NeoCOOP - Supplementary Materials

Brandon Gower-Winter
GWRBRA001@myuct.ac.za
University of Cape Town
Cape Town, South Africa

Geoff Nitschke
gnitschke@cs.uct.ac.za
University of Cape Town
Cape Town, South Africa

ABSTRACT

This document serves to as the supplementary material for *NeoCOOP*[1]. We provide an ODD+D [5] design description (Section 1) and the parameter tuning process (Section 2) using *Optuna*[2].

KEYWORDS

Agent Based Modelling, Machine Learning, ALIFE, Social Simulations

1 ODD+D DESCRIPTION

1. Overview

1.1. Purpose

1.1.a What is the purpose of the study?
The general purpose of NeoCOOP is to be a model by which we can simulate the Paleolithic-Neolithic transitionary period that saw humanity move from largely egalitarian hunter gatherer groups to agrarian super polities typically ruled by a social elite.

1.1.b For whom is the model designed?
This model is designed primarily for Computational Archaeologists. The model will also be of interest to Computational Social Scientists interested in the modelling of complex social phenomena and the Artificial Life community in general.

1.2. Entities, State Variables and Scales

1.2.a What kinds of entities are in the model?
There are four kinds of entities within NeoCOOP:

(1) **Households**: They are the primary decision making entity in NeoCOOP (the agents). They represent a collection of occupants ruled by a patriarchal figure.
(2) **Occupants**: Occupants are contained within households. They are not decision making entities and are only present to determine household resource gathering and consumption levels.
(3) **Settlements**: Settlements represent a collection of households. They are also not decision making entities but are used by several of the model's systems for simulating / restricting agent adaptation.

The model represents its environment as a *nxm* grid-world. Each cell in the grid can technically be thought of as an entity but, their primary purpose is to store local geographical information.

1.2.b By what attributes(i.e. state variables and parameters) are these entities characterized?

Environment Cell:

(1) Soil Moisture: Amount of moisture (mm) in a given environment cell.
(2) Vegetation: Amount of vegetation (kg) in a given cell.
(3) Slope: The slope (∘) in a given cell.
(4) Is Owned: An integer value that indicates if an environmental cell is owned and by whom (-1 for not owned).
(5) Is Settlement: An integer value that indicates if an environmental cell is a settlement (-1 for not a settlement).

Household:

(1) Resources: Amount of resources an agent has.
(2) Load: The amount of (decayed over time) resources the agent has donated to other agents.
(3) Occupants: A list of occupants in the current Household.
(4) Hunger: A value $\in [0, 1]$ that denotes the agent's hunger.
(5) Satisfaction: A value $\in [0, 1]$ that denotes how happy the agent is with its living conditions.
(6) Owned Land: A list of all of the land currently owned by the agent Household.
(7) Storage Decay: A list storing the history of agent resource acquisitions.
(8) Able Workers: The number of able workers the Household has at its disposable (This value is equal to the number of occupants whose age is greater than the age_of_maturity.
(9) Peer Resource Transfer Chance: The likelihood $\in [0, 1]$ of an agent accepting a resource transfer request from a peer agent.
(10) Subordinate Resource Transfer Chance: The likelihood $\in [0, 1]$ of an agent accepting a resource transfer request from a subordinate agent.
(11) Forage Utility: The amount of utility an agent associates with the forage action.
(12) Farm Utility: The amount of utility an agent associates with the farm action.
(13) Stubbornness: Learning rate for individual adaptation.
(14) Conformity: Learning rate for generational adaptation.

Occupant / Individual:

(1) age: How many iterations the occupant has been alive for.

1.2.c What are the exogenous factors/drivers of the model?
Climate Change. Specifically an increasing / decreasing likelihood of drought over time.

1.2.d If applicable, how is space included in the model?
The environment is a grid-world made up of equally sized cells.

[1] *NeoCOOP* is an Agent-based Model implemented in Python 3 available at: https://github.com/BrandonGower-Winter/NeoCOOP
[2] *Optuna* is a Python-based optimization process. Available at: https://optuna.org/

