Supplementary Material: Detailed Description of the Island Resource Exploitation Model (leGUME)

David O'Sullivan George Perry

August 8, 2012

This document provides a full and detailed description (the second 'D' in the ODD protocol) of the workings of the model described in Chapter 8 of *Spatial Simulation: Exploring Pattern and Process* (Forthcoming, 2013, Wiley). It should be read in conjunction with that material and while examining the model itself, full code of which is available at `someurl.org`.

1 Process overview and scheduling

See leGUMEv1.6.nlogo

In spite of the large number of model parameters listed, the overall behaviour of the model is straightforward. The easiest way to get to grips with it is to examine a flowchart of the sequence of events in the model each model time step (see Figure 1). The basic sequence is that groups assess their situation and relocate if necessary, then hunt and gather each month (one iteration in model time) and the collective landscape map is updated. During the relocation step a group may decide to leave the island if hunting is not going well, or to relocate its home camp if local gathering is not going well. To ensure that groups spend at least one year on the island, this step only occurs after the first year has ended. Every twelfth iteration, in other words once a year, additional operations occur: human population growth, and possible group splitting or merging, and landscape resource regrowth.

2 Landscape initialisation

The first stage in landscape initialisation is to set up a two state landscape of which approximately the desired proportion p_H of grid cells have non-zero high value resource capacity, $k_H > 0$. This is done using the SIMMAP modified random clusters (percolation) method (see Chapter 5) with percolation threshold p and assigning the percolation clusters created so that near to p_H of grid cells are high value sites. All patches outside these areas have $k_H = 0$. All grid

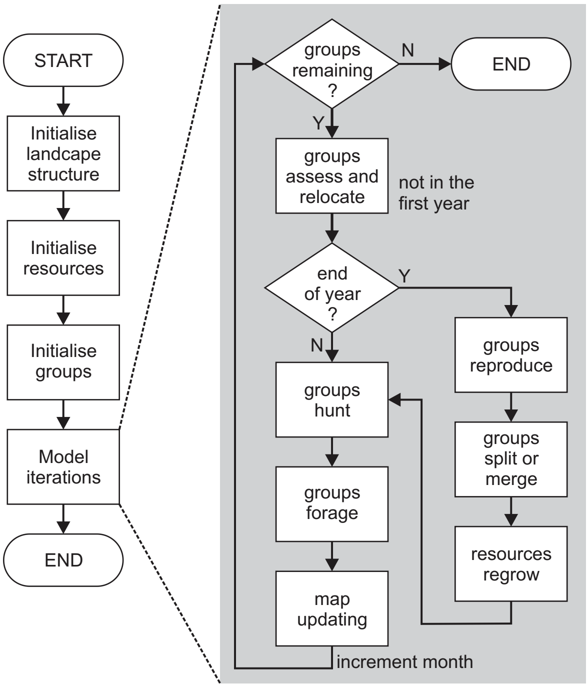

Figure 1: A flowchart showing overview of the model. The left-hand diagram shows model initialisation and running. Right-hand panel shows details for each model iteration.

cells in the high value resource area are then assigned a high value capacity k_H according to

$$k_H^* = k_{\max,H}\mathcal{N}_0^1(\mu = 0.9, \sigma = 0.1) \tag{1}$$

where the sub- and superscripts on the normal random deviate indicate that repeated random draws are made until a value between zero and one is obtained. Next, these values are locally averaged on the von Neumann neighbourhood to give the initial setting for k_H:

$$k_H = \frac{1}{5}\sum_{r=1} k_H \tag{2}$$

Note that this results in a single grid cell 'boundary region' around each patch of high value resource (see Figure 2). Low value resource capacity k_L is set by applying the procedure of Equations 1 and 2 but substituting the low resource capacity parameter $k_{L,\max}$ in place of $k_{H,\max}$.

