Journal of Artificial General Intelligence 11(1) 1-37, 2020
DOI: 10.2478/jagi-2020-0001

Submitted 2018-10-28
Accepted 2019-12-22

Combining Evolution and Learning in Computational Ecosystems

Claes Strannegård CLAES.STRANNEGARD@CHALMERS.SE
Department of Computer Science and Engineering, Chalmers University of Technology, Sweden

Wen Xu WENX@STUDENT.CHALMERS.SE
Department of Computer Science and Engineering, Chalmers University of Technology, Sweden

Niklas Engsner NIKLAS.ENGSNER@GMAIL.COM
Department of Computer Science and Engineering, Chalmers University of Technology, Sweden

John A. Endler JOHN.ENDLER@DEAKIN.EDU.AU
Centre for Integrative Ecology, Deakin University, Waurn Ponds, Australia

Editor: James Marshall

Abstract

Although animals such as spiders, fish, and birds have very different anatomies, the basic mechanisms that govern their perception, decision-making, learning, reproduction, and death have striking similarities. These mechanisms have apparently allowed the development of general intelligence in nature. This led us to the idea of approaching artificial general intelligence (AGI) by constructing a generic artificial animal (animat) with a configurable body and fixed mechanisms of perception, decision-making, learning, reproduction, and death. One instance of this generic animat could be an artificial spider, another an artificial fish, and a third an artificial bird. The goal of all decision-making in this model is to maintain homeostasis. Thus actions are selected that might promote survival and reproduction to varying degrees. All decision-making is based on knowledge that is stored in network structures. Each animat has two such network structures: a genotype and a phenotype. The genotype models the initial nervous system that is encoded in the genome ("the brain at birth"), while the phenotype represents the nervous system in its present form ("the brain at present"). Initially the phenotype and the genotype coincide, but then the phenotype keeps developing as a result of learning, while the genotype essentially remains unchanged. The model is extended to ecosystems populated by animats that develop continuously according to fixed mechanisms for sexual or asexual reproduction, and death. Several examples of simple ecosystems are given. We show that our generic animat model possesses general intelligence in a primitive form. In fact, it can learn simple forms of locomotion, navigation, foraging, language, and arithmetic.

1. Introduction

Stuart Wilson defined *animats* as a variety of artificial animals that includes a model of homeostasis (Wilson, 1986). He also suggested the *animat path to AI*: the idea of creating artificial intelligence (AI) by modeling animal behavior (Wilson, 1991). This makes sense, since human intelligence is a special case of animal intelligence, which in turn is a special case of animal behavior. Moreover, since general intelligence, as it is commonly defined, is a special case of human intelligence, the animat path to AI automatically leads to artificial general intelligence (AGI). Furthermore, the animat path seems to be a natural strategy for reaching AGI, since the most prominent examples of agents with general intelligence belong to the animal kingdom.

In this paper we propose to follow the animat path to AGI by simultaneously modeling two fundamental processes underlying animal intelligence: evolution, which operates at the population level between generations, and learning, which operates at the individual level. Both these processes seem to be fundamental to the ability of animal populations to adapt to new environments, survive, and reproduce. Evolution has generated all known lifeforms and all forms of animal behavior, including intelligence (Futuyma, 2009). Learning enables adaptation to new environments beyond the level of reflex agents, e.g. by learning where resources and dangers are located, and how to find and choose food and mates. Nonetheless, models that combine evolution and learning are relatively rare in AI and also in computational biology, cognitive psychology, and behavioral economics. First let us briefly survey some topics that are relevant to the model.

1.1 Ecosystem models

Ecosystem models include animal models of one kind or another. In analytic approaches to ecosystem modeling, animal populations are frequently modeled with numbers representing biomass and the interaction dynamics between different population groups is modeled with systems of differential equations. A well-known example is the Lotka-Volterra equations for predator-prey dynamics (Lotka, 1925). Biomass plays a central role also in simulation-based ecosystem models. For instance, models created in the Ecopath/Ecosim/Ecospace simulation environment for marine ecosystems (Christensen and Walters, 2004) might include variables for catch, predation, and net migration, all measured in tonnes of certain fish species. Still another form of ecosystem model consists of data alone. For instance, there is a data bank representing the ecosystem of the tropical island of Moorea in French Polynesia, which includes a complete mapping of the biomass and the genome of the island's macroscopic plants and animals above and under the sea, together with detailed data about its topography, temperatures, winds, and currents (Cressey, 2015). In ecosystem models, learning is typically left out of the picture. Yet it is known that the ability to learn is closely correlated to fitness (Ashton et al., 2018). Thus, excluding learning from the ecosystem models might lead to substantial inaccuracy. On the other hand, by including learning, the models become more accurate and animal behavior can be studied in a more realistic setting and at a greater level of detail. Ecology and evolution have been modeled jointly (Hubbell, 2001), but learning was not included in the model. The integration of ecology and cognition has been discussed, e.g. in (Healy and Jones, 2002).

1.2 Artificial animals

Artificial animals have a long tradition of study in the field of artificial life. Classic examples of artificial animals include Braitenberg vehicles (Braitenberg, 1986), cellular automata (Von Neumann and others, 1951), evolving creatures (Sims, 1994), and the previously mentioned animats (Langton, 1997). There are also several computer games that feature ecosystems of artificial animals, e.g. SimLife (Karakotsios and Bremer, 1993) and Spore (Bohannon, 2008).

Now let us briefly survey three methods that have been used for modeling animal cognition and will be used again in the present paper: evolutionary algorithms, neural networks, and reinforcement learning algorithms. Evolutionary algorithms (Coello et al., 2007) imitate aspects of natural evolution such as reproduction, mutation, and selection. Solution candidates to optimization problems are typically represented as binary strings (modeling DNA sequences) that are subject to evolutionary processes. Evolutionary algorithms often rely on manually designed domain-specific fitness functions for selecting solution candidates for reproduction. In real animal populations, fitness can be hard to predict, since it depends on a vast number of factors: who manages to survive until reproductive age, who meets who, who manages to reproduce, who can ensure the survival of the offspring, etc. This makes it difficult to define domain-specific fitness functions manually. On the other hand, in simulation environments it is relatively straight-forward to compute fitness by looking at the offspring. An important class of evolutionary algorithms are the genetic programming algorithms (Koza, 1989, 1994), in which the genome encodes computer programs.

Artificial neural networks, including deep learning networks (LeCun, Bengio, and Hinton, 2015) were originally inspired by nervous systems and have been used for multiple purposes including simulated animals and simulated video game players. Standard neural network models are based on static architectures that do not develop over time. Thus they fail to capture certain aspects of neuroplasticity that are present in natural nervous systems (Draganski and May, 2008). There are also neural network models with dynamic architectures, however, e.g. the cascade-correlation architectures (Fahlman and Lebiere, 1990) that add one hidden neuron at a time and the progressive neural networks (Rusu et al., 2016) that add columns of neurons.

Reinforcement learning is ubiquitous in the animal kingdom and its biological basis is relatively well understood (Niv, 2009). Reinforcement learning algorithms were inspired by animal learning and decision-making. They are powerful tools for learning and decision-making in a general setting (Sutton and Barto, 1998). In particular Q-learning can be used for learning an optimal policy from experience for any Markov Decision Process (Watkins, 1989). Other reinforcement learning algorithms include tree-based methods running in batch-mode (Ernst, Geurts, and Wehenkel, 2005) and a variety of local Q-learning, where Q-values collected from multiple agents are merged into a global Q-value (Russell and Zimdars, 2003). A way of combining conjunctions of basic features in reinforcement learning is described in (Buro, 1998).

1.3 Animats

Stuart Wilson introduced *animats* as a kind of artificial animals that includes a model of homeostasis (Wilson, 1986). Their decision-making aims at maintaining homeostasis (Keramati and Gutkin, 2011, 2014; Yoshida, 2017). In other words, their only goal is to always keep their needs satisfied. This might have several side effects that are positive from the biological fitness perspective. In fact, it may benefit survival, reproduction, parenting, and cooperation.

Homeostatic decision-making has been combined with models for hormonal control (Avila-García and Cañamero, 2005) and cognitive modulation (Bach, 2015). It also supports decision-making influenced by individual personality traits (Bouneffouf, Rish, and Cecchi, 2017). For modeling animals with multiple needs, e.g. water and energy, it may be natural to use multi-objective reinforcement learning (Roijers et al., 2013).

1.4 Main idea and structure of the paper

Although animals such as spiders, fish, and birds have very different anatomies, the basic mechanisms that govern their perception, decision-making, learning, reproduction, and death have striking similarities. These mechanisms have apparently allowed the development of general intelligence in nature. These considerations led us to the idea of approaching AGI by constructing a *generic* homeostatic animat with fixed mechanisms of perception, decision-making, learning, reproduction, and death. The generic animat model, which will be developed below, can take the form of a spider animat, a fish animat, or a bird animat. Our model was heavily inspired by biology, but we do not claim it to be biologically realistic, since we are unaware how cognition actually works in any animal. Rather than speculating on the nature of intelligence, we focus on results of modeled intelligence mechanisms.

Section 2 defines our generic animats and Section 3 introduces the ecosystems that they populate. Sections 4-6 describe, respectively, how the animats perceive, make decisions, and learn, while Section 7 specifies how they reproduce. Section 8 gives several examples of ecosystems that develop over time and Section 9 discusses the generality of the model. Finally, Section 10 draws some conclusions.

2. Generic animats

In this section we define the generic animats.

2.1 Networks

We will model sensory neurons and motor neurons in a simplified way by using Boolean variables. This is in line with the McCulloch–Pitts model of neurons that either fire all out, when the action potential exceeds a threshold, or not at all. Real neurons have a firing frequency that reflects the intensity of their stimuli, but this aspect is not covered in our model. For modeling analogue signals we rely on standard digitization and use one Boolean variable for representing pH 6.00, another for representing pH 6.01, etc. The number of Boolean variables used might depend on the resolution of the sensory apparatus of the animal that one wants to model. For simplicity we assume that time proceeds in discrete ticks. We will use sets for modeling groups of neurons that fire in parallel and lists for modeling sequences of such groups of neurons that fire one after the other. The notation $\{\ldots\}$ will be used for sets and $[\ldots]$ for lists.

Definition 1 (Sensor, sensor set, pattern)

- *A* sensor *is a Boolean variable.*
- *A* sensor set *is a finite set of sensors.*
- *A* pattern *is a finite list of sensor sets.*

Sensors are analogous to receptor neurons with ion channels that could be sensitive to cold temperature, mechanical pressure, muscle tonus, or acidity.

