Effects of Evolutionary and Lifetime Learning on Minds and Bodies in an Artifical Society

T. Buresch **A.E. Eiben** **G. Nitschke** **M.C. Schut**

Free University Amsterdam
{buresch,gusz,nitschke,schut}@cs.vu.nl

Abstract- In this paper we study a population of individuals in a simulated artificial environment. These individuals have a "body" as well as a "mind", i.e., some of their features effect their "physical" properties, like speed and strength, while other features influence their "mental" preferences and choices in interacting with the environment and other agents. We compare two approaches to adapting the minds of individuals. In approach 1, the bodies and the minds develop through evolution, while in appropach 2 only the bodies evolve and the minds are adapted by lifetime-learning. The results indicate that the evolutionary approach is able to sustain larger and more stable agent populations as well as maintain a higher degree of individual success compared to the lifetime learning approach. Furthermore, quite unexpectedly, the method used for mental development has a strong effect on the development of the physical features within the very same environment: The individuals' bodies evolve to completely different segments of the physical feature space under the two regimes.

1 Introduction

Many researchers have adopted the use of learning and artificial evolution mechanisms as a means of adaptation in simulated agent systems [NF02], [NP95], [MF95], [DB94], [PLHF01] though relatively few have focused research efforts on the use of learning or evolutionary mechanisms for agent controller design in the context of collective intelligence systems [WFP02]. The application of learning or evolutionary mechanisms for adaptive controllers [NF99] and the simultaneous evolution of agent body [Sim94] is an endeavor that has also received relatively little research attention in the context of collective intelligence systems.

In this paper we present a comparative study of different adaptive mechanisms in an artificial system containing thousands of agents. For the agents we distinguish minds (controllers, mental features) and bodies (physical features). For instance, a mental feature can be the preference for searching food (as opposed to searching other individuals to mate with). This feature affects an agent's behavior. A physical feature is, for instance, the agent's muscle mass that affects their speed.

The artificial environment is called Artificial Environment with Genetic Inheritance Simulation (AEGIS). Somewhat oversimplifying, it can be perceived as a model of an ecosystem, containing two entirely different types of inhabitants, where one type merely serves as resource (source of energy) for the other type. From this perspective we can call the inhabitants plants, respectively animals. Our system can also be seen as a model of a simple artificial society, where inhabitants of the second type have (obtain through adaptation) a "personality", i.e., individual characteristics that determine their attitude towards eating, mating, and fighting. It is important to note that the system has no abstract fitness function to be optimized, the only driving force is the quest for survival and reproduction. Throughout the rest of this paper we will not take any particular perspective on the AEGIS world and we will use the terms animal, individual, and agent as synonyms; the term inhabitant will stand for either plants of animals.

Previous work with this environment concentrated on the effects of environmental variations on the evolution of the population and the effect of allowing communication between the individuals, cf. [EEvH99]. Our findings can be briefly summarized as follows. First, we observed the best development[1] in those environments that allowed for rapid response. In particular, when the move costs were low (agents could move fast) and when the necessary break between two consecutive reproduction acts was short (the population could evolve quickly). Second, we found that having the possibility of communication can be a matter of life and death: communicating populations survived in demanding environments where mute ones became extint. A surprising outcome was the observation that in an "easy" world, where actions cost little energy, the individuals become more aggressive.

The present investigation is performed in the same artificial environment, although with different research objectives. Our primary objective here is to introduce an alternative to evolutionary adaptation of the agent controllers via individual or lifetime-learning and study the differences between the two approaches. The motivation behind this idea is that evolutionary learning is a *slow* mechanism – it takes place on the time scale of many generations. Individuals that are born with a better set of features tend to produce more offspring and the composition of the population is slowly changing towards superior features. Individual learning is a faster process as it takes place during an individual's lifetime. Each experience, i.e., interaction with the environment and/or another agents, delivers data and the individual can use data mining to develop a model of successful (re)actions. Such a model is then the core of its controller and it is being continuously adapted during the agents lifetime.

