A formal test using agent-based models of the circumscription theory for evolution of social complexity

Williams, A. J. and Mesoudi, A.

Overview, Design concepts and Details (ODD)

1.1 Model description

The model description follows the ODD (Overview, Design concepts, Details) protocol (Grimm *et al*., 2006, 2010). The NetLogo model script can be accessed from: https://github.com/ajw246/Circumscription_ABM.

1.1.1 Purpose of the model

This model has been developed to test the logic of Carneiro's circumscription hypothesis on the emergence of complex society, with the focus on the effects of environmental and resource circumscription. The model also includes the effect of population growth, and therefore population pressure, independent from and in combination with environmental and resource circumscription. This model is intended to be a test of the full hypothesis proposed by Carneiro (1970, 2012a) in an abstract environment.

1.1.2 Entities, state variables, and scales

There are two types of entities in this model: villages and patches. Villages can move, interact with other villages, and form collectives called polities. Villages cannot occupy more than one patch, and patches cannot be occupied by more than one village. Villages can create new villages if there is an empty neighbouring patch for the new village to occupy, and if the *probability.growth* condition is met (see 1.1.4.1). The location, polity identification, and hierarchy level of villages can change over time, but each village will keep a single unique *who* number. The *hierarchy* attribute is used as the measure of social complexity here.

Patches form the environment that the villages act in and are immobile. Each patch will yield a fixed number of resources set at the start of the model which does not change throughout the model run. Patches form two different types of land depending on the number of resources they own (either fertile or less fertile).

The model environment is wrapped horizontally and vertically to form a continuous world such that villages can move off the edge and re-appear on the opposite side. This is to avoid introducing a hard boundary that could affect the results of the geographical conditions being tested.

Arch-polities are agents used in the model to facilitate the identification of different polities during each time step (see Table 1 for the state variables of arch-polities). Arch-polities do not have any direct interaction with either villages or patches. Instead, they record the current state of a polity when relevant for a submodel process (whether there are any villages in that polity, whether the polity is attacking or being attacked by another polity, and how many resources the polity as a whole has access to).

Villages and patches are characterised by different state variables (or attributes which distinguish entities of the same type, Grimm *et al.* 2010), which are listed in Table 1.

Table 1 List of the state variables (or attributes) of villages and patches.

State variables	Value range	Purpose
villages		
who	1 to total number of villages	Unique village identification ('*who*' as a label is specific to the NetLogo program)
polity	1 to total number of villages	Affiliation number shared by villages belonging to the same group to identify each polity and all the villages within it
hierarchy	1 to total number of villages	A positive integer indicating the level of hierarchy of the current village (1 = highest level, >1 means the village is subordinate to a village with the next highest rank within the polity). It is theoretically possible for all villages in the world to form one polity of a continuous chain of hierarchy, so the highest value of *hierarchy* can be the same as the population size, but in practice this is very unlikely.
level-above	*who* number range	The *who* number of the village ranked directly above the current village
level-below	*who* number range	The *who* number of the village ranked directly below the current village (see 1.1.4.2)
resources	0 – 100	The number of resources owned by the village, equal to the *land-resources* of the patch it is occupying
defending	true, false	Used to identify villages which are in the polity being attacked by another polity
benefit-move	0 – 100	Used by each village to calculate the potential gain in resources of moving away if defeated
benefit-remain	0 – 100	Used by each village to calculate the potential gain in resources of remaining in the same location and becoming subordinate. (*benefit-remain* = current *resources* – *tribute* cost)
head-village	true, false	To identify the highest-ranking village in a polity
dominant-dying	true, false	If village dies, this will be labelled true, to facilitate identifying any subordinates of that village to be re-set as autonomous polities
tag	true, false	To identify any subordinates of a village which is dying or rebelling, to facilitate re-labelling of those villages

remaining-head	true, false	To identify the highest-ranking village of a polity, to distinguish that village from any potentially rebelling villages
rebelling	true, false	To identify the rebelling village and its subordinates
rebelling-head	true, false	To identify the village which is rebelling from a polity
potential-rebels	true, false	To identify any subordinates below the *rebelling-head*
hatching-new-village	true, false	To identify a village which will potentially create a new village, with the spatial range *splinter-location*
patches		
land-resources	0 – 100	Number of resources yielded by each patch, used to denote the fertility of the area
village-claim	*who* number range	*who* number of the village which is assessing the patch as a potential location to move to. This number assignment is necessary to avoid multiple villages selecting the same patch
potential-escape	true, false	To identify patches within a given radius of a village's current location which it can assess as a potential area to move to when defeated
territory	true, false	To identify patches within the *conquering-area* radius for a village to find rival neighbouring villages
splinter-location	true, false	To identify patches within the radius *placement-distance* that a new village can potentially be placed on
pxcor* and *pycor	-20 to 21	Coordinates of each patch in the grid
arch-polities		
whole-polity	1 to total number of villages	Used to ensure that only one polity is attacking another at any given time and to identify all villages who share the same polity number (*whole-polity* = *polity*)
target-polity	true, false	To identify the polity being attacked
attacking	true, false	To identify the polity which is attacking another polity
polity-villages	true, false	To eliminate polities without any villages (the number of polities remains the same, but only those polities with villages can participate in the model)
polity-resources	Range depends on total number of villages and the total	Sum of *resources* of all the villages in the polity

	resources in the model world	

Table 2 The parameters in the model, and parameter values used in experiments.

