Lithic Raw Material Procurement and Provisioning March 2015

Jonathan Paige (Arizona State University)

Model Documentation (ODD Protocol)

Overview

This model presents a basic abstraction of the lithic raw material use and provisioning behavior of a group that inhabits a permanent base camp, in an environment populated by some number of raw material sources. Here, forager agents constantly move to random raw material sources, and pick up "cobbles" of lithic raw material until they can carry no more, they then return to the base camp, and deposit these cobbles then repeat the process. During each tick, these turtles perform one unit of work, depleting their stock by some number. Camp dwellers are turtles, who, in the basic setup, stay at the camp and perform work using the available stock of raw material.

Quantitative models of lithic technology tend to focus on hunter-gatherers as subjects of study. This model adopts and modifies some concepts from the literature on the organization of technology among hunter-gatherers to groups that have permanently occupied villages. The model is designed to evaluate the effects of individual mobility, core reduction technology, availability of lithic raw material, and forecasting of raw material shortages on the amount of work individuals are able to perform both at permanent settlements, and during movement across the landscape to and from those settlements. The model features a settlement occupied by agents, which are split into two groups: foragers and camp dwellers. Foragers move between raw material sources and the settlement, performing work and transporting raw material. Camp dwellers occupy the settlement and perform work. Each unit of work performed exhausts some amount (which varies by core reduction efficiency) of raw material. The model allows for evaluation of the conditions in which different technological, or mobility decisions are more likely to be made.

Purpose

The purpose of this model is to evaluate the relationships between raw material conservation, procurement and work output, in order to better understand some of the potential pressures that faced past sedentary users of stone tools. The main unit of measurement here is the difference between potential work value, and actual work performed. The greater difference between these two values, the less effective the system is at providing turtles material they need to perform work.

State variables and scales

The world in this model is 81x81 patches, and does not wrap. The world is inhabited by patches, and turtles. Patches are separated into categories of raw

material source (white) a base camp (red), patches with no characteristics (black) and patches that have had exhausted cobbles dropped into them (yellow). Turtles are separated into foragers (blue), camp dwellers (gray) and temporary foragers (blue). Foragers will turn red if they do not have enough material on them to perform work.

The global variables are:

1. *Odometer.* This tracks the distance each turtle travels and is used to calculate the mean distance travelled by all turtles per run.
2. *Meantravel.* This is the calculated mean travel distance.
3. *Sources*. This value represents the number of sources to be created during setup. This value is modified using a slider.
4. *Distances.* This is a list of the distances from those sources to the base camp
5. *Sd.* This is set to the standard deviation of core values at the base camp during the *up-plot* procedure.
6. *Lcores.* This is set to the number of cores brought to the base camp during the *up-plot* procedure.
7. *Workperformed*. Each time a given turtle has enough raw material to perform work (i.e. that performing work is successful), .01 is added to this variable.
8. *Mobilework.* Each time a foraging turtle has enough raw material to perform work, .01 is added to this variable.
9. *Sedwork.* Each time a camp dweller has enough raw material to perform work, .01 is added to this variable.
10. *Expectedwork.* Each time a turtle could possibly have performed work, .01 is added to this variable, regardless of whether there was enough raw material to perform that work.
11. *Nofturtles.* This variable is altered by a slider, and denotes the number of turtles to be created during the setup procedure.
12. *%.* Denotes the proportion of turtles that are to be set to camp dwellers.
13. *Decay.* This is the measure of how much is subtracted from a cobble per unit of work performed.
14. *Csize.* This value serves as the mean of the normal distribution from which turtles draw new cobble sizes from source patches. Simulations were run keeping cobble size at 80.
15. *Packsize.* This represents the carrying capacity of turtles. Turtles that draw cobbles from sources will draw values until the sum of those values exceeds this value.
16. *Close?* True or false variable that indicates whether sources will be created within 10 patches of the base, or within the world without spatial constraint.
17. *Forecast.* True or false variable that triggers the ability of camp dwellers to identify when available core stocks are low, and change their behavior to temporary foraging.

Each patch has the following state variables:

1. *Cores.* This serves as a list of deposited cobbles at base camp.
2. *source?* This true or false variable serves to initialize patches that are to function as raw material sources
3. *Base?* This true or false variable serves to initialize the patch that is to function as the base camp
4. *Dist*. Denotes the distance between each source patch and the home base.

Turtle variables include:

1. *Forager*. This true or false variable serves to initialize which turtles will function as foragers, and which will serve as camp dwellers.
2. *Temp?* This true or false variable serves to change the behavior of camp dwellers, converting them into temporary foragers, when the global *forecast* is true.
3. *Pack*. This list is analogous to the patch list *cores*, in that when a turtle picks up, or deposits cobbles a value is added to, or subtracted from this list.
4. Var, cond1, cond2, cond3, cond4, and cond5, are all numerical and true or false variables that determine what each foraging, or temporarily foraging turtle does each tick, and what its goals are that tick. The movement submodel will be described in more detail below.
5. *Mpack*. Set to the mean values of the list pack. This serves as a quick indicator of material remaining in the pack.
6. *Target*. Set a goal patch for each turtle, before movement begins.

The temporal scale of the model is in ticks. Each agent runs through one step of a broader routine once per tick, allowing for all agents to be performing the same broad suite of activities in different orders. Here, the temporal extent of the model is limitless, but simulations were performed in the range of 700 to 2000 ticks. The spatial relationships are predominately between turtles, and the camp, or source patches.

