OVERVIEW

Purpose

Swidden Farming is designed to explore the dynamics of agricultural land management strategies. Specifically, 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 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.

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.
move_threshold	If a household's energy falls below this value, then the household will move. Move_threshold = init_energy * init_move_threshold / 100.
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* * *init_energy* / 100) * *patch fertility* / *divisor*) - (*farm_cost* * *init_energy* / 100) - *veg_clear_cost* - ((*distance from patch to farmstead*) / 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 each time a patch is farmed. Households cannot farm when the fertility of a patch is 0.
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.
field	Records if the 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	Value represents the energy cost of clearing vegetation from a patch in order to farm it. It is scaled between tree_clearing_cost and 0, and scaled according to the vegetation value. *Veg_clear_cost* = (*init_energy* * *tree_clearing_cost* / 100) / *vegetation*

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.
fission_energy	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 200% in fission energy 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. Set by the user.
swidden_radius	The radius (in patches) of the area over which a farmstead will claim ownership, if the patches are not already owned by another farmstead. Set by the user.
harvest	Gross energy return from a farmed patch. Given as a percentage of the initial household energy. 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. See net_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. Clearing other vegetation is scaled between this value and 0. Set by the user.
init_move_threshold	If household energy drops below this value, the household will try to move to a new location. Given as a percentage of the initial household energy. 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. Set by the user.
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.
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.
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.

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 establish a farm. The net energy from that farm 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) The most interesting phenomena that emerges as a result of the model is the site distribution pattern. 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, they move to a new location. Their ability to move makes their choice to move their farmstead adaptive.

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 reproduce and prevent 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 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 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.

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 return from farms and the average amount of household energy is recorded at each time step. Furthermore, 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

The model is largely initialized with values specified by the user. First, patches are initialized uniformly, with a vegetation set to 50, no fields, fallow = 0, fertility=1, and no ownership (until agents take ownership). Each grid cell corresponds to a patch. Households are initialized to a random location on a 60x60 grid and a farmstead is set in that location. All households begin with 100 eu (energy units). Next the agent takes possession of all land within a swidden_radius from the farmstead. Other households cannot farm this land while it is owned. Each agent contains identical values corresponding to those set by the user.

Input

No Input data are necessary for the model.

Submodels

<u>Movement Check</u>

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, ((a random number from 1 – 5) **swidden_radius* *2) is chosen as a maximum search distance. Then, if a patch can be found that is not owned, not a farmstead, and with a fertility higher than .8, the agent moves to that patch. If a patch cannot be found, the agent does not move and tries again to move in the next cycle.

<u>Choose Land</u>

The household attempts to farm one patch that it owns, or which is unowned that is within the swidden radius. Households take ownership of the patches within their *swidden_radius* that are unowned and have a *fertility* that is greater than 0. Agents may not place farms on land that is owned by another agent or the patch on which they are located (their farmstead). The patch on which they are located is the location of their dwelling. Farmsteads are identified by their red color. 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. In order for a patch to be farmed, the patch must have a fertility value greater than 0. Each household selects a patch to cultivate each cycle from among owned patches that have the highest potential *net_return*, which is calculated as the harvest value weighted by soil fertility, minus various costs to farm the patch. In other words, the household chooses the patch which will maximize ((*fertility* * *harvest* * *init_energy* / 100) - (*farm_cost* * *init_energy* / 100) - *veg_clear_cost* - ((distance of the patch to the farmstead) / 5). If there are two locations that have the same potential maximum energy, then one is chosen at random. If no location for farming is available, then (0.1 * *init_energy*) is subtracted from the current amount of energy.

<u>Farm and Consume Resources</u>

If a patch is chosen, then the patch is farmed. The cost of clearing vegetation from the patch is calculated based on how much vegetation is on the patch, and the amount of vegetation for the patch is set to 0 (patch turns white). The harvest is weighted by the patch fertility, which declines by *fertility_loss* with each time it is farmed. The energy accumulated through farming is computed. The total amount of energy received by a household after each step in the simulation is the *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, and the distance to the patch from the farmstead. [*net_return* = ((*harvest* * *init_energy* / 100) * patch fertility / *divisor*) - (*farm_cost* * *init_energy* / 100) - *veg_clear_cost* - ((distance from patch to farmstead) / 5)]. The *net_return* value is added to the energy of the farmstead.

The cost of clearing vegetation is scaled according to the cost of clearing forest (set by the user), with forest (*vegetation* = 50) costing the most to clear and nearly bare grass (*vegetation* = 1) the least. [*Veg_clear_cost* = (*init_energy* * *tree_clearing_cost* / 100) / *vegetation*]

<u>*Reproduce*</u>
After a patch 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_energy* / 100). If so, then the agent reproduces a copy of itself. Both households are given half of the original amount of energy. The new 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.

<u>*Check Death*</u>
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.

<u>*Regrow Plots*</u>
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*. *Fertility* cannot exceed 1. For patches that were not cultivated (*field* = 0), and do not have a farmstead (*farmstead* = 0), 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).

<u>*Land Tenure*</u>
If the "tenure" 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.

<u>*Climactic Variability*</u>
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.