Purpose	Parameter	Values	Description
Land type distribution	Distribution	Concentrated or random	The highly fertile patches in a concentrated distribution are all adjacent to other fertile patches to form a continuous band. The width of this band is determined by the *land.width*. The highly fertile patches in a random distribution are allocated random locations anywhere in the model world. The number of more fertile patches is determined by the *number.fertile*.
Social circumscription	*initial.villages*	10	The number of villages at time step 0.
	probability.grow	Main text: 0.1 Figure SM 6: 0.5	Each time step, every village will attempt to create a new village with the likelihood of *probability.grow.* A new village can only be created if there is available space within the radius *village.range.*
	probability.death	0	The probability that a village will become extinct. This parameter is set to 0 in all model runs here, but can be varied between 0-1.
	village.range	Main text: 1 Figure SM 6: 10	The radius (in number of patches) that a village can 'see' to find rival villages (*conquering-area*), create a new village (*placement-distance*), or move to if defeated (*moving-distance*). The three parameters are separate in the model, but because they all determine the range of area that a village in the model is aware of, they have been varied together in these model experiments. Figure 3 shows how the range of patches that a village is aware of can increase by the radius number.

Geographical circumscription	*land.width; number.fertile*	*land.width:* 2, 39 *number.fertile*: 82, 1599	**Concentrated** High circumscription: *land.width* = 2 (82 highly fertile patches in total). Low circumscription: *land.width* = 39 (1599 highly fertile patches in total). The width of land is determined by the parameters *line1* and *line2* in the model code, where *line1* always equals -20 and *line2* is set to either -18 or 19 to vary the *land.width.* **Random** The parameter *number.fertile* is coded as *green-patches* in the model. High circumscription: *green-patches* = 82. Low circumscription: *green-patches* = 1599.
	resource.difference	0.1 , 0.9	The difference in resources between the more and less fertile patch types. Steep resource gradient: the less fertile patches contain 10% of the resources of the more fertile patches. Shallow resource gradient: the less fertile patches contain 90% of the resources of the more fertile patches. The resources of the two land types are determined by the parameters *fertile-land* and *barren-land* in the model code, where *fertile-land* is always 100 and *barren-land* is set to either 10 or 90 to vary the *resource.difference.*
Polity conditions	*tribute*	0.1	Proportion of resources owned by the defeated polity demanded by conquering polity.
	probability.fragment	0.01	Probability of internal polity instability (1% chance of fragmentation).

	probability.attack	1	Chance that one polity will decide to attack another polity (100%), if a village belonging to a different polity is within *village.range*.
General setup	*step*	100	Time measured in steps (1 step is one complete run through the code by every agent).
	random-seed-id		Random seed ID number to allow replication of model runs.

1.1.3 Process overview and scheduling

Each time step allows every village to die or create a new village (see *population-growth* submodel), in a random order. Every polity is then allowed one opportunity to attack a neighbouring polity (see *battle-polities* submodel), and one subordinate village within each polity may decide whether to fragment (see *fragment* submodel) from the main polity or not. The order in which agents run each submodel is random, but all agents must complete a submodel before any agents can begin the next submodel. The time step is complete when these commands have been run by every polity. See the flow diagram in Figure 2 for a simplified illustration of these processes.

1.1.4 Submodels

There are four submodels in this model. The first is to set up the model environment (see 'Initialisation'). The next three submodels are used to allow villages in the model to: 1. attempt to create a new village; 2. attempt to attack their closest neighbour of a different polity; and 3. decide whether to rebel from their current polity respectively. The flow diagram shows a simplified sequence of events.

1.1.4.1 Submodel 1: population-growth

The *population-growth* submodel determines the population dynamics of the model. Each village will potentially die, create another village, or remain as before, with the *probability.grow* parameter determining the likelihood that any of these decisions will be made (see flow diagram of this submodel, Figure 2). Villages which die are removed from the model entirely and any subordinate villages will fragment into autonomous polities. However, in the experiments presented here, there is no likelihood of a village becoming extinct (*probability.death* = 0). This parameter was kept constant at zero because variations in population size or growth rate could be achieved through the *initial.villages* or *probability.grow* parameters, and this model remains abstract.

For the remainder of the *population-growth* submodel, each village will decide whether to create a new village or not. This decision is influenced by two factors: the availability of surrounding, unoccupied patches; and the potential resources of the new location. A patch can only be occupied by one village at a time. If all the surrounding patches that a village can 'see' are occupied by other villages, then it cannot create a new village. If there are available patches, the village will choose one of them, with preference for a patch with more resources (see Equation 1). In this model, there are only two types of patch: those with the maximum level of *land-resources* and those with less than the maximum *land-resources.* The difference between the *land-resources* of the two types of patches is determined by the *resource.difference* parameter (see Table 2). The relative difference in resources between the more and less fertile patches is of importance here because this informs the decisions made by villages. The closer the *land-resources* of a patch is to the maximum potential *land-resources* of a fertile patch, the more likely the village is to interact with that patch (to either move to or create a new village on). The absolute value of *land-resources* of each patch is therefore less relevant to the village than the relative *land-resource* of that patch to any fertile patch.

