Automated Reverse Engineering of Agent Behaviors

Wei Li

Supervisors:

Dr Roderich Groß

Prof Stephen A. Billings

A Thesis Submitted for the Degree of
Doctor of Philosophy

30th September 2015

Artificial evolution
is the end of
engineering's
hegemony.

Kevin Kelly
Out of Control

Abstract

This thesis concerns the automated reverse engineering of agent behaviors. It proposes a metric-free coevolutionary approach—*Turing Learning*, which allows a machine to infer the behaviors of agents (simulated or physical ones), in a fully automated way. In our approach, a population of candidate models competitively coevolves with a population of classifiers. The classifiers observe the models and agents. The fitness of the classifiers depends solely on their ability to discriminate between the models and agents. Conversely, the fitness of the models depends solely on their ability to 'trick' the classifiers into categorizing them as agents. As such, the approach does not require any predefined metrics to quantify the difference between the models and agents.

The merits of *Turing Learning* are demonstrated using three case studies. In the first case study, the machine automatically infers the behavioral rules of a group of homogeneous agents only through observation. A replica, which resembles the agents under investigation in terms of behavioral capabilities, is mixed into the group. The models are to be executed on the replica. The classifiers observe the motion of each individual in the swarm for a fixed time interval. Based on the individual's motion data, a classifier makes a judgment indicating whether the individual is believed to be an agent or the replica. The classifier gets a reward if and only if it makes the correct judgment. The models, on the other hand, are evolved to mimic the behavior of the agents and mislead the judgment of the classifiers. In the second and third case studies, *Turing Learning* is applied to infer deterministic and stochastic behaviors of a single agent through controlled interaction, respectively. In particular, the machine is able to modify the environmental stimuli (to which the agent responds) and thereby interact with the agent. This allows the machine to reveal the agent's entire behavioral repertoire. This interactive approach proves superior to learning through passive observation.

Acknowledgments

Contents

Abstract v

Acknowledgments vii

1 Inferring Individual Behaviors Through Interactive Turing Learning 1

1.1 Introduction . . . 1
1.2 Methodology . . . 2
1.2.1 Models . . . 3
1.2.2 Classifiers . . . 3
1.2.3 Optimization Algorithm . . . 4
1.2.4 Fitness Calculation . . . 4
1.3 Case Study One . . . 5
1.3.1 Deterministic Behaviors . . . 5
1.3.2 Simulation Setup . . . 7
1.3.3 Results . . . 8
1.3.3.1 Analysis of Evolved Models . . . 8
1.3.3.2 Coevolutionary Dynamics . . . 10
1.3.3.3 Analysis of Evolved Classifiers . . . 12
1.3.3.4 Noise Study . . . 17
1.3.3.5 Using a Single-Population Evolutionary Algorithm . . . 18
1.3.3.6 Coevolution of Tests and Models . . . 20
1.4 Case Study Two . . . 21
1.4.1 Stochastic Behaviors . . . 21
1.4.2 Simulation Setup . . . 23
1.4.3 Results: Two States . . . 24
1.4.3.1 Analysis of Evolved Models . . . 24
1.4.3.2 Analysis of Evolved Classifiers . . . 25
1.4.4 Results: Three States . . . 25
1.4.4.1 Analysis of Evolved Models . . . 25

1.4.4.2 Analysis of Evolved Classifiers 26
1.5 Summary 26

1 Inferring Individual Behaviors Through Interactive Turing Learning

1.1 Introduction

In the previous chapters, we demonstrate that how *Turing Learning* can be used for inferring swarm behaviors only through observation. This is based on an implicit assumption that the behavioral repertoire of agents in the swarm could be fully revealed through passive observation. That is, from the perspective of system identification, the target system has high observability. However, when the target system has low observability, some hidden information may not be revealed only through passive observation. Instead, the machine needs to interact with the agent in an active way to explore the hidden information from the agent. Based on this idea, in this chapter we aim to infer such agent behaviors through *Turing Learning* with interaction.

Observation and interaction are widely adapted by ethologists when investigating the behavior of animals [167, 168, 169]. When investigating animals' behavior in their natural habitat, passive observation is preferable as it is difficult to change the environmental stimuli. In this case, inferring the causal relations between the animal's behavior and its environmental stimuli may become challenging, since the stimuli are not under the observer's control. However, when the experiments are carried out in a controlled laboratory, it is possible to actively change the stimuli to interact with the animals under investigation in a meaningful way. In [169], in order to investigate causes of the dung beetle dance, biologists designed various experiments such as the appearance of disturbance or obstacles to interact with the dung beetle and learn how it adapts to the environmental changes.

In this chapter, we investigate whether a machine could infer the agent behaviors through actively interacting with the agents in an autonomous manner using the *Turing Learning* method. To validate this, two case studies are presented: deterministic agent behavior (Section 1.3.1) and stochastic behavior (Section 1.4.1). In these two case studies, the machine is able to control the agent's environmental conditions, which in this thesis corresponds to the intensity of the ambient light. At the same time, it is capable of simulating the actions of the agent. The learning result is a model of the agent that captures its behavior in relation to the environmental stimulus.

The advantages of our approach are twofold:

- Firstly, it does not rely on a pre-defined metric for gauging the resemblance of models to the agent. Rather, such metrics are implicitly defined by the classifiers, and hence incorporated into the evolutionary process.

- Secondly, the machine learns the agent's behavior by interacting with it, rather than simply observing its behavior in a passive manner. This interaction can help the machine to extract all of the agent's behavioral dynamics, as will be shown in the results section.

This chapter is organized as follows. Section 1.2 describes the methodology used, including the agent behaviors (deterministic and stochastic) and the implementation of the metric-free method. Section 1.3.3 presents the results of learning the deterministic agent behaviors, including analyses of the evolved models and classifiers, the coevolutionary fitness dynamics, the effect of noise on the algorithm's behavior. It also presents a comparison of the coevolutionary approach with a single-population evolutionary approach and a approach based on coevolution of tests (inputs) and models. Section 1.4.3 and Section 1.4.4 presents the results of learning stochastic behaviors. Section 1.5 summaries the chapter.