We deliberately restricted the scope of individual learn-

[1] We defined good development by a constantly high number of individuals, that is, large populations with only small and infrequent fluctuations.

ing to mental attributes. Applying it to physical features could be an analogy of sports excercises or any other physical training, but our interest here is the development of controllers. As a consequence, our comparative experiments to test the adaptability of AEGIS agents differ only in the adaptive mechanisms regarding the controllers. In the first experimental series both the bodies and controllers of agents have been evolved. In the second one we kept the evolution of bodies but used lifetime learning to develop the agent controllers. It is important to note that in the latter case we used a combination of learning and evolution, but these were not combined as is typical in many adaptive behavior studies [NF99]. The difference is that in our combined evolution+learning experiments the two adaptive mechanisms act *in a different space*: the individual learning mechanisms act in the space of agent controllers, while evolution acts in the space of physical agent properties.

The rest of the paper is organized as follows. In Section 2 we present the world and the agents of AEGIS. The adaptive mechanisms, evolution and individual learning, are discussed in Section 3. We describe the experimental setup in Section 4. The results are shown in Section 5 and further analyzed in Section 6. Finally, the paper is concluded by Section 7.

2 AEGIS

This section describes the world and the inhabitants of AEGIS. Space limitations prevent giving all details, for a complete description we refer to [Bur04].

2.1 AEGIS - World

Scape The artificial world is a two dimensional grid populated with *plants* and *agents*. The grid is wrapped around at its borders. Each point of the grid is called a *cell*. The scape contains `width` times `height` cells. A cell can hold at most one agent and one plant. The time on the scape is divided into *cycles*. In each cycle plants and agents can perform actions. Agents may perform more than one action in a cycle; plants can perform only one. At initialisation, the scape is populated uniformly with`initialPopulation` number of agents and `initialFlora` number of plants.

Plants Plants are the source of energy in the world. Each plant has a certain amount of `energy`. This energy increases by *growth* and decreases by *eating*. Plants are very simple creatures who can only perform one action per cycle, which action contains *growth* followed by *reproduction*. They are not evolving creatures, they have all the same features which do not change. When a plant is born, it gets a *birth energy*, which is uniformly drawn between `minBirthEnergy` and `maxBirthEnergy`. This interval is the same for all plants. Agents get energy from plants by eating them. A plant dies if its `energy` drops to zero. The plant's `energy` increases by `plantGrowth` in each cycle. If a plant reaches a certain energy level, `multThreshold`, it tries to reproduce to one neighbouring cell if there is no plant there. The neighbours of a cell are the ones at east, west, north and south. If there is no free cell around but the plant's `energy` is above `launchThreshold` then it launches a seed to a random place on the scape and dies. If the seed lands on a place which is occupied by an other plant, then it bounces and is lost. Otherwise it gets initialised with *birth energy*. If the `energyTransfer` flag is *true*, then the parent plant loses the *birth energy* of the newborn plant from its own `energy`.

2.2 AEGIS - Agents

This section briefly overviews the properties of our agents. The main components of our agent are its *body* and *controller*. Additionally, an agent can belong to a particular species (depending on its body characteristics), can perform actions, and has a certain energy, vision, speed, mobility, mortality and it can engage in direct competition by fighting.

Body Agents have 3 inheritable physical properties: gender, muscle and skin.[2] These properties determine their skills and their relations to other agents. Thus, firstly, agents have a `gender`, i.e., male or female. Secondly, they have `skin` and `muscle`, whose strength (skin thickness, resp. muscle mass) is represented by real values between $[0, 1]$. Higher values stand for thicker skin, respectively more muscle. The agents' physical features can evolve, because the `gender`, `skin`, and `muscle` parameters are inhertable as described in Section 3.

Species Based on the `skin` and `muscle` attributes we introduce speciation. Two agents belong to the same species if their `skin` and `muscle` attributes are close to each other: the difference between their `skin` attributes is less than 25% (0.25 on our scale); the same goes for `muscle`. Two agents from the same species can mate and propagate their genes through the offspring. Otherwise they are *foes*, because they consume the same resources but are useless for each other in terms of reproduction. At initialisation, the `skin` and `muscle` attributes are set in such a way that agents from the same species are close to each other geographically.

Controller The controller enables the agent to determine an action to perform in a certain situation. If there is no possible action to perform, then it does *nothing*. The controller is a parameterized decision procedure whose parameters (perceived as mental attitudes) are inheritable and evolve over time together with the physical attributes. Our second experiment is based on acquiring experience during lifetime and to use this experience to develope a controller. In this case no mental features undergo evolution. The exact mechanisms are described in Section 3.

