Model description

1 Purpose

Community-based natural resource management (CBNRM) has been suggested as a method of resource conservation that empowers traditionally marginalized communities. This ABM hopes to contribute to the growing body of ABMs simulating CBNRM to help improve realism of CBNRM ABMs. The model builds on an existing model of community forest management first designed by Bravo [1] and then expanded upon by Vallino [3, 4]. It is implemented in NetLogo [5].

2 Agents and variables

The model is set in a forest environment and is structured as a 50×50 grid, where each patch is a section of the forest described by the traits given in Table 1. The model contains two classes of agents. Logger agents represent loggers in the forest. Logger agent traits are given in Table 2. There is one Institution agent which represents a community management system. The Institution agent traits are given in Table 3. Global state variables, which guide the overall evolution of agents and the environment, are given in Table 4. There are no spatial or temporal units for this model.

3 Model rules

The simulation is broken into periods, where each period consists of 10 ticks. At each tick, Loggers start by moving to a new patch within their *neighborhood* and decreasing their *payoff* by *cost*. This represents the Loggers' cost of living. The Loggers then decide if they should log the forest. Loggers' decisions as to whether they should log a patch are dictated by the *current-institution*, an indication of the CBNRM rules. The *current-institution* starts at 0, indicating that any living patch may be logged at the beginning of the simulation, but changes as the simulation progresses. Loggers go to a random patch within their *neighborhood* with *trees* > *current-institution* and log the patch. When logging a patch, the Logger's *payoff* is increased by the *trees* on the patch, and the patch's *trees* is set to zero (i.e. the Logger cuts down all *trees* on that patch). If there are no patches with *trees* > *current-institution* within their *neighborhood*, the

Table 1: Patch traits

Variable	Description	Type	Value(s)
pxcor, pycor	The x, y-coordinates of a patch, indicating its location on the environment grid.	static	$\{0, 1, \ldots, 50\}$
trees	The tree biomass on the patch. Living patches contain *trees* > 0 and empty patches contain *trees* $= 0$. At the beginning of the simulation, *trees* $\sim U(\frac{1}{2}b_{max}, b_{max})$.	dynamic	$[0, b_{max}]$

Table 2: Logger traits

Variable	Description	Type	Value(s)
xcor, ycor	The x and y-coordinates indicating the patch the Logger is on. The location of each Logger is randomly selected at the beginning of the simulation.	dynamic	$\{0, 1, ..., 50\}$
reference-trees	The fraction of initial tree biomass the Logger believes should be conserved in the forest environment. This value represents how "environmentally-minded" the Logger is. At the beginning of the simulation, for each Logger this value is drawn randomly from a normal distribution with mean 0.5 and standard deviation 0.25.	dynamic*	$[0, 1]$
minimal-cut	The minimal level of tree biomass the Logger believes a patch must contain in order to be logged. Larger *minimal-cut* indicates the Logger is less likely to log the forest. At the beginning of the simulation *minimal-cut*$= 0$ for all Loggers.	dynamic	$[0, b_{max}]$
payoff	How much a Logger earns (or loses) during a period of 10 ticks.	dynamic	$[-10\ cost, 10(b_{max} - cost)]$
old-payoff	Final payoff from the previous period.	dynamic	$[-10\ cost, 10(b_{max} - cost)]$
payoff-satisfaction	Indicates whether the Logger is happy with the state of the forest. This value is set to 1 at the beginning of the simulation (indicating they are content with the state of the forest).	dynamic	{0, 1}
neighborhood	All patches within a 5x5 square centered at the location of the Logger.	dynamic	N/A
prob-cheat (p_c)	The probability a specific Logger will cheat. All Loggers start with p_c = *initial-prob-cheat*.	dynamic	[0,1]

* While this value is dynamic, it rarely changes; only one Logger adjusts their *reference-trees* each period (see rule descriptions).