Example 1 *Some examples of the above notions:*

- *Suppose red, green, and blue are sensors for colors, i.e. simple versions of RGB sensors that can only have two values.*
- *Then* $\{\}$, $\{green\}$, $\{blue,green\}$, *and* $\{red,blue,green\}$ *are sensor sets representing, respectively, black, green, turquoise, and white.*
- *Moreover,* $[\{\}]$, $[\{green\}]$, $[\{blue,green\}]$, *and* $[\{red\},\{red\},\{green\}]$ *are patterns that represent different color sequences.*

Example 2 *Some more examples of patterns:*

- *Tastes. Consider a simplified model with one sensor for each one of the five basic tastes: sweet, salt, sour, bitter, and savory. Then the pattern* $[\{sweet,sour\}]$ *is a list (of length 1) that represents the sweet-and-sour taste.*
- *Scents. The human nose has roughly 400 types of scent receptors that can be roughly modeled by using 400 sensors. Suppose the scent of a certain flower activates the set of sensors* S*. Then the pattern* $[S]$ *(of length 1) represents that smell.*
- *Images. A digital image can be represented as a pattern by using one Boolean variable for each coordinate and each color. For instance, a certain* 2×2 *image with red and green pixels might be represented by the pattern*

  $$[\{red(0,0),red(0,1),red(1,0),green(1,1)\}].$$

  Here $red(i,j)$ *and* $green(i,j)$ *are color sensors for the coordinate* (i,j)*.*
- *Videos. A silent video can be modeled as a list of digital images of the type just mentioned. Hence it is a pattern too.*
- *Music. Fix a set of sensors* S *for the notes that can be produced on a certain digital piano. Then any chord on this piano can be represented by a subset* $S' \subseteq S$*, or equivalently by the pattern* $[S']$*. Any sequence of chords (or single notes) can be represented as a list of such sensor sets, in other words as a pattern. In fact, any piece of digital music can be represented by a pattern.*

Patterns can also be based on sensors for other modalities such as touch. It is also straightforward to define multi-modal patterns, for instance a video merged with its soundtrack. Now let us turn to the motor notions, which are analogous to the sensory notions just defined.

Definition 2 (Motor, motor set, action)

- *A* motor *is a Boolean variable.*

- *A* motor set *is a finite set of motors.*

- *An* action *is a finite list of motor sets.*

Note that propositional variables are used to model both sensors and motors. Also note that patterns and actions are defined in the same way, except that they are based on sensors and motors, respectively.

Example 3 *Some examples of these notions:*

- *Suppose $left$ and $right$ are motors for moving the left and right pectoral fins of a fish robot.*

- *Then $\{\}$, $\{left\}$, and $\{left, right\}$ are motor sets representing, respectively, idleness, moving the left fin only, and moving both fins.*

- *Moreover, $[\{right\},\{left\}]$ and $[\{right, left\}]$ are examples of actions. In the former, one fin is moved at a time and in the latter both fins are moved simultaneously.*

Before moving on, let us note that patterns and actions can both be expressed as simple sentences of modal logic using conjunction and the unary temporal operator 'next' (Gabbay, Hodkinson, and Reynolds, 1994). Let us also note that negations are superfluous in many models. In fact, in the case of sensors, negations are not needed, since one may always extend a set of sensors with new sensors representing the negation of the old ones. For instance, to model $\neg p$ one may add a new sensor q that is set to true if and only if p is false. In the case of motors, negations are also not needed. For instance, activating all motors in the set of motors $\{m, \neg m'\}$ is the same as activating all motors in the set $\{m\}$. In general, animal brains do not seem to be "closed" under negation in the sense that each neuron u has a corresponding neuron $\neg u$ that fires if and only if u does not fire. If they were, then exactly half of the neurons of the brain would always fire. However, this is known to be far from the case, e.g. in zebra fish, whose total neuronal activity fluctuates greatly (Vladimirov et al., 2014). Moreover, had it been true, the energy consumption would have been at a permanently high level –a clear drawback from an evolutionary perspective.

A reflex is an involuntary movement in response to a stimulus. The human patellar reflex is an example: a rubber hammer that hits the knee causes certain sensory receptors at the knee to fire. The nerve signal is transmitted directly to motor neurons that cause the leg muscles to contract and produce a kick. The nerves are wired directly from sensors to motors and the pathway does not pass the brain. Hence this mechanism is not a part of the ordinary decision-making process that involves the brain, e.g. the "conscious" decision to kick a ball. Other examples of reflexes are the sucking and rooting reflexes in babies, both of which facilitate breastfeeding. If the breastfeeding reflexes did not exist, the baby would have to learn to ingest milk by doing trial and error and in the meantime lose valuable time. This example suggests that reflexes might play a key role for survival in situations when it would take too long to learn from trial and error. Reflexes could also be critical when it would be too dangerous to learn from trial and error, e.g. learning to avoid snakes. Considerations such as these led us to include both reflexes and "conscious" decision-making in the animat model.

Definition 3 (Reflex) *A* reflex *is a pair (p, a), where p is a pattern and a is an action.*

Example 4 *Some examples of possible reflexes:*

- $([\{red\}], [\{left\}])$
- $([\{blue, green\}], [\{left, right\}])$
- $([\{blue\}, \{green\}], [\{left, right\}])$

Definition 4 (Network) *A* network *consists of the following data:*

- *a set of sensors*
- *a set of patterns*
- *a set of motors*
- *a set of actions*
- *a set of reflexes*

2.2 Homeostatic variables

Animals typically have several needs, e.g. water, energy, and warmth. These needs often have associated receptors that indicate their status. Examples of needs together with their associated receptors are water (osmoceptors); energy (insulin receptors); protein (amino acid receptors); and oxygen (CO_2 receptors).

Definition 5 (Homeostatic variable) *A homeostatic variable is a variable that takes values in the closed real interval $[0, 1]$.*

For example, *energy* and *water* could be homeostatic variables. We use the convention that 0 and 1 model, respectively the minimum and maximum levels of resource. The lower bound 0 models death from lack of resources. For instance, $energy = 0$ models death from starvation and $water = 0$ models death from dehydration. The upper bound 1, on the other hand, models maximum resource levels. For example, $energy = 1$ models that an animal is full and $water = 1$ that it is unthirsty. Since 1 is a firm upper limit, an animat with $energy = 1$ cannot increase *energy*, e.g. by eating (although it might be able to store energy resources outside its body or inside its body in the form of fat). Similarly, an animat with $water = 1$ cannot increase the value of the homeostatic variable *water*, e.g. by drinking.

Animals typically strive to increase their levels of water and energy, but decrease their level of pain. Pain can be modeled in the present setting by introducing the homeostatic variable *integrity* and letting 0 correspond to minimum integrity (maximum pain) and 1 to maximum integrity (minimum pain). Many animals strive to keep their body temperature in a certain range. This can be modeled by using two homeostatic variables: *cool* and *warmth*. When the body temperature is optimal, both these variables assume the value 1 and when it is warmer or cooler, one of them will be strictly smaller than 1. Some resources, e.g. *energy*, are critical in the sense that low levels can lead to death, whereas others, like *oxytocin*, may not be critical in this sense. To model non-critical resources, one may simply use a homeostatic variable that always takes values in the range $[q,1]$, where $0 < q < 1$.

In Section 5 we will define a decision-making policy that roughly speaking strives to keep all homeostatic variables as high as possible. While homeostatic variables are concerned with the present status of an animat, the experience variables store memories of the past.

2.3 Experience variables

Definition 6 (Experience variable) *The experience variables are the following:*

*age***.** *This variable ranges over natural numbers and represents the number of ticks since the animat was inserted into the ecosystem.*

$LR(h,a,p)$**.** *These are real-valued variables called local R-values. They are indexed by a homeostatic variable h, a pattern p, and an action a. They are used for keeping track of the average reward $r(h,t)$ that follows upon taking action a when the pattern p is top active (see Definition 17).*

$LQ(h,p,a)$**.** *These are real-valued variables called local Q-values. Intuitively $LQ(h,p,a)$ estimates how good it is from the perspective of homeostatic variable h to do action a when pattern p is top active (see Definition 17), taking both immediate and estimated future reward into account.*

It is straightforward to extend the model by adding more experience variables. For instance, one may use a set of variables for keeping track of the transition probabilities (the probability of p' being active at $t+1$ given that p is active at t and action a is taken at t).

2.4 Parameters

A *parameter* is a real number that is used as a constant for regulating processes of learning, decision-making, reproduction, and death. Rather than enumerating them, we shall introduce the parameters gradually as they are needed. The parameters may take values in the set of real numbers $\mathbb{R}$, natural numbers $\mathbb{N} = \{0,1,2,\ldots\}$, or integers $\mathbb{Z} = \{\ldots,-2,-1,0,1,2,\ldots\}$. For example, the parameters $MutationRate$ and $DiscountRate$ take values in $\mathbb{R}$, whereas $MaxAge$ takes values in $\mathbb{N}$. There are also parameters such as Sex that take values in finite domains.

2.5 Configurations

Definition 7 (Configuration) *A* configuration *is a bitstring that encodes the following data:*

- *a network*
- *a set of homeostatic variables*
- *a set of experience variables*
- *a set of parameters*

- *a set of additional physical properties, including a conformation in* $\mathbb{R}^3$.

The exact choice of encoding is not important, so we refrain from specifying it. The last component of this definition can be viewed as a part of the bitstring that is reserved for encoding additional properties of the body such as shape and color. Since our focus here is on the other components of the definition, we do not (need to) go into further details. We merely include it as a resource that can be developed further if needed.

Since configurations are bitstrings that encode data, we can speak interchangeably of configurations as bitstrings and as the data that they represent.

Example 5 *An example of a configuration (with certain data omitted) is given below. This configuration might be suitable for surviving in a world where green objects can be eaten and blue objects drunk.*

- *Sensors: green, blue*
- *Patterns:* $[\{green\}]$, $[\{blue\}]$
- *Motors: eat, drink*
- *Actions:* $[\{eat\}]$, $[\{drink\}]$
- *Reflexes:* $([\{green\}],[\{eat\}])$, $([\{blue\}],[\{drink\}])$
- *Homeostatic variables: energy, water*

Although the definition of configuration is primarily intended for modeling animals with nervous systems, it can also be used for modeling other organisms, e.g. bacteria, fungi, plants, sponges. In fact, one may simply use an empty network and omit most experience variables (except *age*) and parameters (except those that regulate reproduction), while preserving the homeostatic variables and the physical properties.

