

Swidden Farming Version 2.0

by C. Michael Barton 2013

ODD METADATA

MODEL OVERVIEW

Purpose

This model simulates some of the dynamics of household level swidden agriculture (also called shifting cultivation or slash-and-burn agriculture). Agents represent farming households. 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 impacts of a number of social/economic decisions related to farming, including evaluating potential costs and benefits for choosing land to farm, the conditions for abandoning poorly producing land, when a household should fission and create a daughter household, innovation rates, and variation in land tenure rules. It also explores the effects of varying environmental conditions, including harvest returns, farming costs, soil fertility changes, land cover changes, and exogenous environmental change.

The agricultural strategy of the agents results from the combination of six key values (*move-dist*, *move-threshold*, *fission-rate*, *farm-dist*, & *min-fertility*).

In adaptive mode, the model explores the success (or failure) of strategies determined by these variables. Values for these variables are set randomly among the households at the model's initialization, and strategies can change subsequently through innovation in new households created when existing household agents fission.

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.

Several variables set by the user scale as integers from 0-100% to be more understandable, but are converted to floating point values between 0-1 for calculations in the modeling. These are shown below.

$move-rate = init-move-threshold / 100$

$harvest-rate = harvest / 100$

$farm-rate = farm-cost / 100$

$clearing-rate = tree-clearing-cost / 100$

$move-cost-rate = move-cost / 100$

$fertility-loss-rate = fertility-loss / 100$

$restore-rate = fertility-restore / 100$

These floating point rate variables and other state variables are described below in more detail, beginning with agent (household) variables.

<i>energy</i>	The amount of energy each agent has. If it reaches 0 the agent is removed from the simulation. This is initialized as <i>init-energy</i> = 100
<i>move-threshold</i>	If a household's energy falls below this value, the household will move to a new, unoccupied location. $move_threshold = move_rate * init_energy$.
<i>fission-rate</i>	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 <i>fission-rate</i> = 1.5 means that a new household will be created when the household's energy is 150% of the amount of energy the household was initialized with. In adaptive mode, <i>fission-rate</i> can vary randomly by household agent between 1-2.
<i>farm-dist</i>	The maximum 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. <i>farm-dist</i> can vary randomly by household agent from 1-20 in adaptive mode.
<i>min-fertility</i>	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.
<i>move-dist</i>	The distance an agent will move if it must find a new farmstead location. In adaptive mode, this value may vary from agent to agent. The specific value ranges from 1 – 5 and are implemented as multiples of <i>farm-dist</i> .
<i>net-return</i>	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. $Net_return = ((harvest_rate * init_energy) * patch_fertility / divisor) - (farm_rate * init_energy) - veg_clear_cost - ((distance\ from\ patch\ to\ farmstead) / 5)$.

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

<i>vegetation</i>	Numerical value (0-50) that represents the vegetation community on a patch. A value ≥ 40 represents forest, $40 > value \geq 20$ represents shrub, $20 > value > 0$ herbaceous, and value of 0 represents a bare patch.
-------------------	--

<i>fertility</i>	This value (0.0-1.0) represents the agricultural productivity of a patch. It is lowered by <i>fertility-loss-rate</i> each time a patch is farmed. Households cannot farm when the fertility of a patch is 0. <i>fertility-loss-rate</i> is calculated from <i>fertility-loss</i> (set by the user between 0-100) / 100.
<i>farmstead</i>	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. A farmstead patch cannot be farmed.
<i>state</i>	Whether a farmstead is “active”, “unused”, or “abandoned”.
<i>field</i>	Whether a patch is under cultivation or not (1 or 0).
<i>owner</i>	Records which agent owns a given patch (=agent ID). Patches need not be under cultivation to be owned.
<i>fallow</i>	Tracks the number of time steps a perviously cultivated patch has not been cultivated.
<i>veg-clear-cost</i>	The cost of clearing vegetation from a patch for the purpose of farming. As long as the <i>vegetation</i> value and <i>clearing-rate</i> of a patch is > 0, the $veg-clear-cost = ((init-energy * clearing-rate) / vegetation)$. It is updated for all patches at each step.
<i>site</i>	Whether or not a patch is an archaeological site (“True”/“False”)

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

<i>init-households</i>	Initial number of households at the start of the simulation. Set by the user.
<i>init-energy</i>	Initial amount of energy for each household at the start of the simulation. Set to 100 in the code.
<i>harvest-rate</i>	Gross energy return from a farmed patch. Varies from 0-1 and is multiplied by the initial household energy. This partially determines the amount of return a household will receive from farming a patch. See <i>net-return</i> above. Set by the user.
<i>farm-rate</i>	Amount of energy needed to cultivate a patch. Varies from 0-1 and is multiplied by the initial household energy. See <i>net-return</i> above. Set by the user.
<i>clearing-rate</i>	Energy cost of clearing the maximum vegetation cover from a patch (i.e., trees) in order to farm it. Varies from 0-1 and is multiplied by the initial household energy. Clearing other vegetation is scaled between this value and 0. Set by the user.