If a patch is unoccupied, a new village will only be created with a likelihood in proportion to the resources available on that patch (Equation 2). If the chosen patch is rich in resources a new village will very likely be created. If not, then a new village is less likely to appear. The relative richness of resources is scaled compared to fertile patches, which have the maximum *land-resources*. The parameter *probability.grow* therefore only represents the highest possible growth rate and does not guarantee that the population will grow at all.

Equation 1:

$$\text{Probability of choosing a more fertile patch} = \frac{\text{total resources of only the most fertile patches visible to the village}}{\text{total resources of all patches visible to the village}}$$

Equation 2:

$$\text{Probability creating a new village} = \frac{\text{resources of the selected new location}}{\text{highest number of resources that a patch can yield}}$$

When all villages in the model have run through the *population-growth* submodel, and either died, created a new village, or done nothing, then each polity will begin the *battle-polities* submodel.

1.1.4.2 Submodel 2: battle-polities

In the *battle-polities* submodel, each polity is chosen in a random order to run the submodel. When a polity is selected, it will choose a village at random belonging to that polity. That village will then attempt to find a rival village belonging to a different polity with the radius *village.range*. If a village is found, the polities of these two villages enter into conflict. The larger of the two polities is more likely (but not guaranteed) to win the conflict, as calculated by Equation 3.

Equation 3:

$$\text{Probability of winning} = \frac{\text{total resources owned by all villages within the attacking polity}}{(\text{total resources owned by all villages within the attacking polity} + \text{total resources owned by all villages of the defending polity})}$$

If the attacking polity loses, nothing further happens between the two polities. If the attacking polity wins, villages in the defeated polity must decide whether to stay in the same location and become subordinate or move to one of their neighbouring, unoccupied patches and remain in their original polity. The probability of moving is scaled by the relative costs and benefits of each option (Equation 4).

Equation 4:

$$\text{Probability of moving} = \frac{\text{total resources of only the most fertile neighbouring patches of defeated villages}}{(\text{resources of best neighbouring patches of defeated villages} + (\text{current resources of defeated villages} - \text{tribute}))}$$

Each village in the defending polity will choose one of the four adjacent patches to assess, with a preference for patches with more resources (Equation 1). If all four neighbouring patches are occupied, the village cannot move and will become subordinate to the conquering polity. Each village will also calculate the potential cost of becoming subordinate by subtracting the proportion of resources that would be demanded in tribute from the resources of their current patch. The sum of all potential resources from moving from all villages in the polity is compared to the sum of all potential resources from staying put and paying tribute to calculate the probability of moving, as shown in Equation 4. The more resources there are in the neighbouring patches chosen by all villages in the polity, the more likely it is that the polity as a whole will choose to move instead of staying and becoming subordinate, even if that choice is worse for individual villages within the polity.

If the defeated polity decides to stay in their current location, it will become part of the attacking polity and ranked subordinate to the highest-ranking (*hierarchy* = 1) village of attacking polity (see Figure 1 for a diagram of polity structure formation). If the defeated polity decides to move, all the villages of that polity will change location to the neighbouring patch they chose but will otherwise maintain the same polity identification and internal structure between villages of that polity.