2.6 Animats

Now we are ready to define our notion of artificial animal, or animat:

Definition 8 (Animat) *An* animat *is a pair* (G, P)*, where* G *and* P *are configurations.* G *is called the genotype and* P *is called the phenotype.*

In this model the genotype remains constant throughout the lifetime of the animat. It is used for specifying what the animat is like at birth and for partially specifying the offspring. The phenotype, on the other hand, changes constantly throughout the lifetime of the animat. It is used for specifying the animat's cognitive and non-cognitive structures. Several examples of animats will be given in Section 8.

We remarked before that configurations can be used for modeling not only animal anatomies, but also plant anatomies, cell anatomies, etc. Similarly, we interpret animat in a wide sense as an animated object or an artificial organism that may also include robots. For instance, a crude model of a plant can be obtained by disregarding all parts of the animat model that are related to nervous systems and preserving the other parts (including genome, homeostatic variables, and physical conformation).

3. Ecosystems

In this section we will define ecosystems and show how they develop over time.

3.1 Ecosystem definition

Definition 9 (Dead object) *An* dead object *consists of:*

- *a type representing a non-living entity, e.g. rock, earth, sand, air, water, carcass.*
- *a set of physical properties, including a conformation. Which physical properties are used and how they are specified may vary from model to model.*

Definition 10 (Ecosystem) *An* ecosystem *is a pair $(\mathcal{A}, \mathcal{D})$, where $\mathcal{A}$ is a set of animats and $\mathcal{D}$ is a set of dead objects, such that all conformations are pairwise disjoint.*

First let us reiterate that the animat model covers not only animals with nervous systems, but also other organisms. Thus one might model ecosystems that include animals, plants, mountains, and lakes. Several examples of ecosystems will be given in Section 8. By definition, an ecosystem consists of animats and objects that form a subset of the space $\mathbb{R}^3$. Sometimes it is enough to consider subspaces of $\mathbb{R}^3$. For instance, to model life on the surface of a lake, one may want to use the "two-dimensional" set $\mathbb{R} \times \mathbb{R} \times \{0\}$, which can be identified with $\mathbb{R}^2$. Similarly, to model life in a creek, one may want to use the "one-dimensional" set $\mathbb{R} \times \{0\} \times \{0\}$, which can be identified with $\mathbb{R}$. Another way of simplifying the space is to work in subsets like $\mathbb{Z}^3$, $\mathbb{N}^3$, or a finite set such as $\{0,1,2,3,4\}^3$. By combining these simplifications one may work in spaces such as $\mathbb{Z}^2$. This makes it possible to model arcade games like Pac-Man as ecosystems. The Pac-Man character and the four ghosts can be modeled as animats, whereas the maze and the pills can be modeled as objects.

3.2 Ecosystem development

Now let us describe how our ecosystems develop over time. Algorithm 1 shows the main loop of the ecosystem update. Note that this algorithm is generic in the sense that it is common to all animats. Let us explain this algorithm in greater detail. The details of lines 1 and 2 depend on the environment model, e.g. the physics engine, and must be specified separately. At each tick, the animats and dead objects are updated according to simulated laws of physics. For instance, animals and objects may move due to gravity, wind, or currents. Animals may also move due to their own actions. The physical properties of animals may also change, e.g. as a result of their own actions, e.g. locomotion, or other animals' actions, e.g. predation. Line 3 invokes Algorithm 2, which computes whether an animat is dead or alive. Line 4 concerns the genotype update. Here we simply use the identity function. Hence the genotype of an individual animat is not subject to continuous change. Genetic recombination is omitted from the model, while genetic mutation is included, though only in connection with reproduction. Line 5 is the phenotype update, given in Algorithm 3. Line 6 shows what happens if the animat dies and turns into a dead object. Line 7, finally, concerns reproduction, which is described in Section 7.

Algorithm 2 is used for computing whether an animat is dead or alive. Here $MaxAge$ is a parameter whose value is a natural number or ∞. The value ∞ can be used for modeling

Algorithm 1: Ecosystem update.

```
Input: An ecosystem $(\mathcal{A}, \mathcal{D})$
$t = 0$
loop forever
    for $(T, P) \in \mathcal{D}$ do
        Update the type $T$
        Update the properties $P$
    end
    for $(G, P) \in \mathcal{A}$ do
        if $alive(G, P)$ then
            Update the genotype $G$
            Update the phenotype $P$
        else
            Add $(carcass, properties(P))$ to $\mathcal{D}$
            Remove $(G, P)$ from $\mathcal{A}$
        end
    end
    Add new animats to the population $\mathcal{A}$
    $t = t + 1$
end
```

Algorithm 2: Mechanisms of death

```
Input: An animat $(G, P)$
begin
    if $age < MaxAge$  and
    $0 < i(h, t)$ for all homeostatic variables $h \in P$
     then
        $alive = True$
    else
        $alive = False$
    end
end
Output: The Boolean value $alive$
```

animals that might lack mechanisms of aging, cf. the genus *Hydra* (Boehm and others, 2012). The animat is alive unless it has died of senescence (line 1) or of failed homeostasis (line 2). The function $i(h,t)$ is defined in Definition 11.

4. Network activity

In this section we define how certain components of the network (sensors and patterns) are activated by stimuli from the environment and how other components of the network (motors and actions) are activated by reflexes or decisions. Decision-making is the topic of the next section.

Definition 11 *An* input stream *is a function i that assigns a Boolean value $i(s,t)$ to each sensor s and a real value $i(h,t) \in [0,1]$ to each homeostatic variable h at each time t.*

Now we can define a reward function in terms of homeostatic variables.

Definition 12 (Reward) *Suppose h is a homeostatic variable and t is time. Let*

$$r(h,t) = \begin{cases} i(h,t) - i(h,t-1) & \textit{if } t \geq 1 \\ 0 & \textit{otherwise} \end{cases}$$

Note that rewards are in the range $[-1,1]$. Here negative reward can be understood as punishment.

4.1 Pattern activity

First let us describe how the input activity propagates to patterns.

Definition 13 *Given the input stream i, the pattern $[S_0,\ldots,S_n]$ is* active *at t if $i(s,t-n+k)$ is true for all $s \in S_k$ and all k such that $0 \leq k \leq n$.*

Example 6 *Some examples of pattern activity:*

- *The empty pattern $[\{\}]$ is always active.*
- *The gustatory pattern $[\{sweet\}]$ is active whenever the sensor sweet is active.*
- *The visual pattern $[\{green, blue\}]$ (turquoise) is active whenever the sensors green and blue are active.*
- *Let C, D, and E be sensors for musical notes on a piano. Then the auditory pattern $[\{C\},\{C\},\{C\},\{D\},\{E\}]$ (the melody "Row, row, row your boat") is active whenever that particular sequence of notes has been active.*

4.2 Action activity

Actions can be activated in two ways:

Definition 14 (Action activity) *The action a is* activated at *t if*

- *a is activated via a reflex: there is a pattern p and a reflex (p,a) such that p is active at t, or*

- *a is activated via a decision: $\pi(t) = a$ where π is the animat's policy, see Section 5.*

For instance, a kick can be produced in two ways, either via the patellar reflex or via a decision to kick.

Next we will define motor activity, which is a kind of dual to pattern activity. In principle, actions are activated first (by a decision or a reflex) and then those actions activate various motors.

Definition 15 (Motor activity) *The motor m is activated at t if there is a time $t' \leq t$ such that some action $[M_0, \dots, M_n]$ with $m \in M_{t-t'}$ was activated at t'.*

Example 7 *Suppose action $[\{m_0\}, \{m_1\}, \{m_2\}]$ is activated at time t. Then the motor m_0 is activated at t, m_1 is activated at $t+1$, and m_2 is activated at $t+2$.*

4.3 Top activity

Next we will introduce the notion of top activity, which will play a key role later in both decision-making and learning. A child who eats an apple for the first time will perceive a new combination of the five basic tastes, say sweetness=0.6, sourness=0.4, bitterness=0.2, saltiness=0.1, and savoriness=0. If the child forms a memory of this taste, then this particular combination will be stored. Then, when the child eats a similar apple at some later point, this memory will be activated. Then it will be the most detailed among the activated patterns. In our terminology, the apple pattern will be *top active*, whereas the other apple properties (e.g. sweetness=0.6) will only be active. To define top activity formally, we begin by defining a partial order on patterns:

Definition 16 (Extension) *The pattern $[S'_0, \dots, S'_n]$* extends *the pattern $[S_{n-m}, \dots, S_n]$ if $n \geq m$ and $S'_k \supseteq S_k$, for all k such that $n - m \leq k \leq n$.*

If p' extends p, we write $p' \succeq p$. Intuitively, $p' \succeq p$ means that p' is at least as detailed a pattern as p.

Example 8 *Here are two examples of extensions concerning, respectively, tastes and sequences of musical notes:*

- $[\{sweet, sour\}] \succeq [\{sweet\}]$

- $[\{C\}, \{C\}, \{C\}, \{D\}, \{E\}] \succeq [\{D\}, \{E\}]$

Note that $\succeq$ is a partial order on any set of patterns. If $p' \succeq p$, but not $p \succeq p'$, then we write $p' \succ p$. Since patterns and actions are completely analogous, we may use $\succeq$ and $\succ$ also for actions.

Definition 17 (Top activity) *Let P be a set of patterns. A pattern $p \in P$ is* top active *with respect to P at t if*

- *p is active at t and*

- *p is $\succeq$-maximal with respect to the active patterns in P.*

Intuitively, the top active patterns constitute a description of the present situation at a maximum level of detail with respect to a given set of patterns.

Example 9 *Some examples of top activity:*

- *Suppose P consists of the gustatory patterns $[\{sweet\}]$ and $[\{sweet, sour\}]$. Also suppose that the sensors sweet and sour are active. Then $[\{sweet\}]$ and $[\{sweet, sour\}]$ are both active, but only $[\{sweet, sour\}]$ is top active.*

- *Suppose P consists of the musical patterns $[\{D\},\{E\}]$ and $[\{C\},\{C\},\{C\},\{D\},\{E\}]$. Moreover, suppose both of them are active. Then only the latter, more detailed, pattern is top active.*

- *Suppose P consists of the visual patterns $[\{\}]$, $[\{green\}]$, $[\{blue\}]$, $[\{red\}]$ and $[\{green, blue\}]$. Also suppose that the sensors green, blue, and red are active. Then $[\{red\}]$ and $[\{green, blue\}]$ are both top active.*

Obviously, if no patterns are active, then no patterns are top active either. Moreover, several patterns can be top active at the same time, as we just saw. In Section 5 we will see that actions are selected by the decision-making algorithm on the basis of the experience associated with those patterns that are top active at that moment. For example, an animat might decide to swallow when a pattern corresponding to the taste combination of an apple is top active.