Genetic makeup Inheritable properties of the agents are encoded in their chromosome. These chromosomes are subject to evolution for they undergo variation (cf. Section 3) and selection. For the first type of controller (with mental attitudes), the chromosome consists of 6 genes: 3 genes for the physical properties (gender, muscle and skin) and 3 for the mental properties (attack, food, and social). This chro-

[2] Other physical properties, e.g., vision, are not genetically encoded, therefore are not evolvable.

mosome is shown in Figure 1. The chromosome for the second type of controller contains only the 3 genes for physical properties.

Actions Agents can perform the following actions: *move* - move to a neighbouring cell, east, west, north or south; *eat* - eat the plant on its current location; *mate* - perform reproduction with another agent and give life to a child; and *attack* - hit another agent. Agents can perform more actions per cycle.

Energy and eating Agents need energy to live. It is needed to perform actions, however some actions may be performed free of cost. Agents obtain energy by eating plants. The *eat* action is performable if there is a plant and the agent is hungry. An agent is hungry when its energy is below maxEnergy. The energy is transfered from the plant to the agent by a bite. The bite cannot be bigger than the *hunger* or the plant's energy. An *eat action* is cost free. Doing *nothing* costs idleCost energy.

Fighting Agents can fight for resources, i.e., plants. Agents within the same species do not fight against each other. An *attack* action means one hit to a foe. An *attack* action costs fightCost, which is substracted from energy. The *attack* action can be performed if the attacker has more than fightCost energy and there is at least one foe around (east, west, north or south). A hit decreases the attacked's energy by two times fightCost and increases its age by *damage*. If the attacked agent's skin is 0 then it dies.

Vision An agent can see its surrounding square shape area with radius vision. The vision attribute of an agent is drawn uniformly between minVision and maxVision at its birth.

Speed The number of actions an agent can perform is determined by its speed, which depends on skin and muscle. We omit the exact formula here, but the idea is that an agent with more muscle can perform more actions; with thicker skin, it can perform fewer. Each action performed by an agent increases its age. Thus fast agents are get sooner weared and teared.

Mobility Agents explore the world to find food, friends and foes. The *move* action consists of one step to a neighbouring cell (east, west, north or south). It costs moveCost which is substracted from the agent's energy. The *move* action can be performed if there is at least one free direction and the agent has more than moveCost energy.

Mortality Agents have an age attribute which tells how many actions they have performed in their life. The age attribute is increased by one if the agent performs an action. The lifetime is limited by deathAge. If an agent's age reaches deathAge then it dies. When an agent is born, its age is set to 0 and its deathAge is uniformly drawn between controllereathAge and maxDeathAge. An agent may also die when its energy level is 0. At initialisation, energy is uniformly choosen between minInitialEnergy and maxInitialEnergy.

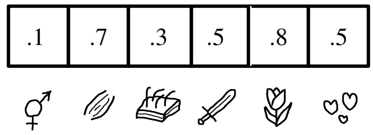

Figure 1: Example chromosome setup of agent with inheritable physical and mental attributes.

3 Adaptive Mechanisms

All agents have the same kind of body as outlined above, but they can have two types of controllers: either based on *mental attitudes* or on an eaction chooser developed by observations. We make this distinction because of the two learning types involved that enable the controller to adapt: evolutionary learning and lifetime learning. Evolutionary learning tweaks the mental attitudes of agents; lifetime learning works directly on an action chooser adjusting it by the observations that an agent makes. In all experiments, the body of the agents is subject to evolutionary learning.

This Section describes the adaptive learning mechanisms for the agent's body (only evolutionary learning) and controllers (both evolutionary and lifetime learning).

3.1 Body

Mating and reproduction

A child is born if two agents mate. Several conditions must hold for them to make reproduction possible: 1) they must be of opposite sex; 2) they must be close enough to each other (maximum distance is 1 cell); 3) they both must be fertile: their age is between beginChildBearingAge and endChildBearingAge; 4) they both must have energy $>$ sexCost; 5) there must be a cell around the initiating parent that can hold the child; 6) their last reproduction action must have been more than sexRecoveryPeriod number of actions ago; 7) they must be from the same species; and 8) they must have similar mental attitudes (explained in detail below).

Chromosomes are real-valued vectors, with gene values between 0 and 1. The child's chromosome is constructed from the parent's chromosomes using 1-point crossover followed by Gaussian mutation with mean zero and standard deviation mutationSigma.

Crossover The crossover operator produces one chromosome. If chromosome length is n then a crossover point cp is uniformly drawn from $[0, n]$. The first cp number of genes are get from the first parent and the remaining genes after position cp are get from the second parent. Who is the first, is randomly choosen.