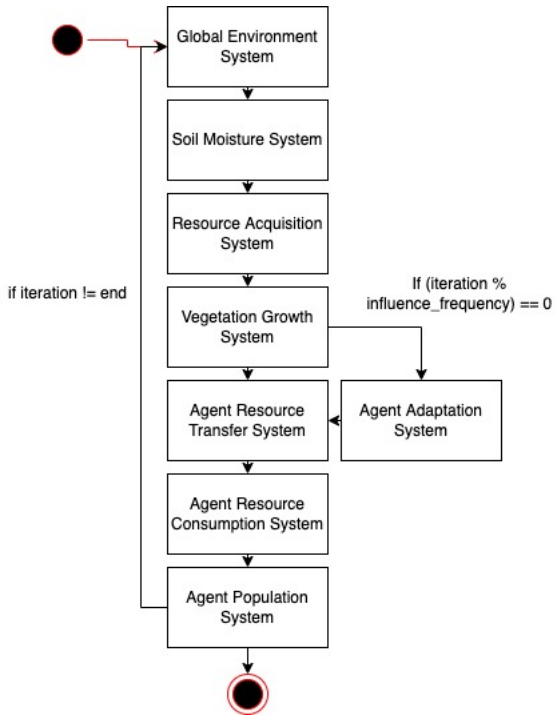

Figure 1: Execution Cycle of *NeoCOOP*

NeoCOOP supports using GIS layers to add environmental information to the environment. This includes but, is not limited to, height, slope, water and sand content data inputted into the model as images where each pixel in the image represents the value of a given attribute at that pixel coordinate in the grid-world.

1.2.e What are the temporal and spacial resolutions and extents of the model?
One iteration in *NeoCOOP* represents a single year. The soil moisture system does calculate total soil moisture (mm) on a per month basis, but it is entirely self-contained to the vegetation model systems (Global Envrionment System, Soil Moisture System and Vegetation Growth Systems).
Each grid cell is 1ha in size and the total size of the grid-world is configurable in height and width *mxn*.

1.3. Process Overview and Scheduling

1.3.a What entity does what and in what order?
The order of execution can be seen in Figure 1.

2. Design Concepts

2.1. Theoretical and Empirical Background

2.1.a Which general theories concepts, theories or hypotheses are underlying the model's design or at the level(s) of the submodel(s) (apart from the decision model)? What is the link to complexity and purpose of the model?
The model heavily relies on the vegetation model description by Xu et al. [10]. This model predicts vegetation net primary production (NPP) which is then converted into a yield for the agents to consume.

The Agent-based component of the model is also loosely based on the work of Chliaoutakis and Chalkiadakis [2, 3] and the model makes use of their self-organization scheme for simulating emergent social hierarchies.

Our Agent's resource trading preferences are probability-based. This is an extension to the typically simple cooperative - defective approach. Each agent has a probability p associated with its resource trading preferences and every time a resource trading request needs to be decided on, a random number is generated $\in [0, 1]$ and the number is less than p, the resource transfer request is granted.

2.1.b On what assumption is/are agents' decision model(s) based?
The model assumes that resource trading preferences can be simulated as a stochastic process. Additionally, it assumes that all households were 'ruled' by a single individual and that personal storage is the preferred method of resource storing (as opposed to collective resource pooling or some other hybrid approach).

2.1.c Why is a/are certain decision model(s) chosen?
Most of the model's input parameters are based on published works or publicly available data. Table 1 provides said references and where no references are made, NeoCOOP is tuned using *Optuna*.

Optuna performs multi-objective optimization maximizing total population and resources levels in the last iteration. These measures are used because a greater population level is indicative of a more successful parameter configuration and we also consider total resources because having a higher population level with more resources is a greater indicator of success than an equally large population with no resources.

2.1.d If the model/ a submodel is based on empirical data, where does that data come from?
See Table 1.

2.1.e At which level of aggregation were the data available?
It varies from source to source. Table 1 clarifies how the data was derived.

2.2. Individual Decision Making

2.2.a What are the subjects and objects of decision-making? On which level of aggregation is decision-making modelled? Are multiple levels of decision making included?
The decision making units are Households.

Every iteration, agents choose to FARM or FORAGE actions equal to the number of able workers (An able worker is an agent who

is older than or equal the *agent_of_maturity* property. Agents are restricted to choosing one action or the other, in fact, agents may choose to have some of their occupants farm, and the rest will forage.

2.2.b What is the basic rationality behind agents' decision-making? Do agents pursue an explicit objective or have other success criteria?
Explicitly, the agents perform Utility maximization. This results in the agents implicitly trying to minimize their hunger and maximizing their social status.
2.2.c How do agents make their decisions?
For resource acquisition, the agents follow the standard e-greedy approach in reinforcement learning. The agents use their hunger as ϵ to determine whether the should take a greedy action, or a random (exploratory) action. Agents will then update the utility values of each action based on the reward (food) received.