1.2 Methodology

In this chapter, we extend *Turing Learning* described in Chapter **??** with interactive ability. The basic idea is the same, that is, the method is comprised of two populations: one of models, and one of classifiers, which coevolve with each other competitively. The fitness of the classifiers depends solely on their ability to distinguish the behavior of the models from the behavior of the agent. The fitness of the models depends solely on their ability to mislead the classifiers into making the wrong judgment, that is, classifying them as the agent. In this following, we will describe the implementation that is related to the work in this chapter.

1.2.1 Models

The models are represented by a set of parameters govern the rules of the agents. The details of these parameters will be described in Section 1.3.1 and Section 1.4.1 As we have argued in the previous chapters, explicit representation (i.e., evolving only the parameters) makes it feasible for us to objectively gauge the quality of the models obtained.

1.2.2 Classifiers

The structure of the classifiers is similar to the one used in Chapter **??** (see Fig. **??**). The only difference is the classifiers take the environmental stimuli and the agent's response as inputs and the outputs can be used to control the environmental stimuli and make judgment. For the proof-of-concept study, we only use one environmental stimulus and stimulate the agent behaviors in one-dimensional environment. Fig. 1.1 shows the structure of the classifiers used in this chapter.

Suppose the agent responses to the level of light intensity in the environment, and we assume that the system could observe the agent's speed corresponding to the light intensity. One of the inputs to the network (classifier) is the light intensity in the environment at time step t, $I^{(t)} \in [0, 1]$, and the other input is the speed $v^{(t)}$. In order to make this setup more feasible to implement, it is assumed that the system cannot directly measure the speed of the individual, but rather its position. The speed of the individual for the classifier's input is then calculated by subtracting the previous estimated position from the current estimated position, and dividing the resulting number by the time interval between two measurements.

In order to make a judgment between a model and the agent, the classifier observes the behavior (speed) over a period of time. In addition, the classifier is also in control of the light intensity in the individual's environment. At time $t = 0$, the value of the light intensity is chosen randomly with a uniform distribution in the range $[0, 1]$. The neural network is then updated, using $I^{(0)}$ and $s^{(0)}$. The value of the light intensity for the next time step is obtained from the classifier's output neuron O_1, and the process repeats.

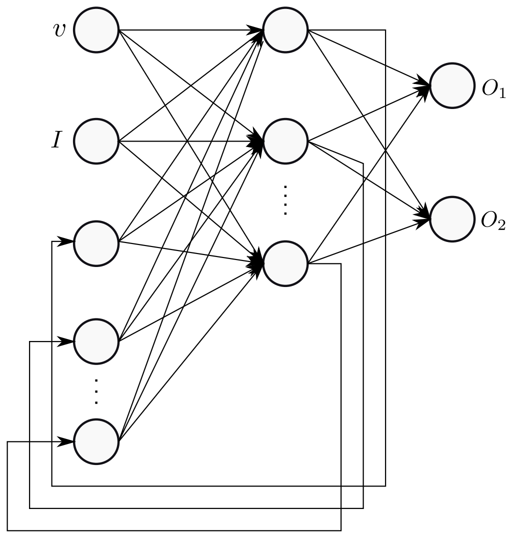


Figure 1.1: This diagram shows the structure of the classifiers used in this chapter. It is a recurrent Elman neural network [151] with two inputs (agent speed, v and environmental stimulus, I), three hidden neurons, and two output neurons (O_1 and O_2). O_1, which controls the stimulus, is fed back into the input; O_2 is used for making a judgment. Two bias neurons with a constant input of 1.0 are connected to each neuron of the hidden and output layers. See the text for details.

After having iterated through all the time steps (a single trial), the final value of output neuron O_2 is used to make a judgment: the network decides on a model if $O_2 < 0.5$, and on the agent if $O_2 \geq 0.5$. The memory (value of hidden neurons) of the classifiers is reset at the end of every trial.

1.2.3 Optimization Algorithm

The algorithm used here is based on a $(\mu + \lambda)$ evolution strategy with self-adaptive mutation strengths [170, 153]. It is the same as the one used in Chapter **??**. For the details of the implementation, see Section **??** of Chapter **??**.

1.2.4 Fitness Calculation

Suppose the population sizes for the model and classifier are N and M, respectively. The fitness of each model is obtained by evaluating it with each of the classifiers in the competing population (N in total). For every classifier that wrongly judges the model as being the agent, the model's fitness increases by $\frac{1}{N}$. The final fitness is in $[0, 1]$.

The fitness of each classifier is obtained by using it to evaluate (i) each model in the competing population (M in total) once, and (ii) the agent L times with different initial light intensities. For each correct judgment of the model and the agent, the classifier's fitness increases by $\frac{1}{2 \cdot M}$ and $\frac{1}{2 \cdot L}$, respectively. The final fitness is in $[0, 1]$.

1.3 Case Study One

To validate our method, we present two case studies: deterministic behaviors and stochastic behaviors. The behaviors to be identified were chosen to serve as a tractable test-bed for proof-of-concept study. While it may loosely correspond to how some 'real' agents react to the stimuli in their environment, it is not intended to mimic any specific animal. In these behaviors, non-trivial interaction with the agent is critical for leading the agent to reveal all of its behavioral repertoire.

1.3.1 Deterministic Behaviors

We simulate a one-dimensional[1] environment in continuous space. The simulation advances in discrete time steps $t \in \{0, 1, 2, \dots\}$. The (ambient) light intensity in the environment, I, can be varied continuously between 0 and 1 (see Fig. 1.2). The agent distinguishes between three levels of light intensity, low ($0 \leq I < I_L$), medium ($I_L \leq I \leq I_H$), and high ($I_H < I \leq 1$). Hereafter, these levels will be referred to as L, M, and H.

If the light intensity is at level M at time t, the speed of the agent, $s^{(t)} \in \mathbb{R}$, varies linearly with $I^{(t)}$ as:

$$s^{(t)} = k \left(I^{(t)} - 0.5 \right), \tag{1.1}$$

where k is a constant.

The agent's behaviors for levels L and H depend on the previous levels of light intensity (i.e. the agent has memory). The two behaviors are symmetrical to each other. Here, we

[1] In principle, the system will still work in higher dimension. As the main focus is to show how a machine interacts with the agent, for the sake of simplicity, we chose one-dimension.

