2020

Modeling Community Resource Management: an Agent-Based Approach

Maya M. Lapp
The College of Wooster, mlapp20@wooster.edu

Modeling community resource management: an agent-based approach

Independent Study Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Bachelor of Arts in the Department of Mathematics and Computer Science at The College of Wooster

by
Maya M Lapp

The College of Wooster
2020

Advised by:

Dr. Colby Long

Abstract

As the human population continues increasing rapidly and climate change accelerates, resource depletion is becoming an international problem. Community-based natural resource management (CBNRM) has been suggested as a method to conserve resources while simultaneously empowering traditionally marginalized communities. Because classical equation-based modeling methods fail to capture the complexity of CBNRM, Agent-Based Modeling (ABM) has emerged as a primary method of modeling these systems. In this investigation, we conduct a sensitivity analysis and thorough evaluation of an existing ABM of community forest management. We then modify the original model by providing a new enforcement mechanism that improves the validity of both the conceptual foundation and emergent behaviors of the model. We finish by conducting a second sensitivity analysis and evaluation of the new model. While this model remains imperfect, we hope future work will be able to use the model to build a greater understanding of CBNRM and apply this understanding to help propagate successful CBNRM systems around the world.

Acknowledgements

I would like to thank my advisor, Dr. Long, for being incredibly patient with me and confident in me throughout the year. His guidance has made this an enjoyable experience, rather than a stressful one (.... for the most part). Thank you to the people of Thailand and my instructors at ISDSI for introducing me to the field of community resource management and inspiring my desire to research this topic. To my academic advisor, Dr. Pasteur, and the rest of the Wooster Math and CS Department, thank you for cultivating my love of math, while encouraging me to explore other areas of interest. Thank you to Gillian from the Writing Center her patience in answering all of my questions (including far too many about semicolons). Last, but certainly not least, thank you to my family and friends for supporting me throughout my time at Wooster and helping me de-stress and have fun when I need it most.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human prosperity relies on natural resources such as forests, clean water, and fertile land. As the human population increases and climate change accelerates, resource depletion is becoming a worldwide problem. During the mid-1900s, researchers such as Garrett Hardin assumed that humanity's selfish nature implies government regulations or resource privatization are the only ways to prevent resource depletion. More recently, the field of community-based natural resource management (CBNRM) has challenged Hardin's conclusions. CBNRM researchers, led by economist Elinor Ostrom, argue that community management systems (CMSs) can be more effective than outside regulation, while simultaneously empowering the local communities that would be marginalized by government intervention and privatization.

Unfortunately, while CBNRM offers a promising solution to resource depletion, the very nature of CBNRM makes it difficult to implement on a large scale. CBNRM focuses on enabling individual communities to form regulations that are compatible with their community and ecosystem. Since each community and ecosystem is unique, a CMS that is successful in one community may be ineffective elsewhere. Therefore, while there have been many case-studies of CBNRM, researchers have struggled to generalize findings and identify overarching themes of what makes a CMS successful and how this success can be propagated to other communities.

Additionally, CBNRM is a social-ecological system (SES), meaning it incorporates interactions both within human communities and between humans and the natural environment. SESs are complex and often non-linear systems that are difficult to describe using classical equation-based modeling. Hence, while statistical analysis has helped identify trends in CBNRM, modeling CBNRM using traditional methods has largely failed to capture the complexity of these systems. This has made it difficult to understand not only what factors influence the success or failure of a CMS, but also why these influences occur and how we can respond to them.

Researchers have begun exploring Agent-Based Modeling (ABM) as an alternative method of understanding CBNRM. ABM is a method of computational simulation that models systems of agents that interact with one another and their environment through a set of defined rules. By modeling individual agents who follow probabalistic rules, ABMs allow for heterogeneous populations, stochastic interactions, and complex, non-linear systems - exactly the kinds of systems that challenge us when modeling CBNRM.

As ABM for CBNRM is a growing field, there remains debate over the most effective way to employ ABMs to investigate CBNRM. While some ABMs have been used to suggest management policies for individual CMSs, most have focused on learning more about the dynamics underlying CMSs. Understanding these dynamics can help us understand key ecological and social factors that tip a CMS toward success or failure. While these models have begun providing interesting insight into CBNRM, this remains an emerging field; hence, many of these models contain unjustified assumptions, are not well validated, and could benefit from additional sensitivity analysis.

In this investigation, we first provide an introduction to Agent-Based Modeling (Chapter 2) and natural resource management (Chapter 3). We then introduce a community forest ABM by Elena Vallino [40] (Chapter 4). We conduct a sensitivity analysis of this model to learn how model parameters impact model outputs. During this process, we identify several implementation errors in the model code and discover that the model's method of enforcing CBNRM rules is not conceptually compatible with real-world CMSs (Chapter 5). We therefore present a new model that fixes implementation errors and modifies the enforcement mechanism to better represent reality (Chapter 6). We then conduct a second sensitivity analysis and analyze new emergent behaviors (Chapter 7). We finish with a conclusion and suggestions for future work (Chapter 8).

Chapter 2

Agent-Based Modeling

Agent-based modeling (ABM) is a method of computational simulation that models systems of agents that interact with one another and their environment through a set of defined rules. In contrast to traditional modeling approaches that focus solely on replicating the macro-level behavior of a system, ABMs are constructed based on micro-scale interactions between agents in hopes of observing emergent behaviors. **Emergent behaviors** are collective behaviors and patterns that appear, often through unobvious means, from the accumulation of smaller interactions [27]. Population dynamics are a classic example of an emergent behavior.

To better understanding ABM methods, in this section we compile information from Borrill and Tesfatsion [5], Crooks and Heppenstall [13], Doran [15], and Macal and North [26] to describe the agents, environments, and rules used to develop ABMs, as well as discuss the advantages and disadvantages of this modeling method. We also provide an example ABM to illustrate these ABM characteristics.

2.1 Agents

While there is no clear-cut definition of what an “agent” is, there are several defining characteristics that are common among agents in most ABMs [13, 26].

1. **Agents are autonomous entities.** While agents interact with each other and their environment, each agent can exist and act in isolation.

2. **Agents have states and behaviors.** Each agent has a set of parameters defining traits of the agent. The values of these parameters make up the state of the agent. The parameters are defined at the beginning of the simulation, but can change as the simulation progresses. The agent’s behaviors are guided by these parameters. For example, an agent may have a parameter that dictates how far the agent moves in a single time-step. Because each agent can have a distinct state, we say the agents can be **heterogeneous**, i.e. not all agents have the same traits and behavior.

3. **Agents are active.** Agents interact with their environment and other agents, both reacting to and influencing their surroundings. Agent activity can take several forms:

    (a) Proactive activity

        i. Agents can have goals, and adjust their behavior to try to reach those goals, often using bounded rationality. **Bounded rationality** recognizes that agents do not have access to all information in the simulation and their decision-making processes may not lead to “optimal” results [43].

ii. Agents can “learn.” In other words, they can maintain a memory of their past in the simulation and use that to change their behavior to accommodate that knowledge.

(b) Reactive agents

i. Agents can simply react to their surroundings with no decision-making process. This is common for inanimate agents.

While most agents contain the characteristics described above, this framework leaves a lot of flexibility as to what agents actually are. They can be anything from individual molecules, to people, organizations, or even countries. Often agents will be grouped into classes of agents that have similar characteristics.

2.2 Environment

In ABMs, agents are embedded within an environment. Environments are characterized by the following:

1. **Environment structure.** Often the environment is representative of a physical environment, in which case it is usually structured as a discrete grid or a continuous plane within which agents can move. Other potential environment structures include three-dimensional environments or social networks, and some ABMs refrain from defining any spatial component to their model entirely [13]. When ABM environments do contain a spatial component, usually one of the agents’ traits is their location within the environment. In a continuous plane or

grid environment, an agent's location is defined by x,y-coordinates. The agent's location within the environment often moves as the simulation progresses.

2. **Environment traits.** Just as agents have traits, the environment of an ABM also has traits. These traits change as the simulation progresses and can be defined heterogeneously throughout the environment.

3. **Agent-environment interactions.** Just as agents interact with other agents, agents also interact with their environment. These are two-way interactions, such that agents influence environment traits and the environment influences agent traits.

4. **Neighborhoods.** Agent "neighborhoods" define what part of the environment the agent can interact with. In a grid environment, potential neighborhoods include the von Neumann neighborhood, which consists of the cell the agent is on and the four cells immediately adjacent to the agent, and the Moore neighborhood, which consists of the cell the agent is on and the eight cells surrounding the agent [26]. In a continuous environment, a neighborhood may be defined by a maximum Euclidean distance from the agent (Fig. 2.1).

Along with the characteristics described above, recently some researchers have started integrating geographic information systems (GIS) into model environments to increase the realism of ABMs [7].

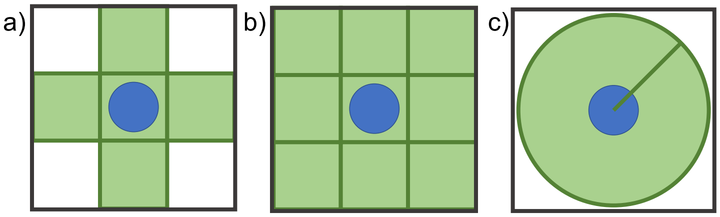


Figure 2.1: Examples of potential agent neighborhoods. Shown are the (a) von Neumann neighborhood, (b) Moore neighborhood, and (c) Euclidean neighborhood.

2.3 Rules

The final component of an ABM is the set of rules defining the behavior of agents within the model. One key aspect of these behaviors is how agents interact with other elements in the model (i.e. other agents and their neighboring environment). Often model rules will be implemented in discrete time-steps. At each time-step, the agent and environment states are updated according the the rules of the ABM. Certain rules may only come into effect at infrequent intervals within the model's progression, perhaps once every 50 time-steps, allowing the model to incorporate discrete events.

Rules can be defined for individual agents, classes of agents, or all agents in the simulation. The rules defining agent behavior are often based on qualitative information, such as expert knowledge, interviews, or stylized facts. **Stylized facts** are patterns that are seen repeatedly in empirical research and simplified into a general rule [21]. For example, a stylized fact in economics may be that individuals with higher incomes spend more money. Historically, ABMs have rarely based rules on empirical data [23]; however, as accessibility to big data sources continues to grow, empirically parameterizing ABMs is becoming more and more common [37, 3, 48, 24].

Rules often introduce **stochasticity**, or randomness, to the model. For example, behaviors may be dependent on sampling from a probability distribution. One common example of this is defining agent movement as a random walk. Other potential sources of stochasticity in the model include stochastic initial states for the agents and environment, as well as "chance encounters" between agents within the model.

2.4 Example ABM: a predator-and-prey model

The NetLogo Model Library contains many example ABMs, including an ABM of a predator-and-prey system [46]. In the predator-and-prey ABM, there are two classes of agents, Sheep and Wolves. Both agent classes contain one trait, *energy*; the value for *energy* is different for each individual agent (i.e. agents are heterogeneous), and this trait is dynamic (i.e. it changes as the simulation progresses). The ABM environment is structured as a grid and represents an open field through which the Sheep and Wolves can roam (Fig. 2.2). Rules for the model are given in Table 2.1. Notice that the rules followed by the Wolf class and the Sheep class are distinct. Also notice that agent movement and reproduction introduce stochasticity to the model.

The population dynamics shown in Fig. 2.3a-b are examples of emergent behaviors; the rules defining individual interactions in Table 2.1 result in large-scale observations about agent population size. Because the model is stochastic, the same initial conditions can cause drastically different emergent behavior. For example, when experimenting with different parameter settings, we found that initiating the model with 50 Sheep and 100 Wolves can lead to

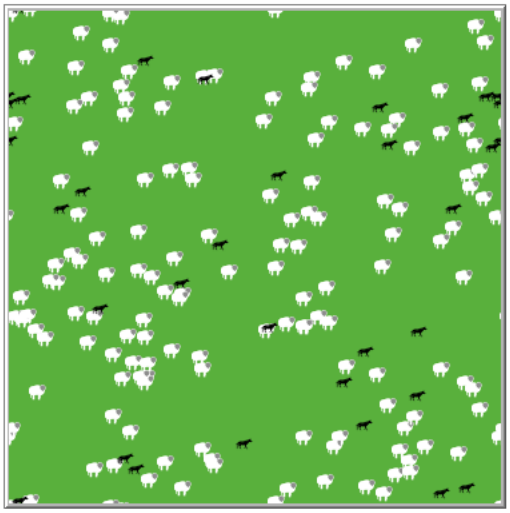


Figure 2.2: Example setup for the NetLogo predator-and-prey model library ABM.

the Sheep population either dying off or exhibiting exponential growth (Fig. 2.3a). Despite this stochasticity, it is possible to observe patterns in the populations dynamics, and these patterns are sensitive to the model's initial conditions. For example, when the simulation begins with 50 Sheep and 100 Wolves, the Sheep population is more likely to die off than experience exponential growth (Fig. 2.3a), whereas when the simulation begins with 50 Sheep and 200 Wolves, the Sheep population consistently ends in exponential growth (Fig. 2.3b).

The predator-and-prey system can also be modeled using equation-based modeling. The Lotka-Volterra equations are a system of differential equations that model the population levels of predators (wolves) and prey (sheep). If x represents sheep and y represents wolves, then the Lotka-Volterra equations are given by

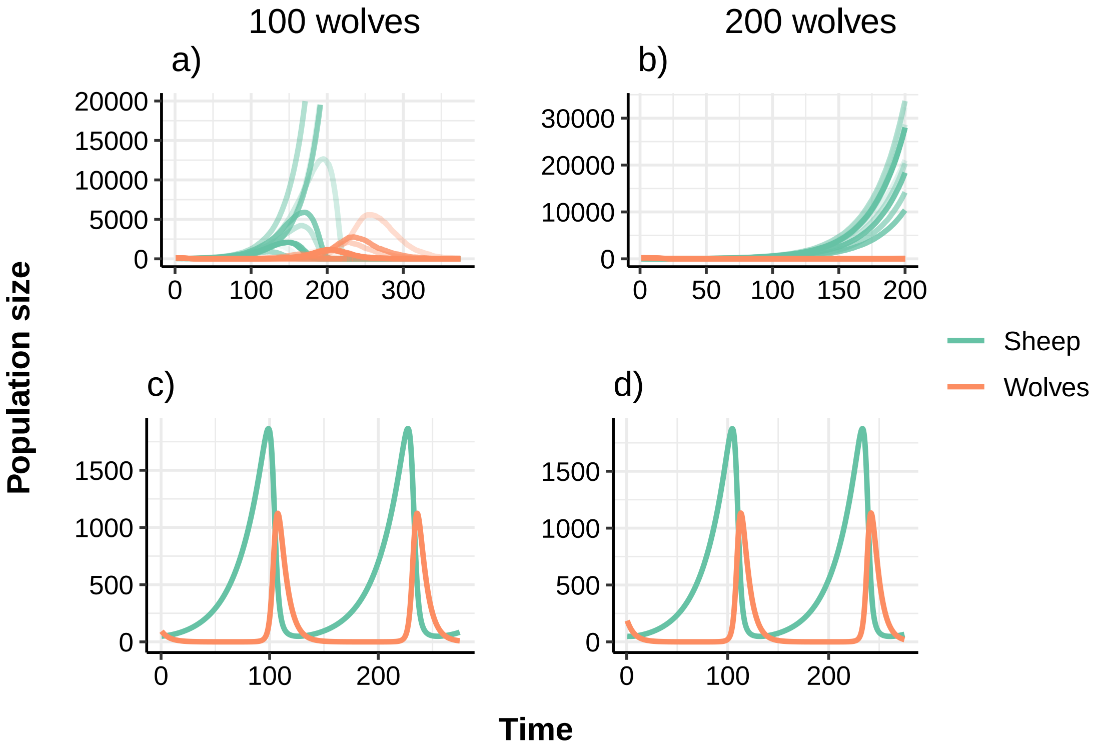


Figure 2.3: Wolf and sheep population dynamics using (a-b) NetLogo's predator-and-prey ABM and (c-d) the Lotka-Volterra equations. The initial conditions are (a, c) 50 sheep and 100 wolves or (b, d) 50 sheep and 200 wolves. For the ABM, 10 runs are shown for each set of initial conditions. These runs are indicated by distinct color saturation. Unlike the Lotka-Volterra equations, the ABM is unstable and exhibits stochastic behavior. Nevertheless, there are patterns in the ABM's emergent behaviors. When there are initially 200 Wolves, the Sheep population always ends with exponential growth, whereas when there are initially 100 Wolves, the Sheep population usually dies off.