For resource transfer, the agents will look at their peer and subordinate resource transfer properties to determine if they are willing to donate resources for a given iteration. The first time an agent is asked (for both peer and sub requests), a random number $\in [0, 1]$ is generated and compared to the aforementioned resource transfer beliefs. If the random number is lower than the resource transfer beliefs, the agent will donate its resources.

For moving, the agent will look at its satisfaction and a generated random number (both $\in [0, 1]$). If the random number is greater than the satisfaction, the agent will move. Agents will first look at neighbouring settlements, if none of them look desirable (average resources of settlement < required resources of agent), the agent will make a new settlement at random location. There is an additional property called lookback_sids which prevents the agent from going back to the same settlements until it has explored additional options.

2.2.d Do the agents adapt their behaviour to changing endogenous and exogenous state variables? And if yes, how?

Yes, as outlined above agents will seek to explore alternative resource acquisition strategies when their current strategy does not work. Similarly, agents will move from one settlement to another when their overall satisfaction is low.

2.2.e Do social norms or cultural values play a role in the decision making process?

Yes, if multiple agents decide to leave a settlement in the same iteration, they will all move to the same location.

2.2.f Do spacial aspects play a role in the decision making process?

Yes, agents can only farm / forage within a specified max acquisition distance. Similarly, the distance an agent can travel when moving from one settlement to another can be controlled by the vision_square property.

2.2.g Do temporal aspects play a role in the decision making process?

Yes, agents only decide whether or not to move every yrs_per_move iterations.

2.2.h To which extent and how is uncertainty included in the agents' decision rules?

Not Applicable.

2.3. Learning
2.3.a Is individual learning included in the decision process? How do agents' change their rules over time as consequence of their experience?

Yes, agents follow a standard reinforcement learning approach to determine which action (farm or forage) to take (See submodels for more details).

2.3.b Is collective learning implemented in the model?

Yes, in the form of generational adaptation. Agents, using a genetic algorithm and a cultural algorithm to exchange information regarding the beliefs (See submodels for more details).

2.4. Individual Sensing

2.4.a What endogenous and exogenous state variables are individuals assumed to sense and consider in their decisions? Is their sensing process erroneous?

It is not erroneous and the agents don't explicitly detect any of the environments properties. What they are aware of though is their hunger, satisfaction and the perceived utility of the forage and farm actions.

2.4.b What state variables of which other individuals can an individual perceive? Is the sensing process erroneous?

Agents are aware of the average resource levels of the neighbouring settlements. They are indirectly aware of the general beliefs held by their settlement (captured in their belief space).

2.4.c What is the spatial scale of sensing?

Agents are able to sense the vegetation density and cells they own within max acquisition distance cells around them.

2.4.d Are the mechanisms by which agents obtain information modelled explicitly, or are individuals simply assumed to know these variables?
It is assumed.
2.4.e Are costs for cognition and costs for gathering information included in the model?

Not explicitly.

2.5. Individual Prediction

Agents do not make any explicit predictions.

2.6. Interaction
2.6.a Are interactions among agents and entities assumed as direct or indirect??

Cultural Influence occurs indirectly while resource transfer is direct.

2.6.b On what do the interactions depend?

Resource transfer requires that agents belong to the same settlement. Cultural Influence depends on the social status of the two 'interacting' agents.

2.6.c If the interactions involve communication, how are such communications represented?

Not Applicable.

2.6.d If a coordination network exists, how does it affect the agent behaviour? Is the structure of the network imposed or emergent?

Not Applicable.

2.7. Collectives
2.7.a Do the individuals form or belong to aggregations that affect and are affected by the individuals? Are these aggregations imposed by the modeller or do they emerge during the simulation?

Yes. As mentioned above agents are may form settlements. The simulation does not enforce settlements (except at initialization). Agents may form new settlemenmts, leave old ones or even move to other settlements every *yrs_per_move* iterations. An agent always needs to belong to a settlement (so that the adaptation submodels can work) but it is entirely possible that a simulation run may result in every agent forming their own settlement. This is equivalent to having no settlements since each hosuehold will adapt individually.

2.7.b How are collectives represented?

As noted above. A collection of one or more households makes a settlement.

2.8. Heterogeneity
2.8.a Are the agents heterogeneous? If yes, which state variables and/or processes differ between the agents?

Yes. All variables listed in Section 1.2.b.

2.8.b Are the agents heterogeneous in their decision-making? If yes, which decision models or decision objects differ between the agents?

Yes. All agent decisions are heterogeneous. This includes resource acquisition, resource transfer and relocation.

2.9. Stochasticity

2.9.a What processes (including initialisation) are modelled by assuming they are random or partly random?