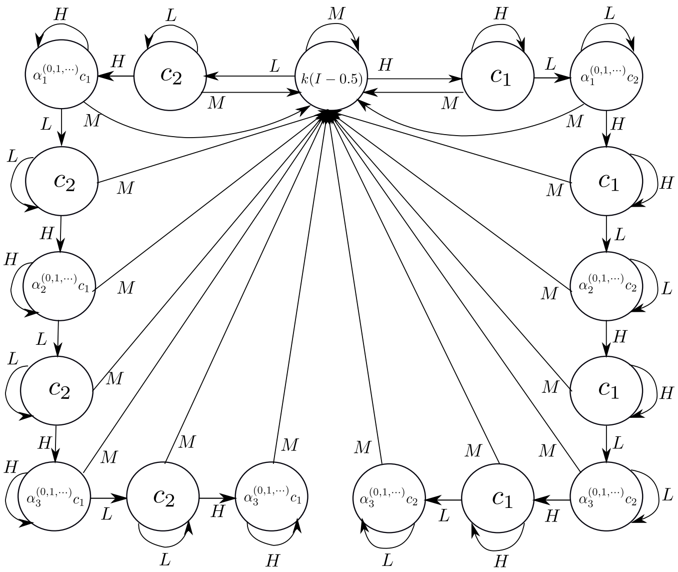


Figure 1.2: The deterministic agent behavior under investigation. It shows how the agent responses to the level of light intensity (L, M and H) in its environment. Each state represents the agent's speed. See texts for details.

Table 1.1: This table shows the change of the agent's speed (shown in Fig. 1.2), for an example sequence of light levels.

level	M	H	L	H	L	L	L	H	H	L	L
speed	$k(I-0.5)$	c_1	c_2	c_1	$\alpha_1^0 c_2$	$\alpha_1^1 c_2$	$\alpha_1^2 c_2$	c_1	$\alpha_1^1 c_1$	c_2	$\alpha_2^1 c_2$

L	H	H	L	L	H	H	H	M	H	L	L
$\alpha_2^2 c_2$	c_1	$\alpha_2^1 c_1$	c_2	$\alpha_3^1 c_2$	c_1	$\alpha_3^1 c_1$	$\alpha_3^2 c_1$	$k(I-0.5)$	c_1	c_2	$\alpha_1^1 c_2$

will describe the behavior for level L; the behavior for level H is obtained by exchanging L with H and I_L with I_H in the following description.

When the light intensity is at level L, the agent's default speed is $c_2 = k\,(I_L - 0.5)$ (which is a constant value as shown in Fig. 1.2), and remains at that value as long as the light intensity remains at level L. If the light intensity is at level H (for any number of time steps), and then immediately changes to level L, and remains at that level for at least one more time step, then the agent's speed decays exponentially with a rate of α_1: that is, in the first time step that the light intensity is at level L, the agent's speed is $\alpha_1^0 k\,(I_L - 0.5)$; it then changes to $\alpha_1^1 k\,(I_L - 0.5)$, $\alpha_1^2 k\,(I_L - 0.5)$, and so on as long as the light intensity remains at level L. The agent now registers that α_1 has been activated. If another $H \rightarrow L \rightarrow L$ sequence is observed, the agent's speed now decays exponentially with a rate of α_2. If further $H \rightarrow L \rightarrow L$ sequences are observed, the exponential decay rate becomes and remains at α_3. Note that at any time, if the agent observes a light intensity at level M, its speed is proportional to the light intensity, as shown in Eq. 1.1, and it forgets all its past observations of the light intensity.

The behavior of the agent can thus be represented by five cases; one where the agent's response to the light intensity is *proportional* (which occurs whenever the light intensity is at level M); one where the agent's response is *constant* (i.e. the agent's speed reaches the lower and upper saturation values (c_1 or c_2) as shown in Fig. 1.2); and three where the agent's response *decays exponentially* with the decay rates α_1, α_2 and α_3, respectively.

Table 1.1 shows an example sequence of light levels, along with the corresponding speed of the agent (i.e., the speed shown in Fig. 1.2).

Here, I_L and I_H are set to 0.1 and 0.9 respectively. k is set to 1.25; hence, the lower and the upper saturation values of the speed are $k\,(I_L - 0.5) = -0.5$ and $k\,(I_H - 0.5) = 0.5$. The exponential decay rates are set to: $\alpha_1 = 0.8$, $\alpha_2 = 0.4$, $\alpha_3 = 0.2$. Thus, in each case, the agent's speed decays exponentially towards zero. Note that these values (k, α_1, α_2 and α_3) are chosen arbitrarily and the coevolutionary method is not sensitive to them.

The system identification task is to learn these four parameters (k, α_1, α_2 and α_3) of the agent. It is worth while to mention again that to learn α_3, the machine needs to at least output the sequence: $H \rightarrow L \rightarrow L \rightarrow H \rightarrow L \rightarrow\rightarrow L \rightarrow H \rightarrow L \rightarrow L$ or $L \rightarrow H \rightarrow H \rightarrow L \rightarrow H \rightarrow H \rightarrow L \rightarrow H \rightarrow H$. Since I_L and I_H are set to a very low and high value in $[0, 1]$, it is very unlikely to randomly generate such sequences.