Mutation After the crossover operator produced the new chromosome, a Gaussian mutation operator is applied to it. The mutation point mp is uniformly drawn from $[1, n]$. The selected gene undergo mutation. A random value drawn from a Normal distribution with mean 0 and standard deviation mutationSigma is added to the gene, cutting its

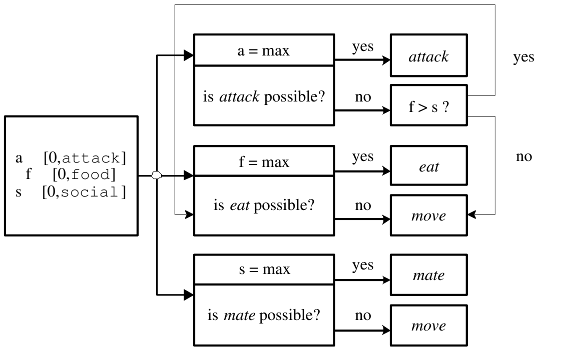

Figure 2: Agent type that performs actions based on mental attitudes.

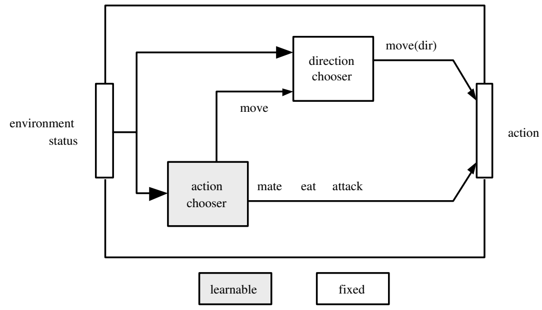

Figure 3: Agent type that performs actions directly based on observations.

value to fall between $[0, 1]$.

$$gene'_{mp} = \max(0, \min(1, gene_{mp} + \sigma))$$

where $\sigma \in N(0, \texttt{mutationSigma})$.

3.2 Controller

3.2.1 Evolution

Consider Figure 2 which shows the agent controller type based on *mental attitudes*. This controller contains the following attitudes: `food` - measure of affinity to eat; `social` - measure of affinity to be social; and `attack` - measure of affinity to fight. These attitudes can take values between $[0, 1]$. The higher the value the agent has higher affinity for a certain action. At initilisation, these mental properties are initialized randomly, drawn uniformly between $[0, 1]$. The agent type is taken from related work on emerging mental features [EEvH99].

Based on its mental attitude, an agent decides to either *move*, *eat*, *mate* or *attack*. The action chooser draws three random numbers uniformly: $a \in [0, \texttt{attack}]$, $f \in [0, \texttt{food}]$ and $s \in [0, \texttt{social}]$. Let f be the agent's current affinity to *eat*, a its affinity to *attack* and s for being social, i.e., to *mate*. Let $m = \max(a, f, s)$. As shown in Figure 2, the agent's controller now prescribes the action that it will perform. If an agent chooses to *move* but cannot perform this (because it has not enough energy), then it does *nothing*. Note that doing *nothing* may has cost.

The same evolutionary process that was explained above, which was used for developing the physical properties (skin and muscle), is applied here for developing the mental attitudes. As mentioned before, evolution in this case works on a chromosome consisting of both physical and mental properties (6 genes) instead of only physical properties (3 genes).

3.2.2 Lifetime Learning

Consider Figure 3 that shows the agent controller type that chooses actions directly based on observations (instead of indirectly via mental attitutes). The agent receives on its input information about observed *food*, *friends* and *foes* around. How much it can observe, depends solely on its `vision`. An agent also knows about its internal state: 1) its `energy`, 2) its `sexRecover`, and 3) whether it is `underAttack`.

Figure 3 shows two major components: the *action chooser* and the *direction chooser*. The *action chooser* can be learned, the *direction chooser* is fixed.

Action chooser The *action chooser* is a decision tree that outputs a particular action to perform in a given situation. At initialisation, the agent gets an handmade decision tree to rank actions. Throughout the agent's life, it collects experience, based on which it builds a new action chooser.

Experience is a set of *observations*. An observation is an (*environment*, *status*, *action*, *success*) tuple. When an agent performs an action it creates an observation and adds it to its experience. An agent can only hold `xpCapacity` number of observations. If it reaches its capacity then a randomly chosen observation is thrown away and the new one is added to its experience. This is called *forgetting*.