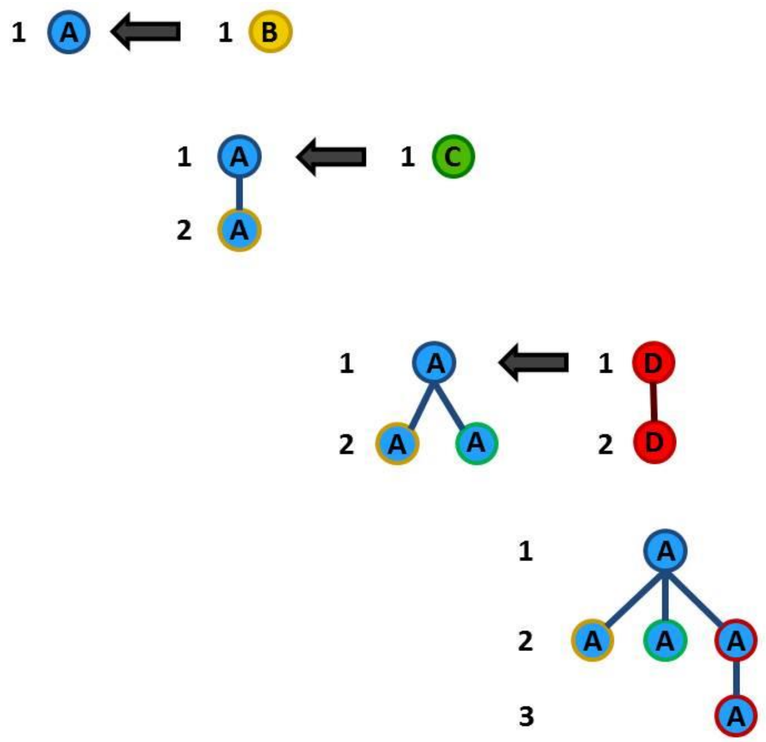
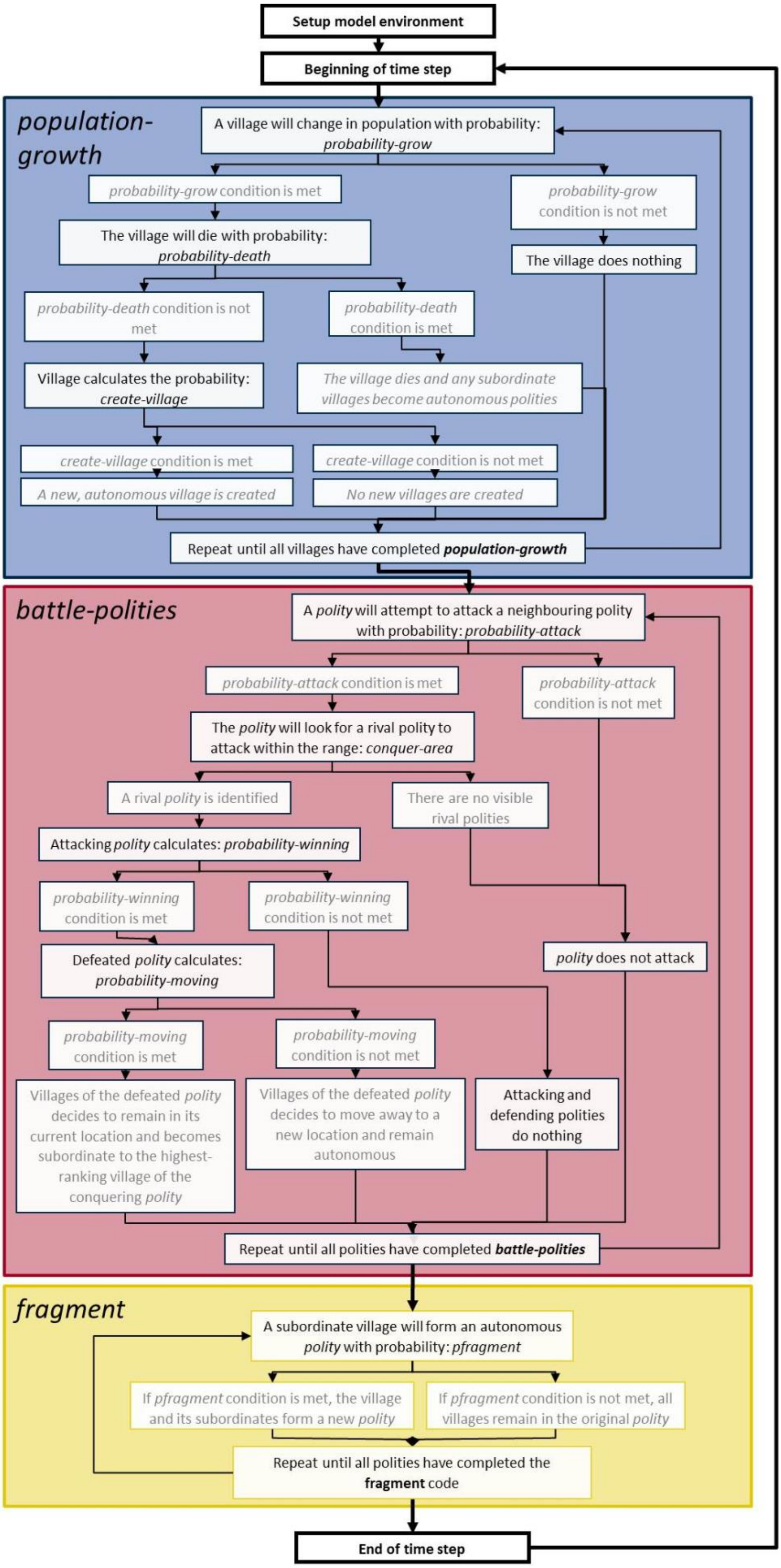


Figure 1 Diagram showing increase in polity size as rival polities are defeated and the corresponding hierarchical structure. Circles, here representing villages, with the same letter belong to the same polity. The hierarchy levels of villages within the same polity are labelled, with 1 being the highest level of hierarchy. The black arrow represents conflict between polities, with the defeated polity becoming subsumed by the victorious polity in the direction of the arrow. In the first conflict, the village labelled polity B (yellow) is defeated by polity A (blue), so polity B becomes subordinate to A and is relabelled as polity A. Polity A now consists of two villages with two levels of hierarchy. In the second conflict, another polity (C, in green) is also defeated by polity A. Polity C does not have any additional subordinate villages, so C becomes directly subordinate to the highest-ranking village in polity A. Although polity A now consists of three villages, the number of levels of hierarchy has not increased because only one village was defeated. In the third conflict, polity D (red) consists of two villages, one subordinate to the other. When polity D is defeated by polity A, both villages become subsumed into polity A below the highest ranking village of polity A, but they maintain their internal structure. Polity A therefore grows into a polity consisting of five villages with three

levels of hierarchy after these three conflicts. This is a simple example of what can happen in the model as a result of conflict.