5. Decision-making

In this section we will describe the decision-making algorithm of the animats. The algorithm presented here can be varied in multiple ways and serves only as a basic example. Our decision-making algorithm evaluates each action from two perspectives: its exploitation value and its exploration value. Then it simply adds those two values into an overall estimate of the action's value and always selects the action with the highest estimate. The decision-making is a part of the phenotype update algorithm shown in Algorithm 3. Here *DevelopmentalTime* is a parameter that models the time it takes from the moment of conception until the sets of sensors, motors, and homeostatic variables are fully developed. Line 1 indicates that the animat does not engage in signal processing before that period has elapsed. Line 2 concerns decision-making and will be explained in this section, whereas lines 3-5 concern learning and will be explained in Section 6.

Now let us focus on decision-making and introduce a function that aggregates several local Q-values into a single global Q-value.

Definition 18 (Global Q-value) *Let*

$$GQ(h,a,t) = \begin{cases} 0 & \text{\textit{if} } TA(t) = \emptyset \\ \frac{\sum_{p \in TA(t)} LQ(h,p,a)}{size(TA(t))} & \text{\textit{otherwise.}} \end{cases}$$

Algorithm 3: Phenotype update.

```
Input: An arbitrary phenotype $P$
age = age + 1
if $age > DevelopmentalTime$ then
    Read the new values to the sensors
    Compute the set of active patterns
    Compute the set of top active patterns
    Select an action
    Update the action activity
    Update the motor activity
    Read the new values of the homeostatic variables
    Update experience variables
    Update the pattern set
    Update the action set
end
Output: The updated phenotype $P$
```

Here $TA(t)$ is the set of patterns that are top active at t and *size* is the cardinality function. This means that GQ is a weighted sum of the local Q-values LQ.

Consider an animat whose pattern set consists of all patterns $[S]$, where S is a set of sensors. Then $TA(t)$ will always consist of exactly one set, namely the pattern defined by those sensors that are active at t. Then the global Q-value will be equal to the local Q-value of that particular pattern.

Definition 19 (Exploit value) *Let*

$$\text{exploit}(h,a,t) = i(h,t) + \gamma_h \cdot GQ(h,a,t).$$

Here γ_h is the parameter $DiscountFactor_h$, which is a real number in $[0,1]$.

Intuitively, exploit(h,a,t) is the estimated h-value of taking action a. It is obtained by adding $i(h,t)$, which is the present h-value (that is independent of what the next action might be) and $\gamma_h \cdot GQ(h,a,t)$, which is a discounted estimate of the future h-value change if a is selected.

The decision-making is handled by a policy that selects exactly one action at every tick, possibly the idle action $[\{\}]$. We will give two alternative policies, one basic and one more refined.

5.1 Basic policy

Let $A(t)$ be the action set at time t. Here is an epsilon-greedy policy:

Definition 20 (Basic policy) *Let* $\pi(t) =$

$$\begin{cases} \arg\max_{a \in A(t)} [\min_h \text{exploit}(h,a,t)] & \textit{if } \rho > \theta \\ \text{a random action from } A(t) & \textit{otherwise} \end{cases}$$

Here ρ is $rand(0,1)$, which is a random number generator that produces a real number in the interval $(0,1)$ with uniform probability. Moreover, θ is the parameter $ExplorationRate$, which is a real number in $[0,1]$.

Note that the policy depends on the values of the homeostatic variables $i(h,t)$ via the exploit function. With just one homeostatic variable h, this feature would be redundant. With more than one homeostatic variable, however, the terms $i(h,t)$ serve a purpose. For instance, an animal that is thirsty but not hungry might prioritize drinking over eating.

As we saw in Section 4, an action can be activated by a decision, a reflex, or both. Agonist and antagonist motors may well be activated simultaneously as in a cramp, possibly resulting in no motion. Extended actions, i.e. long lists of motor sets, might cause motors to activate several steps after the action was initiated. In the meantime additional actions might be activated, causing several motors to activate in parallel.

5.2 Refined policy

One problem with the basic policy is that the animat decides to explore with the same probability regardless of how well it is doing. In animals such behavior could be disastrous. For instance, if a lamprey that is attached to a host fish and sucks small quantities of its blood suddenly decides to leave its host fish in the interest of exploration, the host fish might swim away and the lamprey die from starvation. If the host fish would die, on the other hand, it could be disastrous for the lamprey not to leave it and engage in exploration. Thus it seems reasonable that recent changes to the level of need satisfaction should influence the propensity to explore. The next definition is intended to capture this intuition.

Definition 21 (Sustainability) *Suppose h is a homeostatic variable. Let the* sustainability *at t with respect to h be defined as:*

$$\mathrm{sust}(h,t) = \begin{cases} i(h,t) - i(h,t-s) & \text{if } t \geq s \\ 0 & \text{otherwise} \end{cases}$$

Here $s = SustainabilityHorizon_h$ is a parameter.

Note that $\mathrm{sust}(h,t)$ takes values in $[-1,1]$. Also note that sustainability coincides with reward if $s = 1$. Intuitively, $\mathrm{sust}(h,t)$ is a rough estimate of how well the present strategy of the animat has been working for the last s ticks from the perspective of h. Here we interpret negative values of $\mathrm{sust}(h,t)$ as an indication that the present strategy is unsustainable and needs to be changed. This motivates more intense exploration. If $\mathrm{sust}(h,t)$ is 0 or positive, on the other hand, then the need for exploration is less urgent. Hence the value of exploration depends on the level of sustainability.

Definition 22 (Explore value) *Let*

$$\mathrm{explore}(h,a,t) = rand(0,1) - \mathrm{sust}(h,t).$$

If the sustainability is low, then all actions get relatively high explore values. If sustainability is high, on the other hand, then an action can only get a high explore value if it has been assigned a high random value.

A second problem about the basic policy is that the animat selects its actions randomly when exploring. This means that it sometimes selects actions that it believes to be very bad. Again this behavior would be disastrous for animals. For example, a mountain goat that decides to make a random move might fall off the cliff to its death. This suggests avoiding actions that are believed to be disastrous altogether.

Now we propose to measure the utility of an action from the perspectives of exploration and exploitation on the same scale.

Definition 23 (Utility) *Let* utility(a,t) *be defined as*

$$\min_h \left[\text{exploit}(h,a,t) + \pi_h \cdot \text{explore}(h,a,t)\right].$$

Here π_h *is a parameter in the real interval* $[0,1]$.

In the definition of utility(a,t), note the *min* function. For instance, if an action a is considered excellent from the perspective of one homeostatic variable and disastrous from another, then on the whole, a will still be considered disastrous. This makes sense since death happens as soon as any of the homeostatic variables reaches 0.

A third problem about the basic policy is that the animat selects between existing actions only when exploring. It never tries new combinations of motors. Again this does not seem to be realistic from a biological perspective. A gazelle calf would perhaps never learn to run if it could not test new motor combinations. To address this problem, we select the action at time t from an extended menu:

Definition 24 (Menu of actions) *Recall that* $A(t)$ *is the action set at time* t*. The* menu of actions *at time* t*,* $Menu(t)$*, is defined as follows:*

- *If* $a \in A(t)$*, then* $a \in Menu(t)$
- *If* $[M]$ *and* $[M']$ *are in* $A(t)$*, then* $[M \cup M'] \in Menu(t)$
- *If* a *and* a' *are in* $A(t)$*, then* $concat(a,a') \in Menu(t)$*. Here and in the following* $concat$ *is the concatenation operator.*

Now we are ready to state our refined policy:

Definition 25 (Refined policy) *Let the* policy $\pi(t)$ *be defined as:*

$$\pi(t) = \underset{a \in Menu(t)}{\arg\max}\ \text{utility}(a,t).$$

In a still more refined policy, one may want to balance the fact that there are relatively many actions on the menu that do not belong to the phenotype. In fact, it makes sense to explore the existing actions quite thoroughly before moving on to more complex actions. This might be achieved by introducing a size measure on actions, e.g. the number of motors that they involve, and then penalize actions of large size. Discouraging actions of large size also makes sense from the perspective of energy consumption.

6. Learning

In this section we describe a battery of learning rules for updating the experience variables and for adding and removing patterns and actions. These rules are relatively coarse as they stand and they might be improved in several ways. Rules for associative learning and probabilistic learning in the animat framework are not covered here, but in (Johannesson, Nilsson, and Strannegård, 2018). Other forms of learning that take place outside the neural domain are also not covered.

6.1 Experience variable update

The variable *age* is updated by adding 1 at each tick. The *local reward* value $LR(h,a,p)$ depends on the homeostatic variable h and is updated whenever the pattern p is top active and the action a is taken. Then the average reward received in this situation since birth is updated. For that purpose one may calculate the mean on the fly, as described in (Knuth, 1997). In other situations, $LR(h,a,p)$ is not changed. Here is how the local Q-values are updated:

Learning rule 1 (Local Q-value updates) *Suppose the pattern p is top active at t and the action a_t was performed at t. Then at $t+1$, $LQ(h,p,a_t)$ is updated by letting*

$$LQ(h,a_t,p) = LQ(h,a_t,p) + \alpha_h \cdot [r(h,t+1) + \Delta],$$

where

$$\Delta = \eta_h \cdot \max_{a \in A(t)} GQ(h,a,t+1) - LQ(h,a_t,p).$$

Moreover, α_h and η_h are parameters for, respectively, learning rate and discount rate.

6.2 Pattern addition

Imagine an animal eating a berry of a certain kind for the first time. If eating the berry resulted in an energy reward, then it would benefit the animal to remember the appearance and taste of that berry so that it can find and eat similar berries in the future. On the other hand, if eating the berry resulted in punishment, in the form of stomach pain caused by toxicity, say, then it would also be beneficial for the animal to form a memory of the berry so that it could avoid eating those berries in the future. If eating the berry brought neither reward nor punishment, finally, then it does not seem to be equally meaningful to form a memory of that particular berry. Similarly, an animal that sees a cloud in the sky or a rock on the ground usually does not benefit from forming memories of those particular objects, unless they are hedonistically relevant somehow, such as an indication of rain to an animat with low water level. Now let us try to make these intuitions more precise.

