



# Midterm Presentation Attendance

## Report 2

### on Simulating the Effects of Environmental Complexity in Agent-based Modeling

**report submitter:** Mamanchuk Mykola, SID.202420671  
**for speaker:** Alison R., SID.202320694

August 12, 2024

## 1 Introduction to Research

The introduced concept of **Agent-Based Modeling and Simulation (ABMS)** is a computational approach used to model systems as simulations of autonomously interacting agents. This method is particularly valuable in scenarios where direct experimentation on the systems is impractical or difficult. ABMS has been employed across various domains to better understand complex systems, **particularity where the behavior of the whole system emerges from the interactions of individual agents.**

Referred interactions can be of any complexity, and they drive the overall dynamics of the system being studied. The flexibility of ABMS allows it to be applied to a wide range of systems, which is demonstrated by the research of the speaker.

**A common challenge in ABMS is that many aspects of the real systems are often simplified or abstracted away,** usually due to computational constraints. Introduced **research under review seeks to explore whether reintroducing some of these complexities into the simulation can yield more accurate or insightful results.** By starting with a simple model and gradually adding complexity, the study aims to analyze the effects of these changes on the simulation outcomes, providing a deeper understanding of the systems being modeled.

## 2 Chosen System for Agent-Based Simulation

The **wolf-sheep predation model** is a well-known agent-based simulation that illustrates the interactions within a simple predator-prey ecosystem. This model is **primarily**

used to explore the stability of ecosystems, where stability is defined as the state of equilibrium where no species becomes extinct over time.

## 2.1 Simulation Environment and Agents

**Environment:** The simulation environment is a two-dimensional grid where each cell may contain edible grass, represented visually by different colors. Grass is an essential resource for sheep in the simulation, as it provides the energy needed for their survival.

**Agents:**

- **Sheep:** Each sheep has an energy property that depletes over time and can be replenished by consuming grass. Sheep move randomly across the grid, and if their energy reaches zero, they are removed from the simulation.
- **Wolves:** Wolves function similarly to sheep but must consume sheep to restore their energy. The energy of the wolf increases when it consumes a sheep, which is then removed from the simulation.

## 2.2 Interactions and Dynamics

The **interactions** within this model are based on the consumption of resources:

- **Sheep and Environment:** Sheep consume grass to gain energy, and grass regrows over time.
- **Wolf and Sheep:** Wolves prey on sheep, reducing the sheep population while increasing their energy.

**Reproduction** is another critical aspect here:

- Agents may reproduce if they have sufficient energy, leading to the creation of offspring with an energy level that is a random value, contributing to the dynamics of the population.

## 2.3 Parameters and Phases

Several parameters control the simulation, including:

- Grass growth rate and energy provided
- Initial population counts and maximum energy levels
- Energy consumption rates and reproduction probabilities

These parameters influence the system's dynamics and outcomes, providing a means to test different scenarios and environmental complexities.

## 2.4 Implementation and Simulation Phases

The **research reimplements the wolf-sheep predation model** using the MASON toolkit, allowing for **enhanced complexity and control** over the simulation. The process begins with initializing the environment and agent populations, followed by the simulation of interactions over time, driven by the defined parameters.

This implementation provides a platform to **explore how variations in environmental factors**, such as terrain and resource availability, **affect the stability and behavior** of the ecosystem.

## 3 Simulation Phases

The following two figures describe the initialization and simulation phases of the system.