All stochastic processes are pseudorandom. This is to ensure model reproducibility. A list of stochastic processes during model execution are listed below:

(1) Monthly Global Environment (rainfall, temperature and solar radiation) values.
(2) Agent explorative action selection (e-greedy).
(3) Agent farm / forage resource gathering patch selection.
(4) Household birth.
(5) Household death.
(6) Resource Transfer Requests.
(7) House Split Parent Selection (Genetic Algorithm)
(8) Household Split Mutation (Genetic Algorithm).
(9) Household Influence knowledge source selection (Cultural Algorithm)
(10) Household Move decision.

A list of stochastic processes at initialization are listed below:

(1) Settlement placement.
(2) Agent initial gene creation.
(3) Agent settlement placement.

2.10. Observation
2.10.a What data are collected from the ABM for testing, understanding and analysing it, and how and when are they collected?

Snapshots of the environment are collected every (user-defined) iterations. These snapshots capture all of the necessary aspects for, essentially, recreating the simulation run from the ground up.

All environment data is collected in csv files. All agent and settlement data is collected in JSON files. A log file of the simulation is also recorded which records the result of every stochastic process the model simulates.

2.10.b What key results, outputs or characteristics of the model are emerging from the individuals? (Emergence)

This is heavily reliant on the input parameters of the model but the key emergent properties are list below:

(1) Regardless of initial agent distribution, the agents mean resource trading beliefs (peer and subordinate) will converge to the range $\in [0.4, 0.6]$.
(2) As environmental stress is increased, agents maintain a degree of belief diversity.
(3) Subordinate Ostracization occurs as the frequency of environmental stress increases.

3. Details

3.1. Implementation Details
3.1.a How has the model been implemented?

The model was implemented in Python 3 using the *ECAgent* framework.

3.1.b Is the model accessible, and if so where?

Yes, it is publicly available at: https://github.com/BrandonGower-Winter/NeoCOOP.

3.2. Initialization

3.2.a What is the initial state of the model world, i.e. at time t = 0 of a simulation run?

Agents have been randomly allocated to settlements. Settlements have been randomly placed. All agents have zero resources and zero load. The rest of the initial conditions are based on the decoder file used to create the simulation.

3.2.b Is the initialisation always the same, or is it allowed to vary among simulations?

It varies depending on the seed used. If the same seed is used, the initialization (and model execution) will be exactly the same.

3.2.c Are the initial values chosen arbitrarily or based on data?

Randomly. If agent homogeneity is enforced, the model will not randomly assign agents to each settlement (all settlements will be given the same number of starting agents) but the settlment locations will still be random.

3.3. Input Data

3.3.a Does the model use input from external sources such as data files or other models to represent processes that change over time?

Yes. *NeoCOOP* uses what we call *decoder* files. They share the same structure as table 1 (in JSON format) and are given to the model at initialization. The model also takes in a heightmap, sandcontant map and slopemap (usually derived from the heightmap).

3.4. Submodels

3.4.a What, in detail, are the submodels that represent the processes listed in 'Process overview and scheduling'?

Agent Definition

Each agent represents a Neolithic *household* consisting of one or more *occupants*. These *occupants* are not considered entities and are only tracked for the purposes of determining *household* resource requirements and resource acquisition capabilities. The motivation for this is that typical Neolithic households were ruled by a single patriarchal figure who was responsible for making all of the family's decisions as well as managing their resources. The *NeoCOOP* ABM uses *settlements* to keep track of one or more *households*. The purpose of *settlements* is to store the coordinates of all the agents contained within that settlement.

Unlike most cooperation-based ABMs, *NeoCOOP* allows agents to make decisions based on their social status and the social status of the agents they are interacting with. In *NeoCOOP*, social status is defined as the sum of an agent's available resources and its *load* where *load* is the amount of resources the agent has donated over a period of time. To facilitate social stratification, we use the self-organization scheme described by [2] whereby a relationship type can be determined for every agent pair by comparing their social statuses. We define each of the relationship types as follows:

$$is\_acq(h_1, h_2) = h_1.settlement == h_2.settlement \tag{1}$$

$$is\_peer(h_1, h_2) = \frac{|h_2.ss - h_1.ss|}{max(h_1.ss, h_2.ss)} < L \tag{2}$$

$$is\_auth(h_1, h_2) = \frac{(h_2.ss - h_1.ss)}{max(h_1.ss, h_2.ss)} > L \tag{3}$$

$$is\_sub(h_1, h_2) = is\_auth(h_2, h_1) \tag{4}$$

Where is_acq, is_peer, is_auth, is_sub describe whether household h_2 has an acquaintance, peer, authority or subordinate relationship with household h_1 respectively. $h_n.ss$ is a household's social status. L is the $load\_difference \in [0, 1]$ input parameter which describes how much more social status an agent requires to be considered an authority over another agent. Note that in order for a peer, authority or subordinate relationship to be formed, the two households must be from the same settlement (ie: $is\_acq = true$).