Definition 26 (Approximation) *Suppose θ, v, and v' are real numbers. Let $v \approx_\theta v'$ mean that*

$$\frac{|v-v'|}{|v|+|v'|} \leq \theta.$$

For instance, if $\theta = 0.05$, then $0.52 \approx_\theta 0.5$, but $0.6 \not\approx_\theta 0.5$. We will use the following function for estimating global reward:

Definition 27 (Global reward estimate) *Let*

$$GR(h,a,t) = \begin{cases} 0 & \textit{if } TA(t) = \emptyset \\ \frac{\sum_{p \in TA(t)} LR(h,a,p)}{size(TA(t))} & \textit{otherwise.} \end{cases}$$

Here $TA(t)$ is the set of patterns that are top active at t and $size$ is the cardinality function. Note that GR is a weighted sum of the R-values of the top active patterns.

Now we are ready to define the two pattern addition rules. Intuitively we add new patterns whenever the outcome of an action turns out to be much better or much worse than expected. In fact, we interpret such large prediction errors as a sign that our model of the environment needs to be refined. To make the notion of "large prediction error" precise, we will use real-valued "threshold" parameters θ_i.

Learning rule 2 (Pattern merge) *Suppose $r(h,t) \not\approx_{\theta_1} GR(h,a_t,t)$. If there are at least two different patterns $p \in TA(t)$ of length 1, select two such patterns $[S] \neq [S']$ randomly and add $[S \cup S']$ to the pattern set.*

For instance, this rule can be used for learning tastes and smells, as well as combinations of tastes and smells. Note that Pattern merge only applies if there are at least two top active patterns of length 1 that can be merged. If there is only one top active pattern of length 1, then the following rule may be used:

Learning rule 3 (Pattern concatenation) *Suppose $r(h,t) \not\approx_{\theta_2} GR(h,a_t,t)$. Also suppose there is a unique pattern $[S] \in TA(t+1)$ and $TA(t) \neq \emptyset$. Then select a random pattern $p \in TA(t)$ and add $concat(p,[S])$ to the pattern set.*

For instance, this rule can be used for learning a sound sequence that is associated with reward.

6.3 Pattern forgetting

Next we shall define a rule for removing patterns. The idea is to remove patterns that turn out to be unnecessarily complex.

Learning rule 4 (Pattern forgetting) *Suppose that $p' \succ p$ (see Definition 16) and $LQ(h,a,p') \approx_{\theta_3} LQ(h,a,p)$, for all actions a and homeostatic variables h. Then delete p' from the pattern set.*

For instance, from a hedonistic perspective, the pattern $[\{blueberry, sunny\}]$ might be roughly equivalent to the pattern $[\{blueberry\}]$. Then the Pattern forgetting rule will remove $[\{blueberry, sunny\}]$ from the pattern set.

6.4 Action addition

Now let us define a rule for adding actions. The idea is simply to add actions that bring more value in a given situation than the existing ones.

Learning rule 5 (Action addition) *Suppose $[M] \notin A(t)$ is a new action that was tried at t and there is a homeostatic variable h such that*

$$r(h,t) \not\approx_{\theta_4} GR(h,a,t),$$

for all $a \in A(t)$. Then add $[M]$ to $A(t+1)$.

For instance, consider a whale animat with the only motors $left$ and $right$ for activating, respectively, the left and right pectoral fin. At one point this animat may be engaged in exploration and try the action $[\{left,right\}] \notin A(t)$. This action activates both fins simultaneously and produces a forward motion. Swimming forward (with an open mouth) rather than rotating at the same location increases the chances of finding energy. Thus at one point this action is likely to be accompanied by energy reward. Then the new action $[\{left,right\}]$ will be added to $A(t+1)$.

6.5 Action forgetting

Finally, we define a rule for removing actions that are relatively complex and yet fail to bring much additional value.

Learning rule 6 (Action forgetting) *Suppose $a \in A(t)$ is such that for each homeostatic variable h and pattern p, there is an action a' such that $a \succ a'$ and $LQ(h,a,p) \approx_{\theta_5} LQ(h,a',p)$. Then remove a from $A(t+1)$.*

For instance, this rule may be used for removing actions such as $[\{walk,chew\}]$, which might be roughly equivalent to $[\{walk\}]$.

7. Reproduction

We want our model to be broad enough to cover several types of sexual reproduction, e.g. corals whose gametes meet in the open ocean, mammals whose gametes meet inside the body, sea turtles that bury their fertilized eggs in the sand, and mammals that carry their fertilized eggs inside their body. We also want our model to cover asexual reproduction, which appears in the animal kingdom, e.g. in *Hydra*. Here we will only be discussing sexual reproduction, however. The case of asexual reproduction is similar but simpler, since it involves copying rather than crossover.

In this section we will describe reproduction between the "parents" *Animat* and *Animat′*. For *Animat* we use the letter S for the sensor set, M for the motor set, H for the set of homeostatic variables, P for the pattern set, A for the action set, and R for the reflex set. The corresponding sets of *Animat′* are denoted by S', M', H', P', A', and R'.

7.1 Fertilization

In nature, several conditions must be met for fertilization to take place. For instance, the two parents cannot be too dissimilar genetically. To be able to talk about genetic distance we will use the Hamming distance.

Definition 28 (Hamming distance) *Suppose b and b' are two finite strings. Then $d(b,b')$ is the number of positions in which they differ.*

Essentially, two animals will reproduce at a given moment if they have similar enough genomes, meet physically, are both fertile, have opposite sex, and are lucky in the sense that their encounter leads to fertilization. Algorithm 4 attempts to capture this intuition. If fertilization takes place in the sense that Algorithm 4 returns *True*, then we refer to A and A' as *parents*. Let us remark that the requirement on line 1 can easily be relaxed to make the model more realistic and general. If the condition on line 2 is true, then we let two parameters of A determine if fertilization takes place. Otherwise we go to line 3 and let the corresponding parameters of A' determine the outcome.

Algorithm 4: Fertilization.

```
  Input: Two animats $A$ and $A'$
  begin
      $fertilization = False$
1     if $A$ and $A'$ have the same sensors, motors and homeostatic variables and
      $Sex \neq Sex'$ and
      $age \in [StartFertile, EndFertile]$ and
      $age' \in [StartFertile', EndFertile']$ and
      $conformation(A)$ and $conformation(A')$ touch
       then
2         if $rand(0,1) < 0.5$ then
              if $rand(0,1) < ReprodProb$ and
              $d(A,A') < MaxDistance$
               then
                  $fertilization = True$
              end
3         else if $rand(0,1) < ReprodProb'$ and
          $d(A,A') < MaxDistance'$
           then
              $fertilization = True$
          end
      end
  end
  Output: The Boolean value $fertilization$
```

7.2 Mutation

In this subsection we will describe the mechanisms for mutating the genotype. Algorithm 5 is used for mutating pattern sets.

The algorithm for mutating actions is analogous to Algorithm 5 and is therefore omitted. In fact, recall that patterns and actions have analogous definitions.

Next, we will consider Algorithm 6, which is used for mutating reflex sets. In this

Algorithm 5: Pattern set mutation

```
Input: A pattern set $P$ and three parameters for mutation rate: $\mu_1$, $\mu_2$, and $\mu_3$.
begin
    for $p \in P$ do
        if $rand(0,1) < \mu_1$ then
            Remove $p$ from $P$
        end
    end
    for $([S],[S']) \in P \times P$ do
        if $rand(0,1) < \mu_2$ then
            Add $[S \cup S']$ to $P$
        end
    end
    for $(p,p') \in P \times P$ do
        if $rand(0,1) < \mu_3$ then
            Add $concat(p,p')$ to $P$
        end
    end
end
Output: The mutated pattern set $P$
```

Algorithm 6: Reflex set mutation

```
Input: A reflex set $R$, a pattern set $P$, and an action set $A$
begin
1   if $size(R) = 0$ then
        $m = 1/size(P \times A)$
2   else if $size(R) = size(P \times A)$ then
        $m = (size(P \times A) - 1)/size(P \times A)$
    else
        $m = size(R)/size(P \times A)$
    end
    for $(p,a) \in P \times A$ do
        if $(p,a) \in R$ then
            if $rand(0,1) < 1 - m$ then
                Remove $(p,a)$ from $R$
            end
        else
            if $rand(0,1) < m$ then
                Add $(p,a)$ to $R$
            end
        end
    end
end
Output: The mutated reflex set $R$
```

algorithm, line 1 and 2 represent corner cases that we treat separately in order to enable mutation also in those cases.

To mutate parameters and experience variables, finally, we use the fact that they are of type float and thus can be identified with finite binary strings. This enables us to use ordinary bitflipping for mutating binary strings.

7.3 Crossover

To form new sets of patterns, actions, and reflexes, we need to combine two sets into one. For that we will use Algorithm 7, which is a general algorithm for crossover of two arbitrary sets.

Algorithm 7: Cross-over. Here X and X' might be thought of as parent sets and X^* as a child set. Note that each element of X^* comes either from X or from X'. Moreover, if $X = X'$, then the result will be $X^* = X = X'$.

```
Input: Two arbitrary finite sets $X$ and $X'$
begin
    Let $X^* = \emptyset$.
    for $x \in X \cup X'$ do
        if $rand(0,1) < 0.5$ then
            if $x \in X$ then
                Add $x$ to $X^*$
            end
        else
            if $x \in X'$ then
                Add $x$ to $X^*$
            end
        end
    end
end
Output: The crossover set $X^*$
```

To do crossover on two corresponding parameters or experience variables, we use crossover on bitstrings. More precisely, we use a straight-forward algorithm for crossover of bitstrings that combines two bitstrings into a third, all of the same length.

7.4 Offspring

Immediately after fertilization, a number of new animats (offspring) are formed and introduced into the ecosystem. For instance, the offspring could model a collection of sea turtle eggs in the sand or an embryo in the womb of a mammal. The number of new animats N is determined from the natural number parameter NOO (Number Of Offspring) of the parents by mutation followed by crossover applied to the NOO parameters of the parents. The number $q \in [0,1]$ is created similarly from the parameters $ResourceProportion$ of the parents. Now a new animat A_k is generated separately for each $1 \leq k \leq N$ as follows.