---

**Algorithm 1** Wolf-Sheep Predation System Initialization

---

- 1: **Initialization:**
  - 2: Set *grassInitRatio*  $\leftarrow$  0.2
  - 3: Set *grassRegrowTime*  $\leftarrow$  60
  - 4: Set *grassEnergyAdded*  $\leftarrow$  5
  - 5: Set *wolfInitCount*  $\leftarrow$  50
  - 6: Set *wolfMaxEnergy*  $\leftarrow$  100
  - 7: Set *wolfEnergyConsumption*  $\leftarrow$  1
  - 8: Set *wolfReproduceEnergyThreshold*  $\leftarrow$  50
  - 9: Set *wolfReproduceRate*  $\leftarrow$  5%
  - 10: Set *sheepInitCount*  $\leftarrow$  50
  - 11: Set *sheepMaxEnergy*  $\leftarrow$  40
  - 12: Set *sheepEnergyConsumption*  $\leftarrow$  1
  - 13: Set *sheepReproduceEnergyThreshold*  $\leftarrow$  20
  - 14: Set *sheepReproduceRate*  $\leftarrow$  5%
  - 15: **Proceed with Simulation:** Start the environment simulation algorithm
- 

**Initialization phase** of a Wolf-Sheep Predation System involves setting up initial parameters for the simulation environment, such as the initial ratio of cells containing grass, the time it takes for grass to regrow, and the energy gained by sheep from eating grass. Additionally, it specifies parameters for both wolf and sheep agents, including their initial counts, maximum energy, energy consumption rates, reproduction thresholds, and reproduction rates. After initializing these parameters, the simulation proceeds with the environment simulation algorithm where these agents interact based on the defined rules.

**Simulation phase** describes the simulation process for the System respectively. The simulation starts by initializing a grid environment where cells may contain grass and randomly placing wolf and sheep agents within this grid.

During each simulation tick, the environment and agents are updated as follows:

- **Update Environment:** Each cell in the grid is checked, and if it contains grass, the grass regrows after a specified number of simulation ticks (*grassRegrowTime*).
- **Update Sheep Agents:** Each sheep agent moves to a random neighboring cell. If the cell contains grass, the sheep eats it, gaining energy (*grassEnergyAdded*). The

sheep’s energy decreases by a set amount (`sheepEnergyConsumption`) each tick. If a sheep’s energy falls to zero or below, it is removed from the simulation. Sheep that have energy above a certain threshold (`sheepReproduceEnergyThreshold`) may reproduce with a specified probability (`sheepReproduceRate`).

- **Update Wolf Agents:** Each wolf agent also moves to a random neighboring cell. If the cell contains a sheep, the wolf eats it, gaining energy equal to the sheep’s energy. Similar to sheep, the wolf’s energy decreases by a set amount (`wolfEnergyConsumption`) each tick. Wolves with energy levels at or below zero are removed from the simulation. Wolves that have energy above a certain threshold (`wolfReproduceEnergyThreshold`) may reproduce with a specified probability (`wolfReproduceRate`).

The simulation continues through these updates for each tick, and once complete, population data is collected and analyzed. This process models the dynamics of predator-prey interactions within an ecosystem. The algorithm can be viewed in the Appendix A.

## 4 Complexity Introduction

Further, researcher explores how the introduction of additional complexity in the model affects simulation results. The **experiment modifies the model by adding a new property to each cell in the grid, representing terrain height**, and modifying how neighbors are defined based on this property.

### 4.1 Environment Changes

The modified model introduces terrain height as an additional environmental factor. Each cell in the grid is assigned a height value, with lower heights represented by darker colors. This terrain height affects the movement of agents by creating barriers; cells connected to other cells with a height difference greater than one unit are considered non-neighbors, reducing the number of possible movements.

### 4.2 Experiment Setup

To simulate real-world environmental complexity, the heightmap of a Minecraft world is used as a substitute for actual terrain. This heightmap provides sufficient complexity and naturalness while allowing for easy visualization. The environment is generated with specific parameters, and the simulation is run to observe how the added geographical barriers affect the interaction dynamics between wolves and sheep.

#### 4.2.1 Methodology

The simulation is conducted with the same parameters as the basic model, with the added complexity of terrain. Two different terrain heightmaps are used to study the effects of these barriers on population dynamics. Population charts are generated from the simulation runs, and the results are interpreted.