Table 2.1: Predator-and-prey ABM rules

Rule	Description
1) Movement	Agents turn randomly to face a direction 50° right or left of their current direction. Agents then move forward one unit. Wolves lose one unit of *energy*.
2) Predation	For each Wolf; if there are any Sheep on the Wolf's patch; the Wolf eats one of the Sheep, causing the Wolf's energy to increase.
3) Death	Agents with *energy* < 0 and Sheep that have been eaten die.
4) Reproduction	Agents reproduce with probability *sheep-reproduce* or *wolf-reproduce*. When agents reproduce, they share their energy with their offspring.

$$\frac{dx}{dt} = \alpha x - \beta xy, \qquad \alpha, \beta \geq 0 \tag{2.1}$$

$$\frac{dy}{dt} = \delta xy - \gamma y, \qquad \delta, \gamma \geq 0 \tag{2.2}$$

where α and δ are the growth rates of sheep and wolves, respectively, β is the rate at which wolves consume the sheep, and γ is the death rate of the wolves [22]. Notice that α and δ from the Lotka-Volterra equations are analogous to the probabilities of agent reproduction, *sheep-reproduce* and *wolf-reproduce*, from the ABM. The parameter β corresponds to the probability a Wolf agent will collide with a Sheep agent in the ABM, and γ corresponds to the probability a Wolf agent will run out of energy (and therefore die).

As the ABM and Lotka-Volterra equations model the same system, we expect them to produce comparable population dynamics; to some extent, this is true. For example, when the wolf population is high, the sheep population decreases rapidly for both models, and when the wolf population is low, the sheep population increases rapidly for both models. The inverse is true for the wolf population. This similarity is largely a result of how random behavior can be described probabilistically. For example, in the ABM, *sheep-reproduce* is the probability a Sheep agent reproduces. In an ABM with a large population, if *sheep-reproduce* $= \alpha$ and there are x sheep in the ABM, then on average approximately αx sheep will reproduce, giving us the first term in Eq. 2.1.

However, in practice, the ABM population dynamics are unstable and inevitably lead to the extinction or explosion of the agent populations (Fig. 2.3a-b). In contrast, the Lotka-Volterra equations produce a stable, cyclic relationship between the wolf and sheep populations (Fig. 2.3c-d). This is because the Lotka-Volterra equations allow for a fractional population size, so populations can never die out completely. Organism populations found in the natural world have a very real possibility of extinction, and the ABM allows this possibility to be realized. Modeling much larger populations and environments with the ABM may reduce its instability and enable the ABM to simultaneously produce cyclic emergent behaviors and allow for population extinction.

2.5 Advantages and disadvantages of ABM

In their descriptions of ABM, Borrill and Tesfatsion [5], Crooks and Heppenstall [13], Doran [15], and Macal and North [26] discuss the advantages and disadvantages of this modeling method.

The most powerful advantage of ABM is that it allows us to understand the dynamic interactions of a system and explain how these interactions lead to emergent behaviors. Traditionally, mathematicians have used Occam's Razor to guide model development. This means that, when deciding between two models that are equally good at representing a system or set of data, the simpler model is considered superior, as including additional elements to the model introduces potential areas for error and unnecessary complexity. ABM flips this theory on its head, arguing that increasing model complexity can increase our understanding of the dynamics of the system. By creating a model based on micro-interactions, ABMs can elucidate how individual interactions cumulatively produce large-scale outcomes of the system. This helps us not only understand the current system, but determine how changing these interactions may influence large-scale outcomes.

In addition to understanding how individual interactions influence emergent behaviors, ABMs incorporate several non-traditional components to modeling: heterogeneous agents and environments, stochasticity, and discrete events. These components allow ABMs to model complex non-linear systems, increase the realistic nature of the models, and provide new possible areas of exploration. For example, agent heterogineity allows modelers to examine how populations of outlier agents influence system behavior.

Unfortunately, while these are interesting and exciting aspects of ABMs, they also introduce many challenges. The numerous parameters, stochasticity, and general complexity of ABMs makes them computationally expensive and extremely difficult to analyze. Additionally, determining how much abstraction to include in the model and deciding what rules to use to define model behavior is both critical and, at times, ambiguous. The same real-world behavior can often be represented by drastically different rule choices, and the model may be sensitive to these choices [15]. Because of these difficulties, there is debate over whether results from ABMs should be used for policy-making [36].

Chapter 3

Natural Resource Management

Natural resources such as forests, fisheries, and clean water are essential to human subsistence and prosperity. Depletion of these resources is a major problem for countries around the world. In 2014, the UN outlined an international agreement to decrease deforestation by 50% by 2020; instead, deforestation rates have continued increasing [25]. Overuse of resources has severe impacts on communities that rely on the resources for sustenance and livelihood, and these impacts propagate to the international community. For example, it has been hypothesized that mass deforestation of the Brazilian Amazon will cause water scarcity in places as far flung as regions of Africa and the United States [42].

3.1 Tragedy of the Commons

One reason it is difficult to prevent natural resource depletion is because natural resources are often common-pool resources (CPRs). CPRs have two defining characteristics: (1) CPRs are difficult to regulate, often because they are large or do not have clear boundaries and (2) the more a CPR is used, the less value it retains for other users [17]. Because of the challenges to regulating CPRs, they have long been of interest to economists, environmentalists, and other researchers.

Garrett Hardin first popularized the dilemma of CPR depletion in his piece, *The Tragedy of the Commons*. He explains that CPR users keep 100% of the profit from using a CPR. Although using the CPR also contributes to resource depletion, this cost is split among all CPR users. Therefore, for any given individual, there is always a net benefit to increasing their CPR use. Thus, the "Tragedy of the Commons" is Hardin's conclusion that users will always try to maximize their profit by increasing CPR use, which inevitably leads to resource depletion. These Tragedy of the Commons situations are formalized in game theory as a Prisoner's Dilemma [30].

Hardin argues that there are two ways to avoid a Tragedy of the Commons. Firstly, a governmental authority can initiate regulations on CPR use, thus directly preventing depletion through government legislation and enforcement of that legislation [19]. The drawback to this solution is that outside authorities, such as governments, often do not consider local context when designing regulations on resource use. Regulations are instead designed homogeneously across many communities and geographies, which can be ineffective and even detrimental to individual CPRs.

Hardin's alternative solution is to privatize the CPR by partitioning the CPR and distributing ownership of each part to individual CPR users. Hardin claims that each individual will be invested in the long-term functionality of their portion of the CPR and thus avoid overusing their portion of the resource [19]. Unfortunately, resource privatization almost inevitably leads to consolidation of ownership and massive inequality, as was seen after the privatization of fish stocks first in Iceland and then in parts of New England [28]. Thus, Hardin's solutions to the Tragedy of the Commons do not offer promising escape from this CPR dilemma.

3.2 Community-based natural resource management

Despite the drawbacks of government regulation and privatization, for decades these solutions were accepted by the scholarly community as the only methods to prevent resource depletion; however, beginning in the late 20th century, this belief has been challenged. There are many examples of communities successfully managing CPRs without government regulation or privatization. For example, in southern Thailand, many Community Management Systems (CMSs) have developed to successfully manage coastal mangrove ecosystems. Sudtongkong and Webb find that CMSs are actually *more* successful than state managed parks at protecting mangroves. They speculate that this is because the community members rely on the ecosystem for subsistence and livelihood, and therefore are invested in ensuring its longevity [38].

The mere existence of successful CMSs such as those in southern Thailand challenges Hardin's conclusions that government regulation or privatization are the only ways to prevent a Tragedy of the Commons and offers an alternative solution: community-based natural resource management (CBNRM). CBNRM is a method of preventing resource depletion by allowing members of a community to collectively determine rules regarding resource use and resource management structure.

Economist Elinor Ostrom is a main forerunner in CBNRM research. Ostrom traveled to many small communities throughout the world in the late 20th century, conducting field research to uncover patterns in how communities successfully prevent resource depletion. Through her research, Ostrom develops eight design principals for successful CBNRM: [30]

1. **Clearly defined boundaries** - the people who can use the CPR and the boundaries of the CPR itself must be clearly outlined.

2. **Congruence with the local situation** - the rules governing the CPR must be tailored to the local community and ecosystem.

3. **Collective rule-making** - CMS members must be able to participate in changing CMS rules.

4. **Monitoring** - there is monitoring to detect rule-breaking, and the monitors are CPR users or individuals subject to regulation by CPR users.

5. **Graduated sanctions** - rule-breakers are subject to penalties; the severity of those penalties depends upon the severity of the rule violation; and individuals deciding those sanctions are CPR users or individuals subject to regulation by CPR users.

6. **Conflict-resolution mechanisms** - low-cost conflict resolution mechanisms are available to CMS members.

7. **Recognition by the external governments** - the right of the CPR users to create a CMS is recognized by the government.

8. **Nested enterprises** - when CPRs are a part of a larger system, rules and activities of the CMS are organized in multiple layers.

Because Ostrom's work is based on in-depth case-studies, she is able to observe great detail within individual CMSs and elucidate how community structures and dynamic interactions within CMSs cumulatively lead to their success or failure; however, her reliance on case-studies limits the quantity of CMSs she can consider, and hence limits the generalizability and predictive power of her findings.

Research networks such as the International Forestry Resources and Institutions and the Digital Library of the Commons have helped to facilitate large-*n* analyses of CBNRM by constructing databases of individual CBNRM research projects. In 2010, Cox conducted a thorough review of the literature, confirming that many of Ostrom's principles retain strong support for CMSs around the world [12]. Statistical analyses have also confirmed many of Ostrom's principles, such as the importance of local monitoring and enforcement of CMS rules [11, 18]. These statistical analyses also illuminate new factors that influence CBNRM success, such as resource size [31], dependence on the resource for commercial livelihood [31], leadership [29, 41], organization [41], and social capital [41].

Some researchers use logistic regression to assess factors that influence the probability of CBNRM success [29, 31, 41]; however, others argue that logistic regression is unsuited for CBNRM because, compared to the number of explanatory variables, the sample size remains too small; there are a lot of missing data; and the data are susceptible to selection bias [18, 20]. Additionally, while statistical analysis such as regression models help us identify factors that are important for successful CBNRM, we lose much of our understanding of the dynamics of CMSs that Ostrom was able to evaluate using her case-study methodology.

3.2.1 Challenges for CBNRM research

One of the main challenges to CBNRM research is the sheer complexity of the systems that are being explored. CBNRM research examines social-ecological systems (SESs), which are systems that combine human social networks with ecological networks. Such systems are incredibly complex, because they must consider not only interactions within each subsystem (human and ecological), but also between subsystems. Thus, there are numerous factors that influence resource stability, and these factors likely interact in complex, nonlinear manners [35].

A second challenge of CBNRM research is the many sources of stochasticity in SESs. As in any social science, human behavior is often irrational and unpredictable. Diversity of individuals' experiences, values, and decision-making styles make it difficult to anticipate the evolution of a community's actions. Even ignoring human variability, uncontrollable natural factors such as drought, storms, and invasive species have drastic impact on natural resources and the probability of CBNRM success [4, 10].

Finally, the uniqueness of each CMS makes analyzing CBNRM difficult. As Ostrom describes in her design principles, to have a successful CMS, the management rules must be compatible with the local situation. A management system that works in one location will not necessarily be successful if adopted by another community. Each community has a unique ecosystem and social structure that influences what management methods will be most effective. Thus, generalizing findings to new or future CMSs can be problematic. For this reason, Ostrom warns researchers against trying to create simplistic models of natural resource use, such as the game theory model used to describe Hardin's Tragedy of the Commons situation. She argues that, while such models are not incorrect, they make broad assumptions and oversimplifications that limit the scope of applicability of the model. For example, the Tragedy of the Commons model assumes there is no communication among CPR users, and assumes the CPR users cannot create any rules or agreements over how to use the CPR. In real-world context, these limitations on communication and user agency are not realistic, so this model cannot accurately describe resource use [30].

3.3 ABM for CBNRM

3.3.1 Advantages of ABM for CBNRM

Given the challenges for CBNRM research described in Section 3.2.1 and the strengths of ABM described in Section 2.5, ABM is a logical method to model CBNRM. Communication between agents is an essential part of ABM, and agent activity can easily be translated into agent ability to determine resource management rules. ABMs allow for realistic representation of human communities and behavior through elements such as bounded rationality, heterogeneity, and evolutionary learning. They are also adept at modeling complex, stochastic, nonlinear systems. In fact, interaction between individuals and their environment is an aspect of CBNRM that is embedded within the structure of ABMs. Perhaps most importantly, ABMs allow modelers to analyze the dynamics of agent behavior and understand how the elements of a CMS influence factors such as resource health and prosperity of community members. This makes ABM a useful method to test hypotheses and run simulations of experiments that contain variables that are difficult to control, impractical to conduct (e.g. studying CPR health over centuries), or simply not yet seen in real-world contexts. For these reasons, it is unsurprising that modeling of SESs such as CBNRM is moving away from traditional methods, such as using differential equations, and toward ABM [35].

3.3.2 Past ABMs for CBNRM

Past development of ABMs for CBNRM have focused on two main goals.

Goal 1: Using ABMs to suggest management policies for specific communities implementing CMSs.

Goal 2: Using ABMs to learn about and understand the dynamics and critical factors within CMSs.

ABMs for policy prescription

When using ABMs to develop management policies, the ABM must be a detailed, accurate model of the community and ecosystem of interest. This presents two challenges. First, social-ecological systems are incredibly complex, so determining what characteristics of the system are important to capture in the model and what rules best capture those characteristics is both critical and ambiguous. Secondly, once the ABM rules are defined, the parameter space of the ABM is often enormous, making it difficult to fit parameters to the community of interest. For these reasons, scholars debate whether it is possible for ABMs to accurately model individual SES, and therefore whether they should be used for policy prescription.

Participatory modeling is being explored as a method of improving the applicability of policies suggested by ABMs. When using participatory modeling, researchers involve stakeholders in developing model rules to accurately depict the community of interest. For example, Campo et al. collaborate with stakeholders in a community-forest management system in Palawan, Philippines. Through conversations and behavioral experiments with Palawan stakeholders, Campo et al. develop ABM rules that are specific to the behaviors and ecosystems seen in the Palawan forest [8]. Similar projects have been implemented in Thailand [16] and Vietnam [9]. Allowing stakeholder participation in determining ABM implementation not only increases the validity of the model, it also helps stakeholders understand model outputs and use those outputs to inform future decision-making [47].

ABMs for understanding system dynamics of CBNRM

As ABM sophistication increases through use of new methods such as participatory modeling, GIS, and empirical parameterization, it is possible that ABMs will develop to accurately represent individual real-world communities; however, currently most ABMs are not used for recommending policies. Instead, these models are designed to help us learn about and understand the dynamics and system processes of CMSs. Such models can and should be more abstract than models designed for policy prescription, as they are hoping to represent groups of CMSs, rather than a single community.

These abstract models have helped us understand the role of institutions in managing CPRs, as well as how various ecological and social factors influence CBNRM success. Deadman's ABM of CPR use demonstrates the importance of communication in CPR management [14], while Schill et al. show that cooperation alone cannot prevent CPR depletion - ecological knowledge about the resource is also essential to its preservation [34]. Both of these studies designed their models on behavioral experiments.

Perez and Janssen use ABM to examine how variable resource distribution impacts CPR use. Their study of spatial heterogeneity of resource abundance illustrates how uncontrollable natural factors such as resource abundance can influence CBNRM. In their model, they analyze two outcome variables: resource health and the proportion of cooperative community members. By considering both of these outcomes, this model highlights the importance of both ecological and social sustainability to a successful CMS. They find that there are nonlinear and non-intuitive relationships between resource configuration, agent mobility, resource health, and the level of cooperation. Generally, as agent mobility increases, cooperation decreases; however, when resources are distributed homogeneously, as agent mobility increases, cooperation also increases [32]. The complexity of their findings exemplifies how ABMs can help build a more nuanced understanding of CBNRM.

Perhaps the largest collection of ABMs in CBNRM focus on the role of institutions in CBNRM success. Bravo constructs a community forest ABM to replicate a Tragedy of the Commons situation where uncooperative individuals deplete the forest resource. He then modifies the model to allow agents to collectively determine rules about forest use. This new model successfully prevents forest depletion, replicating how CMSs can effectively alter agent behavior [6]. Vallino builds on Bravo's model to compare the impacts of endogenous and exogenous institutions on forest health [40]. Agrawal et al. also use a forest ABM to study the distinct impacts of formal institutions and informal social norms on resource health [1]. Thus far, ABM research focuses broadly on how the existence of institutions influences CBNRM. Less analysis has been conducted on how differences in institution structure and management impact CBNRM.

3.3.3 Evaluating and validating ABMs

While the field of CBNRM ABM is growing, this area of research still has a long way to go. First and foremost, many of these models need greater verification of model implementation and validation of model outputs. Few model outputs have been validated with real-world observations, and even fewer have been corroborated to determine whether outputs can successfully predict observations that were not used in model development [36]. CBRNM ABMs could also benefit from further sensitivity analysis. In addition to traditional sensitivity analysis of model parameters, little research has been done on how sensitive model outputs are to differences in rule mechanisms. Finally, more work is needed in improving the realistic behavior of model agents, such as refining the implementation of social networks and adaptive behavior within the models [36].