Environment & Vegetation Model

NeoCOOP places agents on a *nxm* grid-world and uses a simple *vegetation model* based on [10], simulating monthly global environment properties (rainfall and temperature) and vegetation growth and decay. The environment comprises three layers, each containing a *nxm* matrix storing the grid-world's sand content (%), soil moisture (mm) and vegetation (kg). Every iteration, the global environment system generates 12 rainfall and temperature values randomly sampled from a range of predetermined values. Using the generated temperature values, the soil moisture system calculates the potential evaporation (PET) using the *Thornthwaite* equation. For each of the 12 rainfall and PET pairs, the soil moisture values of each cell are updated as described by [10]. The vegetation growth system then uses the soil moisture values to calculate vegetation growth and decay for all uninhabited cells.

Resource Acquisition, Transfer and Consumption

Household agents in *NeoCOOP* are utility-based, meaning that every agent associates each action in the model with a utility value $U_{h,t}(a)$. Every iteration, agents choose actions that, based on experience, return the greatest expected reward. *NeoCOOP* has only two actions: *FORAGE* and *FARM* which, when taken, result in an agent foraging or farming respectively. There is only one resource type in *NeoCOOP*, so the only difference between *FORAGE* and *FARM* actions is their prerequisites and quantity of resources returned. If the *FORAGE* action is chosen, the agent will look for a neighbouring cell with the greatest vegetation density and take vegetation directly from this cell equal to a predefined (*forage_consumption_rate)* amount based on the number of *able_workers* the agent has. If the selected action is a *FARM* action, the agent will choose one of it's owned farming cells and gather resources from it. If the agent does not own any farming cells, it will attempt to acquire some from unoccupied neighbouring cells.

In *NeoCOOP*, farming is intended to return a greater surplus of resources. However, it is an action that rewards a sedentary lifestyle and, in times of stress, having access to the diverse set of vegetation cells that are available when foraging can be more beneficial. After the agents have completed their, *FARM* or *FORAGE* actions, they update the utility values they associate with the *FARM-FORAGE* actions in accordance with the following equation:

$$U_{h,t+1}(a) = U_{h,t}(a) + \eta_h(R_{h,t}(a) - U_{h,t}(a)) \tag{5}$$

Where, h is the agent, a is the action, η_h is the *stubbornness* of the agent, t is the iteration, and U and R are the utility and reward functions respectively.

Once resource acquisition is complete, agents determine if they have enough resources to satisfy their needs for the iteration. If not, the agent asks its authority agents if they would be willing to give some of their excess resources to the agent as a donation. For each authority asked, a random value $\in [0, 1]$ is generated and compared to the authority agent's *subordinate_transfer* property. If the generated value is less than the *subordinate_transfer* property, the authority agent is willing to grant donations for that iteration. Whenever a donation is granted, the authority agent has its *load* property increased by the resources donated. If an agent has asked all of its authority agents for resources and it will still go hungry, it then repeats this process for its peer relationships with the donating agent using its *peer_transfer* property to determine if the donation succeeds.

If that is still not sufficient, the agent will then ask all of its subordinates for resources. Given that we are modelling Neolithic households, if a subordinate is asked to give any of its excess resources to an authority agent, it does so with 100% certainty. The *peer_transfer* and *subordinate_transfer* properties allow us to simulate different agent types that exhibit varying degrees of altruistic and selfish behaviour. When resource transfer is complete, agents consume their food and determine their *hunger* using equations 6 and 7.

$$hunger(x) = min(\frac{x.resources}{x.required\_resources}, 1.0) \tag{6}$$

$$inv\_hunger(x) = 1.0 - hunger(x) \tag{7}$$

Where, *inv_hunger* is the likelihood of an agent taking an exploratory action (random) instead of an exploitative (highest utility) action in the next iteration. This allows agents to self-adapt via exploring various avenues of resource acquisition when their preferred method is insufficient.

Population Growth, Loss and Migration Every iteration, households may birth additional occupants in accordance with the *birth_rate* and their *hunger*. If a household reaches *carrying_capacity*, the *split_household* function is called and the household is divided into two separate households. Occupants and resources are split amongst the two new households but load is not. That is, the new household signifies the arrival of a new patriarchal figure within the community and someone who must work to gain the same social status as their parent household.

