

# Alarm Model Paper

*Aviva Blonder*

*October 16, 2019*

## Agent-Based Model (ODD)

### Overview

### Purpose

This model aims to investigate under what ecological conditions social learning should be favored by natural selection over individual learning in the context of heterospecific alarm call recognition. Based on mathematical analyses (discussed above), social learning should always result in increased responsiveness to alarms, decreasing the consumptive costs of predation, but increasing the non-consumptive costs of missed foraging opportunities. This model directly compares the very different costs imposed by predation and missed foraging opportunities.

### State Variables and Scales

The model is composed of a community of animals: agents, predators, and alarm-calling heterospecifics, that are placed in a two dimensional habitat. Agents are characterized by their learning strategy (vigilance only, individual, or social), lifespan, age, location, amount of resources accumulated since last reproduction, whether they are fleeing or not, the direction in which they are fleeing, and learning parameters (depending on the learning algorithm being used in that simulation). Predators are characterized by the number of steps since they last caught an agent and their location. Heterospecifics are characterized by whether they are alarm-calling or not and their location. The habitat has a carrying capacity that limits the number of agents, predators, and heterospecifics that can be alive at any given time. (See *Initialization* for a complete list of parameters and state variables.)

### Process Overview and Scheduling

Time is modeled in discrete steps. On each step, all agents, predators, and heterospecifics execute their behaviors and update their state variables.

[insert figure]

### Design Concepts

*Emergence: Which system-level phenomena truly emerge from individual traits, and which phenomena are merely imposed?*

Table 1: Transformations Associated with the Johnson System

Johnson Family	Transformation	Parameter Conditions	X Condition
$S_B$	$Z = \gamma + \eta \ln\left(\frac{X-\epsilon}{\lambda+\epsilon-X}\right)$	$\eta, \lambda > 0, -\infty < \gamma, \epsilon < \infty$	$\epsilon < X < \epsilon + \lambda$
$S_L$	$Z = \gamma + \eta \ln(X - \epsilon)$	$\eta > 0, -\infty < \gamma, \epsilon < \infty$	$X > \epsilon$
$S_U$	$Z = \gamma + \eta \sinh^{-1}\left(\frac{X-\epsilon}{\lambda}\right)$	$\eta, \lambda > 0, -\infty < \gamma, \epsilon < \infty$	$-\infty < X < \infty$

*Adaptation: What adaptive traits do the model individuals have which directly or indirectly can improve their potential fitness, in response to changes in themselves or their environment?* The only trait allowed to evolve is the agents' learning strategy, which is passed on from parent to offspring with some probability of mutation.

*Fitness: Is fitness-seeking modelled explicitly or implicitly? If explicitly, how do individuals calculate fitness (i.e., what is their fitness measure)?* Fitness is modeled implicitly based on a combination of foraging success and predator avoidance. All living agents have the opportunity to accumulate resources by foraging on each step with some probability of success based on whether or not they are also trying to avoid predators. After enough resources are accumulated an agent will attempt to reproduce, though they only succeed if the population size has not exceeded the environment's carrying capacity. If agents die before the end of their natural lifespan, they lose all remaining opportunities to reproduce.

*Prediction: In estimating future consequences of their decisions, how do individuals predict the future conditions they will experience?* Agents learn from experience whether to focus on foraging or attempt to evade predators. Social and individual learners learn whether to attend to alarms as well as how often to be vigilant in the absence of an alarm based on whether or not predators are present under the two conditions. Social learners also learn from the responses of others - i.e. they are reinforced to flee when they see others fleeing and reinforced to forage when they see others foraging. Vigilance only agents ignore all alarm calls and just learn how often to be vigilant based on how often predators are present.

*Sensing: What internal and environmental state variables are individuals assumed to sense or "know" and consider in their adaptive decisions?* Social and individual learners can detect alarm calls and use those to determine whether to be vigilant on a given step. Vigilance only agents do not detect alarms and just determine whether to be vigilant based on the accumulation of all of their previous experience.

*Interaction: What kinds of interactions among individuals are assumed?* Social and Individual learners can attend to the alarm calls produced by heterospecifics. All agents learn from the presence of predators, and social learners can also observe the behavior of others and learn from them. Heterospecifics can detect predators and produce alarms calls. Predators can detect agents, which they pursue and kill.