Augusiak et al.'s method of ABM "evaludation" offers a comprehensive procedure to evaluate and validate ABMs at each step in this cyclic development process (Fig. 3.1) [2]. These steps consist of:

1. **Conceptual model evaluation.** Reviewing the conceptual decisions and assumptions behind ABM rules to determine whether they accurately represent agent and environment behavior.

2. **Model implementation verification.** Reviewing model code to ensure all rules are implemented as they have been described.

3. **Model output verification.** Analyzing emergent behaviors and other model outputs to determine whether they follow real-world

expectations, as well as assessing how much calibration was required to allow model outputs to successfully match these expectations.

4. **Model analysis.** Conducting sensitivity analysis and determining how emergent behaviors evolve through the simulation process to ensure a thorough understanding of model dynamics.

5. **Model output corroboration.** Comparing model outputs to patterns and data that were not used during model conception or were unknown prior to model conception. This step checks to see if the model can independently corroborate new findings.

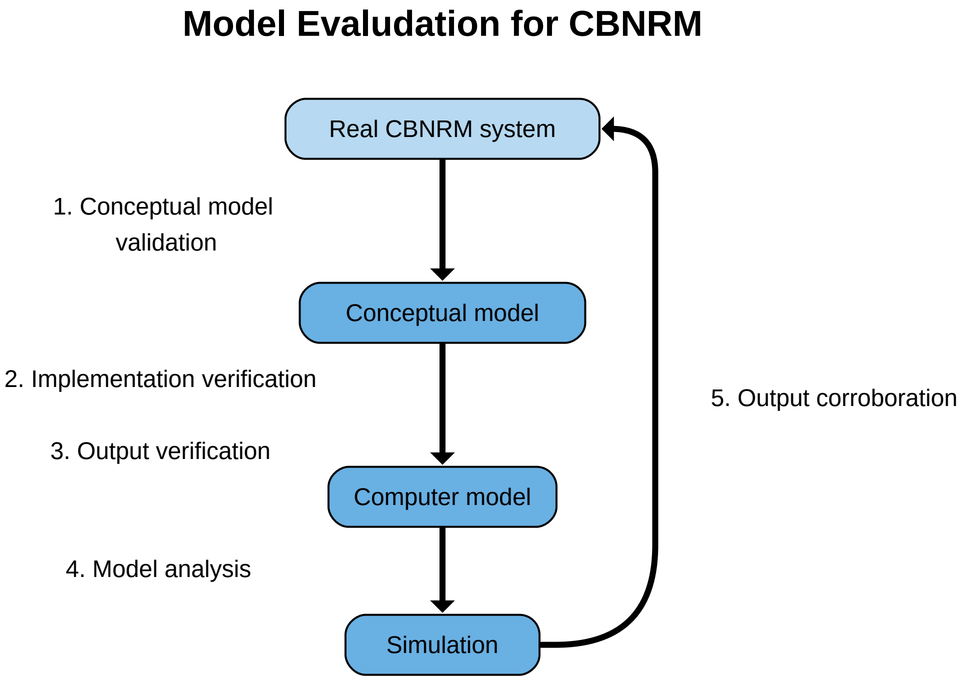


Figure 3.1: The five steps in Augusiak's evaludation process applied to CBNRM. Adapted from Augusiak et al. 2014 [2].

Chapter 4

Model description

For this investigation, we build on an existing open source model of community forest management first designed by Bravo and then expanded upon by Vallino. This model is implemented in NetLogo [45], a software program designed specifically for ABMs. See Table 4.1 for NetLogo terminology that will be used throughout this thesis.

Table 4.1: NetLogo terms

Term	Definition
patch	A cell in NetLogo's grid environment. Each patch contains its own traits.
tick	A single time-step. All model rules are implemented and agent/ patch states are updated during each tick.

BehaviorSpace is a program designed to facilitate experimentation with NetLogo models by repeatedly running models while varying parameters systematically; however, the limited flexibility of BehaviorSpace makes it difficult to implement more complex methods of parameter space exploration and sensitivity analysis. Thus, we will be using the R package RNetLogo [39] to run NetLogo simulations and export model outputs to CSV files that can then be analyzed in R [33]. The tidyverse [44] package was used during analysis.

4.1 Open-Access model

Bravo's base model describes an open-access situation where agents are logging a forest and there are no regulations on logging [6]. The simulation is broken into periods, where each period consists of 10 ticks. The model contains several global variables, which guide the overall evolution of agents and the environment, see Table 4.2. 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 4.3. Model parameters do not have units; thus, the forest area and the timescale of the simulation are not well-defined. Loggers are the only class of agents in the Open-Access model. Logger traits are defined in Table 4.4.

4.1.1 Rules

At each tick, Loggers start by decreasing their *payoff* by *cost*. This represents the Loggers' cost of living. Loggers then go to a random patch within their *neighborhood* with *trees* > *minimal-cut* 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

Table 4.2: 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*
Total-Biomass	The total tree biomass present in the environment.	dynamic	$[0, 2500 \times b_{max}]$
Living-Patches	The number of patches in the environment with *trees* > 0.	dynamic	$\{0, 1, ..., 2500\}$

* Base parameters (described in Section 5.1.1). These are manipulated during model exploration.

Table 4.3: 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	$[0, b_{max}]$

patches with *trees* > *minimal-cut* within their *neighborhood*, the Logger moves to a random patch within their *neighborhood* and does not log.

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

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

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 beliefs about the best way to log the forest. If a Logger's $payoff < old\text{-}payoff$, the Logger's *payoff-satisfaction* is set to 0 with probability

$$q = \frac{payoff - old\text{-}payoff}{|payoff| + |old\text{-}payoff|}. \tag{4.1}$$

Loggers with $payoff\text{-}satisfaction = 0$ then adjust their *minimal-cut*. The

Table 4.4: 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

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

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 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 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 [6].

4.1.2 Model outputs

As expected in a CPR situation where there is no communication among CPR users and no rules dictating CPR use, this ABM results in the depletion of the forest. Figure 4.1 gives a graphic depiction of the forest at the beginning and end of the simulation, clearly illustrating how the forest is all but eliminated in an open-access situation. Not only does the biomass level of the forest decrease to zero, Logger *payoffs*, *minimal-cut*, and *reference-trees* also approach zero, indicating the Loggers have no desire to preserve the forest. This follows intuition. Since there is no cooperation among Loggers, selfish Loggers log the most and are the most prosperous, so this behavior is rewarded and emulated.

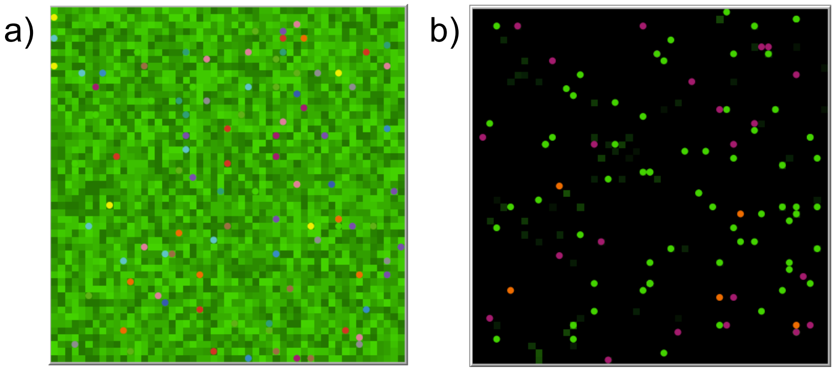


Figure 4.1: Visual representation of the (a) beginning and (b) end of a typical open-access simulation. Loggers are depicted as circles of various colors. Green patches contain biomass; the darker the green, the more biomass on the patch. Black patches contain no biomass. By the end of the simulation, almost no green patches remain.

4.2 Endogenous Institution model

After creating the Open-Access model, Bravo modifies the model to include an "endogenous instituion" (i.e. a CMS) to explore how CBNRM influences forest health. This is done by introducing a new Institution agent class. Each simulation has one Institution agent, which tracks the development of rules for forest management and Logger opinions on the current rules. The definitions of the Institution traits are given in Table 4.5.

Table 4.5: Institution traits

Variable	Description	Type	Value(s)
tolerance-threshold	The value determining when Loggers become unsatisfied with the current rules of the Institution. For this paper, *tolerance-threshold* is 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*$]$
*enforcement-level**	Determines the likelihood a Logger is caught cheating. Loggers are caught cheating with probability 1 − (*enforcement-level*/100).	static	$[0, 100]$

* This variable only exists in the C & E model.

4.2.1 Rules

The most important difference between the Endogenous Institution model and the Open-Access model is that now Loggers' decisions as to whether they should log a patch are dictated by the rules of the Institution, rather than the Loggers' own beliefs. The *minimal-cut* trait of each Logger now does not determine Logger behavior, it only indicates what Loggers believe would be the best behavior for all Loggers in the community. Instead, when Loggers decide whether to log, any and only patches with *trees* > *current-institution* are available to be logged, regardless of the individual Logger's *minimal-cut*.

The *current-institution* starts at 0, indicating that any living patch may be logged at the beginning of the simulation. The *current-institution* is 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.

4.2.2 Model outputs

During Bravo's analysis of the Endogenous Institution model, under the base parameters described in Section 5.1.1, he finds that the mean biomass level at the end of 50 simulation runs is 26.3%. While this is only about a quarter of the original biomass level in the forest, it indicates that the Institution does help prevent complete forest depletion. He also discovers that the average of the Loggers' *minimal-cut* and *reference-trees* also do not decrease to zero in this model, indicating that Loggers are still interested in forest preservation.

A replication of a single run of the Endogenous Institution model is shown in Fig. 4.2. In Fig. 4.2c, we see that biomass declines initially, but then increases and appears to stabilize[1] at around 35% of the original biomass level. This stabilizization occurs around the same time as the change in the *current-institution* (Fig. 4.2c), which suggests that the Institution is essential to the forest's conservation. As found by Bravo, the average *minimal-cut* and *reference-trees* do not decrease to zero in this example run.

4.3 Cheating and Enforcement model

Bravo's model assumes that all Loggers always obey the CMS rules. In reality, there are usually CMS members who violate the resource use rules. Thus, Vallino builds on Bravo's model by introducing Logger cheating and institutional methods to prevent cheating. We call this the Cheating and

[1] There is no clear-cut definition of a "steady-state" for ABMs. Common practice to determine when the model reaches a steady-state appears to be through qualitative observation. This is an important area for future work in ABM.

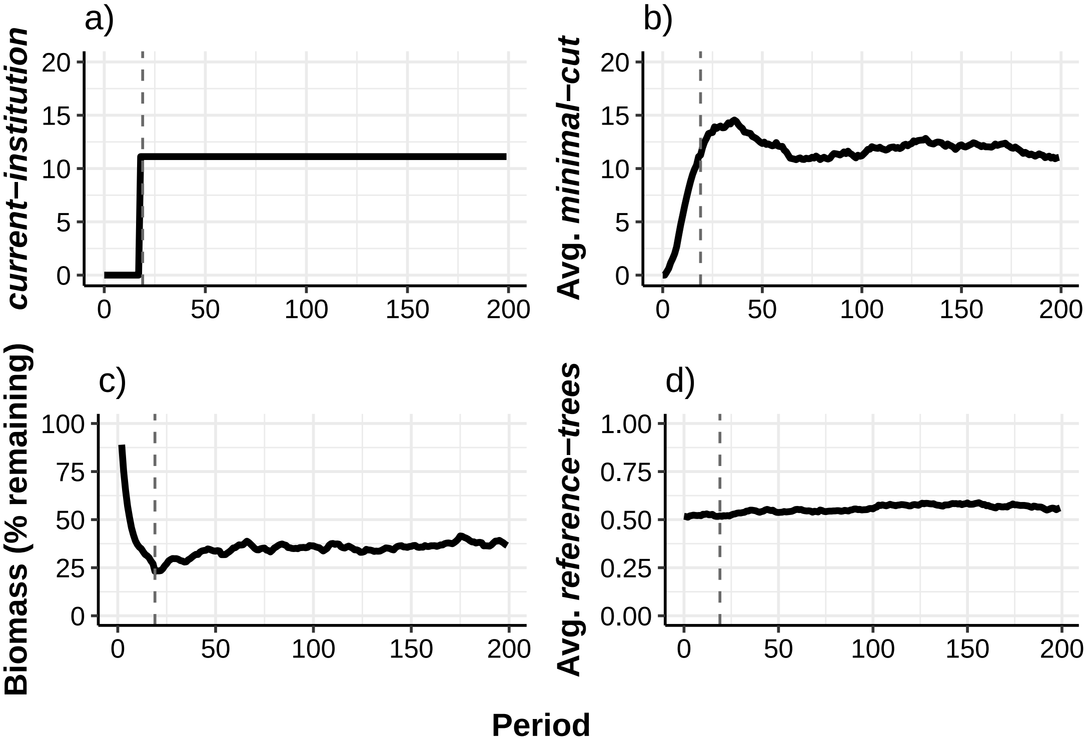


Figure 4.2: Emergent behaviors for a typical simulation using Bravo's Endogenous Institution model. (a) In the 19th period, the *current-institution* is set to 11.12 and remains there for the duration. (b) The average *minimal-cut* for all Loggers increases rapidly to around 11 before leveling out. (c) Biomass decreases sharply until the 19th period before it levels out around 35%. (d) The average *reference-trees* for all Loggers remains around 0.5 throughout the simulation.

Enforcement (C & E) model.

4.3.1 Rules

Loggers cheat when they are unsatisfied. Cheating means that Loggers cut down the *trees* on the patch they occupy even if *trees* < *current-institution*. The Institution agent is given a new trait: *enforcement-level* (Table 4.5). The *enforcement-level* parameter determines how often Loggers who cheat are caught; however, a more intuitive variable would be a transformation of the *enforcement-level* parameter that describes the probability a cheater is caught cheating. Therefore, to facilitate analysis of the model, we define *surveillance-level* to be the probability a cheater is caught,

$$\textit{surveillance-level} = P(\textit{caught}) = 1 - \frac{\textit{enforcement-level}}{100}.$$

When a cheater is caught, they are removed from the model so they are unable to log the forest.

Chapter 5

Analysis of the Cheating and Enforcement model

As discussed previously, many ABMs for CBNRM have not been well validated and could benefit from further sensitivity analysis. In her paper introducing the Cheating and Enforcement model, Vallino tests her model for many different parameter sets, but only reports the outputs of a single run for each of these parameter sets. As ABMs are stochastic models, considering only a single run for each parameter set fails to capture the stochasticity of the model. Thus, we conduct a sensitivity analysis of the C & E model by running the simulation 50 times for each parameter set and analyzing the distribution of outputs over those 50 runs. We then use Augusiak et al.'s evaludation process (described in Section 3.3.3) to conduct a thorough evaluation of Vallino's model.

5.1 Sensitivity analysis

We start analyzing the C & E model by conducting a sensitivity analysis to understand the impact of various model parameters on the magnitude of biomass in the forest and the strictness Institution rules at the end of a simulation.

5.1.1 Sensitivity analysis methods

Sensitivity analysis is conducted by varying one parameter at a time while holding all other parameters at "base" values. The base values are chosen from one set of parameters tested in Bravo's original paper introducing the Endogenous Institution model (Table 5.1). While Bravo gives no justification for his parameter choices, we opt to use one of his parameter sets to facilitate comparison across models. Parameter values are chosen to be equally distributed from 0 to at least 150% of the original value. The primary output values analyzed are

BM% = the percent of tree biomass remaining at the end of the simulation

and

K_f = *final-institution* = *current-institution* at the end of the simulation.

A higher BM% indicates a better preserved forest, and a higher K_f indicates a stricter set of Institution rules at the end of the simulation.

In his paper introducing the Endogenous Institution model, Bravo conducts a sensitivity analysis of his model using 50 runs of each parameter set. Using this as a guideline, we start by testing if 50 runs are enough to capture the stochasticity of model. For the base parameter set, we conduct three trials of 50 runs and compare the results across the three trials to ensure all three trials produce comparable results. As there is little deviation between the distributions of the percentage of biomass remaining at the end of the simulation for the three trials, we conclude that 50 runs are enough to capture the model's stochasticity, and 50 runs are used for the remaining parameter sets. (All 150 runs are used when analyzing base parameters.)

Both Bravo and Vallino run each simulation for 20,000 ticks to make sure the model reaches steady-state; however, running the simulation for this long takes too much computational time. Additionally, while there are no units for the time that passes with each tick, a patch's *trees* can go from 0 to *max-tree-growth* within around 40 ticks. If we make a very conservative estimate that forests require at least 20 years to regrow, 20,000 ticks would represent at least 10,000 years. There is no reason to run the simulation for 10,000 years, as it is very unlikely a community management system (or even the community itself) would last for that long. Therefore, instead of using 20,000 ticks, we allow each simulation to run for 2,000 ticks (at least 1,000 years), which qualitatively appears to take the model well into its steady-state. The exception to this is when varying the number of *initial-loggers*; for large values of *initial-loggers*, the simulation takes longer than 2,000 ticks to reach steady-state. This will be discussed in Section 5.2.1.