Similarly, households may lose one or more occupants in accordance with the *death_rate* and their *hunger*. If a household reaches zero *able_workers*, it is removed from the simulation. Additionally, agents can migrate to another settlement or form a settlement of their own every *yrs_per_move* iterations. This decision is based on the agent's *satisfaction* which is the average *hunger* of the agent over the past *yrs_per_move* iterations. If the *satisfaction* of the agent is low, it has a higher likelihood of moving. When an agent moves, it chooses between all settlements in its vicinity or an unclaimed cell. Typically, an agent will move to the settlement with the most resources. However, if none of the neighbouring settlements have enough resources, the agent will choose to make its own settlement at a new location.

Agent Adaptation

NeoCOOP uses both individual and generational adaptation. Generational adaptation uses two evolutionary algorithms: a *Genetic Algorithm* for vertical generational adaptation and a *Cultural Algorithm* ([7]) for horizontal generational adaptation. Both use an agent's genotype containing six gene values:

(1) **Forage Utility:** Utility value of the FORAGE action.
(2) **Farm Utility:** Utility value of the FARM action.
(3) **Stubbornness:** Individual adaption learning rate.
(4) **Conformity:** Generational adaptation learning rate (genetic and cultural evolution).
(5) **Peer Transfer:** Probability an agent will accept a resource transfer request from a peer agent.
(6) **Subordinate Transfer:** Probability an agent will accept a resource transfer request from a subordinate agent.

Additionally, both the *Genetic Algorithm* (GA) and *Cultural Algorithm* (CA) make use of *influence* when determining best performing settlements (agents). *Influence* describes the probability that two settlements will interact with each other. This is done using the XTENT formula (equation 8, [3]):

$$I(s_1, s_2) = W(s_2)^{\beta} - mD(s_1, s_2) \tag{8}$$

Where, s_1 and s_2 are settlements, $I(s_1, s_2)$ is the influence of s_2 on s_1, $W(s_2)$ is the social status of s_2, $D(s_1, s_1)$ is distance from s_1 to s_2. β and m are coefficients describing required social status of one settlement to influence another. Calculating *influence* of every settlement on a given settlement, gives a probability distribution (equation 9).

$$P(s_1, s_2) = \frac{I(s_1, s_2)}{\sum_{k \in K} I(s_1, s_k)} \tag{9}$$

Where, $P(s_1, s_2)$ is the probability of settlement s_2 influencing settlement s_1 and K is the set of neighbouring settlements that have a positive *influence* value $I(s_1, s_k)$ on s_1.

The GA executes whenever the *split_household* function is called. The child agent is a combination of two parents with the first parent being the household that reached capacity and the second parent gotten via *roulette wheel* selection. This selection uses the social status of other agents within the same settlement of the first parent and from other settlements that have enough influence. The offspring agent is produced using *Uniform crossover* with *Gaussian mutation* for genes 1-4 and random mutation for genes 5 and 6. The CA only executes every *influence_frequency* iterations and Knowledge Sources [8] are used to diversify how agents are influenced. These are as follows:

(1) **Normative:** Influence on an agent's beliefs from its given settlement.
(2) **Spatial:** Influence on an agent's beliefs from another settlement.
(3) **Domain:** Equivalent to GA mutation function, where domain influence mutates one of the agent's beliefs.

Agents are influenced in accordance with the *influence_rate*. If an agent is selected for influencing, a roulette wheel is spun to determine from which knowledge source the influence will come from. Influence from the Domain knowledge source occurs at a rate defined by the *mutation_rate* parameter. Influence from the Normative and Spatial knowledge sources occur with varying probability defined by the following equations 10 and 11.

$$N(s_h, s_i) = max(\frac{s_h.ss}{s_i.ss}, 1.0) \tag{10}$$

$$S(s_h, s_i) = 1 - N(s_h, s_i) \tag{11}$$

Where, N and S are the probability of choosing the *normative* and *spatial* knowledge sources respectively, s_h is the settlement of the agent being influenced, s_i is the settlement that would influence agent h if the spatial knowledge source is selected. s_i is determined by performing roulette wheel selection on all neighbouring settlements with a positive *influence* on settlement s_h. Roulette wheel weights are determined by the values returned by Equation 9.

Each settlement's beliefs are represented by data containers called *Belief Spaces* B_s. Belief Spaces have the same structure as the agent genotype with each property calculated using a weighted average of the corresponding property of all agents within that settlement. The weight an agent contributes to the belief space is determined using its social status relative to its acquaintances. If an agent is influenced by the *normative* knowledge source, the belief space that influences it is the belief space of the settlement the agent belongs to B_{s_h}. If the agent is influenced by the *spatial* knowledge source, the belief space that will influence the agent is the belief space of the settlement selected during roulette wheel selection (B_{s_i}).

