Agents

Agents are the “people” in an artificial society

- each agent has attributes and states
- the attributes are static, fixed for the lifetime of the model (e.g., metabolic rate, vision, gender)
- the states are dynamic, changing in time (e.g., location, wealth, health)

Landscape

The environment in which agents operate, with which they interact.

- a 2D grid of cells containing renewable resources
- each cell has attributes and states
- the attributes are static, fixed for the lifetime of the model (e.g., capacity, regrowth rate)
- the states are dynamic, changing in time (e.g., resource level)

Landscape Coordinate System

An \((x, y)\) integer-valued spatial coordinate system identifies cells

Rotate the coordinate system 90 degrees relative to convention

- \(x\) iterates rows, ↓
- \(y\) iterates columns, →

Characterizing the Landscape

The landscape is an \(X \times Y\) array so that

\[x = 0, 1, 2, \ldots, X - 1 \quad \text{and} \quad y = 0, 1, 2, \ldots, Y - 1\]

(For all figures in Chapter 2, \(X = Y = 50\))

Each \((x, y)\) landscape cell is characterized by:

- \(a(x, y)\): occupancy status of the cell — 1 if occupied, 0 otherwise
- \(r(x, y)\): current resource level at the cell
- \(\rho(x, y)\): resource regrowth rate of the cell
- \(\gamma(x, y)\): resource capacity of the cell

State variables: \(a(x, y)\) and \(r(x, y)\)

Attributes: \(\rho(x, y)\) and \(\gamma(x, y)\)

In the book, \(r(x, y), \rho(x, y), \) and \(\gamma(x, y)\) are integer-valued — why?
Characterizing the Landscape (Cont.)

Each \((x, y)\) landscape cell is characterized by four states/attributes:

- \(\alpha(x, y)\): naturally Boolean (0 or 1)
- \(r(x, y)\): real-valued, non-negative
- \(\rho(x, y)\): real-valued, non-negative
- \(\gamma(x, y)\): real-valued, non-negative

In the book, \(r(x, y)\) is integer-valued with no real justification. \(\rho(x, y)\) and \(\gamma(x, y)\) should agree in type with \(r(x, y)\).

Landscape Resource Capacity

The book uses only one capacity model, but never defines that model.

The model appears to be the two-peak Gaussian

\[
\gamma(x, y) = f(x - X/4, y - Y/4) + f(x - 3X/4, y - 3Y/4)
\]

where

\[
f(x, y) = \psi \exp \left( -\frac{(x/\theta_x)^2 + (y/\theta_y)^2}{2} \right)
\]

with \(\psi = 4.0\), \(\theta_x = 0.3X\) and \(\theta_y = 0.3Y\).

- Is this equation correct?
- If so, are \(\theta_x\) and \(\theta_y\) correct?
- Note \(\gamma(x, y)\) is integer-valued in the book.

Characterizing the Agents

In accordance with behavioral rules, agents move about the landscape gathering and consuming resources.

At any time \(t\), each agent is characterized by:

- \(x(t)\): vertical location at time \(t\)
- \(y(t)\): horizontal location at time \(t\)
- \(u(t)\): resource wealth at time \(t\)
- \(\mu\): metabolic rate
- \(\phi\): vision

State variables \(x(t), y(t),\) and \(u(t)\) are non-negative.

Attributes: \(\mu\) and \(\phi\)

In the book, \(u(t)\) and \(\mu\) are integer-valued — why?

Agent Field of View (FOV)

An agent with vision \(\phi\) can see \(\phi\) cells in the four primary directions.

This collection of \(4\phi\) cells defines an agent’s field of view (FOV).

\(\phi = 3\), if \(\phi = 3\) the agent can move at most 3 units N, S, E or W.
Landscape Regrowth Rule

Rule G: at each \((x,y)\) cell, resource regrows at a rate of \(\rho(x,y)\) units per time interval up to cell capacity.

- an \(\alpha-\epsilon\) rule
- regrowth rate \(\rho(x,y)\) is an attribute
- no inherent reason why \(\rho(x,y)\) can’t depend on cell location (like \(\gamma(x,y)\))

In other words, the resource level at cell \((x,y)\) at time \(t+1\) is

\[
\min \{ \gamma(x,y), r(x,y) + \rho(x,y) \}
\]

where \(r(x,y)\) is the resource level at time \(t\)

\(O(AXY)\) time complexity at each time step

Agent Movement Rule

Rule M: an \(\alpha-\epsilon\) rule

- Look at all cells within FOV defined by \(\phi\)
- Select closest unoccupied cell with maximum resource (break ties randomly)
- Move to the selected cell and collect all the resource

If at time \(t\)

- agent is at cell \((x,y)\)
- agent possesses resource wealth \(w(t)\)
- closest unoccupied cell with max resource is \((x',y')\)

then at time \(t+1\)

- agent moves to \((x',y')\)
- \(w(t+1) = \max\{0, w(t) + r(x',y') - \mu\}\)
- if \(w(t+1) = 0\), the agent dies

\(O(AX)\) complexity at each time step, where \(A\) is number of agents

More Algorithms

What other algorithms are needed at the implementation level?

For each \(t = 1, 2, \ldots\) four algorithms are used in the following order:

- Move the agents consistent with rule M
- Update the landscape consistent with rule G
- Update the agent list, check for any agent deaths, and modify the agent list accordingly
- Shuffle the agent list to randomize the sequential order in which agents will move in the next time step

Evolution of the Model

Given the parameter \(T\) and state variable \(\text{alive}\) where

- \(T\) denotes the number of simulated time steps
- \(\text{alive}\) counts the number of living agents

the artificial society evolves in time according to

\[
\text{// initialize the landscape} \quad * /
\]
\[
\text{// initialize the agents} \quad * /
\]
\[
\text{alive} = A; \quad t = 0;
\]
\[
\text{while} \quad (t < T) \quad \{ \quad * /
\]
\[
\text{// move the agents} \quad * /
\]
\[
\text{// update the landscape} \quad * /
\]
\[
\text{// update the agent list} \quad * /
\]
\[
\text{// randomize (shuffle) the agent list} \quad * /
\]
\[
\text{t++;}
\]
\[
\} \quad * /
\]
\[
\text{// output the value of ‘alive’} \quad * /
\]

For each update of the agent list, \(\text{alive}\) is decreased appropriately