Table 5.1: Parameter values used during sensitivity analysis.

Parameter	Base value	Tested values
cost	5	{0, 2, ... 20}
max-tree-growth	20	{5, 10, ... 30}
initial-loggers	100	{50, 100, ... 300}
reference-threshold	0.5	{0.1, 0.2, ... 1}
enforcement-level	50	{10, 20, ... 100}

5.1.2 Distribution of final biomass level under base parameters

The simulation is first run under the base parameters given in Table 5.1. The distribution of BM% for base parameters is given in Fig. 5.1. Recall that Bravo found his Endogenous Institution model (which does not allow Loggers to cheat) to retain an average of 26.3% of forest biomass. We would expect introducing Logger cheating to decrease BM%, as cheating is detrimental to forest conservation. Instead, the C & E model's mean BM% under base parameters increases to 72.0%.

We also anticipated that, in certain cases, the model's stochasticity would create circumstances in which community members gathered to form an effective Institution, while in other cases the model's stochasticity would fail to produce an effective Institution. This would create clear distinctions between forests that are successfully conserved and forests that are depleted, and lead to a multimodal distribution of BM%. In contrast to our expectations, the BM% distribution is unimodal under base parameters (Fig. 5.1) and for most

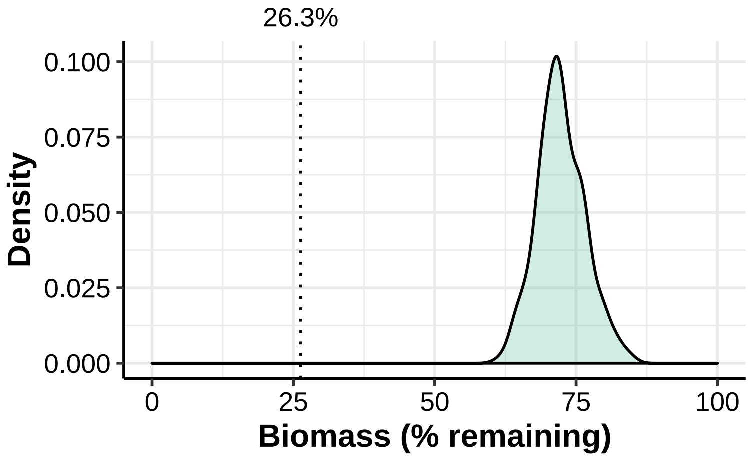


Figure 5.1: Distribution of BM% under base parameters. The distribution is unimodal with mean 72.0%. The mean BM% for the Endogenous Institution model is indicated by a dotted line at 26.3%.

other explored parameter sets. While this makes it more difficult to distinguish between conserved and depleted forests, we have no evidence that biomass should follow a multimodal distribution, so these results do not necessarily indicate a flaw in the model.

5.1.3 Parameter: *reference-threshold*

We know *reference-threshold* is a measure of the environmentalism level of the community. We expect communities that are more environmentally conscious to better preserve the forest. When varying the *reference-threshold* parameter, our intuition is confirmed, and we find a monotonically increasing relationship between *reference-threshold* and BM% (Fig. 5.2). Interestingly, when *reference-threshold* > 0.7, the amount of biomass remaining at the end of the simulation actually surpasses the amount of biomass in the “pristine” forest at the beginning of the simulation. BM% variance appears consistent for

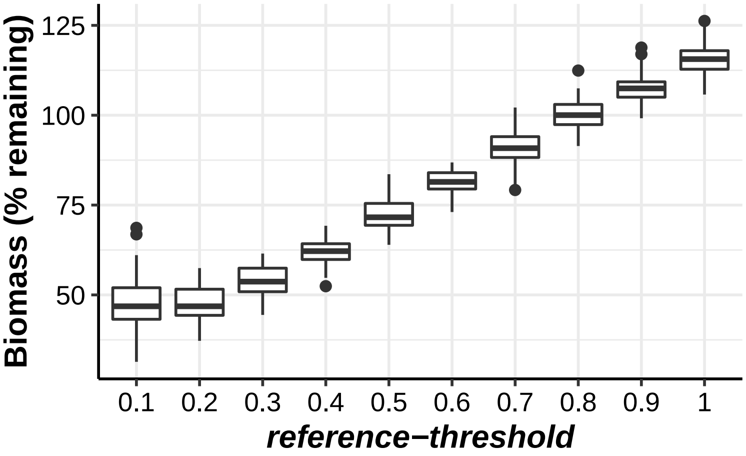


Figure 5.2: Percentage of biomass remaining after 2,000 ticks for various values of *reference-threshold*. There is a strictly increasing relationship between *reference-threshold* and BM%.

all *reference-threshold* values.

5.1.4 Relationship between *current-institution* and *Total-Biomass*

If CMSs have a positive impact on forest conservation, we would expect communities with stricter rules governing forest use to also have higher levels of biomass; this is not seen in the C & E model. In fact, we find no relationship between K_f and BM% (Fig. 5.3). In fact, 97.5% of runs for all parameter sets end with $K_f < 1$. While there are exceptions when *initial-loggers* = 300 or $surveillance - level = 0$, the overwhelming majority of situations lead to a negligible K_f. This indicates that model communities fail to form CMSs, despite their ability to do so.

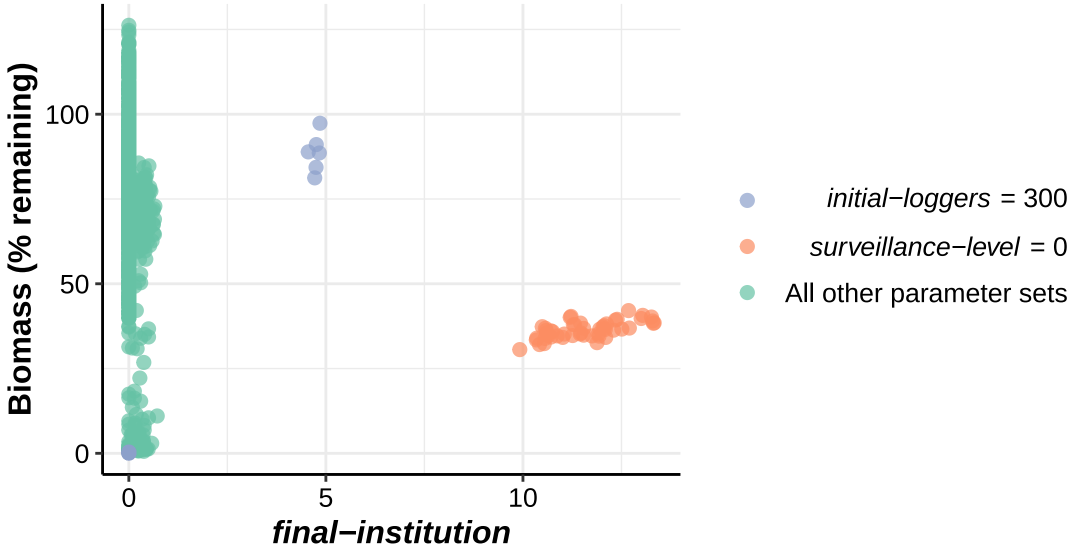


Figure 5.3: There is no relationship between BM% and K_f. For 97.5% of runs, $K_f < 1$. The exceptions occur when the *surveillance-level* = 0 or *initial-loggers* = 300.

5.1.5 Parameter: *enforcement-level*

Gibson et al. find that CMSs with higher monitoring and sanctioning generally have healthier forests [18], and these findings are supported throughout CBNRM literature [12, 41]. Recall that *surveillance-level* is the probability a Logger who cheats is caught cheating. Based on the literature, we would anticipate a clear positive correlation between *surveillance-level* and BM% in the model. On the contrary, in Fig. 5.4 we see that, as long as *surveillance-level* $\neq 0$, the value of *surveillance-level* does not influence BM%. This suggests that the model's enforcement mechanism does not accurately replicate how enforcement plays out in the real-world.

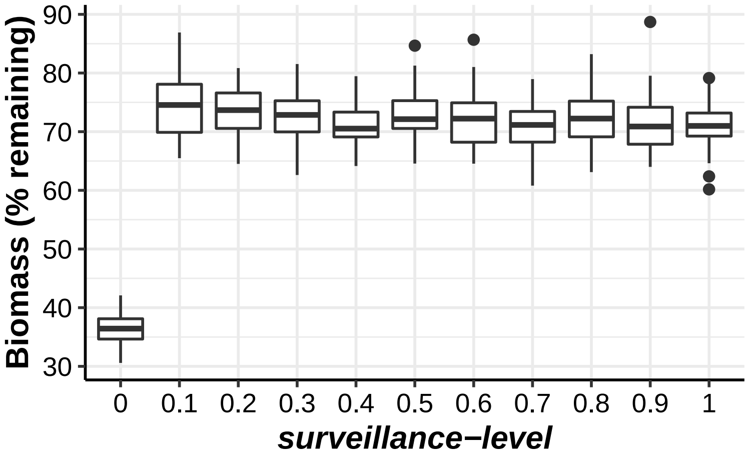


Figure 5.4: Percentage of biomass remaining after 2,000 ticks for various *surveillance-levels*. When *surveillance-level* > 0, the value of *surveillance-level* does not influence BM%.

5.2 Analysis of cheating and enforcement

In order to understand the lack of influence from the *enforcement-level* parameter, we investigate the implementation of cheating and enforcement in the C & E model.

5.2.1 Model evaludation

We use the first three steps of Augusiak et al.'s "evaludation" process (described in Section 3.3.3) to analyze the cheating and enforcement mechanism used in the C & E model. We use this process because it provides a comprehensive analysis of the model's conceptual foundation, accuracy of implementation and coding, and validity of model outputs.

Step 1: Conceptual model evaluation

Common sanctions for cheating on a CMS include reprimands, fines, or, in severe cases, suspension of CPR use privileges. Even when privileges are suspended, the CMS cannot guarantee the Logger will abide by the suspension - in fact, after being ostracized from the CMS, they are likely to completely ignore CMS rules and log whenever they see fit. This is why Ostrom suggests graduated sanctions, determining the punishment for cheaters based on the severity of the violation [30]. In the C & E model, Loggers who are caught cheating are removed from the model entirely, which implies that cheaters are imprisoned, exiled, or executed. This does not make sense in real-life context, and does not follow Ostrom's principles for a successful CMS.

We would expect Loggers to move to a new patch of forest after logging their current patch, so that they have an opportunity to log more biomass in the next tick. In the C & E model, Loggers only move at the end of each tick if they have not logged during that tick. Therefore, if a Logger logs their patch on tick t, then on tick $t + 1$ they are inevitably located on an empty patch and either cannot log or must cheat. There is no reason the Logger would wait to find a new patch of forest, and therefore it would make more sense for all Loggers to change location at each tick.

Finally, we would expect a Logger's *payoff-satisfaction* to depend on their *payoff-satisfaction* from the previous period. For example, if a Logger is unsatisfied in period p because their *payoff* is decreasing and their *payoff* continues to decrease in period $p + 1$, we would expect them to remain unsatisfied in the next period. Instead, Loggers whose *payoff* continues to decrease in period $p + 1$ are unsatisfied with probability q (see Eq. 4.1) regardless of their previous *payoff-satisfaction*.

Step 2: Implementation verification

There are two flaws in the model's implementation of cheating:

1. **Not all Loggers who should cheat actually do cheat.** Vallino claims that all unsatisfied Loggers cheat, but this is not how cheating is coded in the model. Unsatisfied Loggers are Loggers with *payoff-satisfactions* = 0 or $|$*minimal-cut* − *current-institution*$|$ > *tolerance-threshold*. Loggers with *payoff-satisfaction* = 0 do not cheat in the model. At the end of each period, Loggers' *payoff-satisfaction* is updated according to the model rules; however, all Loggers' *payoff-satisfaction* are subsequently reset to 1 before progressing to the next period. Thus, when determining if a Logger will cheat, Loggers that should have *payoff-satisfaction* = 0 are not included, and so do not cheat. We assume this is a coding error.

2. **Some Loggers who are accused of cheating have not actually cheated.** Sometimes, unsatisfied Loggers will be placed on an empty patch and still decide they want to "cheat." A percentage of those Loggers are "caught" cheating and removed from the model despite the fact that they cannot log an empty patch and therefore cannot break the rules.

Step 3: Model output verification

When *Total-Biomass* in the forest is low, more Loggers are unsatisfied with the *current-institution* and cheat; thus, more Loggers are caught cheating and removed, causing the number of Loggers in the simulation to decrease. A good example of this is when *initial-loggers* is large. When there is an overwhelming number of *initial-loggers*, *Total-Biomass* crashes immediately, and the number of Loggers in the simulation begins to decrease. Figure 5.5 is an example of this when *initial-loggers* = 1000, but a similar trend occurs whenever *initial-loggers* > 200.

This decrease in Loggers has several consequences. Firstly, the *current-institution* of the model rarely changes from 0. Recall that the *current-institution* only changes when *unsatisfied* > $\frac{2}{3}$*initial-loggers*. Unsatisfied Loggers are being removed from the model after they cheat, so they are unable to call for a change in the *current-institution*, and consequently the *current-institution* = 0 throughout the simulation. (In fact, it is impossible for the *current-institution* to change once $\frac{1}{3}$ of the Loggers have been removed for cheating.)

Secondly, the decrease in Loggers causes unrealistic behavior in the simulation's *Total-Biomass*. When the simulation has a high number of *initial-loggers*, *Total-Biomass* plummets immediately, and the number of Loggers begins to decline. *Total-Biomass* remains near zero until the number of Loggers reaches a low level, and then *Total-Biomass* jumps back up to a substantial percentage of the original forest biomass level. Even when trying to overwhelm the forest with 1000 Loggers, *Total-Biomass* eventually restores if

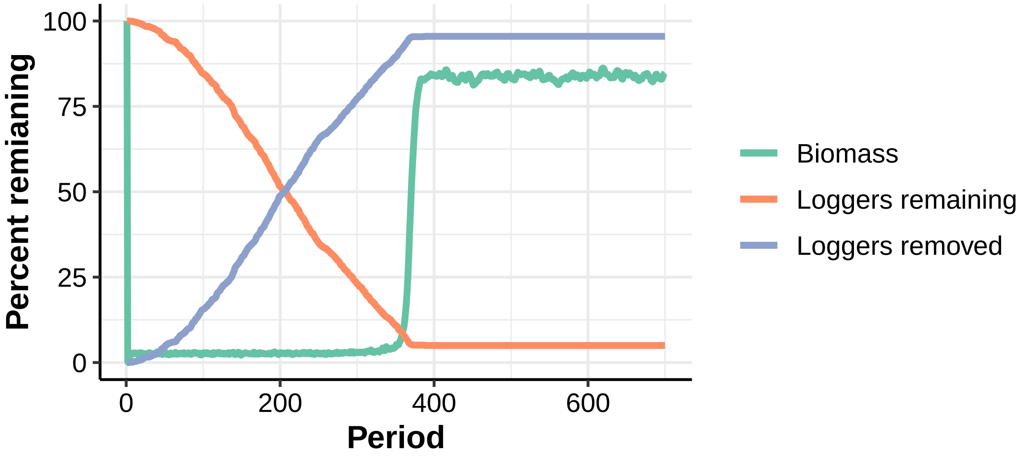


Figure 5.5: *Total-Biomass*, number of Loggers, and number of cheaters removed in the forest over time when *initial-loggers* = 1000. The number of Loggers slowly decreases while *Total-Biomass* is low, due to the removal of cheaters, and then flattens out after around 400 periods (4,000 ticks). *Total-Biomass* plummets immediately, but then jumps back up when the number of Loggers flattens out.

the simulation is allowed to run long enough for all Loggers to be removed from the model (Fig. 5.5).

This behavior does not follow empirical observations. CMSs would never remove the majority of users from a forest, and a depleted forest is unlikely to regenerate without substantial reforestation efforts. It also explains why simulations with high numbers of *initial-loggers* do not reach steady-state before 2,000 ticks. *Total-Biomass* cannot reach steady-state until the number of Loggers also reaches steady-state, and it takes many ticks for the number of Loggers to decrease to a stable number.