1.3.2 Simulation Setup

We use three setups for the metric-free methods. The setup, in which the classifier is in control of the light intensity in the agent's environment, is hereafter referred to as the "Interactive" setup. In order to validate the advantages of the interactive approach, we compared it against the situation where the classifier only observes the agent in a passive manner; that is, it does not control the light intensity in the environment. We considered two such setups: in the first setup (hereafter, "Passive 1") the light intensity is randomly chosen from the uniform distribution in $[0, 1]$, in every time step. In the

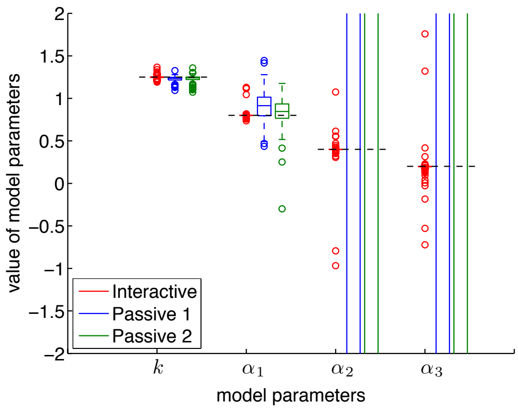


Figure 1.3: This plot shows the distributions of the evolved models with the highest subjective fitness in the 1000$^{\text{th}}$ generation in the coevolutions. Each box corresponds to 100 coevolution runs. The dotted lines correspond to the values of the four parameters that the system is expected to learn (i.e. those of the agent). From top to bottom, these are 1.25, 0.8, 0.4, 0.2, respectively. Note that in order to zoom in on the relevant range, some boxes and outliers are omitted from the plot.

second setup (hereafter, "Passive 2"), the light intensity is randomly chosen only after certain number of time steps (in this setup the number is chosen to be 10). All other aspects of these two setups are identical to the "Interactive" setup.

The population sizes of the models and classifiers are chosen to be 100, respectively. We performed 100 coevolution runs for each setup. Each coevolution run lasts 1000 generation. In one generation, each classifier conducts 100 trials on the agent. In each trial, the classifier observes the agent for 10 s at 0.1 s intervals, that is, a total of 100 time steps.

1.3.3 Results

1.3.3.1 Analysis of Evolved Models

Fig. 1.3 shows a box plot with the distributions of the evolved models with the highest subjective fitness in the 1000$^{\text{th}}$ generation over 100 coevolution runs of the three setups. The passive coevolutions are able to evolve the parameters k and α_1 with a reasonable accuracy; however, they are not able to evolve α_2 and α_3. In the Passive 1 coevolution,

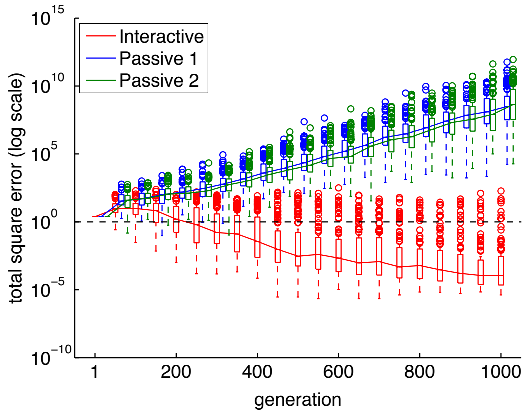


Figure 1.4: This plot shows the total square errors of the evolved model parameters compared to those of the agent over generations. The models with the highest subjective fitness in each generation are selected. Each box corresponds to 100 coevolution runs, and the solid lines correspond to the median error.

the relative errors of the medians of the four evolved parameters $(k, \alpha_1, \alpha_2, \alpha_3)$ with respect to those of the agent are 1.2%, 14.3%, 7.8×10^4%, and 2.3×10^5%, respectively. The Passive 2 coevolution leads to similarly large relative errors in the evolved values of α_2 and α_3. This phenomenon can be explained as follows. If the light intensity changes randomly (either every time step, or every ten time steps), it is unlikely that the $H \to L \to L$ and/or $L \to H \to H$ sequences will occur enough times, without a level of M in between, such that the classifiers can observe the effects of α_2 and α_3. Therefore, the classifiers do not evolve the ability to distinguish the behavior of models from the behavior of the agent with respect to these two parameters, and in turn, these parameters do not converge to their true value in the model population.

In contrast to the passive coevolutions, the Interactive coevolution is able to evolve all the four parameters with a good accuracy. The relative median errors are 0.024%, 0%, 0.025% and 0.15% for k, α_1, α_2 and α_3 respectively. This implies that by the 1000^{th} generation, the classifiers have learned how to control the pattern of the light intensity in such a way that they can distinguish models from the agent based on the effect of any of the four parameters. Therefore, in order to compete for being selected in the population, the models are evolved to behave like the agent in every aspect.

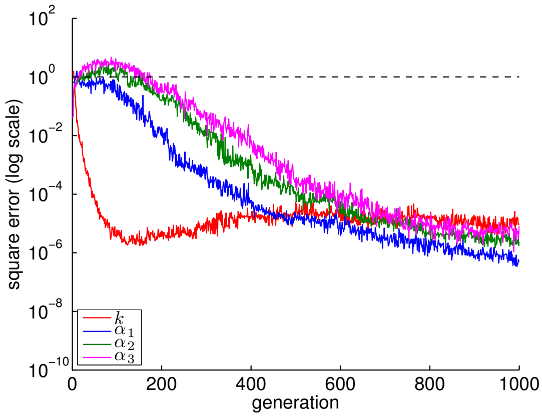
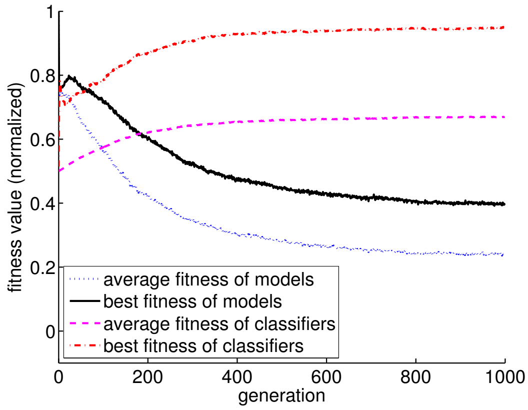


Figure 1.5: This plot shows how the square error in the individual model parameters changes over the generations in the Interactive coevolution. The curves correspond to median values from 100 coevolution runs.

