Model description

Evolution of movement (and) social networks without landscape cues

# Aim and scope

I propose a simple agent based model and associated analyses aimed at understanding the development of social networks in the context of animal movement. The broad underlying question is, how can agents exploit the social information available to them — in the form of their neighbours’ success — to make the best move on a landscape that provides no cues to the location of resources?

This model is inspired by broad similarities in two very different scales of animal movement: long-distance migration, and local-scale foraging movements over inter-tidal mudflats. In both cases, individuals have a general notion of the correct direction of movement, but few clues as to prevailing conditions at their destination.

Nonetheless, agents must undertake movements that are crucial to survival, often without the possibility of accessing landscape cues or memory along the way; examples include migration of terrestrial birds over water, and the pre-foraging movement of waders over still-inundated mudflats.

When agents must repeatedly make such decisions over their lifetime, the condition of other agents might yield valuable information about the best possible move. This information may then be transferred between agents, allowing much more adaptation to changing landscape conditions than is possible through purely genetic evolution.

# Agents

## Agent movement and following

#### Agent movement

The model consists of a population with a constant size; each agent makes a single movement per timestep to travel on a one-dimensional landscape of a fixed size. Agents possess a fixed, heritable, and mutable movement parameter *Mi*, which determines how far from the origin (the high-tide roost, or breeding grounds) they will travel.

This movement parameter *Mi* may be temporarily overruled by a copied movement parameter *Mc*, which is the inherited movement parameter *Mi* of another individual. There is no cost to movement.

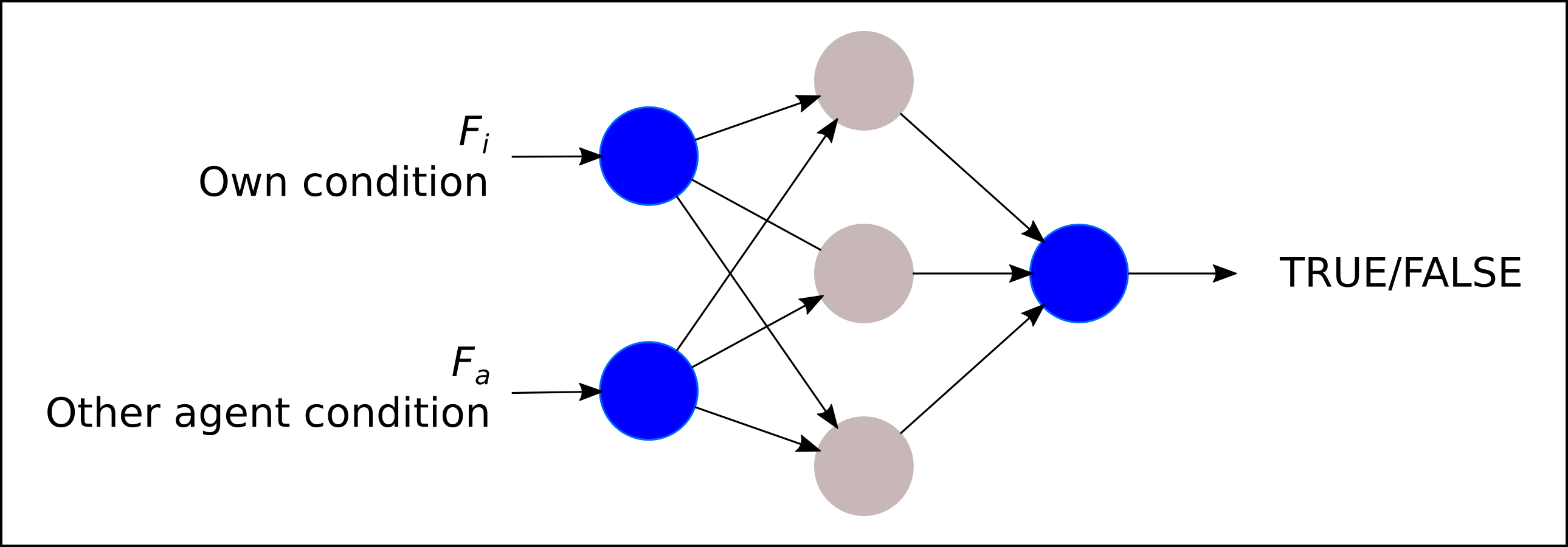
Should either Mi or Mc be greater than the maximum size of the landscape, agents are set on the outer edge of the landscape, where they extract resources (see below). Agents are assumed to return to the origin (0) after each timestep.

#### Agent following

Agents must thus make a single decision in each timestep: whether to follow another agent. Each agent independently and sequentially assesses every other agent in random order, until it decides to follow one of them, or until all agents have been assessed.

Agent assessment is performed by an artificial neural network with 2 input nodes, a single hidden layer of 3 nodes, and one output node. All node values are initialised at 0.

The two input nodes are provided the assessing agent’s own condition, *Fi*, and the assessed agent’s condition *Fa*; *Fi* and *Fa* are the summed intakes of the two agents respectively. These may be thought of as body condition, number of juveniles, return time, or some other indicator of foraging success.