An observation contains the *status* of the agent (`energy`, `sexRecover` and `underAttack`), the state of the *environment*, the *action* it decided on, and the `success` of the action, which is an integer value. A more successful action gets a higher `succes` value. Success is determined by *events* occurred during this and next action. Each event has a weight, which can be positive or negative. This weight is added to `success` every time an event occurs. The following events are recorded: 1) *mate* - an agent takes part in reproduction, which is weighed by `benefitMate`; 2) *attack* - an agent attacks another agent, which is weighed by `benefitAttack`; 3) *being attacked* - an agent is attacked by another one, weighed by `benefitBeingAttacked`; 4) *eat* - an agent eats food, weighed by `benefitEat`; 5) *starvation* - an agent's `energy` is below `maxEnergy` / 2 and is weighed by `benefitStarvation`.

We denote the average *success* of all actions performed by an agent by `kSucc`. (Note that we can measure `kSucc` for agents with mental attitudes as well.)

The action part of the observation contains the action type (*move*, *eat*, *mate* or *attack*). In case of *move*, it also contains the selected *direction*.

The action chooser uses a decision tree to predict the success of an action if performed in a certain situation. The input of this decision tree is based upon the structure of the observations. The action chooser predicts the success of all

Exp	Body	Controller
1.1	evolutionary	evolutionary
1.2	evolutionary	lifetime learning
2.1	evolutionary	constant (best evolved)
2.2	evolutionary	constant (best learned)

Table 1: Experimental design.

Exp	Population size		Avg `kSucc`	
	Mean	Stdev	Mean	Stdev
1.1	8888.64	379.76	3.41	0.32
1.2	1660.93	892.77	1.39	0.71
2.1	9866.40	18.02	4.57	0.05
2.2	4004.24	1254.52	1.69	0.58

Table 2: Statistics over collected data.

performable actions in the given situation using its decision tree, and chooses the most successful action. For learning, Weka's J4.8 decision tree builder[3] was used with default parameters [Qui93].

Direction chooser Direction choosing determines the direction in which an agent *move*s. Each free direction is evaluated by the `evalDirection` function. This function works based on the energy of the agent: if the energy is under 50% of `maxEnergy`, then the agent is inclined to move towards food; if it is over 75% of `maxEnergy`, then it moves towards foes; in all other cases it moves towards friends.

4 Experiments

In the following we describe the experimental setup including the experimental design, the parameters that we fixed throughout all experimental sessions, and the variable parameters that we monitored throughout running the experiments.

4.1 Experimental Setup

Table 1 shows the design of the experiments reported here. The physical body of the agent (`skin` and `muscle`) is always learned evolutionary. As for developing the controllers, we conducted two experiment series, each consisting of two subseries. In the first series the agents adapt their controllers by means of either evolutionary learning (1.1) or lifetime learning (1.2). In two series of control experiments we took the most successful controller from these runs and executed simulations with all individuals having the same controller, not changing during a run. In these cases the agents still adapt their physical properties (skin and muscle) by means of evolutionary learning. For each of these four series, 10 independent runs were conducted.

4.2 Fixed Parameters

We took the experimental parameters for the world and agents from [EEvH99] of which this work is a continuation. These parameters include, among others, initial population size, maximum life time, movement cost, sex recovery, etcetera. For details we refer again to [Bur04].

4.3 Monitors

For all experimental sessions, we recorded basic statistical properties (mean, standard deviation, minimum, maximum) of the following measures in each run: 1) number of agents and plants; 2) number of agent births and deaths; 3) average speed of agents; and 4) average number of different actions performed per cycle. Additionally, we recorded the `skin-muscle` distributions in the last cycle of each run.

[3]Weka is an open source datamining library written in Java. J4.8 is Weka's implementation of the latest C4.5 Revision 8 decision tree builder algorithm.

5 Results

The obtained graph results of the experiments are shown in Figures 4 to 11. Each Figure shows the result of a representative run of a particular session. The results concerning the physical developments of the agents are shown as density maps on the `skin-muscle` plain, where a darker colour means higher density (more agents) in a given region measured in the last population.

Besides these visual representations of the experimental outcomes we also provide some statistical data in Table 2. For each experimental session, we show the population size (number of agents) and the average `kSucc` (explained in Section 3).

6 Analysis

Here we consider the otcomes as shown in Figures 4 to 11 and the statistics in Table 2.