Table 3: Institution traits

Variable	Description	Type	Value(s)
tolerance-threshold	The value determining when Loggers become unsatisfied with the current rules of the Institution. For our analysis, *tolerance-threshold* was set to “high” for all simulation runs.	static	$\frac{2}{3}b_{max}$
current-institution	The minimum level of tree biomass that a patch must have in order for it to be logged. This essentially establishes the rules for when trees can be logged in the CMS. This value is set to 0 at the beginning of the simulation and is subsequently adjusted to follow the mean of all Logger’s *minimal-cuts*.	dynamic	$[0, b_{max}]$
unsatisfied	The number of Loggers unhappy with the *current-institution*. Unsatisfied Loggers have *payoff-satisfaction* = 0 or $\lvert$*minimal-cut* − *current-institution*$\rvert$ > *tolerance-threshold* .	dynamic	[0, *initial-loggers*]
monitoring-level	The probability Loggers will be caught cheating (as a percentage).	static	[0,100]
sanction-level	The effectiveness of sanctions used to discourage Loggers from cheating.	static	[0,1]

Table 4: Global variables

Variable	Description	Type	Value(s)
max-tree-growth (b_{max})	The maximum possible level of biomass on each patch. This establishes a carrying-capacity for the forest.	static	20*
growth-rate	The amount of biomass increase per tick on non-empty patches.	static	0.5
cost	Income needed for sustenance. There is a fixed *cost* agents must pay each round.	static	5*
growth-prob	The probability an empty patch will grow back when all surrounding patches are alive.	static	0.05
initial-loggers	The number of Logger agents at the beginning of the simulation.	static	100*
reference-threshold	The “environmentalism level” of the community. Loggers’ *reference-trees* are chosen from a normal distribution with mean *reference-threshold* and standard deviation 0.25.	static	0.5*
initial-prob-cheat	The initial probability of cheating for all Loggers.	static	[0,1]

* Base parameters. These are manipulated during sensitivity analysis.

Table 5: Definition of *payoff-satisfaction* for the M & S model. Loggers' *payoff-satisfaction* is updated at the end of each period.

Current *payoff-satisfaction*	$d =$ *old-payoff* $-$ *payoff*	New *payoff-satisfaction*
1	$d < 0$	0 with probability q
0	$d < 0$	0
x	$d > 0$	1
x	$d = 0$	x

Logger moves to a random patch within their *neighborhood*. If the patch is not empty, they decide if they should cheat.

Loggers can cheat when they are unsatisfied. Cheating means that Loggers cut down the *trees* on the patch they occupy even if *trees* $<$ *current-institution*. Each Logger starts with $p_c =$ *initial-prob-cheat*, and this value is updated independently for each Logger as the simulation progresses. Unsatisfied Loggers must weigh the consequences of cheating versus not cheating to determine if they should follow Institution rules. We assume the probability a Logger will cheat depends on the Logger's perception of both how likely they are to be caught and the repercussions if they are caught. In other words:

1. Loggers who are caught cheating are less likely to cheat again.
2. Loggers who are not caught cheating are more likely to cheat again.
3. If sanctioning is more effective, the above effects will be greater.

If a Logger chooses to cheat, the *monitoring-level* parameter determines the probability the Logger is caught. When a Logger cheats, p_c is updated by the following equation:

$$\Delta p_c = \begin{cases} -p_c \times \textit{sanction-level} & \text{if caught} \\ (1 - p_c) \times \textit{sanction-level} & \text{if not caught} \end{cases}. \tag{1}$$

The forest also grows as the simulation progresses. All patches with *trees* > 0 are considered "alive". At each tick, the *trees* on all living patches with *trees* $< b_{max}$ increases by *growth-rate*. Empty patches grow *trees* with probability

$$p = \textit{growth-prob} \times \frac{N+1}{9},$$

where N is the number of adjacent or diagonally adjacent patches that are alive. This represents trees propagating from adjacent patches to the empty patch. If an empty patch grows trees, we set *trees* $= 1$ for that patch.

When a period ends (i.e. after every 10 ticks), the Loggers update their satisfaction with the Institution. A Logger's *payoff-satisfaction* depends upon the Logger's *payoff-satisfaction* from the previous round. At the end of a period, all Loggers with *payoff* $>$ *old-payoff* are satisfied. Loggers with *payoff* $=$ *old-payoff* maintain the same *payoff-satisfaction* as in the previous period. If *payoff* $<$ *old-payoff*, Loggers who were unsatisfied with their *payoff* remain unsatisfied, and Loggers who were satisfied become unsatisfied with probability

$$q = \frac{\textit{payoff} - \textit{old-payoff}}{|\textit{payoff}| + |\textit{old-payoff}|}. \tag{2}$$

These rules are outlined in Table 5.