Once a belief space has been determined, each of agent h's properties are influenced as follows:

$$G_h(p) = G_h(p) + \sigma_h B_s(p) \tag{12}$$

Where, p is the agent property (genes 1-6), G is the agent's genotype, σ_h is the *conformity* of the agent and B is the selected belief space (either B_{s_h} or B_{s_i}). Figure 1 presents an overview of the *NeoCOOP* execution cycle.

3.4.b What are the model parameters, their dimensions and reference values?
See Table 1
3.4.c How were the submodels designed or chosen, and how were they parameterised and then tested?
See 3.4.a for descriptions. Parameter tuning the model involved using values derived from other works and, where no parameters could be be found, we performed multi-objective optimization using *Optuna*. The optimization process ran for 119 simulation runs and final input parameters for our model can be seen in Table 1. A report of the optimization process is in this document.

2 OPTUNA

NeoCOOP is parameter tuned using multi-objective optimization in *Optuna*. The process sought to maximize a simulation run's total population and total resources. The motivation for this is because a higher population level is indicative of a more resilient agent and because resources are essential to their survival, agents who are able to maintain a higher degree of surplus resources are more successful that agents that have not. Note that we do not care about the distribution of the resources per agent and only consider mean surplus resources across all agents. This is to ensure that we do not favour one type of organizational scheme over another (Authoritarian vs. Egalitarian)

The optimization process was intended to run for 150 simulations (31 of them did not complete because of a power outage) so the final results of the 119 that did finish are included below: (This can all be replicated using *Optuna.py*):

```
Pareto Optimal Solutions:
[{
    'influence_frequency': 6,
    'influence_rate': 0.0516647,
    'mutation_rate': 0.130285,
    'learning_rate_lower': 0.200033,
    'learning_rate_upper': 0.995248
}.
{
    'influence_frequency': 9,
    'influence_rate': 0.0573575,
    'mutation_rate': 0.0772241,
    'learning_rate_lower': 0.0683736,
    'learning_rate_upper': 0.125681
},
{
    'influence_frequency': 15,
    'influence_rate': 0.195384,
    'mutation_rate': 0.110884,
    'learning_rate_lower': 0.00644802,
    'learning_rate_upper': 0.856429
},
{
    'influence_frequency': 16,
    'influence_rate': 0.179273,
    'mutation_rate': 0.0444818,
    'learning_rate_lower': 0.0820841,
    'learning_rate_upper': 0.763823
},
{
    'influence_frequency': 16,
    'influence_rate': 0.111578,
    'mutation_rate': 0.0978724,
    'learning_rate_lower': 0.17061,
    'learning_rate_upper': 0.786016
},
{
    'influence_frequency': 18,
    'influence_rate': 0.111578,
```

NeoCOOP		
Property	Value	Reference
Model Parameters		
Iterations	10 000	
map_height	100	
map_width	100	
offset_x	0	
offset_y	0	
max_height	1400m	
min_height	0m	
cell_dimensions	100m	This makes the area of a single cell = 1ha
Global Environment System		
Priority	10	
start_rainfall	[32,42]mm	Midpoint of Semi-Arid yearly rainfall [4]
end_rainfall	[0,10]mm	This range represents a range of drought rain conditions. Based on [9].
start_temperature	[13,27]	Arbitrary range to minimize Temperature penalties.
end_temperature	[13,27]	Same as above.
start_solar	[18.67, 21.78]	Taken from [6]
end_solar	[18.67, 21.78]	Same as above.
soil_depth	1000mm	
interpolator_range	10000	Note: Must be equal to the number of iterations.
temperature_interpolator	linear	
rainfall_interpolator	cosine	Set the frequency equal to the environmental stress frequency.
solar_interpolator	linear	
Soil Moisture System		
Priority	9	These values are arbitrary and represent a generic semi-arid climate.
L	9.5 hrs	
N	30 days	
I	86.5 $^{\circ}C$	
Vegetation Growth System		
Priority	7	
init_pop	1050kg	Derived from simulation under perfect conditions.
ideal_moisture	2800mm	[1]
Agent Resource Acquisition System		
Priority	8	
farmers_per_patch	1	
max_acquisition_distance	10	
moisture_consumption_rate	2280	
farming_production_rate	400kg	
foraging_production_rate	150kg	
forage_production_multiplier	2	Multiplied by foraging to get 300kg
storage_yrs	3	
Agent IE Adaptation System		
Priority	7	
influence_rate	0.1	Parameter Tuned
influence_type	AVG	
persist_belief_space	true	
frequency	15	Parameter Tuned
Agent Resource Transfer System		
Priority	6	
load_decay	0.8	

Table 1: A comprehensive list of *NeoCOOPs* model properties.