*Stochasticity: Is stochasticity part of the model? What are the reasons?* Almost all outcomes are determined stochastically based on set probabilities to account for the variability found in life. This includes agents' foraging success, the likelihood of mutations in their offspring, predators' success in catching an agent once they have landed on it, and agents' behavior - whether they focus on foraging or attempt to evade predators.

*Observation: How are data collected from the IBM for testing, understanding, and analyzing it?* Whenever an observed variable changes, it is updated for the simulation as a whole (see table for more details). The metrics for the simulation are then recorded at set intervals.

Variable	Observation
Number of individuals (by strategy)	The count for each learning strategy is incremented when an agent with that strategy is born and decremented whenever an agent with that strategy dies.
Response to alarm (by strategy)	If an agent detects an alarm, its previous probability of fleeing in the presence of an alarm is subtracted from the total for its learning strategy, and its new probability is added to the total. Before data is recorded, it is divided by the total number of individuals of that learning strategy that have been in the simulation so far to get an average.
Response to the absence of an alarm (by strategy)	If an agent does not detect an alarm, its previous probability of foraging in the absence of an alarm is subtracted from the total for its learning strategy, and its new probability is added to the total. Before data is recorded, it is divided by the total number of individuals of that learning strategy that have been in the simulation so far to get an average.
Cost of predation (by strategy)	If an agent dies from predation, the cost of predation for its learning strategy is incremented. Before data is recorded, it is divided by the total number of individuals of that learning strategy that have been in the simulation so far to get an average.

Variable	Observation
Cost of not foraging (by strategy)	If an agent does not successfully forage, the cost of not foraging for its learning strategy is incremented. Before data is recorded, it is divided by the total number of individuals of that learning strategy that have been in the simulation so far to get an average.
Reproductive success (by strategy)	If an agent successfully reproduces, the reproductive success for its learning strategy is incremented. Before data is recorded, it is divided by the total number of individuals of that learning strategy that have been in the simulation so far to get an average.
Lifespan (by strategy)	When an agent dies, its age is added to the total lifespan for its learning strategy. Before data is recorded, it is divided by the total number of individuals of that learning strategy that have been in the simulation so far to get an average.
Predation rate	If an agent dies from predation, the total predation rate is incremented. Before data is recorded, it is divided by the total number of steps so far to get an average.
Reproduction rate	If an agent successfully reproduces, the total reproduction rate is incremented. Before data is recorded, it is divided by the total number of steps so far to get an average.
False alarm rate	If an agent detects an alarm, but neither a predator nor a fleeing agent, the total false alarm rate is incremented. Before data is recorded, it is divided by the total number of steps so far to get an average.

## Details

### Initialization

First, all of the test parameters were initialized according to an input file.

Parameter	Initialization
Total steps	30000
Result recording interval	150 steps
Random seed	Integer from 1 to 10 (inclusive)
Grid dimensions	100 X 100 squares
Predator frequency (relative to agents)	Value between 0 and 1
Steps between predation attempts	Integer from 1 to 30000
Heterospecific frequency (relative to agents)	.1
Probability that a heterospecific will give an alarm call if it detects a predator	1
Probability that a heterospecific will give an alarm call if it does not detect a predator	0
Moore distance at which agents can detect predators	2 squares
Moore distance at which agents can detect alarms	4 squares
Moore distance at which agents can detect other agents fleeing	2 squares
Moore distance at which predators can detect agents	2 squares
Moore distance at which heterospecifics can detect predators	4 squares
Maximum population size	1000 agents
Probability that an agent will flee independent of learning	.001
Whether there will be vigilance only agents in that simulation	True or False
Initial proportion of vigilance only agents	Value between 0 and .334
Initial proportion of individual learning agents	Value from .333 to .5
Initial proportion of social learning agents	Value from .333 to .5
Mutation rate	.001

Parameter	Initialization
Foraging success threshold for reproduction	25
Probability of foraging successfully while not fleeing	Value between 0 and 1
Probability of successfully evading a predator while fleeing	Value between 0 and 1
How much lower the probability of foraging successfully is while fleeing	Value between 0 and 1
Mean lifespan	125 steps
Lifespan range	50 steps
Learning algorithm	Rescorla-Wagner or Neural Network
Learning rate	.01

When a new model was constructed, the observed variables were also all initialized to zero.