Process overview and scheduling.

Go triggers two separate procedures: one for foragers, or temporary foragers (*forageprocedure)*, and one for camp-dwellers (*camp-work)*, then a tick, and the *up-plot* procedure. The *forageprocedure* is the most elaborate of the model. Here, turtles *go* in random order. The order of operations within the forageprocedure is determined by a suite of IF statements, the conditions for which are triggered by set variables during the preceding operation. This ensures that each turtle performs one operation within the *foragerprocedure* independently per tick.

Design concepts

The *basic principle* of this model is that it evaluates how much work agents are able to perform with some availability of raw material, and some efficiency of raw material use, relative to the greatest possible amount of work they are able to perform if they had limitless raw material access. This serves to evaluate how availability of raw material, and amount of raw material provisioning may act to conserve stocks of material. Here, differential patterns of raw material exhaustion and work output emerge from these behaviors. The adaptive behaviors here include 1. Ensuring that as much material as possible is procured from sources, and 2. Responding to resource depletion by making additional forays for nearby raw materials.
The first adaptive behavior is evaluated to explore whether foraging by mobile individuals is enough to provide a base camp with enough raw material. The second behavior is evaluated to explore whether identification and response to low stocks is sufficient enough to stave off raw material crashes.

In cases where both are not sufficient, minimizing decay rate, or abstraction of efficiency of core reduction, may be seen as one alternative strategy to ensure that all have enough material to use.

There is no learning in this model, and all agents know the location of all source patches. Only the camp dwellers are able to evaluate the amount of stock remaining at the camp. Interaction between agents is indirect, in that all agents depend to some degree on the provisioning behavior of other agents to supply a stock of material for their use. The availability of material for each agent is dependant on the cumulative behaviors of all other agents.

The stochasticity of the model lays in the distribution of raw material sources, the distribution of possible cobble sizes, and in which cobbles are chosen to deposit, or reduce during depositing, or working routines.

Initialization

Setting state variables that denote which turtles will be camp dwellers or foragers occurs during the setup phase, as well as setting state variables that differentiate the base, and ram material sources.

Input data

There are no input data for this model.

Submodels

Forageprocedure

1. *Var*. A numerical variable used to trigger targeting of patches, provisioning, and deposition behaviors. As this numerical variable advances 0 to 5, it triggers the x+1 condition described below.
2. *Cond1*. Triggered by var = 5. This is true when a foraging, or temporarily foraging turtle has returned. This triggers the *deposit* and *provision-self* submodels and sets var to 1

3. *Cond2*. Triggered by var = 1. This is true when a turtle has just deposited cores at the base camp and provisioned itself. This triggers a *target* patch to be set, and var to be set to 2.
4. *Cond3.* Triggered by var = 2. This is true when a forager has targeted a patch. This condition triggers the *gototarget* and *work* routines and is repeated until the target patch is reached, at which point the *stock-up* routine is triggered, and var is set to 3.
5. *Cond4.* triggered by var = 3. After a forager stocks up at a source patch, this triggers the agent's setting a new *target,* the base patch, and sets var to 4.
6. *Cond5.* triggered by var = 4. Turtle moves forward and performs work. This is repeated until the target patch is reached, at which time var is set to 5.
7. *Setvariables*. Serves to ensure that var changes and triggering of cond variables, happens once per tick, so that each forager moves, and performs work once per tick.

Gototarget

This procedure is performed under cond3. During each tick, the heading is set to the target again, to ensure that turtles do not overshoot the target patch. Once the patch is reached then if the target is the base, var is set to 3, and if the patch is a source, the *stock-up* routine is run, and the var is set to 3.

Deposit. Performed under cond1. The length of the turtle's pack is evaluated, and a random cobble within that pack is selected for donation. The turtle then adds the value of that donation to the patch list *core* and subtracts that donation from its own pack. This is repeated until the turtle has one core left.

Stock-up. Performed under cond3. A value taken from a random distribution with a mean of *Csize* and standard deviation of 5 is added to the pack. This is repeated until the sum of values within pack exceeds the packsize, at which point the lowest value is selected for removal.

Work. At the start of this routine, .01 is added to *expectedwork* regardless of whether work is performed.

Here, if there is a cobble in the turtle's pack, and if the mean value of the pack is greater than 16, then if the core selected to work is less than 15, then it is dropped on the current patch, turning it yellow if it is not a camp, or source, at which point the *work* routine is triggered again. Otherwise, if the selected core is greater than 15, then that cobble is replaced with another value in the list that equals (cobble - *decayed*). .01 is added to *workperformed* to indicate that work was successful this tick, and .01 is added to *mobilework.*

Camp-work. At the start of this routine, .01 is added to *expectedwork* regardless of whether work is performed or not. This routine is similar to the *work* routine except when forecast = true, and cores at the camp are sufficiently small, camp dwellers are converted into temporary foragers (temp? = true) and the forager procedure is triggered. When work is successfully performed, .01 is added to *workperformed*, and .01 is added to *sedwork*.

Provision-self. At the start of this routine, .01 is added to *expectedwork.*
Here, a core is selected from the list *cores* at the home patch, which is added to the *pack* list, and subtracted from *cores*.
Only turtles that are dedicated foragers perform this routine.