Starting with the genotype of A_k, its sets of sensors, motors, and homeostatic variables are inherited from the parents (they need to be the same in both parents for reproduction to take place). All sensors and motors are set to $False$. The value of homeostatic variable h is obtained by adding the corresponding homeostatic variables of the parents and then multiplying the result by q/N. Thus only the fraction q of the parents' joint resources is being passed on to the N offspring and evenly divided among them. Hence the offspring have relatively low values of their homeostatic variables when they are conceived. On the other hand, their burn-rate of resources is typically relatively low in the beginning. The pattern sets are formed individually for each offspring by first applying mutation (Algorithm 5) to the pattern sets of the parents and then crossover (Algorithm 7), again to the pattern sets of the parents. The action set is formed analogously. The reflex set, finally, is formed by using reflex mutation (Algorithm 6) followed by crossover (Algorithm 7) on the reflex sets of the parents. As a final step, reflexes that involve patterns or actions that do not belong to the offspring's own pattern and action sets are removed.

Parameters and experience variables are formed from the corresponding variables of the parents by bitstring mutation followed by bitstring crossover. This concludes the description of the genotype of A_k.

The initial phenotype of A_k is defined to be identical to the genotype of A_k. The conformation of A_k, finally, is left to be specified in each case. This concludes the definition of the new animat A_k. As soon as A_k has been conceived, it is inserted into the ecosystem and becomes subject to the updates of Algorithm 1.

Before moving on, let us note that the reproduction model just described enables new "species" to develop. In fact, reproduction involves mutation, which has an element of randomness and may lead from a population of animats that can interbreed into a population whose genomes are no longer sufficiently similar for general interbreeding.

8. Examples of ecosystems

In this section we will look at five concrete examples of ecosystems. The purpose is to show how the animat model works and at the same time illustrate its generality. Open sourced code for versions of the animat model along with some simple ecosystems can be found at (Mäkeläinen, Torén, and Strannegård, 2018a).

8.1 The frog in the forest

Consider the frog world shown in Figure 1. This world is populated by a single frog animat with the following genotype:

- Sensors: $green$, $blue$
- Patterns: $[\{green\}]$, $[\{blue\}]$
- Motors: eat, $drink$
- Actions: $[\{eat\}]$, $[\{drink\}]$
- Reflexes: none.

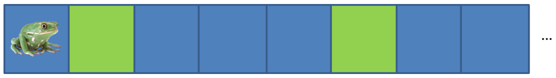

Figure 1: The frog in the forest. This ecosystem consists of a frog on an infinite path of cells that are green (edible insects) or blue (shallow water). The frog jumps forward at each tick automatically. It has two homeostatic variables (energy and water), whose values increase when the frog eats on green and drinks on blue cells, respectively. (Real frogs soak up water rather than drink, but this animat frog drinks.) At each tick, small quantities of energy and water are dissipated due to metabolism. This means that the animat eventually dies if it makes suboptimal actions too often.

- Homeostatic variables: *energy*, *water*

- Experience variables: The local Q-values and transition probabilities are set to 0.

- parameters: The parameters *LearningRate* and *DiscountRate* are set to standard values.

This frog animat follows the basic policy and quickly learns to eat on green and drink on blue cells. In fact, after a few ticks, we have

$$\begin{aligned} LQ(energy, [\{green\}], [\{eat\}]) &> 0 \\ LQ(energy, [\{green\}], [\{drink\}]) &< 0 \\ LQ(water, [\{blue\}], [\{drink\}]) &> 0 \\ LQ(water, [\{blue\}], [\{eat\}]) &< 0. \end{aligned}$$

This leads to the desired behavior (almost exactly as in Q-learning). In this case the phenotype pattern set will remain the same throughout the lifetime of the animat. In fact, it will never get surprised and no rules for adding or removing patterns will ever be triggered.

8.2 The frog in the swamp

Now consider a slightly different frog world, shown in Figure 2. This world is also populated by a frog animat, whose genotype looks exactly as in the previous example. This time the pattern set will change, however. In fact, when eating on a turquoise square for the first time, the frog will get punished, e.g. by losing energy due to vomiting. This punishment means that the animat gets surprised and the surprise triggers the rule *Pattern merge*, which in this case leads to the formation of the new pattern $[\{green, blue\}]$, representing turquoise. The animat will remember the bad experience now, since:

$$LQ(water, [\{green, blue\}], [\{drink\}]) < 0.$$

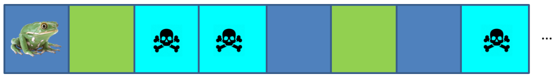

Figure 2: The frog in the swamp. This ecosystem is similar and consists of a frog on an infinite path of cells that are green (edible insects), turquoise (toxic insects), or blue (water). The frog jumps forward at each tick automatically. It has two homeostatic variables (energy and water) that go up when it eats on green and drinks on blue cells, respectively. If it eats on a turquoise cell (which is both blue and green), it will throw up and thus lose energy and water.

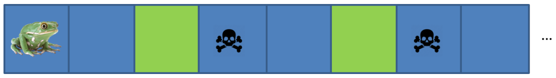

Figure 3: The frog in the creek. This example is similar to the other ones, but the twist this time is that the water to the right (immediately downstream) of green cells is toxic.

8.3 The frog in the creek

Now let us consider a third frog world, this time as in Figure 3. This world is again populated by a frog animat, whose genotype is the same as before. Again the animat gets surprised when it drinks toxic water for the first time. This surprise in turn triggers the rule *Pattern concatenation*. In fact, *Pattern merge* does not apply. This leads to the formation of the new pattern $[\{green\}, \{blue\}]$, representing green followed by blue. Again the animat will remember the bad experience, since

$$LQ(water, [\{green\}, \{blue\}], [\{drink\}]) < 0.$$

8.4 The sheep world

Now let us consider a two-dimensional ecosystem with sheep. Figures 4, 5, and 6 show this ecosystem at time 0, 97, and 123, respectively.

8.5 The sheep and wolves world

The sheep and wolves world at time 0 is shown in Figure 7. Figure 8 shows scaled biomass (total weight) curves for this ecosystem, indicating how the populations of sheep and wolves develop over time. Figures 9 and 10 show the learning curves for one of the sheep and one of the wolves, respectively.

9. Generality of the animat model

In this section we will analyze the generality of the animat model from three different perspectives.

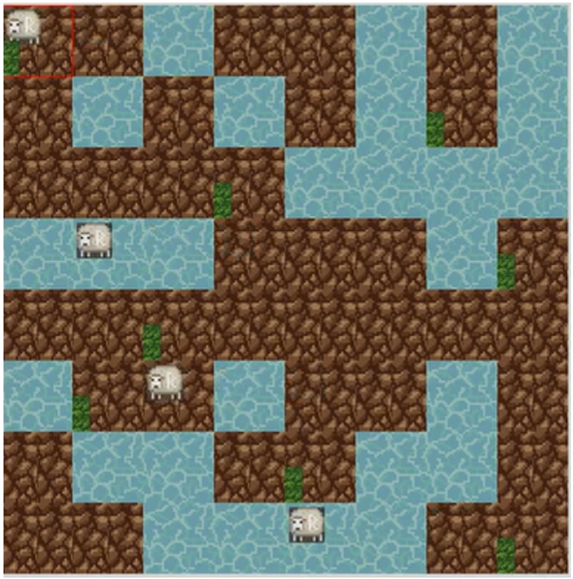

Figure 4: The sheep world at time 0. This world consists of green cells (grass), blue cells (water), and brown cells (sand). The world is populated with sheep, two males and two females. The sheep can move up, down, left, and right. They can also idle, eat, and drink. Moreover, they have sensors for green, blue and brown. Again their homeostatic variables are energy and water. Eating and drinking on green and blue cells give, respectively, energy and water reward. Eating or drinking on other cells yields a slight punishment in terms of both energy and water. The grass grows by propagation to adjacent cells and gets dissipated when the sheep graze there.

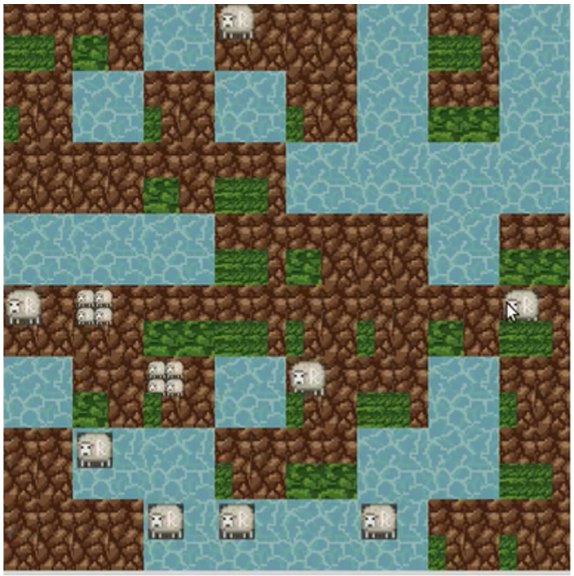

Figure 5: The sheep world at time 97. Here we see that the grass has grown and the sheep have reproduced. The new symbol here represents a pregnant female with offspring inside her body.

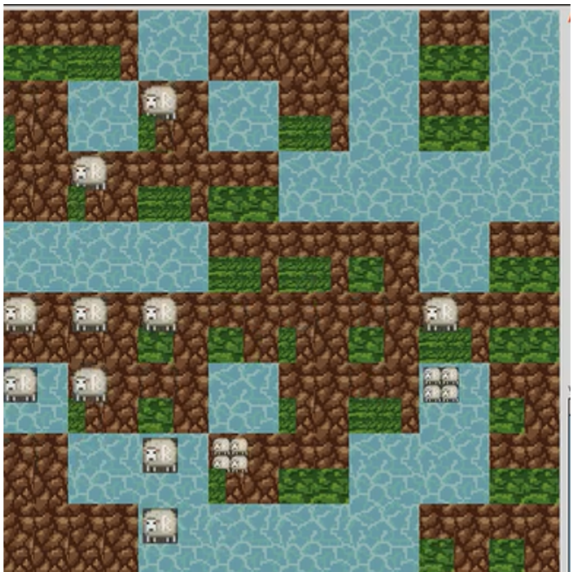

Figure 6: The sheep world at time 123. As the sheep multiply, the impact of competition gets more pronounced. For instance, the two sheep to the left might have difficulty finding grass. At this point several sheep have already died from lack of resources and others have died out of senescence. Several generations have also been born.