Animals were then initialized according to the provided maximum population size, predator frequency, and heterospecific frequency.

FUNCTION INITIALIZE\_ANIMALS:

```

FOR i = 1 to maximum population size:
  IF i < maximum population size * initial proportion of vigilance only agents:
    CREATE vigilance only Agent
  ELSE IF i < maximum population size * (initial proportion of vigilance only agents + initial proportion of individual learning agents):
    CREATE individual learning Agent
  ELSE:
    CREATE social learning Agent
END FOR

FOR i = 1 to predator frequency * maximum population size:
  CREATE: Predator
END FOR

FOR i = 1 to heterospecific frequency * maximum population size:
  CREATE: Heterospecific
END FOR

```

END INITIALIZE\_ANIMALS

When a new Animal was created, its state variables were initialized.

Variable	Initialization
<b>All Animals</b>	
Location	x and y coordinates drawn from a uniform distribution over the length and width of the grid
<b>Agent</b>	
Response to alarm	.5
Response to the absence of an alarm	.5
Accumulated resources	0
Lifespan	Drawn from a normal distribution with standard deviation equal to the provided lifespan range, and mean equal to the provided mean lifespan.
Age	0
Neural network weights	Two arrays of values drawn from a normal distribution with standard deviation .1 and mean 0.
Rescorla-Wagner stimulus strength	.5 (for both alarms and the absence of an alarm)
Whether the agent is fleeing	False

Variable	Initialization
<b>Predator</b>	
Steps since last successful predation attempt	0
<b>Heterospecific</b>	
Whether it is producing an alarm call	False

## Input

The model was run under a range of pre-set conditions described in the experiments and *Initialization* above, which did not change over the course of a simulation. The random number generator for each simulation was seeded with values from 0 to 10 for each replicate.

## Submodels

**Aging:** Each step, each agent's age is incremented by one. When an agent's age exceeds its predetermined lifespan, the agent is considered dead and removed from the simulation.

**Detection:** Agents can detect predators, alarm calling heterospecifics, and other agents. Predators can detect agents, and heterospecifics can detect predators. All of these processes use the same function, which takes in the class to search for (agent, predator, or heterospecific), and whether to specifically look for a signalling individual (such as a fleeing agent or an alarm calling heterospecific) or any individual, and returns the location of the nearest such individual, if one is present within the searching animal's detection range. Using Mason's getMooreNeighbors function, it iteratively checks all occupied squares within the searching animal's detection range, and stores the nearest individual and its location. If specifically looking for signalling individuals, it only stores the nearest signalling individual. Each subsequent individual is compared to the nearest one, and, if closer, replaces it as the nearest, until all squares have been checked.

### FUNCTION DETECT:

```

INITIALIZE nearest to empty
FOR n = each object in this Animal's detection range:
  IF n is of the type being searched for:
    IF this Animal isn't looking for a signalling Animal OR n is signalling:
      IF nearest is empty OR the Euclidian distance to n < Euclidian distance to nearest:
        SET nearest = n
END FOR
RETURN nearest
END DETECT

```

**Moving:** Agents can move to flee from Predators and Predators can move to pursue Agents. The Animal's x and y coordinates are incremented or decremented by 1 in accordance with the sign of the provided distance on that axis. If the distance is 0, then the Animal moves randomly. If the Animal's x or y value exceeds the height or width of the grid, it moves in the opposite direction on that axis.

### FUNCTION MOVE:

```

IF the provided distance = 0 on both axes:
  Move one square in a random direction.
ELSE:
  IF the provided x distance is positive:
    Increment this Animal's x coordinate by 1
  ELSE IF the provided x distance is negative:
    Decrement this Animal's x coordinate by 1
  IF the provided y distance is positive:
    Increment this Animal's y coordinate by 1
  ELSE IF the provided y distance is negative:
    Decrement this Animal's y coordinate by 1

```

```

IF this Animal's x coordinate < 0:
    SET this Animal's x coordinate = 1
ELSE IF this Animal's x coordinate > 100:
    SET this Animal's x coordinate = 99
IF this Animal's y coordinate < 0:
    SET this Animal's y coordinate = 1
ELSE IF this Animal's y coordinate > 100:
    SET this Animal's y coordinate = 99
END MOVE