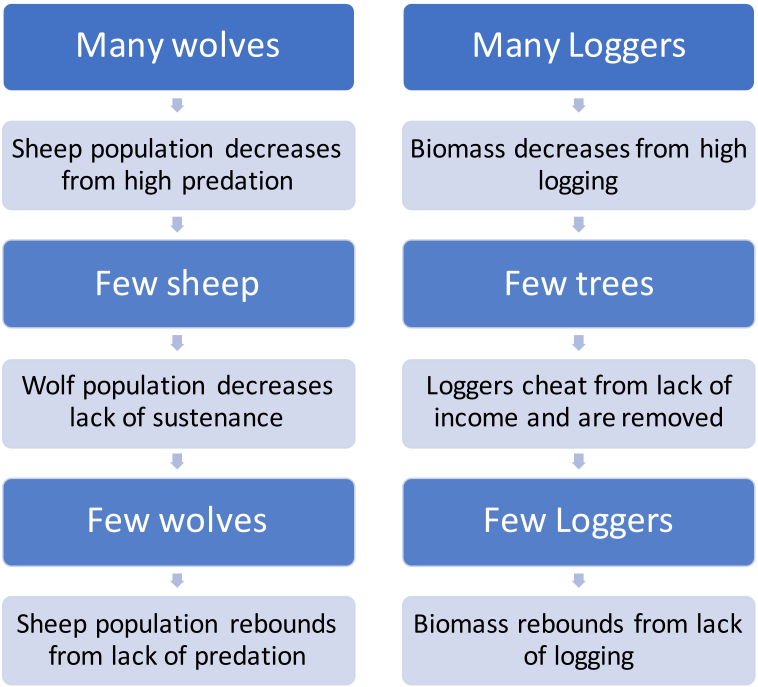

Many wolves
Sheep population decreases from high predation
Few sheep
Wolf population decreases lack of sustenance
Few wolves
Sheep population rebounds from lack of predation
Many Loggers
Biomass decreases from high logging
Few trees
Loggers cheat from lack of income and are removed
Few Loggers
Biomass rebounds from lack of logging

Figure 5.6: A comparison of the predator-and-prey system and the C & E model when *initial-loggers* is large.

5.2.2 Comparison with a predator-and-prey system

When *initial-loggers* is large, the behavior of the model is reminiscent of the predator-and-prey system discussed in Section 2.4, where Loggers represent Wolves and *Total-Biomass* represents Sheep (Fig. 5.6). In this section, we compare the C & E model outputs with outputs from the the Lotka-Volterra equations. We use a single ABM run where *initial-loggers* = 1000, because the model's stochasticity is minimal with this initial condition. *Total-Biomass* and number of Loggers are analyzed as a percentage of their initial values. The Lotka-Volterra equations are solved numerically in R. Since new Loggers are never introduced into the model, we set $\delta = 0$. All other parameters are chosen to replicate ABM outputs.

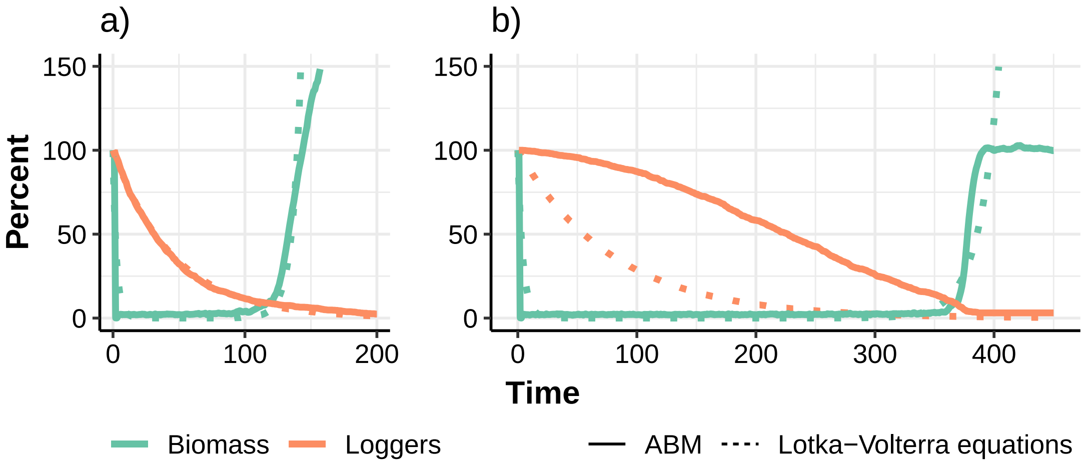


Figure 5.7: Loggers and biomass in the forest as a percentage of the original amount for (a) the simplified ABM replicating a predator-and-prey model, along with the Lotka-Volterra equations with parameters chosen to match the ABM and (b) the C & E model with *initial-loggers* = 1000, along with the Lotka-Volterra equations with parameters chosen to match C & E model.

If we remove the model's forest carrying capacity and change Logger behavior so that all Loggers cheat with a probability of 0.5%, we can choose parameters for the Lotka-Voltera equations so that the equations closely follow the ABM (Fig. 5.7a). Thus, under these modifications, the model is essentially an ABM approximation of a specific instance of the predator-and-prey differential equations.

If we reintroduce the forest carrying capacity, *Total-Biomass* now levels off at around 4,000 ticks, while the Lotka-Volterra equations allow biomass to increase indefinitely. Additionally, if Loggers are allowed to cheat according to C & E model rules instead of simply cheating 0.5% of the time, the Lotka-Volterra equations are not successful at modeling the number of Loggers in the ABM (Fig. 5.7b). When biomass levels are near zero in the

Lotka-Volterra equations, we have $x \approx 0$, so Eq. 2.2 reduces to

$$\frac{dy}{dt} \approx -\gamma y. \tag{5.1}$$

This is the differential equation for exponential decay; thus, using the Lotka-Volterra equations, we would expect exponential decay of Loggers. Instead, the number of Loggers appears to decrease approximately linearly (Fig. 5.7b). Since the rate of Logger decrease is consistent regardless of the number of Loggers left in the model, we conclude that the probability any individual Logger will cheat increases as the simulation progresses. In the Lotka-Volterra equations, this would be equivalent to replacing the constant γ with $\gamma(t)$, where $\gamma(t)$ is an increasing function of time.

Chapter 6

Monitoring and Sanctioning model description

In Chapter 5, we outlined the conceptual drawbacks of removing all cheaters from the Cheating and Enforcement model and identified several implementation errors within the model's code. Given these limitations, in this chapter we modify the C & E model to fix implementation errors and offer a new mechanism for cheating and enforcement that we believe better represents community management systems seen in the real world. We will call this model the Monitoring and Sanctioning (M & S) model.

6.1 Modifying Logger movement

In contrast to the C & E model, in the new M & S model all Loggers move at the beginning of each tick. This ensures Loggers who successfully logged in the last tick have a chance to be located on a patch with biomass. The rules governing how Loggers choose a new patch remain unchanged.

6.2 Preventing punishment of innocent Loggers

In the C & E model, Loggers were permitted to "cheat" even if they were located on an empty patch. Because of this, Loggers were punished for cheating even when they had not removed any biomass from the forest. In the M & S model, Loggers cannot opt to cheat when they are located on an empty patch.

6.3 Modifying *payoff-satisfaction* implementation

Recall that Loggers with *payoff-satisfaction* = 0 do not cheat in the C & E model, even though these Loggers are considered unsatisfied. In the M & S model, this implementation error is fixed, ensuring that *payoff-satisfaction* is not reset to 1 at the end of each period. Thus, Loggers who are unsatisfied with their *payoff* are now permitted to cheat.

The M & S model also changes the updating process of *payoff-satisfaction* so it 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 (see Eq. 4.1). These rules are outlined in Table 6.1.

Table 6.1: 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

6.4 Revising the cheating and enforcement mechanism

In the C & E model, every unsatisfied Logger cheats on the *current-institution*; however, human populations are heterogeneous. Individual people have distinct moral codes and unique experiences, so Loggers' cheating behavior should reflect these differences. This would suggest that each unsatisfied Logger should have a distinct probability of cheating, and that their experiences throughout the simulation should influence that probability of cheating. Therefore, in our model, we introduce a new dynamic Logger trait, *prob-cheat* (p_c), which is the probability a Logger will cheat if they are unsatisfied. This value is updated throughout the simulation to account for how the Loggers' experiences influence their decision-making process.

Ostrom's fourth and fifth design principles say that it is important to have monitoring to detect rule-breaking and graduated sanctions to punish Loggers in accordance with the severity of their violations [30]. To better align with these principles, the *enforcement-level* trait of the Institution agent from the C & E model is replaced by two traits, *monitoring-level* and *sanction-level*. All

Table 6.2: Monitoring and Sanctioning model variables

Variable	Description	Type	Value(s)
num-cheaters	Tracks the number of cheaters at each time step.	dynamic, Institution	[0, *initial-loggers*]
initial-prob-cheat	The initial probability of cheating for all Loggers.	static, global	[0,1]
monitoring-level	The probability Loggers will be caught cheating (as a percentage).	static, Institution	[0,100]
sanction-level	The effectiveness of sanctions used to discourage Loggers from cheating.	static, Institution	[0,1]
prob-cheat (p_c)	The probability a specific Logger will cheat. All Loggers start with p_c = *initial-prob-cheat*.	dynamic, Logger	[0,1]

new M & S model parameters are defined in Table 6.2.

6.4.1 Rules

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.

Therefore, when a Logger cheats, p_c is updated by the following equation:

$$\Delta p_c = \begin{cases} -p_c \times \textit{sanction-level} & \text{if caught} \\ (1 - p_c) \times \textit{sanction-level} & \text{if not caught} \end{cases}. \tag{6.1}$$

As an example, suppose

$$\textit{initial-prob-cheat} = \textit{monitoring-level} = \textit{sanction-level} = \frac{1}{2},$$

and we examine a Logger who is indefinitely unsatisfied. The Logger's p_c over time can be represented by the graph of the Markov chain in Fig. 6.1.

Some may object that the *sanction-level* should only affect Δp_c when a Logger is caught; however, using *sanction-level* to scale Δp_c when the Logger is not caught allows us to ensure that p_c is primarily determined by the Logger's most recent experiences. If the Logger has most recently been caught cheating, $p_c < 0.5$, and if the Logger has most recently gotten away with cheating, $p_c > 0.5$. Nevertheless, all previous experiences retain some impact on the Logger's p_c.

Example 1. *Suppose we are in a forest where*

$$\textit{initial-prob-cheat} = \textit{monitoring-level} = \textit{sanction-level} = \frac{1}{2}.$$

Logger A cheats once and is not caught. Logger B cheats twice and is caught the first

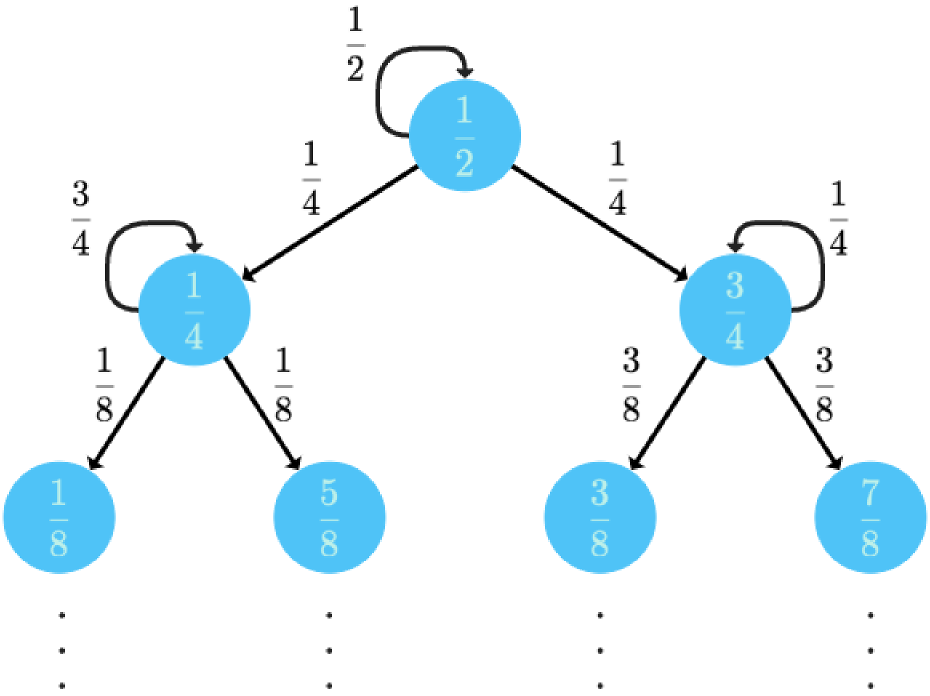


Figure 6.1: Graph representing the infinite Markov chain for an indefinitely unsatisfied Logger with *initial-prob-cheat* = *monitoring-level* = *sanction-level* = $\frac{1}{2}$. Fractions within blue nodes represent potential states (values of p_c) for the Logger. Fractions along edges represent the probability of transitioning from one state to another. Notice that a Loggers who is successively caught cheating ends with p_c near 0, while a Logger who is successively not caught cheating ends with p_c near 1.

time. Therefore,

$$p_{c,A,final} = \frac{1}{2} + \left(1 - \frac{1}{2}\right) \times \frac{1}{2} = \frac{3}{4}$$

$$p_{c,B,t=1} = \frac{1}{2} - \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$$

$$p_{c,B,final} = \frac{1}{4} + \left(1 - \frac{1}{4}\right) \times \frac{1}{2} = \frac{5}{8}$$

Although $p_{c,final} > 0.5$, for both Loggers, Logger B has a memory of being caught once, and therefore $p_{c,A,final} > p_{c,B,final}$.

6.4.2 Markov chain simulation of *prob-cheat*

To get an idea of how *monitoring-level* and *sanction-level* affect the probability Loggers will cheat in the model, we use R to create a simulation of the Markov chain given in Fig. 6.1. The simulation starts with 1,000 Loggers with $p_c = 0.5$ and runs for 1,000 time-steps. We assume these Loggers are perpetually unsatisfied and have the option to cheat at each time-step. We let *monitoring-level* = *sanction-level* = 0.5, and then vary the *monitoring-level* and *sanction-level* parameters one at a time.

Results from the Markov chain simulation are given in Fig. 6.2. When *monitoring-level* = *sanction-level* = 0.5, the distribution of p_c after 1,000 time-steps is skewed right, such that Loggers are much more likely to have p_c near 0, but some Loggers now have $p_c > 0.5$ (Fig. 6.2a). This skewness occurs because once a Logger has a low p_c, they are unlikely to cheat, and therefore their p_c is unlikely to change, causing them to get "stuck" with a low p_c.

As expected, in simulations with a high *monitoring-level* or *sanction-level*,

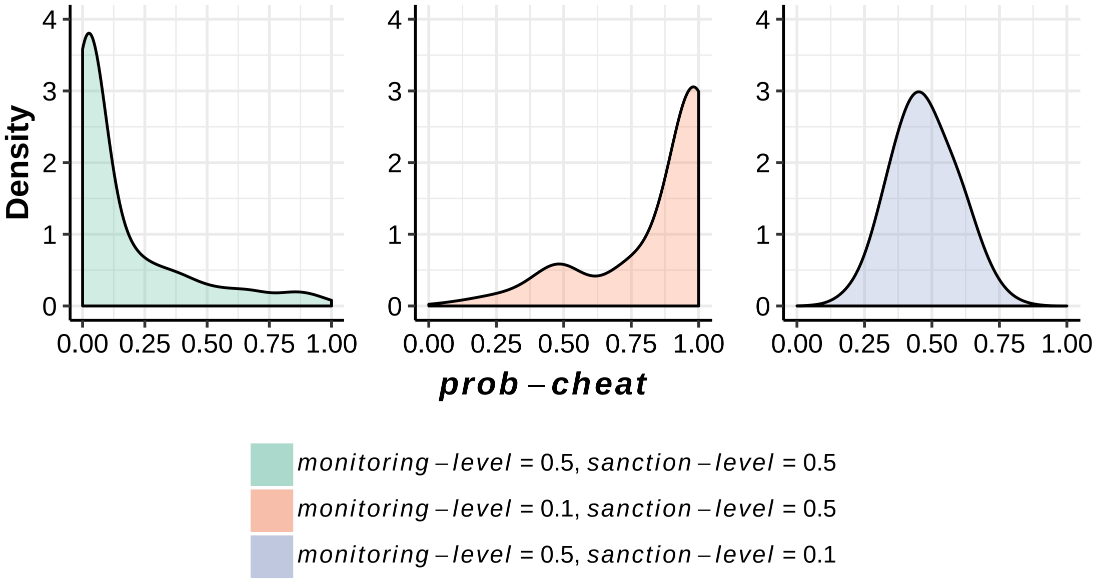


Figure 6.2: Distribution of p_c for a Markov chain simulation of 1,000 perpetually unsatisfied Loggers with *initial-prob-cheat* = 0.5 after 1,000 time-steps. (a) When *monitoring-level* = *sanction-level* = 0.5, the distribution is skewed right, such that most Loggers have a low p_c. (b) When *monitoring-level* = 0.1 and *sanction-level* = 0.5, the distribution is skewed left, such that most Loggers have a high p_c. (c) When *monitoring-level* = 0.5 and *sanction-level* = 0.1, the distribution is approximately symmetric and centered about 0.5.

most Loggers have a very low p_c. In fact, when either *monitoring-level* or *sanction-level* is 0.9, over 95% of Loggers have $p_c < 0.01$ (not shown). Also as expected, decreasing *monitoring-level* or *sanction-level* bellow 0.5 increases the number of cheaters with a high p_c. When *monitoring-level* = 0.1, the distribution of p_c is skewed left, such that Loggers are much more likely to have p_c near 1 (Fig. 6.2b). When *sanction-level* = 0.1, the distribution of p_c is approximately symmetric about 0.5 (Fig. 6.2c). This indicates that monitoring has a greater impact than sanctioning on Loggers' p_c

Chapter 7

Analysis of the Monitoring and Sanctioning model

The Monitoring and Sanctioning model both fixes implementation errors from Vallino's original model and offers a new mechanism for cheating and enforcement. We therefore conduct a second sensitivity analysis to understand the influence of parameters in the new model and determine how the rule changes impact model dynamics.