  
Figure 1: Artificial neural network (ANN) architecture used in this model. Agents assess their own condition and that of every other agent in the population to determine whether or not one of the assessees should be followed. If none of the agents in the population is to be followed, the agent makes a move encoded by its inherited movement parameter Mi.

The single output node returns a floating point value which is assessed as either TRUE (follow) when greater than 0.0, and FALSE (do not follow) otherwise. When an agent chooses to follow another agent, its movement distance is the same as that used by its leader. When an agent rejects all other agents as leaders, its movement distance is determined by the movement parameter *Mi*.

Agents assess the population as described above in each timestep; the choice to follow another agent is not retained between timesteps, and must be made anew each time.

## Leadership resolution

#### Linear chain resolution

When a ‘leadership chain’ of more than two agents forms, i.e., agent 1 follows agent 2, which in turn follows agent 3, agent 1 effectively follows agent 3, the ‘ultimate leader’ of the chain, while retaining as its leader identity agent 2, the ‘proximate leader’.

This might correspond to a situation in which agent 1 physically follows agent 2, and does not simply copy a cue provided by agent 2. As long as agent 2 moves (following agent 3), agent 1 follows.

#### Loop resolution

A leadership chain may loop on itself since agents make their following decision independently, and agnostic to the follower status of other agents. To continue with the above example, a loop would form if agents 2 or 3 chose to follow agent 1.

Such a case would result in an infinite loop in a simulation. I propose to resolve leadership loops by choosing an ‘ultimate leader’ from among these agents at random, and linearising the loop from that agent onwards.

#### Timesteps

Agents cannot benefit from assessing each other if they can only do so once, and that at a stage when individual differences in condition are not evident. I propose to implement repeated trials, or iterations of the above procedure within each generation. Each iteration lasts a single timestep, after which agents are returned to the origin.

## Agent intake and reproduction

#### Intake

Agents position themselves on the landscape based on their used movement parameter (*Mi*, if independent, or *Mc*, if following another agent). This position is represented by a floating point number that is discretised by rounding down to the nearest integer, *x*.

Agent intake is calculated as the resource value at the *x*-th position of a sequence of landscape grid cells. When there are multiple agents at the *x*-th position on the landscape, they each receive an equal share of the resource. The sum of agent intake, or resources accumulated, is initialised at 0, and increases in each timestep as described above.

#### Reproduction

Agents are modelled as asexually reproducing, with a fixed population size. Total agent intake over a generation is used as a proxy for fitness, and is always >= 0. A new population is generated before the old one is destroyed.

The neural network node weights and inherited movement parameter of an agent *j* in the new population are set to be the same as those of an agent *i* in the old population, which is considered to be the parent. The probability of any agent *I* being chosen as the parent is proportional to its share of the total intake of the population.

The neural network weights and movement parameter values of the new generation undergo random mutation at a very low rate, 0.001. The value by which each weight, or the movement value is mutated is drawn from a Cauchy distribution around the original value.

# Resource landscape

The resource landscape consists of a linear sequence of *X* grid cells: *x0, x1, x2...xn*. This may be thought of as a transect from a roost-site to the low-tide line in the case of shorebirds, or as a simplified representation of a migratory route.

#### Landscape replenishment

Cells have a resource value *r*, and a replenishment value *R*. Each cell *x* also holds information on the total number of agents over a generation whose discretised position placed them on *x*. Cells are initialised with *r* = 1.0, and *R* = 1.0. Cells accumulate resource at the rate of *R*/generation, such that a cell with *r* = 1.0 in generation 1 would have *r* = 2.0 in generation 2. The maximum value of *r* per cell is capped at 10.