With the resource capacities set, initial random resource availability levels are set according to

$$\begin{aligned} z_H &= k_H\mathcal{N}_0^1(0.9, 0.1) \\ z_L &= k_L\mathcal{N}_0^1(0.9, 0.1) \end{aligned} \tag{3}$$

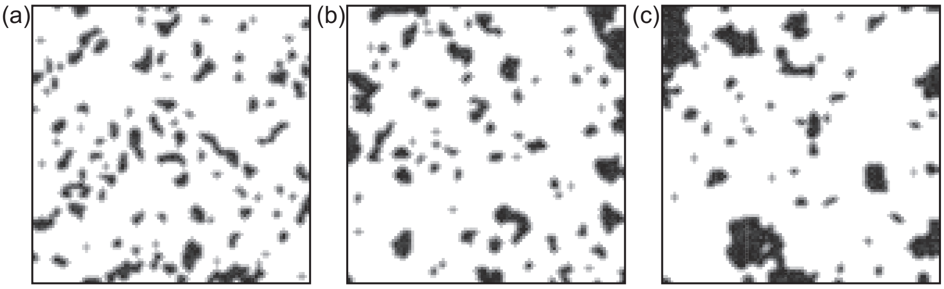

Figure 2: Three initial landscapes, each with $p_H = 0.15$ and with (a) $p = 0.4$, (b) $p = 0.52$, and (c) $p = 0.58$. These snapshots are for a smaller 100×100 island. Dark areas are high value resource patches.

Finally, ten iterations of the annual resource regrowth process (see the next section) are applied to give the initial landscape. Three example initial landscapes are shown in Figure 2.

3 Landscape regrowth

As landscape regrowth is part of the model initialisation process, we describe it here. Resource regrowth is modelled using a standard logistic growth model. Thus, the levels of availability of each resource change each year, that is every twelfth model iteration according to

$$z(t+1)^* = z(t) + \mathcal{N}(r,\sigma)z(t)\left[\frac{k - z(t)}{k}\right]\Bigg|_0^k \tag{4}$$

where the z, k, r and σ parameters are either the high or low value resources versions as appropriate. The final 0 and k sub- and superscripts indicate that the resulting new resource level is constrained to lie between zero and the grid cell capacity k, that is it is set to k if Equation 4 produces a value greater than this. Following this regrowth, the low value resource (only) undergoes diffusion by local averaging with its four near neighbours according to

$$z_L = (1-w)z_L^* + \frac{w}{4}\sum_{r=1} z_L^* \tag{5}$$

where z_L^* is the intermediate value resulting from Equation 4 applied to the low value resource. For the high value resource, if z_H^* is less than the minimum sustainable level $z_{H,\mathrm{min}}$, then the resource is deemed to have been eliminated at that location, and set to $z_H = 0$. High and low value resources exist completely independently from one another so there are no interactions in their growth processes.

4 Collective ‘map’ of the island

All groups in the model maintain a shared map of the island. This is crudely represented by maintaining an integer index value $V_{\mathbf{x}}$ for each grid cell location $\mathbf{x}$ such that $1 \leqslant V_{\mathbf{x}} \leqslant 100$, where a value of one means that the location is familiar to groups on the island, while 100 means that it is completely unknown. When the model is initialised, all locations are equally unknown ($V_{\mathbf{x}} = 100$) except for the local area A around the home-camp location, where the index is initialised to one. Any grid cell visited by a group has its index value reset to one, and every month all grid cells other than the home-camp location have their index value incremented by one. In addition, all cells perform a local averaging step over the von Neumann neighbours:

$$V_{\mathbf{x},i}(t+1) = 0.95V_{\mathbf{x},i} + 0.05 \sum_{d_{ij}=1} V_{\mathbf{x},j} \tag{6}$$

This makes the map ‘memory’ of locations that have not been visited recently fade, and also has the effect of slowly spreading the effect of known locations, so that, for example, grid cells adjacent to a recently travelled route become better known. Note that unlike the hunting memory of groups which are exclusive and not shared, the map of the island is held in common.

This index plays a role in the search behaviour and relocation behaviour of groups during model execution, as described in the relevant sections. The map ensures that regions of the island near currently active hunting grounds, along recent search paths and around the current and recent home camp locations are less favoured for search hunting trips.

5 Initialisation of groups

A single group is placed on the island in a location on the edge of the grid, such that it is not on a high value resource patch, but is adjacent to one. This represents the idea that a group will arrive on the shoreline of the island, and that they are likely initially to be close to useful resources.