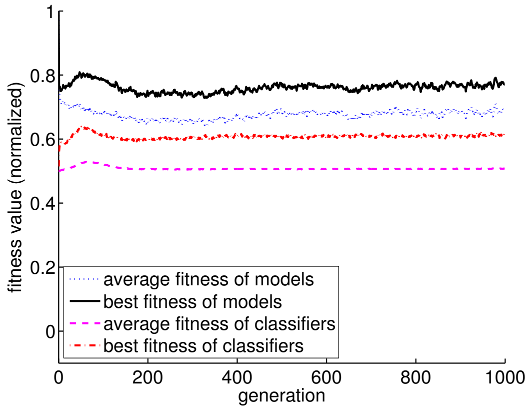
1.3.3.2 Coevolutionary Dynamics

Fig. 1.4 shows the dynamics of the coevolutionary algorithms. The horizontal axis shows the generation, whereas the vertical axis shows the total square error of the model parameters, that is, the sum of the square errors in the four parameters (of the individual with the highest subjective fitness in each generation) with respect to their true values. In the case of the Interactive coevolution, the median error starts to reduce after around the 100^{th} generation, and keeps decreasing until the last generation where it reaches a value of 10^{-4}. In contrast, in the case of the passive coevolutions, not only does the median error not decrease, but it increases to a value of 10^8 by the 1000^{th} generation.

We now analyze how the four individual parameters evolve during the course of the Interactive coevolution, which is the only fully-successful setup. The plot shown in Fig. 1.5 reveals how the learning proceeds in the coevolution. Parameter k is the first to be learnt, followed by α_1, while parameters α_2 and α_3 take a longer time to approximate the true values. This means that the classifiers first learn to distinguish models from the agent on the basis of k and α_1. This ability of the classifiers drives the model population to evolve k and α_1, in order to mislead the classifiers. Eventually, the classifiers also learn to exploit the effects of α_2 and α_3 in order to make the right decision; thereby driving the model population to evolve these two parameters accurately. After about the 600^{th} generation, the learning of the four parameters proceeds with approximately identical rates.



(a) Interactive coevolution



(b) Passive 1 coevolution

Figure 1.6: This plot shows the subjective fitness (normalized) of the classifiers and the models in (a) the Interactive coevolution, and (b) the Passive 1 coevolution. The curves show the average fitness across 100 coevolution runs.

In order to analyze why the Interactive coevolution is successful while the passive ones are not, we can look at the dynamics of the subjective fitnesses of the classifiers and the models (as defined in Section 1.2.4) during the course of the coevolution. As both of the passive coevolutions fail to converge, we present the analysis of fitness dynamics only for Passive 1 coevolution (the other case was found to have similar dynamics). Fig. 1.6 shows the fitness dynamics of the Interactive and the Passive 1 coevolutions. In the case of the Interactive coevolution (see Fig. 1.6(a)), the average fitness of the classifiers starts off at 0.5, which means that the classifiers make decisions that are no better than random decisions. However, the classifiers quickly improve in fitness, which in turn causes the fitness of the models to decrease. This increases the selective pressure on the models. After about 600 generations both the fitness of classifiers and models reach a steady state, which according to Fig. 1.5 corresponds to the region where the four parameters evolve with virtually identical rates. In the case of the Passive 1 coevolution (see Fig. 1.6(b)), the average fitness of the classifiers also starts off at 0.5. In the first few generations, this increases slightly, because the classifiers learn how to distinguish models from the agent on the basis of parameters k and α_1. However, the models quickly adapt to this new ability of the classifiers. Now, as the classifiers are unlikely to have the opportunity to observe the effects of α_2 and α_3, their average fitness returns to 0.5. This leads to a disengagement phenomenon, in which there is no more meaningful selection in the model population, therefore leading the parameters α_2 and α_3 to drift, the effect of which can be seen in Fig. 1.4.

1.3.3.3 Analysis of Evolved Classifiers

The model described in Section 1.2 is defined by four parameters: $\{k, \alpha_1, \alpha_2, \alpha_3\}$. In order to evaluate the quality of the evolved classifiers we performed a grid search over the space of the amount of disturbance (i.e. noise) injected into each of the four parameters. We used 11 noise magnitude settings per parameter, $M \in \{0, 0.1, \ldots, 1\}$, with the noise being added to the parameters as follows:

$$p' = p(1 + \mathcal{U}(-M, M)), \tag{1.2}$$

where $p \in \{k, \alpha_1, \alpha_2, \alpha_3\}$ represents any of the four parameters, and $\mathcal{U}(-M, M)$ denotes a uniform distribution on the interval $(-M, M)$. Note that $M = 0$ corresponds to no noise, whereas $M = 1$ means that the noisy value of the parameter can be between 0 and twice its actual value.

For the sake of simplicity, we only analyzed the classifiers with the highest subjective fitness in the last generation of the 100 coevolutions. For each of the 100 classifiers, and for each combination of noise magnitudes, we conducted 100 trials. In other words, we conducted $100 \cdot 11^4 \cdot 100 = 146,410,000$ trials in total. In each trial, we modulated the agent's parameters according to Eq. 1.2. Each trial was run for T = 100 time steps (the same setting used within the coevolutions). For each combination of magnitudes employed, the overall performance U was computed as the sum of the final judgments

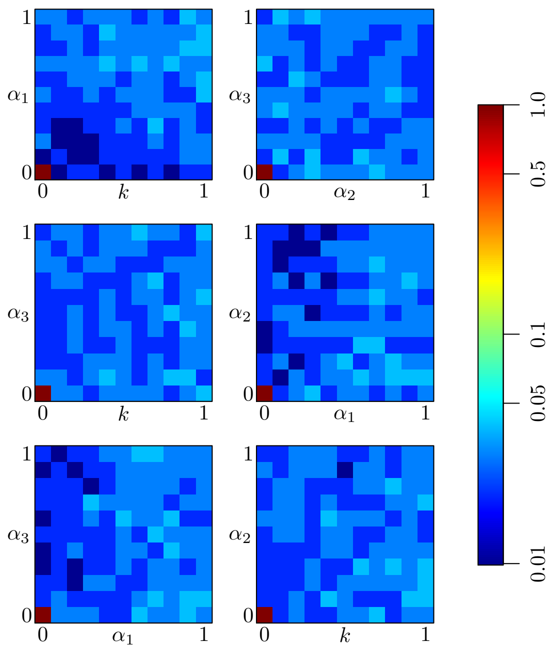


Figure 1.7: This figure shows the landscape of U^* (log scale) for the overall best classifier over the six sub-spaces with two parameters as degrees of freedom. Each axis in each plot ranges between 0 and 1, corresponding to the minimum and maximum magnitude of noise added into each parameter respectively. See the text for more details.