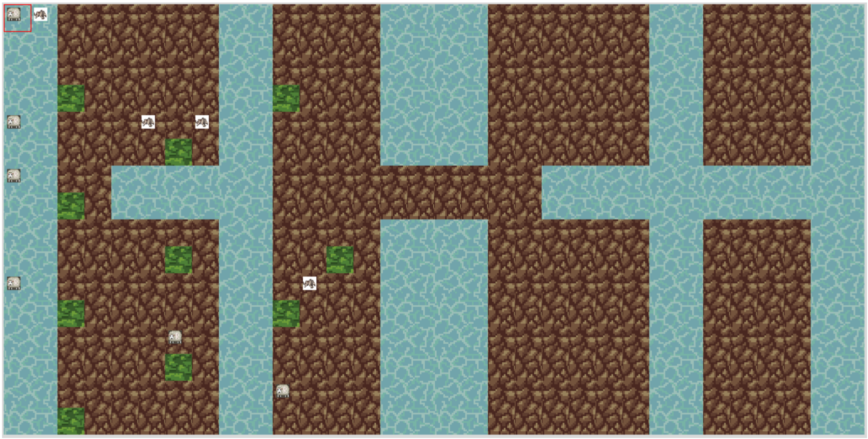

Figure 7: The sheep and wolves world at time 0. This world is initialized with 6 sheep, 4 wolves and 10 patches of grass. Both sheep and wolves have energy and water as their homeostatic variables. Both get their water from drinking at the blue cells. The sheep get their energy from grazing, whereas the wolves get theirs from predation, while eating on a cell that is shared with a sheep.

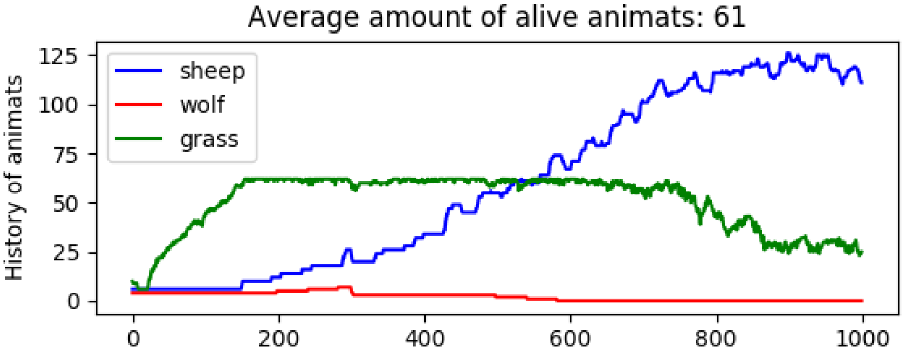

Figure 8: Biomass curve. These three curves show the development over time of the biomass for sheep, wolves, and grass, respectively.

9.1 Versatility

All animals including humans arguably evolved for solving a specific class of problems: surviving and reproducing in a wide class of ecosystems. This in turn presupposes the ability to solve a great number of naturally occurring tasks every day: find food, find water, find the way, avoid predators, find a mate, and find shelter. Humans evolved primarily for solving such tasks. Our ability to play Go, solve IQ-test problems, or prove mathematical theorems only arose as a secondary effect. Systems aiming at human-level general intelligence must nevertheless be able to learn to solve problems that are commonly associated with human intelligence in domains such as games, IQ-tests, and mathematics.

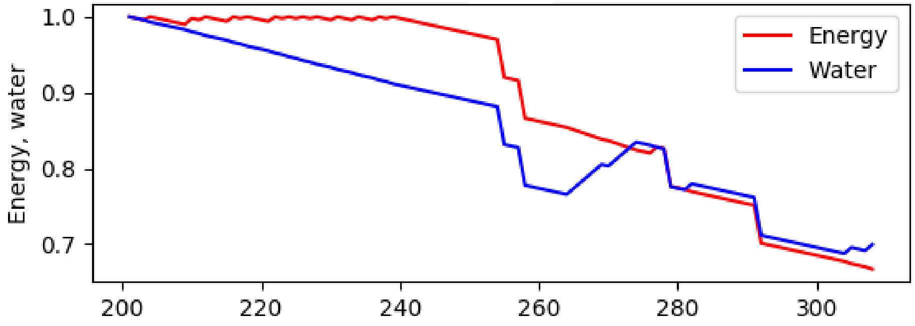

Figure 9: Learning curve for one of the sheep. This sheep was born at tick 200 with maximum levels of energy and water. After several ticks the sheep tried to eat grass and learned to eat. After tick 240 the sheep started to explore the world. We observe sharply declining curves between tick 252 and tick 258. The decline was caused by either inappropriate actions, e.g. eating or drinking in a sand cell, or being bitten by a wolf. Both would lead to significant decreases. After tick 260 the sheep drank water several times and quickly learned to drink to increase its level of water. Still, it has difficulty finding resources after about 282 steps.

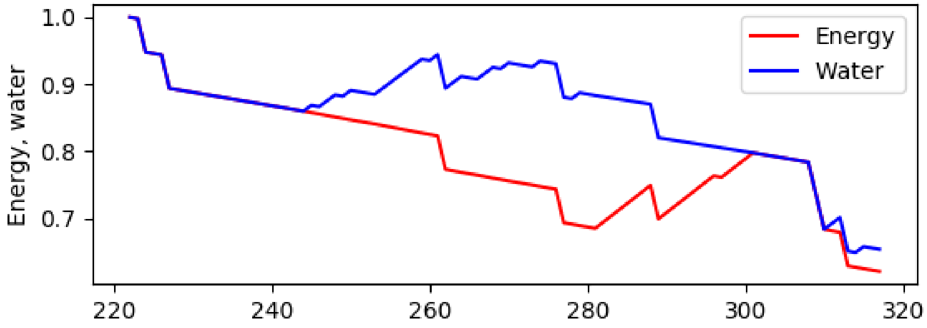

Figure 10: Learning curve for one of the wolves. This wolf was born at tick 220. After 20 steps it tried to drink water and learned it in a short time. At tick 260 and 276 the wolf took inappropriate actions in wrong cells which led to significant loss of energy and water. At tick 280 the wolf tried to eat for the first time and learned to eat in a few ticks. The wolf started to explore again after tick 300 when the curve began to decline.

Given the goal of constructing a fully autonomous system with general intelligence, continuous learning is fundamental, but reprogramming, tweaking, or knowledge injection by human engineers is not allowed once the system has been deployed. One and the same design should be able to learn to survive in a broad class of ecosystems, but also to master domains where humans excel, e.g. language, logic, and mathematics.

The versatility of the animat model has been explored in several studies, see below. Not surprisingly, the animat model typically performs below the level of domain-specific programs in each specific domain, but its focus is on breadth rather than depth. Below are some examples of domains that illustrate the versatility of the animat model.

Locomotion Different forms of animat locomotion, including frog animats that learn to jump by using both hind legs simultaneously and toad animats that learn to crawl by using its hind legs alternately were studied in (Strannegård et al., 2017).

Navigation Different forms of animat navigation, including landmark navigation and gradient-based navigation, were studied in (Carlsson, 2018).

Foraging The learning and decision-making algorithms of the animat model were improved significantly in (Mäkeläinen, Torén, and Strannegård, 2018b). Moreover, the ability of the animat to survive in large and noisy state spaces by learning to eat certain objects and avoid others was explored. The performance of the animat model was compared to that of a deep Q-network with a simple manually optimized architecture and found to be similar.

Language Basic language learning in animats was considered in (Johannesson, Nilsson, and Strannegård, 2018): learning to produce phonemes, to remember phonetic sequences, and to associate phonetic sequences to other sensory patterns. For instance, the animat learned to associate the phonetic sequence "cold" to the pattern $[\{cold\}]$, where *cold* is a sensor for cold temperature. Additional experience variables that model associations were used for this purpose.

Arithmetic Steps towards learning simple arithmetic, logic and grammar in the animat setting were taken in (Grund Pihlgren and Lallo, 2018) and also in (Strannegård et al., 2016). The latter approach depends on the addition of a working memory and the use of abstraction variables. The model learned algebraic axioms tentatively by generalizing from examples and used them for computing previously unseen arithmetic expressions. For instance, it learned $x \cdot 0 = 0$ from examples such as $2 \cdot 0 = 0$ and $3 \cdot 0 = 0$. It also learned other algebraic axioms like $x + y = y + x$ from examples. Then it used those axioms to compute previously unseen expressions such as $75 \cdot 3$.

9.2 Computational modeling

Now let us show how the animat model can emulate several common computational models.

Cellular automata As an example, consider a simple ecosystem with a 10×10 grid of cells (with or without torus topology). Each cell is either green or red (more formally: contains an object of type green or red). Each cell also contains an animat. These animats are all identical and have sensors sensitive to the color of their own cell and their neighboring cells. They should also have motors that enable them to 'paint' their own cells. They have reflexes that encode rules of an arbitrary cellular automaton. For example, one of the reflexes could be

$$([\{NR, EG, SG, WG\}], [\{paint\_green\}]).$$

Here N, E, S, W represent the cardinal directions, whereas R and G represent, respectively, red and green. Moreover, *paint_green* is a motor that causes the animat's own cell to turn green.

Genetic algorithms A version of genetic algorithms can be realized in any ecosystem with animats whose learning parameters are set to 0. This process will evolve a population of reflex agents. The process can be steered in the desired direction by providing reward to those agents that behave in the desired way. For simplicity the ecosystem can be defined so that all animats touch each other and never move. Thus they will always be close enough to each other to interbreed.

Swarm algorithms Consider a swarm of animats, each represented by a point in $\mathbb{Z}^3$. Similarly to cellular automata, the animats all have the same design. They have sensors that register the speed, course and distance of their immediate neighbors and perhaps also sensors that indicate the main direction to the center of the swarm. Their motors enable them to move forward, turn left, or turn right. Thus it is possible to model swarm-like behavior of a population of animats.

Ant algorithms Animats may leave a 'pheromone' trace of objects behind them whose 'smell' decreases over time. More formally, they may leave objects in their trail that are replaced by other objects over time. These animats may be equipped with sensors that indicate the main direction to the scent gradient. They may also have reflexes that cause them to steer in that direction. More details can be found in (Strannegård et al., 2017).

Bacteria cultures Growth and spread of bacteria (or grass) can be modeled by working in a two-dimensional grid of cells and starting with an arbitrary distribution of animats and food on the cells. The animats are all identical and they have only one homeostatic variable: *energy*. Their sets of sensors, patterns, motors, actions, and reflexes are all empty, reflecting the fact that bacteria lack nervous systems. When reproducing, an animat places a copy of itself on each of its four neighboring cells. Thus the animats will start spreading on the grid and when the food sources have been consumed, the population will go extinct.

Turing machines Let M be a one-tape Turing machine. Define an ecosystem consisting of a tape T with one object, a symbol from the alphabet of M, in each cell. The ecosystem contains a single animat A, which is a reflex agent capable of moving left or right on the tape and of replacing objects in cells. A uses special colors (more formally: objects of type red, green, etc.) for representing its internal states. When it perceives a certain pair of objects (symbol, color) in a cell, a reflex causes it to replace the objects of the next cell with a new pair of objects (symbol', color'), while moving or standing still. In this way, A will transform the tape just like M would have done.