#### 4.2.2 Experiment entry 1: Flat Terrain with Geological Barriers (A)

Terrain A is a relatively flat area with a steep geographical barrier towards the sea. The addition of geographical movement barriers prevents the extinction of sheep agents, **leading to a more stable ecosystem**. The sheep agents find safe habitats in areas protected by cliffs, where they can grow undisturbed by wolves. Periodically, some sheep wander out of these habitats and grow their population until they encounter wolves, leading to a cyclical pattern of population growth and decline.

#### 4.2.3 Experiment Entry 2: Terrain with Steeper Contours but no Distinct Barriers (B)

Terrain B has steeper contours but lacks extreme geographical barriers like Terrain A. The simulation on Terrain B demonstrates frequent near-extinctions of sheep agents, but they manage to rebound each time, **leading to cyclical population dynamics**. The sheep agents' ability to escape wolves by navigating slight geographical features prevents total extinction, leading to a more dynamic but less stable ecosystem.

### 4.3 Key Observations

The results of these experiments suggest that **adding environmental complexity significantly impacts the outcomes of agent-based simulations**. Terrain A's geographical barriers create safe zones for sheep agents, allowing them to grow undisturbed, whereas Terrain B's lack of significant barriers leads to more dynamic but cyclical population changes. These findings provide insights into how real-world geographical features might influence population stability and predator-prey dynamics in ecosystems.

## 5 Future Prospects

In the "Future Work" section, the researcher emphasizes the importance of exploring more complex models to better understand the effects of increased simulation complexity on ABMS. The **experiments suggest that adding complexity to simulations can provide deeper insights into the systems being modeled**, making the models more representative of real-world scenarios. The researcher plans to further investigate the introduction of more complex agent behaviors and adaptive agents, aiming to create simulations that closely mimic real-world dynamics.

Additionally, it is acknowledged that increased complexity in simulations will require more computational resources. To address this, future research will explore strategies such as parallelization and optimization techniques to manage the computational demands, ensuring that the simulations remain efficient and scalable as complexity grows.

## References

1. Mamanchuk N., University of Tsukuba, Github, August 12, 2024. Available online: <https://github.com/RIFLE>
2. Alison R., University of Tsukuba, on Simulating the Effects of Environmental Complexity in Agent-based Modeling, 2024. Available online: <https://www.cs.>

[tsukuba.ac.jp/lecture/midterm/local/24\\_data/mov/202320694.mp4](https://tsukuba.ac.jp/lecture/midterm/local/24_data/mov/202320694.mp4) [Accessed:  
2024-08-12]

## Appendix A. WS Simulation Stage Algorithm

---

**Algorithm 2** Wolf-Sheep Predation System Simulation

---

```
1: Simulation Start:
2: Initialize the grid environment with cells containing grass
3: Place wolf and sheep agents randomly on the grid
4: for each simulation tick do
5:   Update Environment:
6:   for each cell in the grid do
7:     if cell contains grass then
8:       grass regrows after grassRegrowTime ticks
9:     end if
10:  end for
11:  Update Agents:
12:  for each sheep agent do
13:    Move to a random neighboring cell
14:    if cell contains grass then
15:      Eat grass and increase energy by grassEnergyAdded
16:    end if
17:    Decrease energy by sheepEnergyConsumption
18:    if energy  $\leq 0$  then
19:      Remove sheep from simulation
20:    end if
21:    if energy  $\geq$  sheepReproduceEnergyThreshold then
22:      Reproduce with probability sheepReproduceRate
23:    end if
24:  end for
25:  for each wolf agent do
26:    Move to a random neighboring cell
27:    if cell contains a sheep then
28:      Eat sheep and increase energy by sheep's energy
29:    end if
30:    Decrease energy by wolfEnergyConsumption
31:    if energy  $\leq 0$  then
32:      Remove wolf from simulation
33:    end if
34:    if energy  $\geq$  wolfReproduceEnergyThreshold then
35:      Reproduce with probability wolfReproduceRate
36:    end if
37:  end for
38: end for
39: End Simulation: Collect and analyze population data
```

---