The initial home camp $\mathbf{c}$ is set to this location. The collective map index value for all grid cells is set to 100, except for those grid cells in the local area A (defined according to Equation **??**) where $V_{\mathbf{c}} = 1$. Thus, the island starts as essentially unknown territory to the newly arrived group.

The group’s initial population n is drawn from a Poisson distribution with mean $\lambda = n_{\max}/2$. Hunting memory H is set to be empty, and the search tortuosity of the group is set to an initial value $s_{p,0}$.

6 Group assessment of situation and relocation

In every iteration after the first year of model time, the first action taken by groups is to determine whether they wish to relocate. First they assess how good the hunting has been in recent times. The total yield associated with all hunting spots in their memory $\sum_H Y_i$ is determined, and if it is equal to zero, then the hunting is deemed bad and the group elects to leave the island. If they are the last remaining group, then this will end the model run.

Next the local area A is assessed to determine if the group will relocate because the foraging locally is poor. Here the total currently available low value resource $\sum_A z_L$ is determined and if it is less than half the group's total annual resource requirement $nZ/2$ they decide to relocate. This formulation implicitly assumes that a group hopes to collect up to half of its annual resource needs locally. We note that this threshold may be rather low, causing groups to relocate relatively often, but no more often than once a month.

When a group decides to relocate two possible mechanisms are used. A particular model run will be set up to use just one of them. Under the first method the group first moves to the hunting spot in memory with the highest map index of 'unknownness', $V_{\mathbf{x}}$, that is the least familiar location. From that intermediate location, the new home camp is chosen to the be the nearest location, measured as $\max(\Delta_x, \Delta_y)$ which has no currently available high value resource, that is $z_H = 0$. This distance metric which for convenience we will denote d_8, corresponds to the number of eight-direction movement steps across the lattice which it would take to reach the location. This represents a deliberate attempt to move to a location where good hunting is known to be available, consistent with wishing to explore the island (the high $V_{\mathbf{x}}$ requirement). The second alternative, less 'rational' method selects the least known site (the one with highest $V_{\mathbf{x}}$) with $z_H = 0$ within d_8 of the current home camp, such that $\lfloor 2r \rfloor < d_M < \lceil 3r \rceil$.

Both methods assume that a group will not locate directly in area of high value resource. d_8 distances are used because the random walk component of the group searching behaviour operates on the lattice with eight directions of movement. We also explored the effect of using four nearest neighbour movement, but it makes no qualitative difference to model outcomes. After relocation, the search tortuosity of the group is reset to its initial value, $s_p = s_{p,0}$ and the map index values $V_{\mathbf{x}}$ of the new home camp $\mathbf{c}$ and its local area A are set to one.

7 Hunting behaviour

Hunting and the associated decision making occurs every model iteration. The sequence of operations is shown in Figure 3.

8 Deciding how many hunting trips

The first step is to decide how many hunting trips will be undertaken this month. The maximum possible number of hunting trips is determined according to

$$n_{X,\max} = \left\lceil \frac{n}{2} \frac{f_X}{n_X} \right\rceil \tag{7}$$

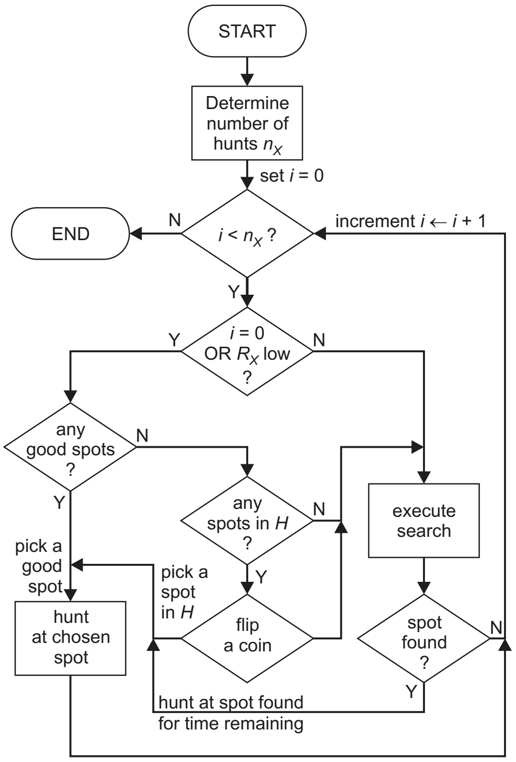

Figure 3: A flowchart showing the decision making process for hunting.

