the simulation. Set by the user.
init-energy	The initial amount of energy for each household at the start of the simulation. Set to 100 in the code.
harvest	Gross energy return from a farmed patch. Given as a percentage of the initial household energy and converted into *harvest-rate* (0-1) for calculations in the simulation. This partially determines the amount of return a household will receive from farming a patch. See *net-return* above. Set by the user.
farm-cost	Amount of energy needed to cultivate a patch. Given as a percentage of the initial household energy and converted to *farm-rate* (0-1) for calculations in the simulation. See n*et-return above*. Set by the user.
tree-clearing-cost	Energy cost of clearing the maximum vegetation cover from a patch (i.e., trees) in order to farm it. Given as a percentage of the initial household energy and converted to *clearing-rate* (0-1) for calculations in the simulation. Clearing other vegetation is scaled between this value and 0. Set by the user.
move-cost	The energy cost incurred for moving a distance of 1 patch. Given as a percentage of the initial household energy and converted to *move-cost-rate* (0-1) for calculations in the simulation. Set by the user.
init-move-threshold	For controlled mode only, if a household's energy falls below this value, then the household will move. In controlled mode, the user sets *init-move-threshold* (0-100), which is converted into *move-rate* (which varies from 0-1).

fertility-loss	The amount of fertility that is lost from a patch each cycle if it is farmed. The value does not go below 0. Set by the user from 0-100% and converted to *fertility-loss-rate* (0-1) for use in calculations in the simulation.
fertility-restore	The amount of fertility that is restored to each patch after a step of the program if it is not farmed (i.e., it is left fallow). Set by the user (0-100) and converted to *restore-rate* (0-1) for use by the simulation.
bad-years	The approximate percentage of cycles, chosen randomly, in which the harvest is half the normal amount. Set by the user.
max-fallow	Used in conjunction with the limit land tenure option. The maximum number of cycles a patch can remain unfarmed (i.e., fallow) before it is released as unowned. Set by the user.
divisor	Variable used in the calculation of returns from farms. If the year is a bad year, then *divisor* = 2. If it is not a bad year, *divisor* =1.
innovation-rate	This variable is used to define the rate at which an agent household will innovate, or randomly choose new values for some of its properties. It ranges from 1-500 and is set by the user at the inception of the program.

Process Overview and Scheduling
Essentially the simulation processes revolve around the creation of farmsteads and farm plots by each agent. The specific processes evoked and their scheduling is as follows: check to see if an agent should move, choice of a patch to farm, farming of a patch, consumption of farm returns, possible reproduction, and possible death. Each household agent is chosen at random and the subsequent processes are called. First, the household checks to see if it needs to move to a new location based on their energy. Next agents choose the best patch in their swidden radius on which to cultivate. The net energy from the cultivated patches is calculated and added to each agent's total energy and a basic living cost is subtracted from the household energy. Following this, the possibility of agent reproduction and the possibility of agent death are evaluated based on the agent's energy. The simulation continues until all agents are removed from the simulation or until the user stops it.

DESIGN CONCEPTS

Emergence (a summary of emergent phenomena from the interaction of the agents)
Agent behavior is intertwined with the values of six adaptive characteristics. These characteristics are as follows: the preferred farming distance (1-20), the maximum distance an agent will look in order to establish a new household (farming distance * 1-5), the amount of energy required to fission and produce a daughter household (1-200), and the minimum fertility within the swidden radius around a new patch that is acceptable in order to establish a new farmstead and to cultivate a patch (0.1-1.0). All daughter households will have the same values as the parent household EXCEPT if they innovate when during fissioning (i.e., "hatching" a new agent). The user sets the chance of innovation. If an innovation occurs, a new random value for the particular characteristic

is chosen (within the aforementioned ranges). As the simulation progresses, agents with successful adaptive strategies fission more frequently. The specific strategies that emerge as successful can be highly variable.

In addition, the model creates interesting site distribution patterns. The specific site distribution and the regularity of the distribution, is the result of multiple parameters and agent actions within those parameters. For example, simulation runs without land tenure, essentially contain territories, which create more regular settlement patterns than those with land tenure.