Population Dynamics From experiments 1.1 and 1.2 we observed that both the evolutionary and lifetime learning populations can lead to oscillating and stabilising populations. However, the majority of evolutionary populations stabilises, whereas the majority of lifetime populations keeps oscillating. This is illustrated in Figures 4, 8 and Figures 6, 10 that show typical runs. It can also be derived from Table 2 where the ratios between mean and standard deviations are much smaller for the evolutionary run than for the lifetime learning run. For comparing sessions 1 and 2, we look particularly at *phase transitions* which are the time points at which an oscillating population changes into a stabilising one. We see that the a phase transition for evolutionary learning occurs earlier in the control experiment (2.1) than in the standard experiment (1.1), giving an extra indication of the superiority of the evolved controller used in experiment (2.1).

Statistics Firstly, we observe that the evolutionary populations in both the standard (by factor 4) and control experiments (by factor 2) are much larger than the lifetime learning populations. Secondly, the evolutionary populations are more stable with max standard deviation of 5%, while the lifetime learning populations have standard deviations up to 50%. Thirdly, we observe that evolutionary learning obtains more successfull controllers. This is surprising, since it only implicitly optimizes `kSucc`, whereas lifetime learning explicitly optimizes it. Finally, for the control experiments (2.1 and 2.2), we see that both mechanisms beat the standard ones (1.1 and 1.2) - this means that in both cases something was indeed learned.

Physical Development To our surprise, we observed that the two mechanisms lead to different physical bodies even though the environment remained the same. This can be concluded from the density maps that show the `muscle-skin` distributions. Additionally, we observe that the evolutionary controllers in experiments 1.1 and 2.1 are consistent with each other - both obtain physical bodies in the lower-right quadrant of the `skin-muscle` density maps. We also observe that the lifetime learning controllers are not consistent with each other in experiments 1.2 and 2.2. This leads us to conclude that lifetime learning may not be able to develop stable physical bodies.

7 Conclusions

From a very coarse-graded perspective we can observe that our system does have a stable state with large, constant populations, cf. Figures 4 and 6. Whether or not such a stable state is reached depends on the ”personalities” of the inhabitants, that is, on the agent controllers which, in turn, depend on the applied mechanism of adaptation. To this end see Figures 8 and 10.

From the perspective of comparing the the evolutionary approach with the lifetime learning approach to empower mental development (that is, the adaptation of the agent controllers) the following can be noted. In the AEGIS word, as used here, the evolutionary approach was clearly more successful in the sense that agent populations whose controllers have been evolved were larger and more stable than those whose controllers have been learned. Additionally, these evolved agents obtained a higher degree of success, cf. Table 2. These differences were observable in the experiments with the adaptation mechanisms working on-the-fly, as well as in the control experiments that utilized the results of these in a nonadaptive fashion.

An intresting angle for evaluating our experimental data is that of the development of the bodies. Considering that the environment (properties of plants, etc.) and the adaptation mechanism applied to the bodies of the agents were the same in both types of experiments, we expected no significant difference between the emerged physical properties of the agents. However, the results clearly show that the mechanism utilized to derive an agent controller strongly affects the development of agent body. Further research is required to investigate the scope of this effect, to study under which circumstances, i.e., when, it happens and to find explanations clarifying why it happens.

Bibliography

[Bur04] T. Buresch. Evolutionary versus lifetime learning of agent’s controllers on a gridworld. Master’s thesis, Free University Amsterdam, 2004.

[DB94] F. Dellaert and R. Beer. Toward an evolvable model of development for autonomous agent synthesis. In *Artificial Life IV*, pages 246–253, Cambridge, USA, 1994. MIT Press.

[EEvH99] A.E. Eiben, D. Elia, and J.I. van Hemert. Population dynamics and emerging features in AEGIS. In W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M. Jakiela, and R.E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, pages 1257–1264. Morgan Kaufmann, 1999.

[MF95] M. Mitchell and S. Forrest. Genetic algorithms and artificial life. In *Artificial Life: An Overview*. MIT Press, Cambridge, USA, 1995.

[NF99] S. Nolfi and D. Floreano. Learning and evolution. *Autonomous Robots*, 7:89–113, 1999.

[NF02] S. Nolfi and D. Floreano. Synthesis of autonomous robots through evolution. *Trends in Cognitive Science*, 6:31–36, 2002.