NeoCOOP		
Property	Value	Reference
Model Parameters		
Agent Resource Consumption System		
Priority	5	
Agent Population System		
Priority	4	
birth_rate	0.15%	
death_rate	0.10%	
yrs_per_move	5 iterations	. [2]
init_settlements	10	
cell_capacity	100	
Agents		
number	100	
age_of_maturity	10	
consumption_rate	250kg	[2]
child_factor	0.5	Children eat less than adults, this takes that into account.
init_occupants	5	
init_age_range	10<	
vision_square	10000	So agents can see whole map.
move_lookback	1	
load_difference	0.6	[2]
init_preference	250	An approximation of the actual max reward for foraging.
learning_rate_range	[0.15, 0.7]	Parameter Tuned
conformity_range	[0.15, 0.7]	Parameter Tuned
mutation_rate	0.1	Parameter Tuned
b	1.5	[2]
m	0.005	[2]

Table 2: A comprehensive list of *NeoCOOPs* model properties.

```
    'mutation_rate': 0.0978724,
    'learning_rate_lower': 0.334526,
    'learning_rate_upper': 0.760217
}]

Average Results:
{
    'influence_frequency': 13.333333333333334,
    'influence_rate': 0.11780586666666665,
    'mutation_rate': 0.09310328333333333,
    'learning_rate_lower': 0.14367912,
    'learning_rate_upper': 0.714569
}
```

For simplicity, each value was rounded to nearest appropriate number:

```
Final Results:
{
    'influence_frequency': 15,
    'influence_rate': 0.1,
    'mutation_rate': 0.1,
    'learning_rate_lower': 0.15,
    'learning_rate_upper': 0.70
}
```

REFERENCES

[1] NR Bhat, VS Lekha, MK Suleiman, B Thomas, SI Ali, P George, and L Al-Mulla. 2012. Estimation of water requirements for young date palms under arid climatic conditions of Kuwait. *World Journal of Agricultural Sciences* 8, 5 (2012), 448–452.

[2] Angelos Chliaoutakis and Georgios Chalkiadakis. 2016. Agent-based modeling of ancient societies and their organization structure. *Autonomous Agents and Multi-Agent Systems* 30, 6 (2016), 1072–1116.

[3] Angelos Chliaoutakis, Georgios Chalkiadakis, et al. 2020. An Agent-Based Model for Simulating Inter-Settlement Trade in Past Societies. *Journal of Artificial Societies and Social Simulation* 23, 3 (2020), 1–10.

[4] Milica Kašanin-Grubin, Francesca Vergari, Francesco Troiani, and Marta Della Seta. 2018. The role of lithology: Parent material controls on badland development. In *Badlands Dynamics in a Context of Global Change*. Elsevier, 61–109.

[5] Birgit Müller, Friedrich Angermueller, Romina Drees, Gunnar Dressler, Jürgen Groeneveld, Christian Klassert, Maja Schlüter, Jule Schulze, Hanna Weise, and Nina Schwarz. 2012. Describing Human Decisions in Agent-Based Social-Ecological Models-ODD+ D an Extension of the ODD Protocol. *Available at SSRN 2044736* (2012).

[6] MA Omran. 2000. Analysis of solar radiation over Egypt. *Theoretical and applied climatology* 67, 3 (2000), 225–240.

[7] Robert G Reynolds. 1994. An introduction to cultural algorithms. In *Proceedings of the third annual conference on evolutionary programming*. World Scientific, 131–139.

[8] Robert G Reynolds and Bin Peng. 2004. Cultural Algorithms: Modeling of How Cultures Learn to Solve Problems. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*. 166–172.

[9] L Shanan, NH Tadmor, M Evenari, and P Reiniger. 1970. Runoff Farming in the Desert. III. Microcatchments for Improvement of Desert Range 1. *Agronomy Journal* 62, 4 (1970), 445–449.

[10] Duanyang Xu, Alin Song, Hefeng Tong, Hongyan Ren, Yunfeng Hu, and Quanqin Shao. 2016. A spatial system dynamic model for regional desertification simulation–A case study of Ordos, China. *Environmental Modelling & Software*

83 (2016), 179–192.