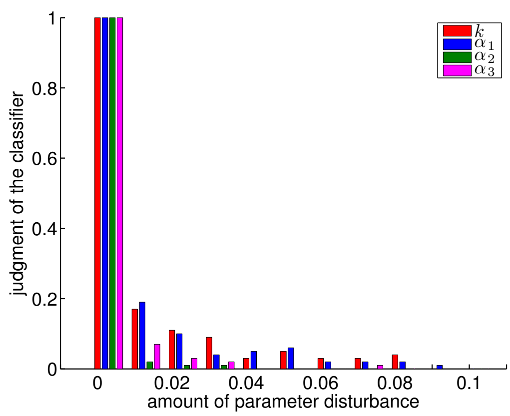


Figure 1.8: This plot shows the average judgment [U, see Eq. 1.3] of the best evolved classifier over 100 trials when the magnitude of noise added into each parameter of the agent is within 0.1.

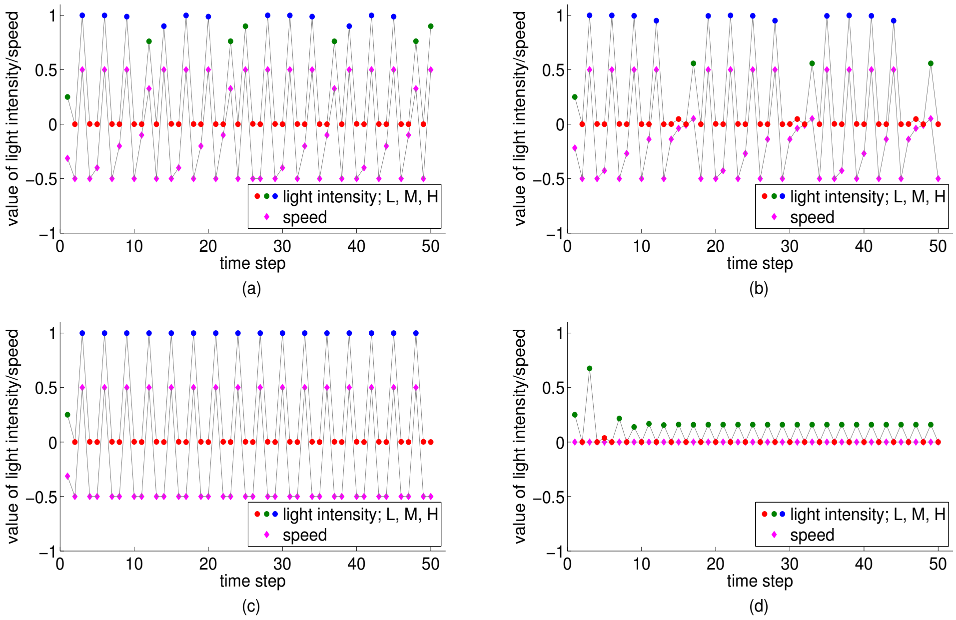


Figure 1.9: This plot shows the light intensity sequences as output by the overall best classifier (circular points), along with the corresponding speeds of the agents (diamond-shaped points) in four trials conducted on different agents: (a) the agent; (b) a model similar to the agent, generated randomly according to Eq. 1.2 with $M = 0.5$; (c) a model whose speed is linear to the light intensity; and (d) a static model, whose speed is always 0 (i.e. $k = 0$). The colors of the circular points (red, green, blue) correspond to light intensities at levels L, M and H, respectively.

of the classifier in each trial, divided by the total number of trials, N:

$$U = \sum_{i=1}^{N} J_i/N, \tag{1.3}$$

where, $J_i \in \{0, 1\}$ is the final judgment of the classifier in trial i, and N is the total number of trials conducted. Note that $J_i = 0$ and $J_i = 1$ imply that the classifier has judged the behavior as being that of a model and the agent, respectively.

In order to find the overall best classifier from the 100 classifiers analyzed, we defined the following metric:

$$W = [1 - U(0,0,0,0)] + \frac{1}{\Omega} \sum\sum\sum\sum_{\substack{i,j,k,\ell \in \{0,0.1,\ldots,1\} \\ i^2+j^2+k^2+\ell^2 \neq 0}} (i+j+k+\ell)\, U(i,j,k,\ell), \tag{1.4}$$

where $\Omega = 29282$ is the maximum value that the quadruple sum can achieve (i.e. if all the U's are equal to 1). The first term in Eq. 1.4 penalizes the classifier if $U < 1$ when there is no noise on the parameters; they are thus undisturbed and identical to the ones of the agent. The second term penalizes the classifier if $U > 0$ for any non-zero noise combination, with increasing penalties being applied to higher noise magnitudes. The normalization of the second term by Ω serves to make the two terms contribute equally to W, which can take values in $[0, 2]$. Note that the minimum value of $W = 0$ can only be achieved by the perfect classifier, that is, one that outputs 1 if $i = j = k = \ell = 0$ and 0 otherwise. In our case, the best classifier achieved a value of $W = 3.7 \cdot 10^{-3}$.

The performance landscape of the models is 5-dimensional (4 model parameters plus performance measure), and cannot be visualized directly. Therefore, we considered each combination of two model parameters ($\binom{4}{2} = 6$ combinations) as a sub-space, and for each point in this sub-space, we calculated the performance measure as the maximum value over the sub-space spanned by the remaining two parameters. For instance, on the sub-space (k, α_1), the performance measure $U^*(k, \alpha_1)$ was calculated as:

$$U^*(k, \alpha_1) = \max_{\alpha_2,\alpha_3} U(k, \alpha_1, \alpha_2, \alpha_3). \tag{1.5}$$

Note that for all the points except $(0,0,0,0)$, Eq. 1.5 corresponds to the worst-case scenario for the classifiers, because the value of U for the ideal classifier at these points is 0. For the point $(0,0,0,0)$, the value of U for the ideal classifier is 1.

Fig. 1.7 shows the landscape of U^* (log scale) for the overall best classifier over the six sub-spaces. When interacting with the agent (i.e. $i = j = k = \ell = 0$), the output of the classifier is always 1. This corresponds to the point $U^*(0,0)$ in the six sub-spaces of Fig. 1.7 (here only the maximum is shown). When interacting with the models (i.e. when the parameters of the agent are perturbed), for any combination of noise magnitudes, the average output of the classifier is below 0.05 for the six sub-spaces.