7.1 Sensitivity analysis methods

Sensitivity analysis was conducted using the same methods described in Section 5.1.1. Base parameter values and tested values were the same as in Table 5.1, except for *initial-loggers* and new parameters. New base parameters and tested parameters are given in Table 7.1. The value *initial-loggers* was varied between 60 and 200, because when *initial-loggers* > 200 in the M & S

Table 7.1: Parameter values used during sensitivity analysis of the M & S model.

Parameter	Base value	Tested values
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}

model, the forest became depleted immediately in all simulations and did not rebound.

7.2 Distribution of final biomass level under base parameters

Recall that Bravo found the mean BM% under base parameters for the Endogenous Institution model (which does not allow Logger cheating) to be 26.3% [6]. We expected introducing cheating and enforcement to Bravo's model would decrease BM%; however, the C & E model actually increased the mean BM% to 72.0%. After modifying the cheating and enforcement mechanisms to create the M & S model, the mean BM% is 11.9%. This is just under half the BM% found for the Endogenous Institution model, and satisifies our intuition that cheating should decrease forest health (Fig. 7.1).

When testing the C & E model, we were surprised to find that BM% followed a unimodal distribution. We thought the stochasticity of the model would lead to some communities forming successful Institutions and other

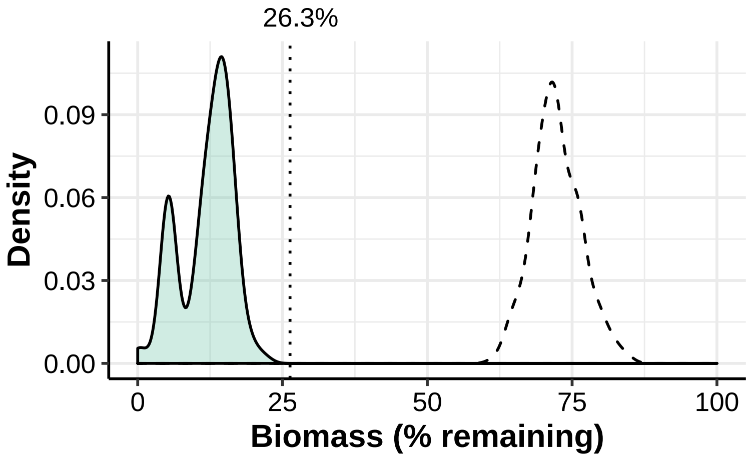


Figure 7.1: Distribution of BM% under base parameters. The distribution is bimodal with mean 11.9%. The mean BM% for the Endogenous Institution model is indicated by a dotted line at 26.3%. The BM% distribution under base parameters for the C & E model is shown as a dashed line.

communities failing to form successful Institutions, resulting in a multimodal distribution for BM%. The new model's distribution of BM% is bimodal, which aligns better with our intuition, and makes it easier to distinguish between depleted and conserved forests. The trough of the distribution occurs at 8.2%, so we define "depleted" forests to be those with BM%< 8.2% an "conserved" forests to be those with BM%≥ 8.2%.

7.3 Relationship between *current-institution* and *Total-Biomass*

Because Loggers who were caught cheating were removed from the C & E model, the *current-institution* rarely changed from zero, and therefore there was no relationship between the *final-institution* (K_f) and BM%. In the new model, we now see a strong positive correlation between K_f and BM%.

Under base parameters the correlation between K_f and BM% is $\rho = 0.947$ (Fig. 7.2). This indicates that model communities that form a stricter set of rules governing forest logging are more successful at conserving the forest, and communities that fail to form adequate rules fail to conserve the forest. Additionally, the two cluster of points in Fig. 7.2 show that K_f is bimodal, and the modes of K_f align with the modes of BM%. Finally, the figure shows that communities that have met more recently are more likely to lie in the higher cluster, indicating that they have both a high K_f and a high BM%. This suggests that it is important for communities to continue meeting and adjusting the *current-institution* even after they have established a preliminary set of rules. Ostrom's third design principle states that it is important for individuals using a community resource to participate in modifying CMS rules [30]. This implies that allowing users to change CMS rules over time is important to CMS success. This behavior was not included during model conception, which indicates that our model is able to corroborate real-world patterns that were not included during the development of the model.

The positive correlation between K_f and BM% extends to almost all parameter sets tested ($\rho = 0.855$ when including all simulation runs), although the relationship between these variables no longer appears linear (Fig. 7.3). The exception is when *initial-loggers* = 60. In this case, there are not enough Loggers to deplete the forest, and therefore the *current-institution* is irrelevant. If we exclude simulation runs where *initial-loggers* = 60, the correlation between K_f and BM% is $\rho = 0.945$, which is similar to the correlation found using only base parameters.

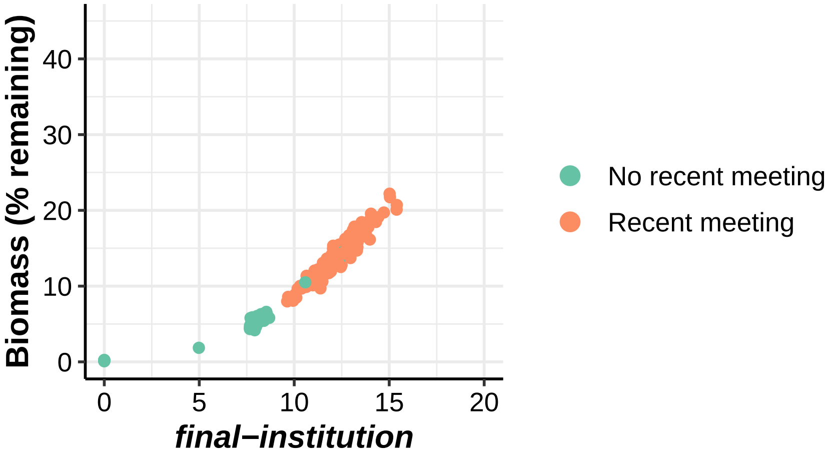


Figure 7.2: There is a strong positive correlation between K_f and BM% under base parameters ($\rho = 0.947$). Both K_f and BM% exhibit bimodal behavior, and the modes of these distributions align, as evidenced by the two clusters of points seen in the figure. Communities that fail to meet consistently throughout the simulation are more likely to end in the lower cluster, which has both a lower K_f and lower BM%.

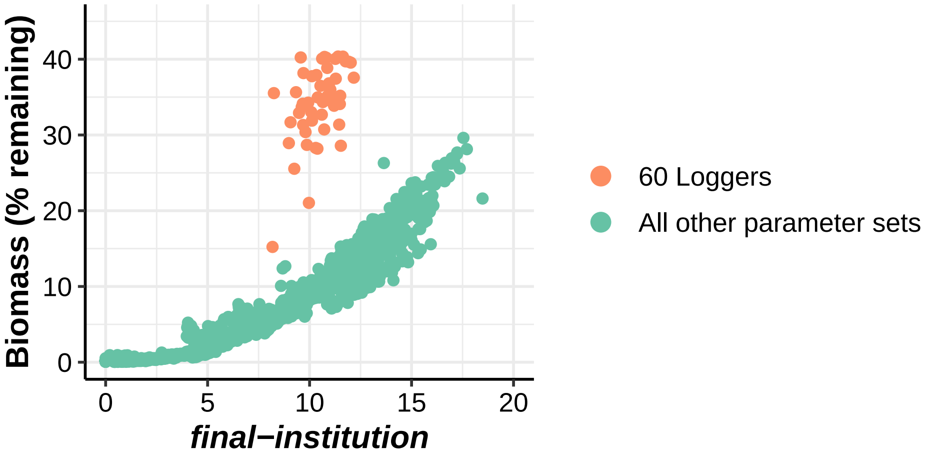


Figure 7.3: There is a strong positive correlation between K_f and BM% for almost all parameter sets ($\rho = 0.855$). The exception is when there are only 60 Loggers in the simulation; in this case there are not enough Loggers to deplete the forest, and K_f is irrelevant. When excluding simulations with 60 Loggers, $\rho = 0.945$.

Not only do we find a correlation between the *current-institution* and biomass at the end of the simulation, we also find that changes in the *current-institution* have a direct impact on the biomass in the forest as the simulation progresses. Figure 7.4a-b shows an example of the biomass and *current-institution* over time for a typical simulation under base parameters. We see that changes in the *current-institution* cause direct changes in the biomass level. At the beginning of a simulation, when biomass is decreasing rapidly, the change in the *current-institution* stops forest depletion and helps biomass level-out. After 59 periods, when the *current-institution* jumps up drastically again, biomass level also increases sharply.

This relationship between changes in the *current-institution* and biomass level is consistent across most simulation runs under all parameter sets. Because biomass levels are stochastic and small fluctuations in biomass are expected, we define an “increase” or “decrease” in biomass to be when the mean biomass for the most recent 3 periods is at least 1,000 units larger or smaller than the mean biomass for the 3 preceding periods. Since the effect of changes in the *current-institution* may lag behind the adoption of new rules, when the *current-institution* increases, we check whether this change is followed by an increase of biomass within 3 periods. A “leveling-out” of biomass occurs when there have been at least 3 periods of biomass decrease, followed in the next 3 periods by at least 1 period where there is no change in biomass. We believe these choices allow us to capture most instances when the *current-institution* impacts biomass, while ensuring random fluctuations in biomass are not included. Using this method for all runs over all parameter sets, we find that an increase in the *current-institution* causes a leveling-out of biomass 50.4% of the time; an increase in biomass 22.8% of the time; and no quantified impact 26.9% of the time (Fig. 7.4c).

7.4 Parameter: *reference-threshold*

Recall that *reference-threshold* is a measure of the environmentalism level of the community. We would expect a monotonically increasing relationship between *reference-threshold* and BM%, where communities with higher environmentalism levels are better at conserving the forest. This was confirmed in the C & E model, but is no longer found in the M & S model (Fig. 7.5). BM% initially increases with *reference-threshold*, but drops for *reference-threshold* = 6 and remains around 6.4% for *reference-threshold* > 6. Interestingly, parameter sets where BM% is higher also have higher variance in BM%. This suggests that unfavorable circumstances lead almost inevitably to forest depletion, but favorable circumstances are not a sufficient condition to guarantee forest conservation.

We next investigate the reason for the unexpected non-monotonic relationship between BM% and *reference-threshold*. In Fig. 7.6, when *reference-threshold* < 3, the community usually meets 1-2 times, and these meetings occur near the beginning of the simulation, causing K_f (and therefore BM%) to remain low. When *reference-threshold* > 6, the community is more environmentally conscious, so they set a higher *current-institution* early in the simulation; however, because of the high initial institution, they only meet 1-2 times and these meetings occur early in the simulation. The community fails to increase the *current-institution* as the simulation progresses, and they end

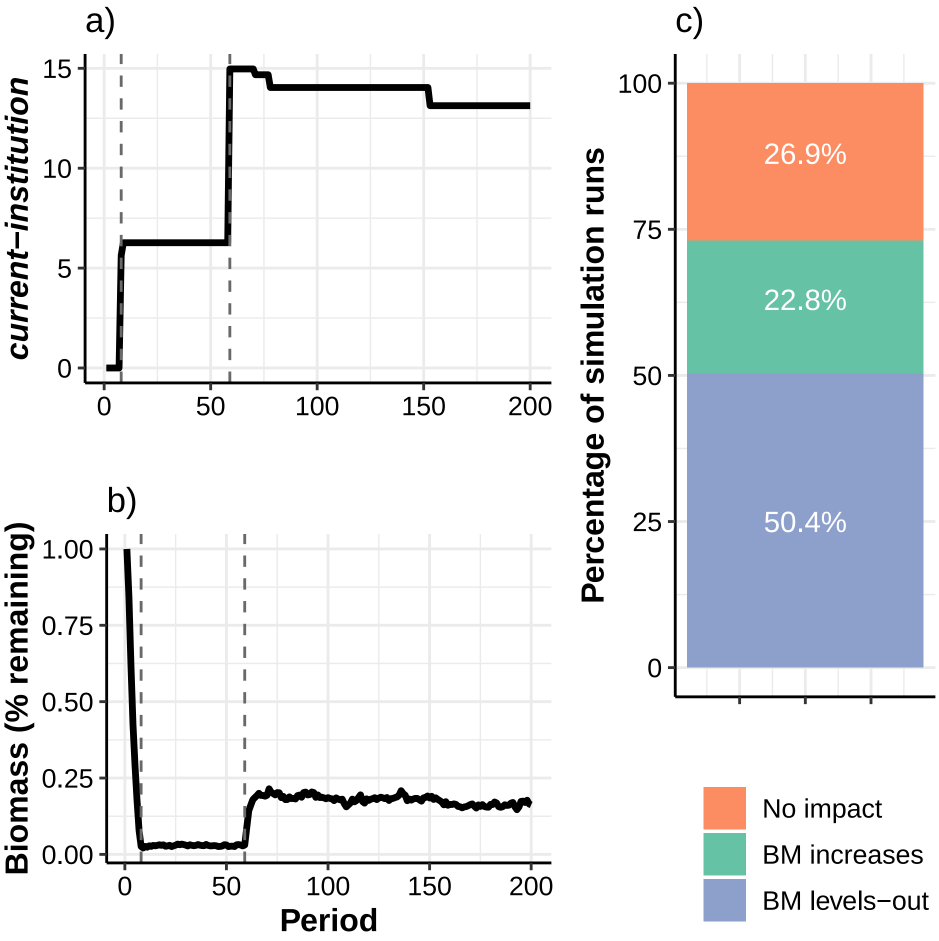


Figure 7.4: For a typical M & S model under base parameters, (a) the *current-institution* and (b) the percentage of biomass remaining over time. The *current-institution* increases drastically after 8 periods, causing a leveling off in biomass. After 59 periods, the *current-institution* increases sharply again, causing an increase in biomass. (c) This relationship between the *current-institution* and biomass levels is consistent across runs for all parameter sets. A change in the *current-institution* ends a sharp decline in biomass 50.4% of the time, causes an increase in biomass 22.8% of the time, and has little to no effect on biomass 26.9% of the time.

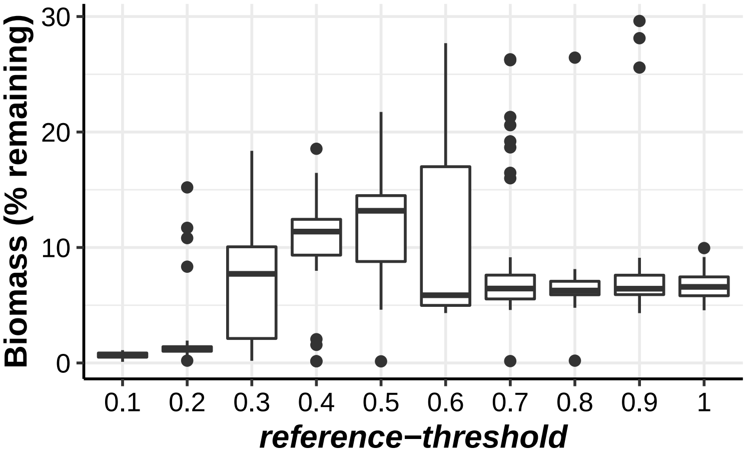


Figure 7.5: Distribution of BM% for various values of *reference-threshold*. Initially, increasing *reference-threshold* increases BM%; however, when *reference-threshold* = 6, BM% drops, and for *reference-threshold* > 6, BM% remains around 6.4%.

with a relatively low K_f. In contrast, when 3 ≤ *reference-threshold* < 6, the community usually meets at least three times and these meetings continue until late in the simulation, which leads to a high K_f and, hence, a high BM%.

7.5 Parameters: *monitoring-level* and *sanction-level*

One reason we chose to modify the C & E model is because the *enforcement-level* had no impact on BM%. Unfortunately, the *monitoring-level* and *sanction-level* parameters continue to have no meaningful impact on BM% in the M & S model (Fig. 7.7a-b). The lack of influence of these two parameters on BM% suggests that further work is needed in establishing an enforcement mechanism for this model.