1.1.4.3 Submodel 3: fragment

After every polity has run the *battle-polities* submodel, each polity will again be chosen at random to select one subordinate village within each polity at random to potentially rebel from the rest of the polity. Polities consisting of only one village cannot run this submodel. The probability that the chosen subordinate village will rebel is determined by the *probability.fragment* parameter (see Table 2 for details of parameter settings). If the village does not rebel, no further action is taken by any villages in the polity. If the village does rebel, it will become the highest-ranking village of a new polity, taking all its subordinate villages with it into the new polity. After one village from each polity with subordinate villages has made this decision to rebel or not, the time step is complete (see the flow diagram in Figure 2).



Setup model environment
Beginning of time step
population-growth
A village will change in population with probability: probability-grow
probability-grow condition is met
probability-grow condition is not met
The village will die with probability: probability-death
The village does nothing
probability-death condition is not met
probability-death condition is met
Village calculates the probability: create-village
The village dies and any subordinate villages become autonomous polities
create-village condition is met
create-village condition is not met
A new, autonomous village is created
No new villages are created
Repeat until all villages have completed population-growth
battle-polities
A polity will attempt to attack a neighbouring polity with probability: probability-attack
probability-attack condition is met
probability-attack condition is not met
The polity will look for a rival polity to attack within the range: conquer-area
A rival polity is identified
There are no visible rival polities
Attacking polity calculates: probability-winning
probability-winning condition is met
probability-winning condition is not met
polity does not attack
Defeated polity calculates: probability-moving
probability-moving condition is met
probability-moving condition is not met
Villages of the defeated polity decides to remain in its current location and becomes subordinate to the highest-ranking village of the conquering polity
Villages of the defeated polity decides to move away to a new location and remain autonomous
Attacking and defending polities do nothing
Repeat until all polities have completed battle-polities
fragment
A subordinate village will form an autonomous polity with probability: pfragment
If pfragment condition is met, the village and its subordinates form a new polity
If pfragment condition is not met, all villages remain in the original polity
Repeat until all polities have completed the fragment code
End of time step

Figure 2 Sequence of events in this model. The initialisation and beginning and end of the time step are in bold boxes. The population-growth *submodel is indicated by the navy coloured background box, the* battle-polities *submodel is indicated by the rose coloured background box, and the* fragment *submodel is indicated by the yellow background box. Decision-making criteria are in black text. Potential decision outcomes for villages and polities are in light grey text (see Equation 1, Equation 2, Equation 3, and Equation 4 for details on calculating the probabilities).*

1.1.5 Model parameters

The parameter values discussed in the main text are limited to values which most clearly show the relative effects of environmental, resource, and social circumscription.

Preliminary tests of the effect of environmental and resource circumscription included varying parameters for: (1) the area of fertile land (*land.width/number.fertile*), as a proxy for environmental circumscription; (2) the resource gradient between land types (*resource.difference*), as a proxy for resource circumscription; (3) the distribution of fertile patches (concentrated or random); (4) the frequency of conflict/warfare between polities (*probability.attack*); (5) the area around a village in which other villages are visible (*conquering.area*); (6) the population size in number of villages (*initial.villages*); (7) the cost of becoming subordinate (*tribute*); (8) the stability of the hierarchical structure connecting villages in the same polity (*probability.fragment*). Population size does not vary within model runs in these experiments. These tests were conducted using an earlier version of the model which does not include population growth parameters (see the NetLogo model script for this model can be seen in 'Model_One.nlogo' at: https://github.com/ajw246/Thesis_code.git). The model processes remain almost identical to those discussed in the current model. Results from varying each of these parameters are discussed in Williams (2019).

The full parameter list is shown in Table 2, and parameters are grouped together by their purpose. Parameters which could potentially affect the level of experienced social circumscription by villages in the model (that is, the degree to which any village is surrounded by other villages) include: parameters which can influence the population size at a given time step (*initial.villages, probability.grow,* and *probability.death*); and the parameter which influences how far villages can see and move (*village.range*). The parameter *village.range* covers the distance that newly-created villages can move to (*placement.distance*), the distance that a village may consider moving to if defeated (*moving.distance*), and how far away villages are willing to search to find rival villages to attack (*conquering.distance*). A village may be willing to travel further to escape defeat than to initiate conflict, but here we assume that the distance a village is willing to act within is the same for each of these three parameters. We have therefore varied these parameters in conjunction as the *village.range* parameter.

Parameters which determine the environmental and resource conditions are: the distribution of patches (concentrated or random), the area of fertile and (*land.width* and *number.fertile*), and the difference in resources between the more and less fertile patches (*resource.difference*). The remaining parameters (*tribute, probability.fragment,* and *probability.attack*) are linked to the social conditions of polities.

1.2 Design concepts

1.2.1 Basic principles

This model has been built to test the logic of Carneiro's environmental circumscription hypothesis in an abstract environment. The aim of the model is therefore to observe the impact of varying the severity of environmental conditions on polity formation over time.

1.2.2 Emergence

The size of polities (determined by the number of villages within the polity) and number of levels of hierarchy connecting those villages in the same polity are emergent phenomena in this model. The pattern of polity size increase and average level of hierarchy increase begins to emerge from the end of the first time step.

1.2.3 Adaptation

Villages will weigh up the costs and benefits of whether to become subordinate or potentially loose resources by moving to a new area if they are defeated in conflict. Villages, and polities as collections of villages, therefore adapt to the situation they find themselves in. As a side effect, polities which are larger (include more villages) are more powerful in conflict and so are more likely to defeat and subsume other polities. Larger polities therefore tend to emerge.