Adaptation (how the agents adapt their behavior to their and their environments current state)
The agents adapt to their environments based on the productivity of the land around them. They attempt to farm the patches nearby their farmstead, if they are unable to do so or the returns from farming are too low, they move to a new location. Their ability to move makes their choice to move their farmstead adaptive. A process of adaptation occurs as agents with more productive strategies propagate more frequently. However, the agents themselves do not direct their adaptation. This process is affected by random innovation (see Reproduction for more information).

Fitness/Objectives (a summary of the agents' goals)
The goal of agents is to collect energy by farming nearby patches. They attempt to maximize the amount of energy they collect each step. Ultimately, their goal is to obtain enough energy to fission and avoid death.

Prediction (how the agents predict the consequences of their decisions)
The agents do not predict the consequences of their decisions.

Sensing (environmental variables perceived by the agents, which might include their own variables)
Agents or households are able to detect a variety of environmental variables. They are able to detect if they or another agent owns a patch of land. They are able to detect the fertility of a given patch as well as the amount of vegetation on an individual patch. These variables are crucial in the decision of an agent about which patch(s) to farm.

Interactions
The interaction of agents is not specifically modeled. Agents interact with their environment by placing farms on nearby patches. Thus agents affect each other indirectly through the landscape.

Stochasticity
Stochasticity is achieved in part through environmental variability. A temporally variable environment can be simulated. The user can set a *bad-years* value. This represents the approximate percentage of total cycles that there are poor harvests. Currently, a poor harvest is half its normal value. A random function selects cycles as a 'bad year'. All households are affected equally by bad years, although this could easily be changed to randomly select individual household to have a poor harvest in any particular cycle. The distance moved by an agent also has a random component.

Stochasticity is also found in the variability within agents' properties. Six agent characteristics vary randomly at the initialization of the model. As the model progresses, these agents reproduce new copies of themselves. Each newly created agent has a random chance of innovation. Innovation causes the agent to randomly assign a new value to one of the characteristics inherited from its parent.

Collectives (whether the agents are grouped socially)
The agents are not grouped socially.

Observation (how data are gathered from the model).
The total numbers of agents or households are recorded. In addition, the average value for each of the adaptive rules is recorded and displayed. The percent of each vegetation type represented on the grid is also observed. Data may be gathered by the BehaviorSpace tool included in Netlogo and may viewed on plots updated after each step of the simulation.

DETAILS

Initialization
A number of households selected by the user are placed randomly on a landscape. All households begin with 100 eu (energy units). Initially, all patches are covered with forest (vegetation = 50) and have a maximum fertility value (= 1.0). All households use a minimum amount of energy to continue to live, even if they don't farm (currently set to 1% of the initial energy per model cycle [tick]). Adaptive variables are set randomly for each initial household agent within their accepted ranges

Following the initialization of the adaptive variables, the patch on which the agent is located becomes a farmstead and its fertility is reduced to 0. Next the agent takes possession of all land within a *farm*-dist from the farmstead. Other households cannot farm this land while it is owned.

Input
No Input data are necessary for the model.

Submodels

Movement Check
If the household's energy is below the *move-threshold*, then the household will look for a new location to move to. If the household needs to move, ((*farm-dist* * *move-dist*) + 1) is chosen as a maximum search radius. Of the patches in this maximum search distance, only patches with *fertility* >= the agent's *min-fertility,* no farmstead and unowned are recorded as potential move locations. One of these available patches is chosen at random.

At this point the patch values are changed to reflect the moving farmstead. The old farmstead location is removed and ownership of the nearby patches is relinquished and all old patches owned have *fallow* reset to 0. Next a new farmstead is put on the empty patch and the agent takes ownership of all unowned patches within the *farm-dist* of the new farmstead and *fallow* is set to 0 for all of these patches. A cost is incurred for movement and (*init-energy* * *move-cost-rate*) - ((distance from the old to the new farmstead) / 5) is subtracted from the agent's energy. *Move-threshold* is reduced by ((*init-energy* * *move-cost-rate*) + 1). The purpose of this is to reset the *move-threshhold* to be lower than the current energy level so that a household doesn't keep moving after it first moves.

Choose Land
The household attempts to farm one patch that it owns, or which is unowned and within the *farm-dist* radius. Agents may not place farms on land that is owned by another agent or the patch on which they are located (their farmstead identified by their red color and house icon). When a

household moves or dies, it relinquishes ownership of its patches. An abandoned farmstead turns violet in color; a farmstead where the household dies turns magenta.