Although *monitoring-level* and *sanction-level* do not influence BM% in the

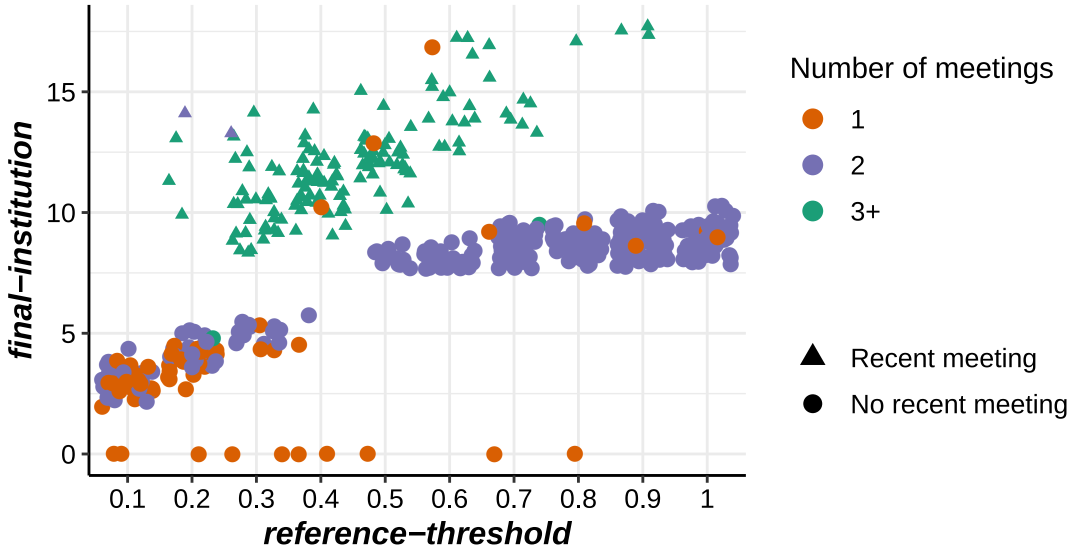


Figure 7.6: The *final-institution* for various *reference-thresholds*. When *reference-threshold* is extremely low or extremely high, communities meet only 1-2 times near the beginning of simulation, leading to lower K_f.

forest, higher levels for these parameters do decrease the number of Loggers who cheat in the final period of the simulation (Fig. 7.7c-d), a relationship not seen between *enforcement-level* and the number of cheaters in the C & E model. This suggests that simulations with higher monitoring and sanctioning do exhibit more cooperative communities.

7.6 Relationship between the number of cheaters and *final-institution*

Initially, it seems odd that parameter sets with fewer cheaters do not have higher BM% in Fig. 7.7a-b; however, further analysis of the mechanics behind Institution behavior explains this incongruity. The *current-institution* can only

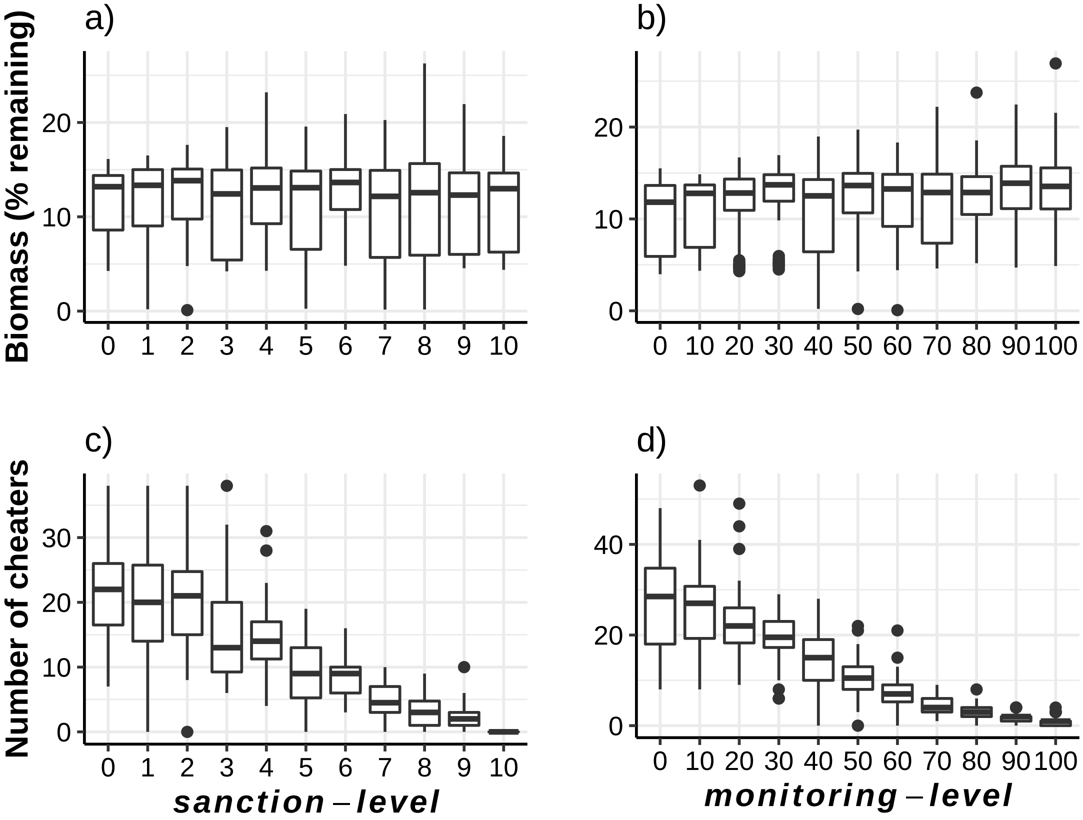


Figure 7.7: The level of monitoring and sanctioning appear to have no impact on BM% (see a-b); however, increasing either of these parameters does decrease the number of cheaters in the final period of the simulation (see c-d).

change when $\frac{2}{3}$ of all Loggers are unsatisfied, and unsatisfied Loggers are also the Loggers that have the potential to cheat. Thus, it follows that communities with a high number of unsatisfied Loggers will have both a high K_f (and therefore a high BM%) and a high number of cheaters. In fact, when looking at all parameter sets tested, a moderate positive correlation exists between K_f and the number of cheaters during the final period ($\rho = 0.58$) (Fig. 7.8). This behavior does not follow empirical observations and further indicates that the model's enforcement mechanism needs additional modification.

However, also note that, when K_f is high, "cheaters" may be Logging patches with relatively high biomass. For example, suppose Cheater A lives in a community with $K_f = 11$ and logged a patch with $trees = 10$, whereas Cheater B lives in a community with $K_f = 3$ and logged a patch with $trees = 2$. It is naïve to consider Cheater A to be as culpable as Cheater B. Thus, although communities with higher K_f may have more cheaters, these cheaters may be logging patches with more biomass than the cheaters in a community with a low K_f.

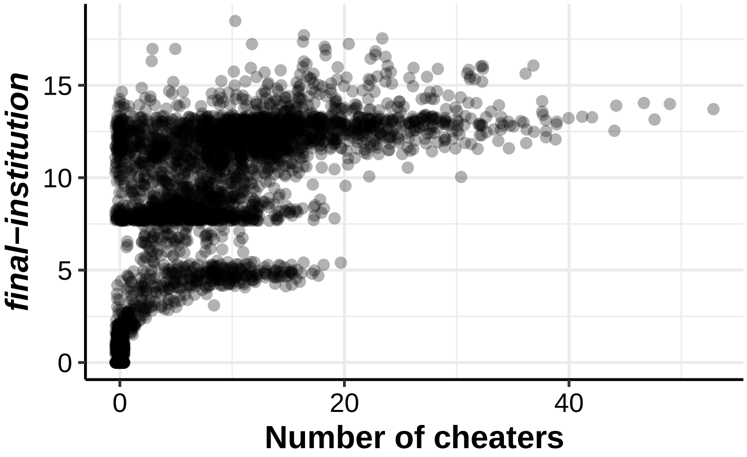


Figure 7.8: There is a positive correlation between the number of cheaters during the final period of a simulation and K_f, the *final-institution*.

Chapter 8

Conclusion

In an era when our population appears to be exceeding sustainable levels, resources are squandered flippantly, and climate change threatens to further limit resource abundance, natural resource management has become a subject of primary concern both internationally and for individual communities. As ecologists and social scientists learn more about ecosystem conservation and human behavior, we have come to realize that small-scale community management systems can be a powerful method of conserving natural resources; however, understanding the complex social-ecological systems surrounding community-based natural resource management can be extremely challenging. The introduction of Agent-Based Modeling to the field of CBNRM offers a promising method to learn about the dynamic interactions within community management systems and provide insight into how such dynamics impact the success of resource conservation.

8.1 An Agent-Based Model of community-based natural resource management

During this investigation, we examine an existing ABM first developed by Bravo and then expanded upon by Vallino. The original model simulates a community management system in which a community establishes rules regulating how Logger agents may extract wood from a forest. Vallino's modifications allow Loggers who are unsatisfied with the Institution rules to cheat. The model also establishes an enforcement mechanism which punishes Loggers who are caught cheating.

Upon analyzing Vallino's Cheating and Enforcement model, we found that the enforcement mechanism does not accurately represent enforcement procedures found in real-world CMSs. Loggers who are caught cheating are sanctioned by being removed from the model. This suggests that the punishment for cheating is exile, imprisonment, or execution, none of which are realistic sanctions. Additionally, the enforcement mechanism leads to unrealistic patterns in forest biomass levels over time; the level of enforcement appears to have no meaningful impact on forest health; and the model found no correlation between the strictness of Institution rules and forest conservation.

8.2 Model modifications and findings

We propose a new Monitoring and Sanctioning model that fixes several implementation errors within the C & E model and offers a new set of rules to model Logger cheating and Institution enforcement. We believe these modifications make the ABM rules more realistic and in closer alignment with Ostrom's design principles for CBNRM. Additionally, the new model exhibits emergent behaviors in which changes in Institution rules have a direct impact on forest biomass levels, leading to a positive correlation between the strictness of Institution rules and forest conservation. This behavior aligns with CBNRM literature. Moreover, the model independently corroborates the importance of communities meeting consistently and adapting CMS rules over time. Finally, unlike Vallino's C & E model, in the M & S model, increasing monitoring and/or sanctioning successfully decreases the number of cheaters in the model.

8.3 Limitations and future work

Unfortunately, just as the level of enforcement has no meaningful impact on forest conservation in the C & E model, the level of monitoring and sanctioning continues to have no meaningful impact on forest conservation in the M & S model. This suggests that the cheating and enforcement mechanisms require further modification.

One possible method of modifying the cheating and enforcement mechanism is by only allowing Loggers to cheat when they are unsatisfied because they believe the *current-institution* is too high. Currently, all unsatisfied Loggers have the potential to cheat, but we would expect Loggers who believe the Institution rules are not strict enough would not cheat on those rules. Allowing these Loggers to remain unsatisfied but not cheat would better replicate human behavior and decrease the number of cheaters in the model while retaining enough unsatisfied Loggers to allow the *current-institution* to change. This may also alleviate the unexpected positive correlation between the number of cheaters in the simulation and the biomass levels in the forest.

Other potential changes to the model include introducing more heterogeneity to agent behavior. For example, not all individuals respond identically to sanctioning, so it may be logical to have each Logger have their own (randomized) *sanction-level* trait. Additionally, in a real CMS, individuals who have recently been caught cheating may be subjected to increased monitoring, which would suggest that Loggers should also have their own *monitoring-level* trait, and that trait should increase when they are caught cheating. Finally, a Logger's probability of cheating should be based on their perception of how "fair" the Institution is. Thus, a Logger's *prob-cheat* could be influenced by the number of cheaters in the community, as well as how often those cheaters are caught.

As ABMs are extremely complex models, the process of evaluating and validating these models is never complete. For example, after implementing the new M & S model, we noticed that the Logger selection process appears flawed. The selection process occurs at the end of each period, when the Logger with the lowest *payoff* is replaced by a copy of the Logger with the highest *payoff*, to simulate unsuccessful Loggers adopting the behavior and beliefs of successful Loggers. However, during this process, the unsuccessful Logger also copies the *payoff* of the most successful Logger, which does not make sense. Thus, we would suggest that the unsuccessful Logger should retain their own *payoff* when they copy the successful Logger's beliefs. While this is only one example of a model rule that should be adjusted, there are no doubt many such changes that will need to be made as validation of the model continues and as the model develops in sophistication and complexity.

In addition to modifying the model itself, future work should expand the breadth of analysis for this model. During our analysis, we focused almost exclusively on biomass levels as a measure of a CMS's "success"; however, social sustainability is just as important as ecological sustainability for CBNRM success. In past explorations of this model, Vallino and Bravo incorporated Logger *payoff* into their analysis. Similarly, in their model of CPR use, Perez and Janssen examined both resource health and the proportion of cooperative community members [32]. While the number of cheaters in the M & S model was briefly discussed, future work should provide further analysis of the social indicators in model communities.

During our investigation, we found no clear way to define the steady-state of an ABM. As ABMs are stochastic and have many model outputs, random fluctuations in output values are always possible, and even if one output appears to reach a steady-state, this is no guarantee that the model itself has reached a steady-state. Thus, future work establishing a method of defining when an ABM reaches steady-state could be beneficial both to this model and ABMs in general.

It is clear from our investigation that CBNRM ABMs are still in their beginning stages. CMSs are extremely complex systems, and although ABMs have the capacity to model complex systems, developing such models is a long and cyclic process, whereby a model is created, tested, and refined over and over. As with any computational model, there is no way to ensure that coding errors have been entirely eliminated, and the conceptual validity of the model will be challenged and improved upon over time. These problems are more prevalent with new models, such as the community forest ABM we investigated, as they have not yet gone through as thorough of a vetting process. This does not mean that ABMs are an unacceptable means of modeling CBNRM, but it does mean current investigations of these models may wish to focus on model improvement before drawing concrete conclusions or policy suggestions from model emergent behaviors.

Appendix A

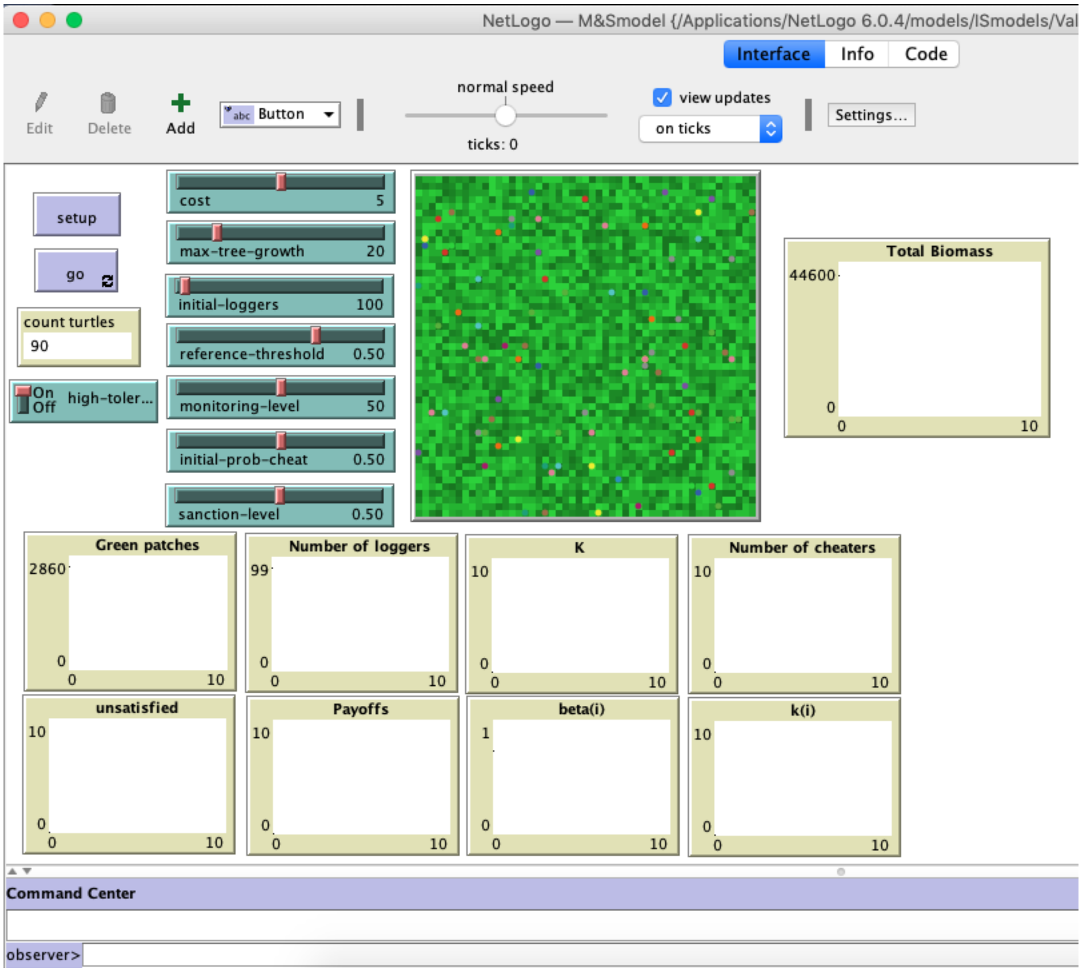
Monitoring and Sanctioning model code and interface