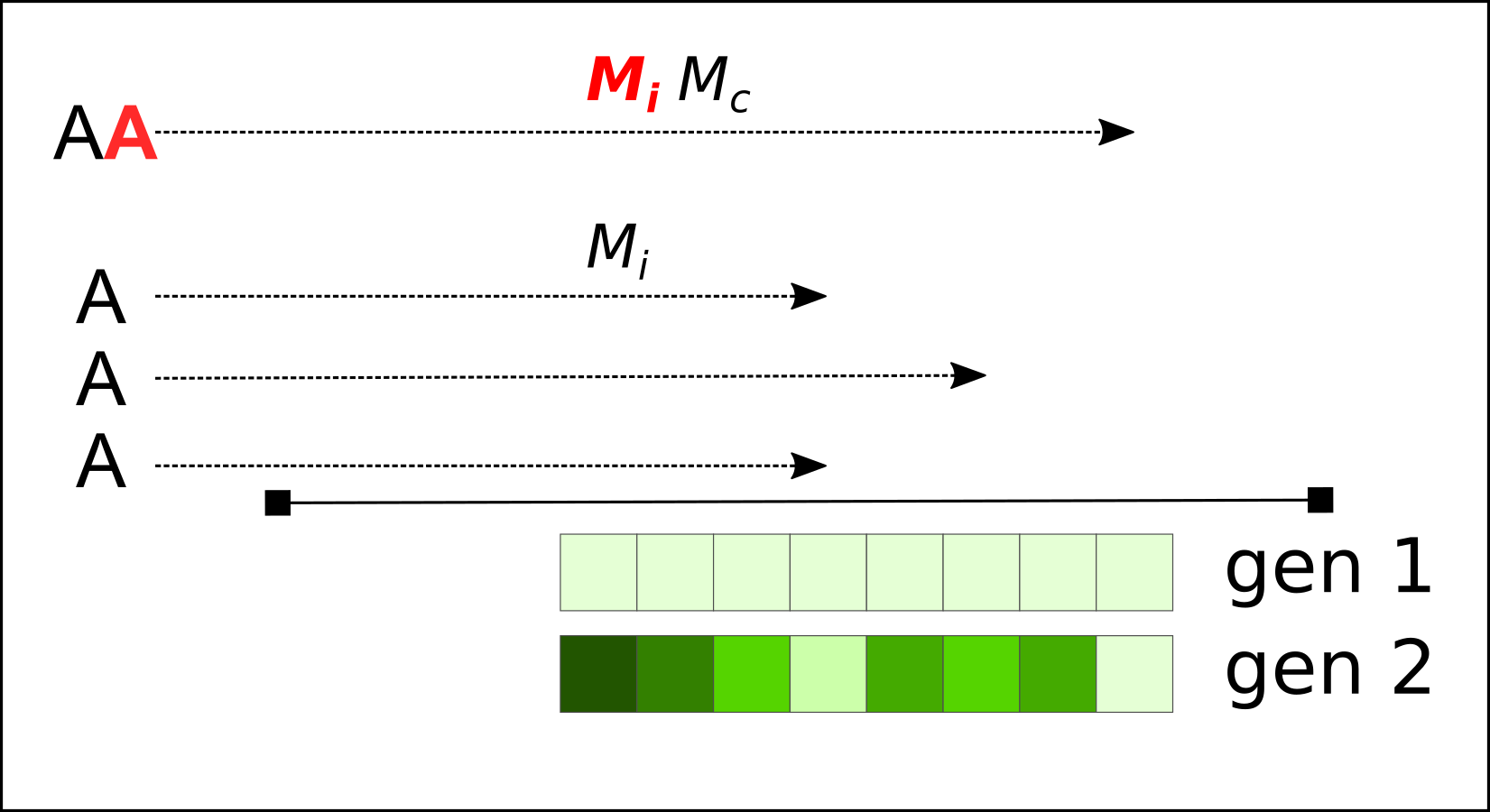
Cell resources are not depleted within a generation; agents extract *rx*/*Nx* resources per timestep from a grid cell *x*, where *Nx* is the number of agents on that grid cell.

#### Landscape depletion

However, in each generation, a cell’s replenishment value *R* is decreased from the maximum (1.0) by an amount corresponding to the proportion of total agent visits it received:

1 – sum(*Nx*) / (*Nagents* \* *timesteps*)

Consequently, a cell starting at *r* = 1.0 and receiving half (0.5) of all visits in generation 1 would have *r* = 1.5 in generation 2; this is a reduction of 0.5 from a case in which the cell was never visited by agents. This is justified as a reduction in landscape productivity due to the direct or indirect effects of agent foraging.

  
Figure 2: Movement and depletion on a mixed continuous-discrete landscape. Agents (A) move with an intrinsic movement parameter Mi. When agents choose to follow another agent in the population (top agents, AA), the follower (black A) adopts the Mi of the leader (red A) as Mc. Agent visitation of the landscape depletes it over generations; the replenishment rate of each landscape grid cell is reduced by the proportion of total agent visits that it received. This is shown by the differences in cell colour between generations 1 and 2. Cells with multiple agent visits are replenished less (lighter green) than cells with few or no agent visits (darker green).

# Output and analysis

The model produces text files with the following output: generation, timestep, agent identity, agent *Mi*, agent *Mc*, agent proximate leader, agent ultimate leader, agent leadership chain-length, sum of agent intake.

I propose to answer the following questions:

1. How does the proportion of followers to independents change over ecological and evolutionary timescales?
2. How can interaction (leader-followership) networks be quantified over timescales in terms of agent-specific and *Mi* -bin value specific nodes, and their degree distribution?
3. Do certain environmental conditions promote the evolution/development of characteristic leader-followership network structures?

# Model parameters

|  |  |
| --- | --- |
| Parameter | Value |
| **Population** |  |
| Generations | 1000 |
| Timesteps per generation | 25 |
| Population size | 100; linked to landscape size for required density |
| **Landscape** |  |
| Landscape size | 20; linked to population size for required density |
| Agent density | 5 per cell; variable between 0.1 – 10 |
| Maximum resource value | 10.0f; variable between 1.0f – 10.0f |
| Initial resource regrowth value | 1.0f; variable between 0.1f – 1.0f |
| Resource extracted value | Cell value / number of agents |
| **Artificial neural network** |  |
| ANN weights | 0.0f |
| ANN weight mutation value | 0.001f |
| ANN weight mutation shape | 0.1 Cauchy distribution |
| ANN architecture | 2 – 3 – 1 |
| Node function | Rectified linear activation |
| Network type | Simple feed-forward |
|  |  |