In order to further analyze how sensitive the overall best classifier is when the parameters of the agent are only slightly perturbed, we set the maximum magnitude of noise added to each parameter to 0.1, and used a resolution of 0.01. In this evaluation, we only injected noise into one parameter at a time. For each noise magnitude, the classifier was evaluated in 100 trials (with different initial light intensities, and randomly-generated noise), and the average performance measure was computed according to Eq. 1.3.

Fig. 1.8 shows the average judgment [U, see Eq. 1.3] of the best classifier, which was evaluated in 100 trials. With increasing noise magnitudes, the average judgment of the classifier approaches zero, which means that the classifier can identify the agents as models more consistently. The classifier has different sensitivities to the four parameters; the effects of noise on α_2 and α_3 on its judgment are higher than those of k and α_1. For instance, in the case of α_2, even with $M = 0.01$, the classifier correctly judges the behavior as being that of a model in 98% of the trials. As we have seen, k and α_1 are the first two parameters to be identified in the coevolution. α_2 and α_3 are addressed in the later generations of the coevolutionary process, and it seems that the classifier is more sensitive to these two parameters.

We now investigate how the overall best classifier interacts with the agents. We conducted four trials with four different agents: (a) the agent; (b) a model similar to the agent, generated randomly according to Eq. 1.2 with $M = 0.5$; (c) a model whose speed is linear to the light intensity without exponential decay (i.e. $k = 1.25$, $\alpha_1 = \alpha_2 = \alpha_3 = 1$); and (d) a static model, whose speed is always 0 (i.e. $k = 0$). In each trial, the initial light intensity was set to 0.25. The trials lasted for 100 time steps.

Fig. 1.9 shows the sequences of light intensity output by the classifier, along with the speed of the agents in the four trials, for the first 50 time steps (we observed that the last 50 time steps constitute a repetition of the first 50). As we can see, the classifier outputs different sequences of light intensity in order to interact with the different agents. The greater the difference between a model and the agent, the more varied is the sequence of light intensities that the classifier outputs when interacting with it as compared to the one it outputs when interacting with the agent. For the agent, the classifier repeatedly outputs a $H \rightarrow L \rightarrow L$ sequence, in order to fully reveal the behavior (this is analyzed in detail in the following paragraph). For the model that is similar to the agent (see Fig. 1.9(b)), the sequence of light intensities is similar to the one produced for the agent, but it is not identical. Interestingly, for the model without exponential decay (see Fig. 1.9(c)), the classifier produces the $H \rightarrow L \rightarrow L$ sequence even more often than it does for the agent, but it does not set the light intensity to level M. For the static model (see Fig. 1.9(d)), the classifier never outputs a $H \rightarrow L \rightarrow L$ sequence; instead, it outputs a sequence that alternates between M and L.

We now focus on analyzing the behavior of the agent in Fig. 1.9(a). Initially, the classifier outputs the sequence $L \rightarrow H \rightarrow L \rightarrow L \rightarrow H \rightarrow L \rightarrow L \rightarrow H \rightarrow L \rightarrow L \rightarrow M$, which means that by the 12^{th} time step, it has already observed the effects of all four parameters. Interestingly, the classifier then (essentially) repeats this sequence and thus observes the effect of all the parameters for another seven times (four repetitions occur in the last 50 time steps, which are not shown in Fig. 1.9(a)). We conjecture that the repetitions make the classifier more robust in determining whether the behavior under observation is that of a model or the agent. These results suggest that the classifier succeeds in controlling the level of ambient light in a way that helps distinguish between the agent and the models, revealing all aspects of the underlying behavior. Note that the classifiers are not provided with any prior information about the agent, including its structure. Thus, all knowledge is gained through an evolutionary process that is driven by the accuracy of their judgments.

1.3.3.4 Noise Study

We now consider the situation where, during the coevolutionary process, noise is injected into the agent's behavior, and the agent's positions as measured by the system. This makes the overall setup more realistic as the locomotion of agents and any tracking system will be affected by noise and measurement error. Since the passive coevolutions fail even in the noiseless case, we consider only the Interactive coevolution for the sake of simplicity. We performed 100 coevolutionary runs with the following settings. The light intensity perceived by the agent at time t is obtained by multiplying the actual intensity by a random number generated uniformly in $(0.95, 1.05)$, and capping the perceived intensity to 1 if it exceeds this value. Noise is also applied to the speed of the agent by multiplying the original speed with a random number generated uniformly in $(0.95, 1.05)$. Noise on the estimated position on the agent is applied by adding a random number generated from a normal distribution: $\mathcal{N}(0, 0.005)$.

Fig. 1.10 shows a box plot with the distributions of the evolved models with the highest subjective fitness in the 1000^{th} generation of the Interactive coevolution with noise. The effect of the noise is to widen the distribution of the evolved parameters across the 100

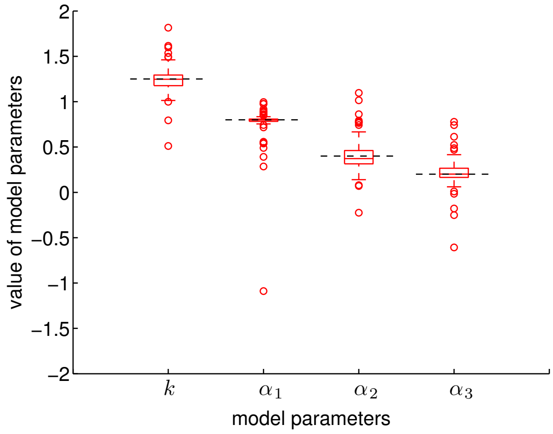


Figure 1.10: This plot shows the distributions of the evolved models with the highest subjective fitness in the 1000th generation of the Interactive coevolution with noise (for a comparison to the case without noise, see Fig. 1.3). The dotted lines correspond to the values of the four parameters that the system is expected to learn (i.e. those of the agent). From top to bottom, these are 1.25, 0.8, 0.4, 0.2, respectively.