NetLogo code used for the Monitoring and Sanctioning model. All parameters that were varied during sensitivity analysis are controlled by the sliders shown in the screenshot of the NetLogo interface for the M & S model (Fig. A.1).

```
; global variables (including Institution variables)
globals [
  growth-prob
  satisfaction
  current-institution
  tolerance-threshold
  unsatisfied
  num-cheaters
]

; Logger variables
turtles-own [
  payoff
  minimal-cut
  reference-trees
  old-payoff
  payoff-satisfaction
  prob-cheat
  cheater?
]

; patch variables
patches-own [
  trees
  living-neighbors
]



; setting up the simulation
to setup
  clear-all

  ; used to match old version (NetLogo 4.3) default world size
  resize-world 0 50 0 50
  set-patch-size 5

  ; setting original values and model vizuals
  set growth-prob .05
  set current-institution 0
  set unsatisfied 0

  set-default-shape turtles "circle"
  ifelse high-tolerance = true
    [set tolerance-threshold  (2 * max-tree-growth) / 3]
    [set tolerance-threshold  max-tree-growth / 3]
  ask patches [
    set trees random (max-tree-growth - max-tree-growth / 2) +
       max-tree-growth / 2 + 1
    set pcolor 60 + 5 * (trees / max-tree-growth)
  ]

  ; initializing Loggers
  crt initial-loggers
  ask turtles [
    setxy random-pxcor random-pycor
    set payoff 0
    set old-payoff 0
    set minimal-cut 0
    set payoff-satisfaction 1
    set reference-trees  (max-pxcor * max-pycor) *
       random-normal reference-threshold 0.25

    set prob-cheat initial-prob-cheat
    set cheater? FALSE
  ]
  compute-satisfaction

  reset-ticks
end

; steps in one tick of the model
to go
  ; update tick (+ 1)
  tick
  tree-growth
  turtle-actions

  ; if we have reached the end a period, implement rules that occur
  ; at the end of a period
  if ticks mod 10 = 0 [compute-satisfaction]

  ; stop simulation after 20,000 ticks
  if ticks = 20000 [stop]
```

```
end



; rules governing tree growth
to tree-growth
  ; calculate living neighbors for each patch
  ask patches [set living-neighbors count neighbors with [trees > 0]]

  ; see if an empty patch should regrow
  ask patches with [trees = 0]
  [
    if (random 1001 / 1000) < growth-prob * ((living-neighbors + 1) / 9)
    [
      set trees 1
      set pcolor 60 + 5 * (trees / max-tree-growth)
    ]
  ]

  ; allow living patches to grow
  ; (if they have not reached max-tree-growth)
  ask patches with [trees > 0 and trees < max-tree-growth] [
    set trees trees + .5
    set pcolor 60 + 5 * (trees / max-tree-growth)
    ]
end



; logger actions for each tick
to turtle-actions
  ask turtles [

    ; Model change:
    ; make all Loggers move at the beginning of their actions
    move-turtles
    ; update payoff (living costs)
    set payoff payoff - cost

    ; if rules permit, log the patch they are on
    ifelse ([trees] of patch-here > current-institution)
    [log-here]

    ; If rules to not permit, see if the Logger will cheat

    ; Model changes:
    ;     Loggers can only cheat when [trees] of patch-here != 0
    ;     Monitoring and sanctioning are introduced
    [if ((abs (minimal-cut - current-institution) > tolerance-threshold
       or payoff-satisfaction = 0) and [trees] of patch-here != 0)
      ; check if Logger will cheat
      [if ((random 100) / 100 < prob-cheat)
        [log-here

         ; track cheaters
         set cheater? TRUE

         ; monitoring and sanctioning
         ifelse random 100 < monitoring-level ; is cheater caught?
             [set prob-cheat prob-cheat -
                prob-cheat * sanction-level]
             [set prob-cheat prob-cheat +
                 (1 - prob-cheat) * sanction-level]
         ]

       ]
    ]
  ]

end
```

```
; rules for Logging
to log-here
  set payoff payoff + [trees] of patch-here
  ask patch-here [
      set trees 0
      set pcolor black]
end

; rules for Logger movement
to move-turtles
  let green-patches patches in-radius 2.83 with
     [trees > current-institution]
  ifelse count green-patches = 0
    [let newX xcor + (random 3 - 1) * 2
    let newY ycor + (random 3 - 1) * 2
    set xcor newX mod (max-pxcor + 1)
    set ycor newY mod (max-pycor + 1)
    ]
    [let newX [pxcor] of one-of green-patches
    let newY [pycor] of one-of green-patches
    set xcor newX
    set ycor newY
    ]
end

; rules implemented at the end of each period including:
; Logger payoff-satisfaction update
; Institution updates
; plotting model outputs
; Logger "selection" process
; resetting payoff
to compute-satisfaction

  ; Model change: how payoff-satisfaction is computed
  ask turtles with [payoff > old-payoff][
    set payoff-satisfaction 1
   ]

  ask turtles with [payoff < old-payoff] [
    if payoff-satisfaction = 1[

    let q (payoff - old-payoff) / (abs payoff + abs old-payoff)
    if (- random-float 1) > q [
      set payoff-satisfaction 0
      ifelse count patches with [trees > 0] > reference-trees
        [set minimal-cut max list (0) (minimal-cut - random 10)]
        [set minimal-cut min list (minimal-cut + random 10)
            (max-tree-growth + 1)]
    ]
  ]
    ]


 ; "endogenous institution" rules and evolution
  set unsatisfied count  turtles with
     [abs (minimal-cut - current-institution) > tolerance-threshold
     or payoff-satisfaction = 0]
  if  unsatisfied > (0.66666 * initial-loggers) [
    set current-institution mean [minimal-cut] of turtles
  ]


  ; plotting emergent behaviors
  set-current-plot "K"
  plot current-institution
  set-current-plot "Green patches"
  plot count patches with [trees > 0]
  set-current-plot "Total Biomass"
  plot sum [trees] of patches
  set-current-plot "Payoffs"
  plot mean [payoff] of turtles
  set-current-plot "k(i)"
  plot mean [minimal-cut] of turtles
  set-current-plot "beta(i)"
  plot mean [reference-trees] of turtles / 2500
  set-current-plot "unsatisfied"
  plot unsatisfied
  set-current-plot "Number of Loggers"
  plot count turtles

  set num-cheaters count turtles with [cheater?]
  set-current-plot "Number of cheaters"
  plot num-cheaters
  ask turtles [set cheater? FALSE]


  ask one-of turtles with [payoff = min [payoff] of turtles] [ die ]
  ask one-of turtles with [payoff = max [payoff] of turtles] [
    hatch 1 [
      setxy random-pxcor random-pycor
      set minimal-cut 0
      ]
  ]

  ask turtles [
    set old-payoff payoff
    set payoff 0
  ]
end
```



NetLogo — M&Smodel {/Applications/NetLogo 6.0.4/models/ISmodels/Val

normal speed

☑ view updates

on ticks

ticks: 0

count turtles 90

On Off high-toler...

cost 5

max-tree-growth 20

initial-loggers 100

reference-threshold 0.50

monitoring-level 50

initial-prob-cheat 0.50

sanction-level 0.50

Command Center

observer>

Figure A.1: Screenshot of the M & S model NetLogo interface. All parameters that were varied during sensitivity analysis are controlled by sliders.

Bibliography

[1] Arun Agrawal, Daniel G. Brown, Gautam Rao, Rick Riolo, Derek T. Robinson, and Michael Bommarito. Interactions between organizations and networks in common-pool resource governance. *Environmental Science & Policy*, 25:138–146, January 2013.

[2] Jacqueline Augusiak, Paul J. Van den Brink, and Volker Grimm. Merging validation and evaluation of ecological models to ‘evaludation’: A review of terminology and a practical approach. *Ecological Modelling*, 280:117–128, May 2014.

[3] David Bell and Chidozie Mgbemena. Data-driven agent-based exploration of customer behavior. *SIMULATION*, 94(3):195–212, March 2018.

[4] Fikret Berkes. Environmental Governance for the Anthropocene? Social-Ecological Systems, Resilience, and Collaborative Learning. *Sustainability*, 9(7):1232, July 2017. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

[5] Paul L. Borrill and Leigh Tesfatsion. Agent-based Modeling: The Right Mathematics for the Social Sciences? In *The Elgar Companion to Recent Economic Methodology*. Edward Elgar Publishing, 2011.

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

[7] Daniel G. Brown, Rick Riolo, Derek T. Robinson, Michael North, and William Rand. Spatial process and data models: Toward integration of agent-based models and GIS. *Journal of Geographical Systems*, 7(1):25–47, May 2005.

[8] Paolo C. Campo, Guillermo A. Mendoza, Philippe Guizol, Teodoro R. Villanueva, and François Bousquet. Exploring management strategies for community-based forests using multi-agent systems: A case study in Palawan, Philippines. *Journal of Environmental Management*, 90(11):3607–3615, August 2009.

[9] J. C. Castella, N. H. Trung, and S. Boissau. Participatory simulation of land-use changes in the northern mountains of Vietnam: the combined use of an agent-based model, a role-playing game, and a geographic information system. *Ecology and Society*, 10(1):1–32, 2005.

[10] Brian Chaffin, Hannah Gosnell, and Barbara Cosens. A decade of adaptive governance scholarship: synthesis and future directions. *Ecology and Society*, 19(3), September 2014. Publisher: The Resilience Alliance.

[11] Ashwini Chhatre and Arun AgrawaI. Forest commons and local enforcement. *Proceedings of the National Academy of Sciences of the United States of America*, 105(36):13286–13291, September 2008.

[12] Michael Cox, Gwen Arnold, and Sergio Villamayor Tomás. A Review of Design Principles for Community-based Natural Resource Management. *Ecology and Society*, 15(4), December 2010.

[13] Andrew T. Crooks and Alison J. Heppenstall. Introduction to Agent-Based Modelling. In Alison J. Heppenstall, Andrew T. Crooks, Linda M. See, and Michael Batty, editors, *Agent-Based Models of Geographical Systems*, pages 85–105. Springer Netherlands, Dordrecht, 2012.

[14] Peter J. Deadman, Edella Schlager, and Randy Gimblett. Simulating common pool resource management experiments with adaptive agents employing alternate communication routines. *JASSS*, 3(2), March 2000.

[15] Jim Doran. Agent-Based Modelling of Ecosystems for Sustainable Resource Management. In Michael Luck, Vladimír Mařík, Olga Štěpánková, and Robert Trappl, editors, *Multi-Agent Systems and Applications: 9th ECCAI Advanced Course, ACAI 2001 and Agent Link's 3rd European Agent Systems Summer School, EASSS 2001 Prague, Czech Republic, July 2–13, 2001 Selected Tutorial Papers*, Lecture Notes in Computer Science, pages 383–403. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[16] Pongchai Dumrongrojwatthana, Christophe Le Page, Nantana Gajaseni, and Guy Trébuil. Co-constructing an agent-based model to mediate land use conflict between herders and foresters in northern Thailand. *Journal of Land Use Science*, 6(2-3):101–120, June 2011.

[17] Roy Gardner, Elinor Ostrom, and James M. Walker. The Nature of Common-Pool Resource Problems. *Rationality and Society*, 2(3):335–358, July 1990.

[18] Clark C. Gibson, John T. Williams, and Elinor Ostrom. Local Enforcement and Better Forests. *World Development*, 33(2):273–284, February 2005.

[19] Garrett Hardin. The Tragedy of the Commons. *Science*, 162(3859):1243–1248, December 1968.

[20] Tanya M. Hayes. Parks, People, and Forest Protection: An Institutional Assessment of the Effectiveness of Protected Areas. *World Development*, 34(12):2064–2075, December 2006.

[21] Daniel Hirschman. Stylized Facts in the Social Sciences. *Sociological Science*, 3:604–626, July 2016.

[22] Josef Hofbauer, Karl Sigmund, and Professor Institute of Mathematics Karl Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, May 1998.

[23] Marco Janssen and Elinor Ostrom. Empirically Based, Agent-based models. *Ecology and Society*, 11(2), December 2006.

[24] Hamdi Kavak, Jose J. Padilla, Christopher J. Lynch, and Saikou Y. Diallo. Big Data, Agents, and Machine Learning: Towards a Data-driven Agent-based Modeling Approach. In *Proceedings of the Annual Simulation Symposium*, ANSS '18, pages 12:1–12:12, San Diego, CA, USA, 2018. Society for Computer Simulation International. event-place: Baltimore, Maryland.

[25] Mark Kinver. World 'losing battle against deforestation'. *BBC News*, September 2019.

[26] C M Macal and M J North. Tutorial on agent-based modelling and simulation. *Journal of Simulation*, 4(3):151–162, September 2010.

[27] Tyson Macaulay. Chapter 12 - Threats and Impacts to the IoT. In Tyson Macaulay, editor, *RIoT Control*, pages 221–278. Morgan Kaufmann, Boston, January 2017.

[28] Jonathan Mussman. Cod is Dead, 2018.

[29] Sanchayan Nath. Leadership and Trade-Offs Between Social-Ecological Outcomes in Community-Governed Forests. *Society & Natural Resources*, 31(7):790–806, July 2018.

[30] Elinor Ostrom. *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press, November 1990. Google-Books-ID: 4xg6oUobMz4C.

[31] Lauren Persha, Arun Agrawal, and Ashwini Chhatre. Social and Ecological Synergy: Local Rulemaking, Forest Livelihoods, and Biodiversity Conservation. *Science*, 331(6024):1606–1608, March 2011.

[32] Irene Pérez and Marco A. Janssen. The effect of spatial heterogeneity and mobility on the performance of social–ecological systems. *Ecological Modelling*, 296:1–11, January 2015.

[33] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.

[34] Caroline Schill, Nanda Wijermans, Maja Schlüter, and Therese Lindahl. Cooperation Is Not Enough—Exploring Social-Ecological Micro-Foundations for Sustainable Common-Pool Resource Use. *PLOS ONE*, 11(8):e0157796, August 2016.

[35] Michael Schoon and Sander Van der Leeuw. The shift toward social-ecological systems perspectives: insights into the human-nature relationship. *Natures Sciences Societes*, Vol. 23(2):166–174, July 2015.

[36] Jule Schulze, Birgit Müller, Jürgen Groeneveld, and Volker Grimm. Agent-Based Modelling of Social-Ecological Systems: Achievements, Challenges, and a Way Forward. *Journal of Artificial Societies and Social Simulation*, 20(2):8, 2017.

[37] Alex Smajgl, Daniel G. Brown, Diego Valbuena, and Marco G. A. Huigen. Empirical characterisation of agent behaviours in socio-ecological systems. *Environmental Modelling & Software*, 26(7):837–844, July 2011.

[38] Chanyut Sudtongkong and Edward Webb. Outcomes of State- vs. Community-Based Mangrove Management in Southern Thailand. *Ecology and Society*, 13, December 2008.

[39] Jan C. Thiele, Winfried Kurth, and Volker Grimm. RNetLogo: An R package for running and exploring individual-based models implemented in NetLogo. *Methods in Ecology and Evolution*, 3(3):480–483, 2012.

[40] 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.

[41] Frank Van Laerhoven. Governing community forests and the challenge of solving two-level collective action dilemmas—A large-N perspective. *Global Environmental Change*, 20(3):539–546, August 2010.

[42] Craig Welch. How Amazon forest loss may affect water—and climate—far away, August 2019.

[43] Gregory Wheeler. Bounded Rationality. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2019 edition, 2019.

[44] Hadley Wickham. *tidyverse: Easily Install and Load the 'Tidyverse'*, 2017. R package version 1.2.1.

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

[46] U. Wilensky. NetLogo Wolf Sheep Predation (Docked Hybrid) model. *Center for Connected Learning and Computer-Based Modeling*, Northwestern University, Evanston, IL, 2005.

[47] Moira L. Zellner, Leilah B. Lyons, Charles J. Hoch, Jennifer Weizeorick, Carl Kunda, and Daniel C. Milz. Modeling, Learning, and Planning Together: An Application of Participatory Agent-based Modeling to Environmental Planning. *Journal of the Urban & Regional Information Systems Association*, 24(1):77–92, January 2012.

[48] Haifeng Zhang, Yevgeniy Vorobeychik, Joshua Letchford, and Kiran Lakkaraju. Data-driven agent-based modeling, with application to rooftop solar adoption. *Autonomous Agents and Multi-Agent Systems*, 30(6):1023–1049, November 2016.