```

**Fleeing:** Agents calculate the probability of fleeing based on whether they detected an alarm calling heterospecific, using the learning algorithm set for that simulation (see *Neural Network* and *Rescorla-Wagner* below). The agent then decides whether to flee by drawing from a uniform distribution from 0 to 1. If the chosen value is less than the calculated probability of fleeing, it decides to flee, and otherwise it decides not to flee. If an agent decides to flee, it attempts to evade predators by moving. If it detects a predator, it moves in the opposite direction away from the nearest predator. Otherwise, if it detects an alarm call, it moves away from the nearest alarm calling heterospecific. If it neither detects a predator nor an alarm calling heterospecific, it moves in the same direction as the nearest fleeing conspecific. Otherwise, if it detects none of these, the agent moves randomly.

FUNCTION FLEE:

```

IF this Agent is vigilance only:
    Assume no alarm is present
ELSE:
    DETECT alarm calling (signalling) Heterospecifics

```

CALCULATE probability of fleeing using the learning algorithm, based on the whether an alarm was detected

IF a draw from a uniform distribution from 0 to 1 < probability of fleeing:

```

    INITIALIZE x and y distance = 0
    IF this Agent DETECTs a Predator:
        SET x distance = this Agent's x coordinate - the Predator's x coordinate
        SET y distance = this Agent's y coordinate - the Predator's y coordinate
    ELSE IF this Agent is not vigilance only AND DETECTs an alarm-calling Heterospecific:
        SET x distance = this Agent's x coordinate - the Heterospecific's x coordinate
        SET y distance = this Agent's y coordinate - the Heterospecific's y coordinate
    ELSE IF this Agent is a social learner AND DETECTs a fleeing Agent:
        SET x and y distance = that Agent's direction of movement
    MOVE according to distance

```

END FLEE

*Neural network:* In simulations using the neural network learning algorithm, agents make decisions and learn using a simple two layer neural network. The input layer contains only one simulated neuron, representing the agent's sensory apparatus, and the output layer contains two neurons, representing the agent's decision. The activation of the first output neuron represents a decision not to flee and the activation of the second output neuron represents the decision to flee. The detection of an alarm is represented as an input of 1 to the network, and the absence of an alarm is represented as an input of -1. The activation of the output neurons is then calculated based on the input, the learned weights from the input neuron to the output neurons, and the learned biases of the output neurons.

$$output_{i=0,1} = input * weight_i + bias_i$$

The agent's action is determined based on the relative activations of the output neurons, such that the probability of fleeing is:

$$P(flee) = \frac{e^{output_1}}{e^{output_0} + e^{output_1}}.$$

[INSERT FIGURE]

```
FUNCTION CALCULATE_NERUAL_NETWORK:
  IF an alarm-calling Heterospecific was detected:
    SET input = 1
  ELSE
    SET input = -1

  SET output 0 = input*weight 0 + bias 0
  SET output 1 = input*weight 1 + bias 1
  SET probability of fleeing = exp(output 1)/(exp(output 0) + exp(output 1))

  RETURN probability of fleeing
END FUNCTION DECIDE_NEURAL_NETWORK
```

*Rescorla-Wagner:* In simulations using the Rescorla-Wagner learning algorithm, Agents decide whether to flee based on the learned association between an alarm or the absence of an alarm and fleeing, depending on whether or not the Agent detected an alarm. The strength of the association is used as the probability of not fleeing, so that the probability of fleeing is:

$$P(flee) = 1 - association.strength.$$

```
FUNCTION CALCULATE_RESCORLA_WAGNER:
  IF an alarm-calling Heterospecific was detected:
    SET probability of fleeing = 1 - association strength of an alarm
  ELSE
    SET probability of fleeing = 1 - association strength of the absence of an alarm

  RETURN probability of fleeing
END CALCULATE_RESCORLA_WAGNER
```

**Foraging:** Each step, all agents have the opportunity to forage. The probability of foraging successfully is based on whether the agent decided to flee. If the agent is fleeing it has one probability of foraging successfully, and if it is foraging, it has an equal or greater probability of foraging successfully. These success rates are set for each simulation. If an agent forages successfully, it accumulates one unit of resource.

```
FUNCTION FORAGE:
  IF this Agent is fleeing:
    SET probability of foraging successfully = probability of foraging successfully while not fleeing -
  ELSE:
    SET probability of foraging successfully = probability of foraging successfully while not fleeing

  IF a draw from a uniform distribution from 0 to 1 < probability of foraging successfully:
    Increment the amount of resources accumulated by this Agent
END FORAGE
```