which is half the group population (assumed to be hunters) multiplied by the max number of hunts per person in a month, f_X, and divided by the hunting party size, n_X. This result is rounded up, which means that the maximum is never less than one hunting trip. In the first month of the model running this maximum value is the actual number of hunts which will be undertaken, so that $n_X = n_{X,\max}$. If at any later time there are no spots in the hunting memory, then the actual number of hunts is also set to the maximum value. In later months, the result of Equation 7 is combined with an assessment of the likely hunting success, which is calculated from the fraction of hunting spots in memory with non-zero yields, that is

$$\hat{p}_s = \frac{|\{H : Y_i > 0\}|}{|H|} \tag{8}$$

The number of hunting trips to be undertaken is then determined as a Poisson random deviate

$$n_X = \mathrm{Pois}\left(\lambda = \hat{p}_s n_{X,\max}\right)\Big|_0^{n_{X,\max}} \tag{9}$$

with repeated draws to ensure that the result falls in the indicated range. Note that it may be determined that no hunting trips will be attempted.

9 Deciding what to do on each hunting trip

With n_X now known, the group repeats a process of deciding to consult their hunt memory, or whether instead to search, based on hunting success in the year to date, and the state of the hunt memory H.

For the first hunting trip of the month, and also subsequent ones if the total return from hunting in the year so far is less than the level aimed for, then the left hand side of the flowchart in Figure 3 is followed. This choice uses an expectation of hunting resource yield per month of $E(R_X) = nZ/24$, which is half the total resource requirement per month of the group. A group will consult its hunt memory to identify an already known hunting spot if the resources taken from hunting in the year to date R_X are less than this amount multiplied by the number of months in the year so far, in other words if they are 'below quota' for hunting. If the decision is to consult memory rather than conduct a new search, then, if there are any 'good spots' in the hunt memory, which are those with a known yield $Y_i > nz/24$, which are also within range, such that the d_8 distance to them is less than the hunting range t_X, then one is chosen (see below) and the group goes there to hunt. If no good spots are available, but there are any spots in hunt memory within range, then with equal probability, the group will choose one of those spots and hunt there, or will instead opt to search for a new spot.

For the second and subsequent hunting trips in month, or if resource collection is going reasonably well this year, that is the overall take R_X is running at more than $nZ/24$ per month, then the group will decide to search for new hunting grounds.

10 Going hunting

For hunting and searching the group has available a number of 'steps' given by the hunting range t_X. If hunting, a spot must be chosen from good spots or spots in memory within range. The chosen spot is that which has the highest value of yield divided by its d_8 distance from home camp plus one, $Y_i/(d_M + 1)$. This is similar to the simple rule followed in the foraging model discussed on pages **??** ff. Having decided the spot, a route to it is randomly generated by moving in single steps towards it parallel to either the x or y axes at each step, in a d_8 shortest path (see Figure 4). Each grid cell $\mathbf{x}$ visited *en route* has its value of $V_{\mathbf{x}}$ set to one. The d_8 length of the route is deducted from t_X to determine the remaining available steps for resource exploitation at the hunting ground. Note that this route may involve crossing narrow inlets if these intervene between the home-camp and the hunting spot, because we assume that these would not have represented a serious obstacle.

Having arrived at the spot, the group exploits available high value resources in the grid cell and then moves to whichever of the eight nearest neighbours has the highest remaining z_H. This exploit-move behaviour is repeated for the remaining available number of steps, after which the group instantaneously

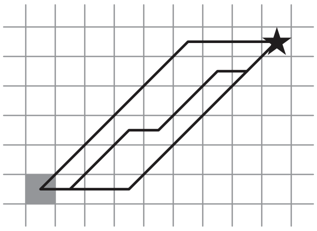

Figure 4: Three d_8 shortest paths from home camp (grey square) to a hunting spot (star). Note that no path will go outside the parallelogram shaped region between home camp and the hunting spot, and that progress is made every step closer to the hunting spot with no backtracking.

returns to home camp–it is assumed along the same route taken to get there.