1.2.4 Objectives

The objective of all individual villages is to maximise their resource gain, through either conquering neighbouring villages or deciding on their most cost effective option if defeated.

1.2.5 Learning

Villages do not learn or alter their behaviour over time. They respond to the environmental and social conditions they are presented with.

1.2.6 Prediction

Villages predict the costs of becoming subordinate by subtracting the cost of tribute as a proportion of their current resources. The villages do not pay the cost if they become subordinate but will make the decision of moving or staying assuming they will have to pay the cost of tribute. Villages cannot look further ahead to assess the potential cost of being subordinate over time.

1.2.7 Sensing

Villages can locate other villages within the radius *village.range* (see Figure 3 for an illustration of this distance). Villages are also aware of the resources contained and occupancy of the four directly adjacent patches to their current location.

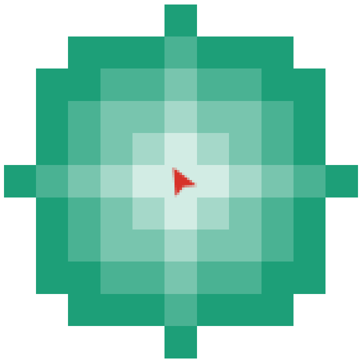


Figure 3 Diagram to show the distance around a village (red triangle) in patches that it can find other villages if village.range *= 1 (lightest teal) to* village.range *= 5 (darkest teal).*

1.2.8 Interaction

Polities will interact through conflict. In each time step, one village from each polity will find a village of a rival polity within the radius *conquering.area* to attack. All other villages in the polities of the attacking and defending villages become involved. If the defending polity is defeated, the whole polity becomes subordinate to the attacking polity, with the highest-ranking village of the defending polity ranked directly below the highest-ranking village of the attacking polity. The internal hierarchy of the defending polity remains the same within the new polity (see Figure 1 for an illustration of this process).

1.2.9 Stochasticity

Villages are initially located at random on any green patch (patches with *land.resources* = 100). One village within a polity will be chosen at random to check its surrounding environment within a radius of *village.range* for neighbouring villages of a different polity to attack. One subordinate village within a polity will also be chosen at random to decide whether to rebel from the polity or not.

1.2.10 Collectives

Villages form collectives as polities. This is emergent to the extent that the size of polities can change over time, but the total number of independent polities possible at any one time is equal to the total number of villages (*initial.villages*) because each village can form a polity of one village.

1.2.11 Observation

The level of hierarchy of each village is recorded over time. The average level of hierarchy, level of experienced social circumscription and level of experienced environmental circumscription for all villages at the end of each time step is recorded for model analysis.

Level of experienced social circumscription:

	Low level of experienced social circumscription	**Medium level of experienced social circumscription**	**High level of experienced social circumscription**
	If there are no other villages within range, the level of experienced social circumscription will be 0.	If there are some villages within range, the level of experienced circumscription will be between 0 and 1.	If the area within range around a village is completely full, then the level of experienced social circumscription will be 1.

***village.range* = 1 patch**

***village.range* = 10 patches**

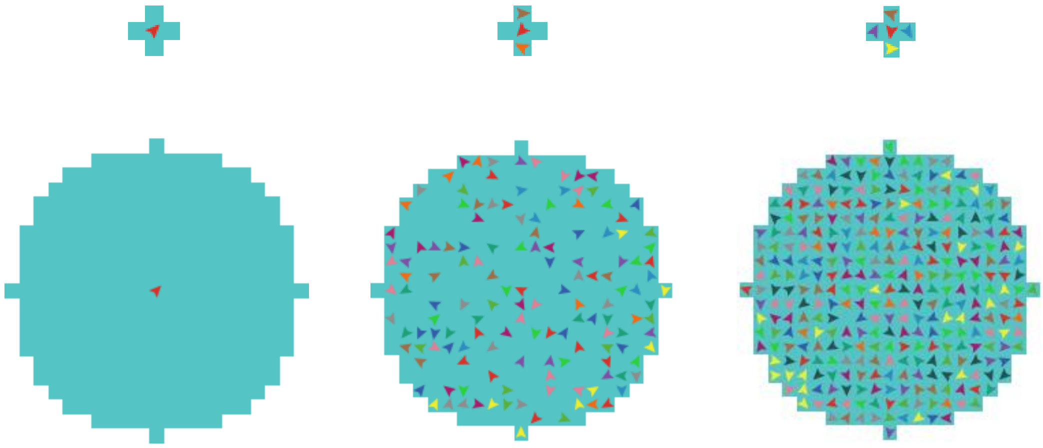


Figure 4 Examples of different levels of social circumscription, as experienced by an individual village. Each village will record the proportion of patches occupied by other villages within the distance village.range*, at each time step. The level of experienced social circumscription is the average proportion of occupied land across every village for each time step. Note: this is not the same as the total number of villages because villages may share neighbours.*

Level of experienced environmental circumscription:

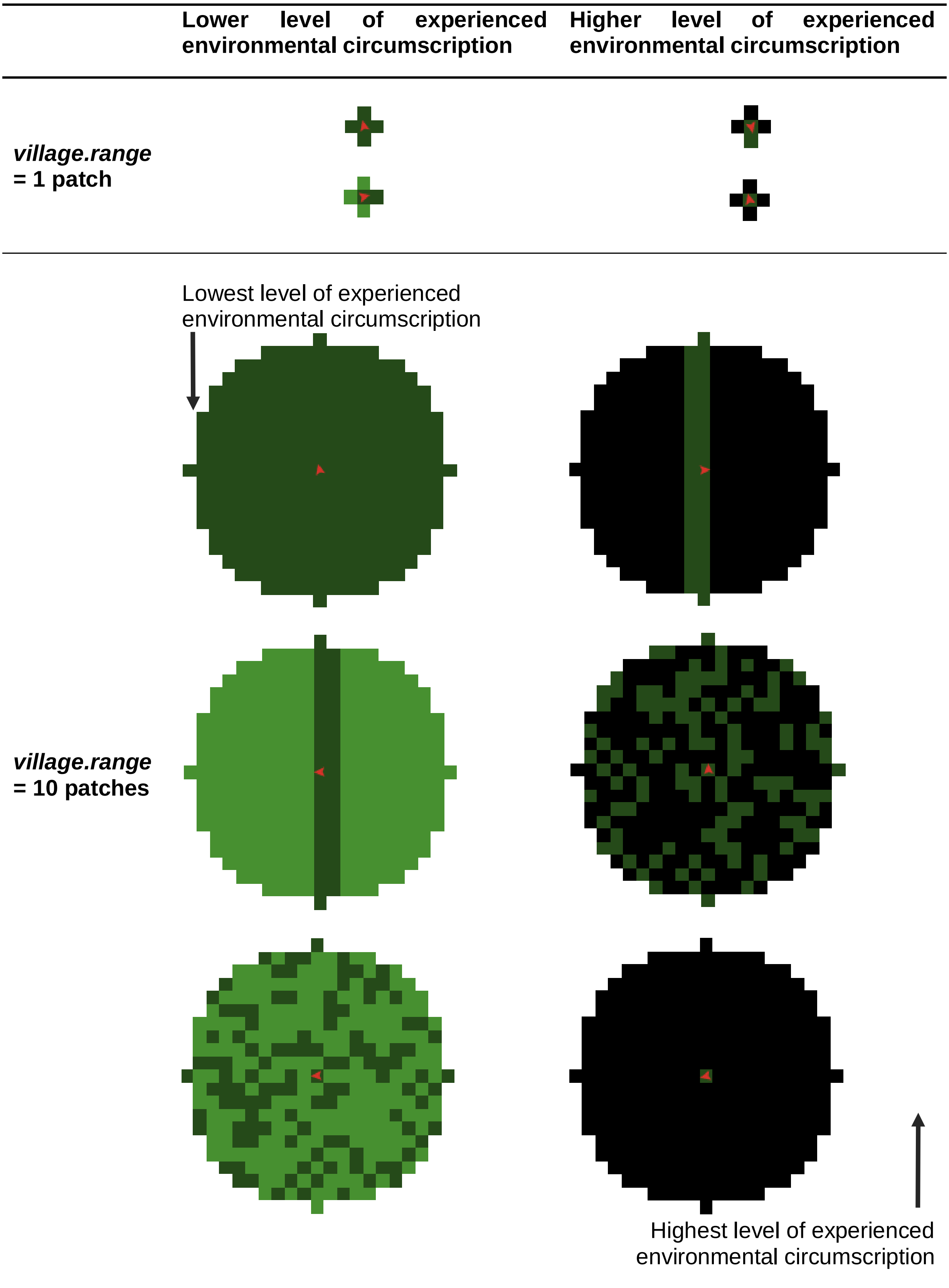


Figure 5 Examples of different levels of experienced environmental circumscription, as experienced by an individual village. Each village will calculate the sum of resources across all the patches visible within the distance village-range. *The sum of visible resources is then compared to the maximum potential resources that would be possible if each of those patches had the maximum number of* land-resources *to give a measure of experienced environmental*

circumscription. This measure combines the effect of both environmental circumscription (the number of more fertile patches) and resource circumscription (the difference in resources between the more and less fertile patches) as both affect the number of potential resources available. Note: this measure is not the same as the total resource of the whole model world because villages can share neighbouring areas. The average experienced environmental circumscription across all villages is taken for each time step. This provides a scale of experienced environmental circumscription from 0 (all surrounding patches are uniformly fertile, therefore there is no experienced environmental circumscription) to 1 (all surrounding patches have no resources, so the village is very highly environmentally circumscribed).

1.3 Initialisation

The model world is set up by dividing patches into either more or less fertile land, with the difference in resources between the two types determined by the *resource.difference* parameter. The resources owned by each patch does not change over one model run. The spatial distribution of green patches is either concentrated in a continuous band or random spread across the model world. The number of more fertile patches is varied between model runs by the *land.width* or *number.fertile* parameters, depending on the patch distribution (Figure 6). Villages are then created to populate the model world to the number set in *initial.villages*, and placed on a randomly chosen, unoccupied patch with the highest proportion of resources. Each village starts as an independent polity.