[NP95] S. Nolfi and D. Parisi. Learning to adapt to changing environments in evolving neural networks. Technical Report 95-15, Institute of Psychology, National Research Council, Rome, Italy, 1995.

[PLHF01] J. Pollack, H. Lipson, G. Hornby, and P. Funes. Three generations of automatically designed robots. *Artificial Life*, 7:1–11, 2001.

[Qui93] J.R. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann, San Francisco, 1993.

[Sim94] K. Sims. Evolving virtual creatures. In *Computer Graphics (SIGGRAPH’94) Annual Conference Proceedings*, pages 15–22, New York, USA, 1994. ACM SIGGRAPH Press.

[WFP02] R. Watson, S. Ficici, and J. Pollack. Embodied evolution: distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems*, 39:1–18, 2002.

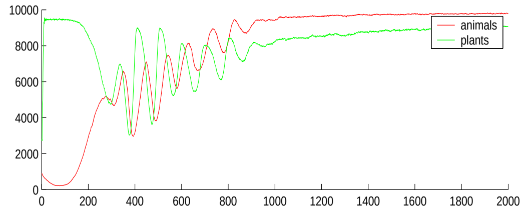

Figure 4: 1.1A - Population dynamics using evolutionary learning for bodies and minds

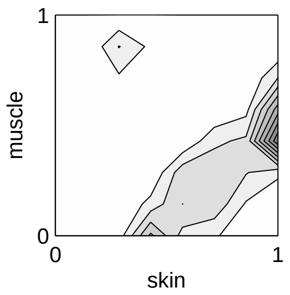

Figure 5: 1.1B - Emerged body features using evolutionary learning for bodies and minds

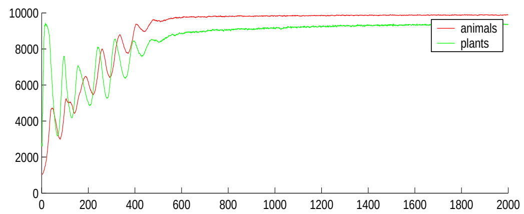

Figure 6: 2.1A - Population dynamics using the best evolved controller in the minds and evolutionary learning for bodies

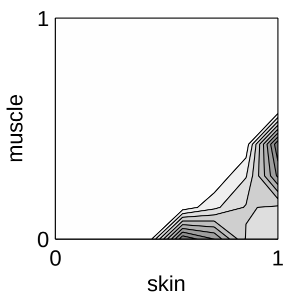

Figure 7: 2.1B - Emerged body features using the best evolved controller in the minds and evolutionary learning for bodies

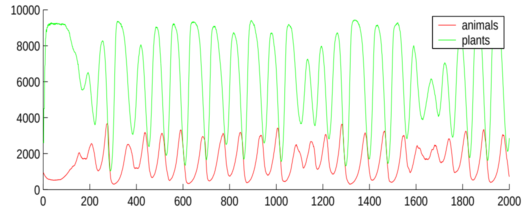

Figure 8: 1.2A - Population dynamics using evolutionary learning for bodies and lifetime learning for minds

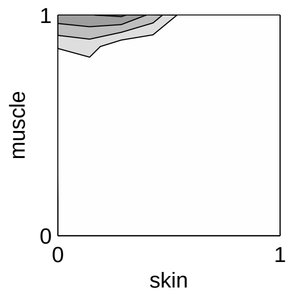

Figure 9: 1.2B - Emerged body features using evolutionary learning for bodies and lifetime learning for minds

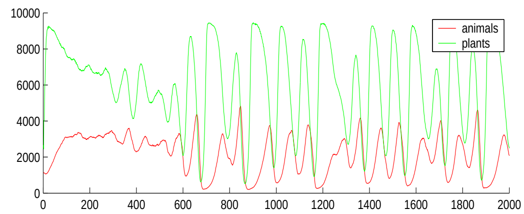

Figure 10: 2.2A - Population dynamics using the best learned controller in the minds and evolutionary learning for bodies

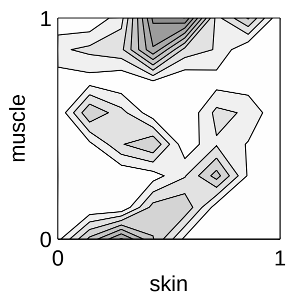

Figure 11: 2.2B Emerged body features using the best learned controller in the minds and evolutionary learning for bodies