Loggers with *payoff-satisfaction* $= 0$ then adjust their *minimal-cut*. The adjustment depends upon the number of *Living-Patches* in the forest. When *Living-Patches* $<$ *reference-trees*, their *minimal-cut* increases by $X \sim U(0, 9)$, and when *Living-Patches* $>$ *reference-trees* their *minimal-cut* decreases by $X \sim U(0, 9)$. If the amount of forest remaining is less than the amount the Logger believes is appropriate, the Logger attributes the decreased payoff to depletion of the forest, causing them to become more environmentally-minded and increase their *minimal-cut*. In contrast, if the amount of forest remaining is more than the

Table 6: Parameter values used during sensitivity analysis.

Parameter	Base value	Tested values
cost	5	{0, 2, ... 20}
max-tree-growth	20	{5, 10, ... 30}
reference-threshold	0.5	{0.1, 0.2, ... 1}
initial-loggers	100	{60, 80, ... 200}
monitoring-level	50	{0, 10, ... 100}
sanction-level	0.5	{0, 0.1, ... 1}
initial-prob-cheat	0.5	{0, 0.1, ... 1}

amount the Logger believes is appropriate, the Logger attributes the decreased payoff to not logging enough of the forest, and they decrease their *minimal-cut*.

The *current-institution* is also updated at the end of each period if

$$\textit{unsatisfied} > \frac{2}{3}\ \textit{initial-loggers}.$$

Loggers become unsatisfied when their *payoff-satisfaction* $= 0$ or $|\textit{minimal-cut} - \textit{current-institution}| > \textit{tolerance-threshold}$. When $\frac{2}{3}$ of the Loggers are unsatisfied with the Institution, the *current-institution* is set to the mean of all Loggers' *minimal cuts*. Since *minimal-cut* is a Logger's belief about how much biomass there should be on a patch before the patch can be logged, the mean *minimal-cut* of the community represents a compromise among community members regarding how much biomass should be present for a patch to be logged. This assumes that all Loggers have equal weight in the decision-making process.

The end of each period also allows for a "selection process" among the Loggers. The Logger with the lowest *payoff* is replaced by a copy of the Logger with the highest *payoff*. The new Logger is placed on a random patch and their *minimal-cut* is set to zero. The new Logger now has the same *reference-trees* and *payoff* as the most successful Logger. This represents unsuccessful Loggers adopting the behavior of successful Loggers.

At the end of the period, for each Logger, *old-payoff* is set to the *payoff* of the most recent period, and *payoff* is reset to zero. For more justification of the rules and setup of the model, see Bravo 2011 [1] and Vallino 2014 [3].

4 Sensitivity analysis

A sensitivity analysis and analysis of model emergent behaviors is found in Lapp 2020 [2]. Variable values used during analysis are given in Table 6. Each parameter set was run 50 times and allowed to continue for 2,000 ticks, where they reached a steady-state.

References

[1] Giangiacomo Bravo. Agents' beliefs and the evolution of institutions for common-pool resource management. *Rationality and Society*, 23(1):117–152, February 2011.

[2] Maya Lapp. Modeling community resource management: An agent-based approach [unpublished bachelor's thesis]. March 2020. The College of Wooster.

[3] Elena Vallino. The Tragedy of the Park: an Agent-based Model of Endogenous and Exogenous Institutions for Forest Management. *Ecology and Society*, 19(1), March 2014.

[4] Elena Vallino. The Tragedy of the Park: an Agent-based Model of Endogenous and Exogenous Institutions for Forest Management. *CoMSES Computational Model Library*, April 2018. Retrieve from: https://www.comses.net/codebases/3004/releases/1.2.0/.

[5] U Wilensky. NetLogo. *Center for Connected Learning and Computer-Based Modeling*, Northwestern University, Evanston, IL., 1999.