**Reproduction:** When an agent has accumulated a set amount of resources, it attempts to reproduce. If the population is below the carrying capacity, it succeeds, creating an agent with the same genotype (learning strategy) in a random location on the grid, with a set probability of mutating to another genotype.

```
FUNCTION REPRODUCE:
  IF the amount of resources accumulated by this Agent > threshold for reproduction:
    SET amount of resources accumulated by this Agent = 0
  IF total number of Agents in the population < maximum population size:
    IF a draw from a uniform distribution from 0 to 1 < mutation rate:
      CREATE a new Agent of a different learning strategy from its parent (randomly chosen)
    ELSE:
      CREATE a new Agent of the same learning strategy as its parent
```

END REPRODUCE

**Learning:** Each step, all agents learn based on the current state of their surroundings. Vigilance only agents just detect predators and increase their probability of fleeing if a predator is present, or otherwise decrease their probability of fleeing in accordance with the learning algorithm. Individual and social learning agents detect whether an alarm-calling heterospecific is present. If an agent detects an alarm call, it adjusts its likelihood of fleeing in the presence of an alarm, and if an agent does not detect an alarm call, it adjust its likelihood of fleeing in the absence of an alarm. Individual learning agents detect predators and increase their probability of fleeing if a predator is present and otherwise decrease their probability of fleeing, while social learning agents detect both predators and other fleeing agents, and increase their probability of fleeing if either is present, and otherwise decrease their probability of fleeing.

FUNCTION LEARN:

IF this Agent is vigilance only:

Assume no alarm is present

ELSE:

DETECT alarm calling (signalling) Heterospecifics

IF this Agent DETECTs Predators OR this is a social learning Agent and DETECTs fleeing Agents:

REINFORCE fleeing under the current conditions (whether an alarm is present or not) using the learn

ELSE:

REINFORCE not fleeing under the current conditions using the learning algorithm for this simulation

END LEARN

*Neural network:* In simulations using the neural network learning algorithm, Agents learn by adjusting the weights and biases of the neural network. Reinforcement is converted into an array that represents the ideal behavior under the provided circumstances. If the Agent is reinforced to flee, the ideal behavior is represented as (0,1), which would result in a nearly one hundred percent change of fleeing, and if it is reinforced not to flee, the ideal behavior is represented as (1,0), which would result in a nearly zero percent chance of fleeing. The ideal response is then compared with the actual output of the network under the circumstances, to calculate the error.

$$error_i = ideal_i - output_i$$

The error is used to adjust the weights and biases to each output node in accordance with the backpropagation equation.

$$\Delta weight_i = learning.rate * error_i * input$$

$$\Delta bias_i = learning.rate * error_i$$

FUNCTION REINFORCE\_NEURAL\_NETWORK:

IF this Agent DETECTs an alarm-calling Heterospecific:

SET input = 1

ELSE:

SET input = -1

IF this Agent is reinforced to flee:

SET ideal response = [0, 1]

ELSE:

SET ideal response = [1, 0]

FOR i = 0,1:

SET error i = ideal response i - output i

SET weight i = weight i + learning rate \* error i \* input

SET bias i = bias i + learning rate \* error



END FOR  
END REINFORCE\_NEURAL\_NETWORK

*Rescorla-Wagner:* In simulations using the Rescorla-Wagner learning algorithm, Agents learn by adjusting the strength of the association between the presence or absence of an alarm and not fleeing.

**Density check:** Each step, all predators and heterospecifics check to ensure that the ratio of their type (predators or heterospecifics) to agents is within one individual of the initial ratio. If there are too few individuals, a new individual is initialized at a random location, and if there are too many this individual is removed from the population.

**Predation:** When a predator has gone a designated number of steps without successfully catching an agent, it will detect agents and if successful, it will move toward the nearest agent, and otherwise moves randomly. If a predator lands on the same space as an agent, it attempts to catch the agent. If it succeeds, the agent is removed from the population and the predator's step counter is reset. Each predator can only catch one agent on each step.

**Alarm calling:** Each step, all heterospecifics attempt to detect predators. If successful, they signal that they are alarm calling.