The amount of resource killed at each grid cell visited during the hunt is determined by the hunting party size, the hunting range, and the availability of resources.

$$\Delta z_K = \left(\frac{1}{t_X} \times \Delta_K \min\left(n_X, \frac{n}{2}\right) \times \frac{z_H}{k_{H,\max}} \right)\Bigg|_0^{z_H} \tag{10}$$

The first term ensures that the total kill over a whole hunt is no more than $\Delta_K n_X$. The second term means that each person in the hunting party will kill up Δ_K of resources during the whole hunt, and that the hunting party size will be restricted to only $n/2$ if the group population is less than $2n_X$. The third term modifies the kill due to the increasing difficulty of making kills when the prey population is low. Note that the maximum kill is constrained to be between 0 and the total resource in the cell z_H.

Over the course of the exploit-move behaviour each grid cell has the amount Δz_K removed from its available level of resources. A cell may be visited more than once during a hunt, as the movement is a simple random walk, albeit biased by the requirement to move to the highest neighbouring level of z_H at each step. The total kill made $\sum \Delta_K$ is accumulated for the group over the duration of the hunt, but when the hunt ends, the total resource added to the total and hunt collection totals for the year is limited to only $\min(n_X, n/2)\Delta_X$ reflecting the fact that the hunting party's ability to return home with kill may be lower than its ability to make those kills in the first place. Note that this limit is not affected by the duration of the hunt, only by the party size.

Each cell visited has its value of $V_\mathbf{x}$ set to one. If a spot in the hunt memory is encountered during the hunt, then it is removed from memory, since the information about this hunting ground is shortly to be updated on departure. When the available time for hunting is up, the grid cell visited during the hunt with the highest *remaining* value of z_H is recorded and stored in the hunt memory, with an associated yield $\sum \Delta_K$, that is the total kill (from all sites) for this trip. The current search tortuosity value s_p is also adjusted according to

$$s_p \leftarrow \min\left(s_p + \Delta_s, s_{p,\max}\right) \tag{11}$$

so that the next searching trip will be a more tortuous and ‘thorough’ exploration.

11 Searching

Search behaviour follows a biased random walk. Starting from the home-camp a walk of t_X steps begins. The first step is made to a randomly selected orthogonally or diagonally adjacent grid cell, where the selection is weighted by the $V_{\mathbf{x}}$ map index values of the grid cells. This means that less well known grid cells are more likely to be chosen, thus favouring exploration of the unknown. The bias is linear, so that the probability of a cell with (say) $V_{\mathbf{x}} = 35$ being chosen is five times greater than one with $V_{\mathbf{x}} = 7$. Of course on the first search of a particular month, all grid cells close to home-camp are likely to have similar index values so the initial direction of the search is more or less a simple random choice. On subsequent searches in the same month, the ‘road less travelled’ close to home becomes more likely to be chosen, and as a search ventures into the unknown, previously univisited cells are highly favoured for exploration.

Subsequent steps of the walk are in the same direction except that with probability s_p a change of direction will occur. When the direction changes it is again based on $V_{\mathbf{x}}$-weighted random selection, and immediate backtracking to the previous grid cell is disallowed. At each step, the $V_{\mathbf{x}}$ index value of the grid cell occupied is set to one. If a grid cell is entered with $z_H > 0$, then a hunt is launched at that location with the number of steps available for resource exploitation given by t_X minus the number of steps taken in the search so far. The hunt proceeds exactly as described in the previous section, except that the group is already ‘on site’, so there is no need for a route to the location to be generated. If the search process ends with no hunting ground found, then the search tortuosity is adjusted according to

$$s_p \leftarrow \max\left(s_p - \Delta_s, s_{p,\min}\right) \tag{12}$$

so that the next search undertaken will be more directed and so likely to range further from the home camp.

12 Local foraging

Local foraging is conducted in the local area A around the home camp location $\mathbf{c}$. The amount of ‘person-months’ of effort available for foraging is determined from

$$n_F = n - \left\lceil \min(\frac{n}{2}, n_X)\frac{n_X}{n_{X,\max}} \right\rceil \tag{13}$$

so that the more hunting that has been done in a month the less effort is available for local foraging. The minimum available foraging effort will be close to $\frac{n}{2}$.