<i>move-cost-rate</i>	The energy cost incurred for moving a distance of 1 patch. Varies from 0-1 and is multiplied by the initial household energy. Set by the user.
<i>fertility-loss-rate</i>	The amount of fertility that is lost from a patch each cycle if it is farmed. Varies from 0-1.
<i>fertility-restore-rate</i>	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). Varies from 0-1.
<i>bad-years</i>	The approximate percentage of cycles, chosen randomly, in which the harvest is half the normal amount. Simulates exogenous environmental variation. Set by the user.
<i>max-fallow</i>	Used in conjunction with the <i>transfer-ownership</i> option. The maximum number of cycles a patch can remain unfarmed (i.e., fallow) before it becomes available for use by another household. Set by the user.
<i>divisor</i>	Variable used in the calculation of returns from farms. If the year is a bad year, then <i>divisor</i> = 2. If it is not a bad year, <i>divisor</i> = 1.
<i>innovation-rate</i>	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 0.001-0.5 and is set by the user.

Process Overview and Scheduling

Essentially the simulation processes revolve around the creation of farmsteads, choice of farm plots, and farming by each household agent. For each household active in the simulation, the specific processes evoked and their scheduling is as follows:

- 1) The household checks to see if it needs to move to a new location based on current energy level.
- 2) The household chooses the best available (unowned by another household) patch to farm within the radius of *farm-dist*. The net energy from the farmed patch is calculated and added to each household's total energy.
- 3) The potential for a household to fission or to die are evaluated based on the household's energy level. The simulation continues until all households die or until stopped by the user.

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 maximum farming distance (*farm-dist*), the

energy level that will prompt a move to a new location (*move-threshold*), the maximum distance an agent will move in order to establish a new household [$(farm-dist * move-dist) + 1$], the amount of energy required to fission and produce a daughter household (*fission-rate*), 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 (*min-fertility*). All daughter households will have the same values as the parent household EXCEPT if they innovate when during fissioning (set with *innovation-rate*). If an innovation occurs, a new randomly selected value for one of the above variables is chosen. As the simulation progresses, agents with successful adaptive strategies become more numerous than those with less successful strategies.

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.

Finally, changing land tenure rules (*transfer-ownership* and *max-fallow*) can produce dramatic changes in household population over time.

Adaptation (how the agents adapt their behavior to their and their environments current state).

The agents attempt to farm the patches near 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 (especially in adaptive mode).

Fitness/Objectives (a summary of the agents' goals).

The proximate goal of agents is to maximize energy levels by farming nearby patches. 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).

Household agents are able to detect who owns a patch of land, the distance to a patch of land, the fertility of a patch, and the amount of vegetation on a patch.

Interactions

Household agents do not directly interact. However, households indirectly interact with other households by preventing them from farming a patch (i.e., owning a patch) and by making a previously owned patch available to other agents.

Stochasticity

In controlled mode, there is stochasticity over time in harvest returns (*bad-years*), as well as in the choice of a farmstead and the specific patch to farm. In adaptive mode, there is

also stochasticity in many of the key variables determining agent behavior (see Emergence above).

Collectives (whether the agents are grouped socially).

The agents are not grouped socially.

Observation (how data are gathered from the model).

Graphs in the interface display the total number of household agents active, the average net-return, the average household energy, the average fertility of patches, the percent of different kinds of natural vegetation, and the percent of patches farmed/fallowed. Monitors also display the average fission energy, average farming distance, average move threshold, and average move distance. Any variable can be recorded in BehaviorSpace.

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]). If adaptive mode is selected, relevant variables are set randomly for each initial household agent within their accepted ranges. The patch on which each household 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

Household Movement (*move*). 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* \geq the agent's *min-fertility*, with no farmstead, and unowned are 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-threshold* to be lower than the current energy level so that a household doesn't keep moving after it first moves.

Choosing Land to Farm (*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 farm patches 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 (*farm*). 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*, but cannot go below 0. Once the patches are updated, the *net-return* from the farmed patch is calculated and added to the energy of the farmstead.

Household Fissioning (*reproduce*). 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. In adaptive mode, 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 based on the *innovation-rate*.

Households Die (*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.

Vegetation Succession and Fertility Restoration (*regrow-patch*). 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 are 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 farming sets vegetation to 0.

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 *transfer-ownership* switch is on, households have limited land tenure. If land that is not a farmstead has lain fallow for more than the number of years set by the user in the *max-fallow* slider, it reverts to unclaimed and any agent (including the original owner) can claim it in the next cycle.

Innovation of New Farming Strategies (*innovate*). If a new daughter household is allowed to innovate (random selection based on the *innovation-rate* setting), then ONE of the following variable is changed randomly: *move-dist* (0-4), *move-rate* (0-1), *fission-rate* (0-1 + 1), *farm-dist* (0-20), *min-fertility* (0-1). The remaining variables will be identical to the parent household agent's values.

CREDITS AND REFERENCES

Copyright C. Michael Barton, Arizona State University, 2013

Associated publication:

Barton, C. M. (2013). Complexity, Social Complexity, and Modeling. *Journal of Archaeological Method and Theory*, 1–19. doi:10.1007/s10816-013-9187-2