9.3 Combining evolution and learning

Let us justify the fact that our model includes both evolution and learning by showing that one of these adaptive processes cannot compensate for the other. First let us show that learning is sometimes necessary for avoiding extinction of animat populations.

Example 10 *Consider the following genotype of a stationary barnacle animat.*

- *Homeostatic variable: energy*
- *Sensor: green, red*
- *Patterns:* $[\{green\}]$, $[\{red\}]$,
- *Motor: eat*
- *Actions:* $[\{eat\}]$, $[\{\}]$
- *Reflexes: none.*

Imagine the barnacle animat being attached to a rock on the sea floor and perceiving nearby objects that could either move by themselves or drift in the currents. Let us model an environment of this kind as a one-dimensional infinite tape with odd cells red and even cells green. We assume that the green cells at the beginning of the tape are edible, but gradually become more toxic. The red cells have the opposite properties: they are toxic at the beginning of the tape, but gradually become more edible. Since our animat model is able to learn, it can survive in this changing world by learning to eat green cells first and then gradually switching to red cells. In contrast, an animat without the ability to learn is like a reflex agent with a fixed behavior. Regardless of its eating behavior –whether it eats green or red objects or both or none– it will not survive in worlds that require adaptation.

Now let us show that the ability to learn is generally not enough for avoiding extinction of animat populations.

Example 11 *We may use the same setting as in the previous example, but this time we assume that the red cells are deadly to eat. Animats of the above kind will have to explore at some point and try to eat a red cell. When they do they will die.*

If the above animats are capable of reproduction (and mutation), on the other hand, then some with the following set of reflexes might eventually arise:

- *Reflexes:* $([\{green\}],[\{eat\}])$ *and* $([\{red\}],[\{\}])$.

These animats might live forever in the new dangerous world and always avoid the deadly threats.

10. Conclusion

We have presented an ecosystem model that emphasizes animal information processing. The model includes fully automatic genotype and phenotype development based on generic mechanisms of perception, decision-making, learning, reproduction, and death. In more technical terms, the model combines homeostatic decision-making, reinforcement learning, lifelong learning, one-shot learning, and architecture learning with evolution of parameters and networks. These properties set our model apart from other approaches that combine evolution and learning, e.g. (Ackley and Littman, 1992), (Yaeger, 1994), and (Such et al., 2017).

Our generic model is fully autonomous and enables continuous adaptation to new environments and survival in a range of simple worlds by learning basic forms of locomotion, foraging, and navigation. The model is also capable of learning simple cases of arithmetic and natural language. This indicates that the mechanisms of genotype and phenotype development that were modeled in this paper are capable of jointly producing rudimentary forms of general intelligence.

The model is still in an early phase and much work remains to be done: improving the learning and decision mechanisms, testing a wider class of ecosystems, and increasing the model's performance and generality.

Acknowledgement

This research was supported by the Torsten Söderberg Foundation ÖT17/18.

References

Ackley, D., and Littman, M. 1992. Interactions between Learning and Evolution. In Langton, C. G.; Taylor, C.; Farmer, C. D.; and Rasmussen, S., eds., *Artificial Life II, SFI Studies in the Sciences of Complexity.* Reading, MA, USA: Addison-Wesley. 487–509.

Ashton, B. J.; Ridley, A. R.; Edwards, E. K.; and Thornton, A. 2018. Cognitive performance is linked to group size and affects fitness in Australian magpies. *Nature* 554(7692):364–367.

Avila-García, O., and Cañamero, L. 2005. Hormonal modulation of perception in motivation-based action selection architectures. In *Proc. of the Symposium on Agents that Want and Like.*

Bach, J. 2015. Modeling motivation in MicroPsi 2. In *Proceedings of the 8th International Conference on Artificial General Intelligence*, 3–13. Springer.

Boehm, A.-M., et al. 2012. FoxO is a critical regulator of stem cell maintenance in immortal Hydra. *Proceedings of the National Academy of Sciences* 109(48):19697–19702.

Bohannon, J. 2008. Flunking spore. *Science* 322(5901):531–531.

Bouneffouf, D.; Rish, I.; and Cecchi, G. 2017. Bandit Models of Human Behavior: Reward Processing in Mental Disorders. In *Proc. of the 10th International Conference on Artificial General Intelligence*, 237–248. Springer.

Braitenberg, V. 1986. *Vehicles: Experiments in synthetic psychology.* MIT press.

Buro, M. 1998. From simple features to sophisticated evaluation functions. In *International Conference on Computers and Games*, 126–145. Springer.

Carlsson, M. 2018. Animat Navigation Using Landmarks. Master's thesis, Chalmers University of Technology.

Christensen, V., and Walters, C. J. 2004. Ecopath with Ecosim: methods, capabilities and limitations. *Ecological modelling* 172(2-4):109–139.

Coello, C. A. C.; Lamont, G. B.; Van Veldhuizen, D. A.; et al. 2007. *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer.

Cressey, D. 2015. Tropical paradise inspires virtual ecology lab. *Nature* 517(7534):255–256.

Draganski, B., and May, A. 2008. Training-induced structural changes in the adult human brain. *Behavioural brain research* 192(1):137–142.

Ernst, D.; Geurts, P.; and Wehenkel, L. 2005. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research* 6(Apr):503–556.

Fahlman, S. E., and Lebiere, C. 1990. The cascade-correlation learning architecture. In *Advances in neural information processing systems*, 524–532.

Futuyma, D. 2009. *Evolution.* Sinauer Associates.

Gabbay, D. M.; Hodkinson, I.; and Reynolds, M. 1994. *Temporal logic (vol. 1): mathematical foundations and computational aspects.* Oxford University Press, Inc.

Grund Pihlgren, G., and Lallo, N. 2018. Rule-Based Sequence Learning Extension for Animats. Master's thesis, Chalmers University of Technology.

Healy, S. D., and Jones, C. M. 2002. Animal learning and memory: an integration of cognition and ecology1. *Zoology* 105(4):321–327.

Hubbell, S. P. 2001. *The unified neutral theory of biodiversity and biogeography.* Monographs in population biology. Princeton University Press.

Johannesson, L.; Nilsson, M.; and Strannegård, C. 2018. Basic Language Learning in Artificial Animals. In *Biologically Inspired Cognitive Architectures Meeting*, 155–161. Springer.

Karakotsios, K., and Bremer, M. 1993. *SimLife: The official strategy guide.* Prima Pub.

Keramati, M., and Gutkin, B. 2011. A reinforcement learning theory for homeostatic regulation. In *Advances in neural information processing systems*, 82–90.

Keramati, M., and Gutkin, B. 2014. Homeostatic reinforcement learning for integrating reward collection and physiological stability. *Elife* 3.

Knuth, D. E. 1997. *The Art of Computer Programming*, volume 2. Pearson Education.

Koza, J. R. 1989. Hierarchical Genetic Algorithms Operating on Populations of Computer Programs. In *IJCAI*, volume 89, 768–774.

Koza, J. R. 1994. Genetic programming as a means for programming computers by natural selection. *Statistics and computing* 4(2):87–112.

Langton, C. G. 1997. *Artificial life: An overview.* MIT Press.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *Nature* 521(7553):436.

Lotka, A. J. 1925. *Elements of Physical Biology, by Alfred J. Lotka.* Williams & Wilkins.

Mäkeläinen, F.; Torén, H.; and Strannegård, C. 2018a. Code for Homeostatic Agents with Dynamic State Representation. `gitlab.com/fredrikma/aaa_survivability`.

Mäkeläinen, F.; Torén, H.; and Strannegård, C. 2018b. Efficient Concept Formation in Large State Spaces. In *Proc. of the 11th International Conference on Artificial General Intelligence, Prague, Czech Republic*, 140–150. Springer.

Niv, Y. 2009. Reinforcement learning in the brain. *Journal of Mathematical Psychology* 53(3):139–154.

Roijers, D. M.; Vamplew, P.; Whiteson, S.; Dazeley, R.; et al. 2013. A Survey of Multi-Objective Sequential Decision-Making. *J. Artif. Intell. Res.(JAIR)* 48:67–113.

Russell, S. J., and Zimdars, A. 2003. Q-decomposition for reinforcement learning agents. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 656–663.

Rusu, A. A.; Rabinowitz, N. C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; and Hadsell, R. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671.*

Sims, K. 1994. Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, 15–22. ACM.

Strannegård, C.; Nizamani, A. R.; Juel, J.; and Persson, U. 2016. Learning and Reasoning in Unknown Domains. *Journal of Artificial General Intelligence* 7(1):104–127.

Strannegård, C.; Svangård, N.; Lindström, D.; Bach, J.; and Steunebrink, B. 2017. The Animat Path to Artificial General Intelligence. In *Proceedings of the Workshop on Architectures for Generality and Autonomy 2017.*

Such, F. P.; Madhavan, V.; Conti, E.; Lehman, J.; Stanley, K. O.; and Clune, J. 2017. Deep Neuroevolution: Genetic Algorithms are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning. *arXiv preprint arXiv:1712.06567.*

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction.* MIT press.

Vladimirov, N.; Mu, Y.; Kawashima, T.; Bennett, D. V.; Yang, C.-T.; Looger, L. L.; Keller, P. J.; Freeman, J.; and Ahrens, M. B. 2014. Light-sheet functional imaging in fictively behaving zebrafish. *Nature methods* 11(9):883.

Von Neumann, J., et al. 1951. The general and logical theory of automata. *Cerebral mechanisms in behavior* 1(41):1–2.

Watkins, C. J. C. H. 1989. *Learning from delayed rewards.* Ph.D. Dissertation, King's College, Cambridge.

Wilson, S. W. 1986. Knowledge growth in an artificial animal. In *Adaptive and Learning Systems.* Springer. 255–264.

Wilson, S. W. 1991. The animat path to AI. In Meyer, J. A., and Wilson, S. W., eds., *From animals to animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, 15–21. MIT Press.

Yaeger, L. 1994. Computational Genetics, Physiology, Metabolism, Neural Systems, Learning, Vision, and Behavior or PolyWorld: Life in a New Context. In *Proceedings of the Artificial Life III conference*, 263–298. Addison-Wesley.

Yoshida, N. 2017. Homeostatic Agent for General Environment. *Journal of Artificial General Intelligence* 8(1):1–22.