First, households examine the patches within their *farm-dist* that are theirs or are unowned and have a *fertility* that is greater than or equal to the agent's *min-fertility*. From these patches, each household selects the patches to cultivate which have the highest potential *net-return*, as described above. In other words, the household chooses the patch which will maximize returns and minimize costs. Patches chosen have a field placed on them and their ownership is set to the current agent if the patch was previously unowned.

Farm and Consume Resources

The patches which were chosen are now farmed. The patches chosen for farming have their *vegetation* and *fallow* count set to 0. The *net-return* for each patch is set as described above. The fertility of the patch is then adjusted to reflect the farming which has occurred on the patch. The fertility of the patch is reduced by *fertility-loss-rate.* The fertility of the patch cannot go below 0.

Once the patches are updated, the total return from the farms is calculated. The sum of the *veg_clear_cost* and the sum of the *farm_dist* for each farmed patch are calculated. The total amount of energy received by a household after each step in the simulation is the *net_return*. The net_return is calculated as the (sum of *return* for each patch being farmed) – (the sum of the *veg_clear_cost* for the patches) – (the sum of the farm_dist for the patches) – (the actual number of patches farmed* *farm_cost* * *init_energy* / 100). The *net_return* value is added to the energy of the farmstead.

Reproduce

After a farm has been farmed, the possibility of reproduction is evaluated. Households may reproduce if the current amount of energy a household has is greater than (*init-energy* * *fission-rate*). If so, then the household agent reproduces a copy of itself. The new household agent moves to a new location if it can (see 'Choose Land' above). If the new agent cannot move (i.e., cannot find an unowned patch with fertile soil), it stays in the parent household patch until the next cycle. This is the only case in which more than one household can occupy the same patch.

Both households are given half of the original amount of energy. The newly created agent may not be an identical clone of its parent. The new household agent may innovate and change one of the parent household's characteristics. A random number is chosen from 0-1000, if it is less than or equal to *innovation-rate*, then the characteristic is randomly changed.

Check Death

Farmsteads are removed from the simulation if their energy drops below 1. Land that the agent owned is released so new households may take ownership of the land and the removed agent's farmstead is also removed.

Regrow Patches

The purpose of this process is to update the patches after agents have cultivated their farms. If patch fertility is less than 1, the *fertility* of a patch is increased by *restore-rate*. *Fertility* cannot exceed 1. For patches that were not cultivated (*field* = 0) and do not have a farmstead (*farmstead* = 0) and are not a site (*site*=False), then the following statement is evaluated for vegetation. If *vegetation* is less than 50, the value is increased by 1. *Vegetation* is not allowed to exceed 50.

Vegetation increments by a value of 1 each cycle of the program until it reaches 50 (forest). Clearing for farms decreases this value.

Next, fields that are owned by an agent and were not under cultivation (*field* = 0) have their *fallow* value incremented by 1. Fallow records the number of steps that a field has remained fallow. All patches are then marked as not farmed (*field*=0) and the Land Tenure process is evaluated if it has been switched on (see Land Tenure process below).

Finally, the *veg-clear-cost* is updated for each patch. If both the *tree_clearing_cost* and *vegetation* are greater than 0, the *veg-clear-cost* = ((init_energy * tree_clearing_cost / 100) / vegetation). If *vegetation* is equal to 0, then the *veg-clear-cost* = 0. The *veg-clear-cost* cannot be less than 0.

Land Tenure

If the t*ransfer-ownership* switch is on, households have limited land tenure. If land has lain fallow for more than the number of years set by the user in the fallow slider, it reverts to unclaimed and any agent (including the original owner) can claim it in the next cycle. If a patch has been fallow for longer than *max-fallow* and is not a farmstead, then the farming household no longer owns the patch.

Environmental Variability

The user selects the approximate percentage of *bad-years*. A random number is chosen and if the value is <= *bad_years* the yield of fields are reduced by half for all agents. If the year is chosen to be a bad year, *divisor* = 2. If it is not, then *divisor* = 1.

CREDITS AND REFERENCES

Associated publication:

Barton, C. M. (2013). Complexity, social complexity, and modeling. Journal of Archaeological Method and Theory.