The foraging effort is applied by once, for each unit of effort available, selecting the grid cell in A with the highest low resource availability z_L, and taking

$$\Delta z_L = \Delta_L \frac{z_L}{k_{L,\max}}\Bigg|_0^{z_L} \tag{14}$$

meaning that (subject to availability) successive visits to the same location will yield less resource as foraging becomes harder due to lack of resources. This amount is deducted from the z_L level of the grid cell targeted and added to the total resource accumulation for the group.

13 Human demography

Once a year, the human population is adjusted in accordance with overall success at resource collection, and groups may split if they become too large.

14 Reproduction

Population changes are determined using an exponential growth model with noise.

$$r_\mu = r_G \, \frac{R_T}{nZ}\Bigg|_0^{m_{\max}} \tag{15}$$

$$\Delta_n = \| n(t)\mathcal{N}(r_\mu, r_\sigma) \| \tag{16}$$

$$n(t+1) = n(t) + \Delta_n \tag{17}$$

Thus a mean birth rate is determined, which is the baseline birth rate r_G multiplied by a factor reflecting the success during the past year at resource collection. This mean birth rate is used to draw a random normally distributed actual birth rate, and the population change that results is rounded to the nearest whole number. Over time, if the group is successful at collecting the annually required quantity of resources nZ then it is expected to grow in population as the birth rate will consistently be high and positive. Note that a negative birth rate is possible, so that group populations may fall as well as rise.

15 Group merging

If after reproduction $n \leqslant n_{\min}$ then the group will merge with the nearest (measured by d_8 distance) other group. The populations of the two groups and their hunting spot memories are combined. If the number of spots in the merged group memory exceeds the maximum allowed, then the required number of spots with the lowest yields are forgotten. Note that the new merged group may subsequently also undergo the reproduction process with its augmented (post-merger) population, so that a small number of the population contribute twice in a year to possible population growth, but this will be unusual (it will only occur if the small group has experience negative growth to fall below $n_{\min}$) and the effect is expected to be very small.

16 Group splitting

If after reproduction and possible group merges, the population of any group exceeds the maximum allowed, so that $n > n_{\max}$ then the group splits into two groups. Each member of the population is assigned with equal probability to one of the two groups. The hunt memory of the original group is subdivided by determining the yield-weighted centre location of the spots in memory according to

$$\mathbf{h} = \frac{\sum_H \mathbf{x} Y_i}{\sum_H Y_i} \tag{18}$$

Then, the orientation of the spots in the memory is determined, that is whether it extends further parallel to the x or y axes of the model space. The hunting spots are then divided into two sets perpendicular to the longer axis at the centroid location $\mathbf{h}$. Thus an east-west oriented set of hunting spots will be split into eastern and western subsets. Note that if one spot has particularly high yield and many or all the others have zero yield, then $\mathbf{h}$ may be located close to or even on the 'edge' of the region covered by the hunting spots. In such cases one of the subsets may include many spots with close to zero yield while the other consists of only one spot, the good one. This does not seem an unreasonable scenario, although whichever group is assigned the poor hunting spots may struggle to succeed in subsequent months.

Having split the hunting spots, the subset with higher total yield is assigned to the larger of the two groups, and the lower yielding spots are assigned to the smaller group. Both groups then relocate from the current home camp location according to the usual process for group relocation (see pages 4 ff.), using their new acquired hunting spots in their memory if the hunting-spot based method is in use.

17 Hunt memory

The hunt memory H of each group plays an important role in its behaviour, as is clear from preceding sections. Here we summarise the ways in which the memory may change during model operation, so that it is clearer.

- New hunting spot tuples $\langle \mathbf{x}_i, Y_i \rangle$are added to H at the end of a hunting trip.
- If adding a spot to the memory at any time increases its size $|H|$ to greater than the hunt memory length n_H, then spots are removed from memory by removing the lowest yield spots first. If two spots have equally low yield, then one is chosen at random.
- If a spot already in the memory is encountered while hunting in the exploit-move phase, then it is removed from memory, as it will soon be replaced by an updated record for this hunting ground.

- When groups merge, their hunting memories are combined and if $|H| > n_H$ then the required number of lowest yielding spots are forgotten to ensure that $|H| = n_H$.
- When a group splits, its hunting memory is also split between the group and its 'offshoot' group as detailed in the previous section.