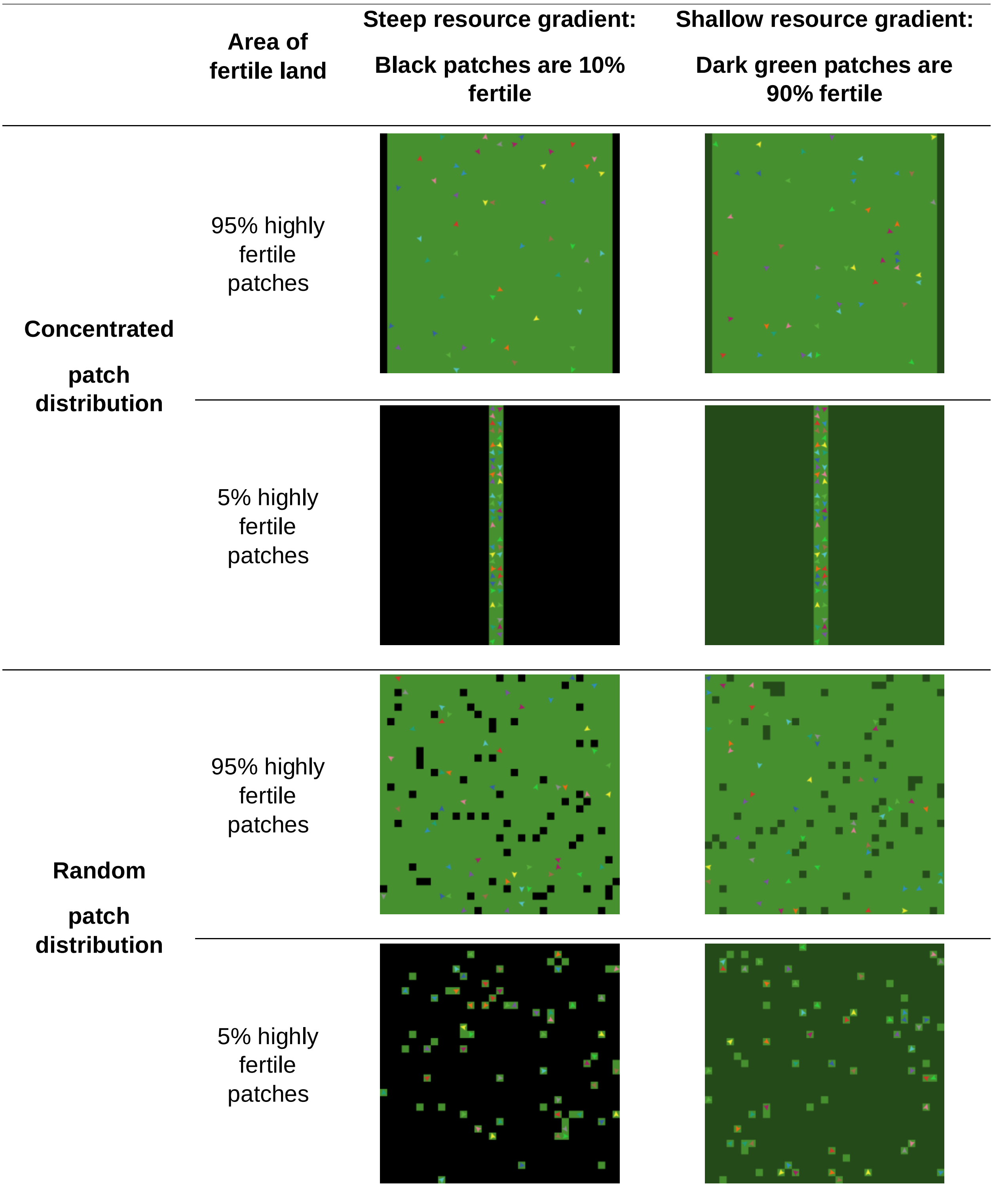


*Figure 6 Example model environments illustrating the layout of patches and villages in different conditions. The area of fertile land is varied by widening the strip of the more fertile patches in the concentrated patch environment (*land.width*), and by increasing the number of more fertile patches in the random environment (*number.fertile*). The* land-resources *of fertile patches is kept constant, but the relative resources of the less fertile patches can be varied from low (10 percent of the more fertile patches) to nearly as high as the fertile patches (90 percent of the more fertile patches), as determined by the* resource.difference *parameter. Villages (coloured triangles) are always initially located on the fertile patches, but may move to the less fertile patches.*

Concentrated patch distribution*: all patches are clustered into continuous bands such that each land type will have the same land type on at least three sides. The area of more fertile land is varied by* land.width*. High and low levels of environmental circumscription will have the equivalent number of fertile patches as set by the* number.fertile *parameter in the random patch distribution.*

Random patch distribution*: patches of either land type can be located anywhere on the grid. The number of more fertile patches is varied by* number.fertile*.*

Steep resource gradient*: Less fertile patches (black) contain 10 percent of the land-resources of more fertile patches (*resource.difference = *0.1).*

Shallow resource gradient*: Less fertile patches (dark green) contain 90 percent of the land-resources of more fertile patches (*resource.difference = *0.9).*

1.4 Input data

The model does not use input data to represent time-varying processes.