Table 1.2: This table shows a comparison of all approaches. The numbers show the relative errors of the evolved parameters (median values over 100 runs) with respect to the parameters of the agent (in absolute percentage).

	k	α_1	α_2	α_3
Interactive (metric-free)	0.024	0	0.025	0.15
Passive 1 (metric-free)	1.2	14.3	7.8×10^4	2.3×10^5
Passive 2 (metric-free)	0.7	5.74	1.3×10^4	3.3×10^5
Passive 1 (SPEA)	0	0	1.2	48.7
Passive 2 (SPEA)	0	0	9.8	48.5
CoEATM	1.1×10^{-6}	7.5×10^{-6}	0.3	0.08

coevolutionary runs; however, the median values of the evolved parameters are still very close to the true values. Interestingly, the Interactive coevolution does not seem to learn α_2 and α_3 significantly worse than it does learn k and α_1.

1.3.3.5 Using a Single-Population Evolutionary Algorithm

In order to compare the metric-free coevolutionary method against a more traditional approach, we used a simple evolution where a single population of models evolves. We

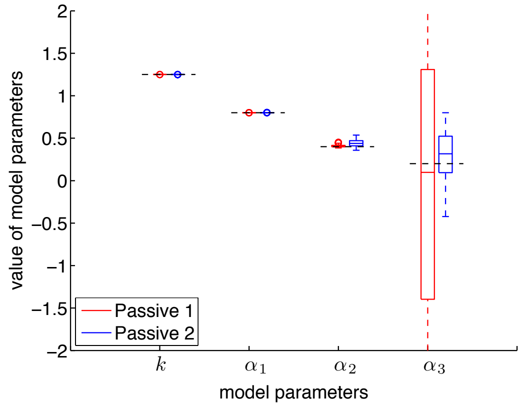


Figure 1.11: This plots shows the distributions of the evolved models with the highest fitness in the 100000$^{\text{th}}$ generation in the simple evolutions with a single population. Each box corresponds to 100 evolutionary runs. The dotted lines correspond to the values of the four parameters that the system is expected to learn (i.e. those of the agent). From top to bottom, these are 1.25, 0.8, 0.4, 0.2, respectively. Note that in order to zoom in on the relevant range, some boxes and outliers are omitted from the plot.

call this method SPEA. As there are now no classifiers, an interactive approach is not possible, and thus we conducted 100 evolutionary runs for the Passive 1 and Passive 2 methods of changing the light intensity in the agent's environment. The structure of the evolution is identical to the sub-algorithms used in the coevolution, except for the fitness evaluation step. Now, in each generation, 100 experiments are performed on the agent using 100 randomly generated intensity patterns. The 100 intensity patterns are used to evaluate a model 100 times. The average square error between the model's and the agent's speed sequences is used as the model's fitness. Each evolutionary run lasts 100,000 generations. In other words, the number and duration of experiments on the agent is kept the same as that in the coevolutionary approach, as outlined in Section 1.3.2.

Fig. 1.11 reveals that the evolution is able to identify parameters k, α_1, α_2, but not α_3. Note that, apart from the single-population evolutions not being able to consistently identify the parameter α_3, they also rely on a pre-defined metric for their operation; in this case, computed as the square error between the model's and the agent's speed sequences.

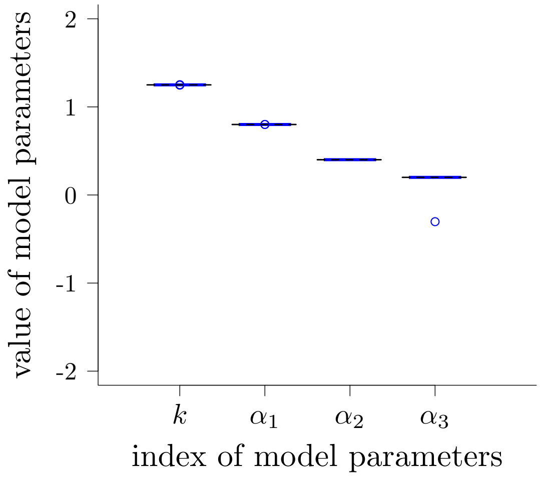


Figure 1.12: This plots shows the distributions of the evolved models with the highest fitness in the 100000$^{\text{th}}$ generation in the simple evolutions with a single population. Each box corresponds to 100 evolutionary runs. The dotted lines correspond to the values of the four parameters that the system is expected to learn (i.e. those of the agent). From top to bottom, these are 1.25, 0.8, 0.4, 0.2, respectively.

1.3.3.6 Coevolution of Tests and Models

Although we have shown that the metric-free method can learn deterministic agent behaviors, it is still possible to obtain the same results through coevolution of inputs and models, which is similar to the *estimation-exploration algorithm* [14]. Different from [14], we use the classifiers to generate the inputs (tests), and the inputs are used for controlling the stimulus (in this case, light intensity). The models are optimized through minimizing the square error of speed between the agent and models, given the same sequence of inputs generated by the classifiers. The classifiers compete with the models through generating the inputs that maximize the model errors. Note that in this case the classifiers are only used for generating the inputs. We call this method CoEATM. Fig. 1.12 shows the results of CoEATM. The models are learned with a high accuracy. In other words, there is no 'true' interaction between the agents and classifiers during the experiments. The classifiers only need to learn how to output a fixed sequence to extract all the information from the agent. For a comparison of all approaches, see Table 1.2.

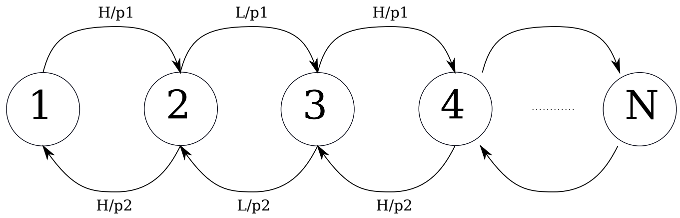


Figure 1.13: The finite state machine which represents the stochastic behavior of the agent under investigation. The starting state is 1. p_1, p_2 are probabilities. H and L represents the levels of the ambient light intensity that the agent responses to. For all the other cases (which are not shown), the agent keeps staying in the same state.

In order to further validate and highlight the benefit of the metric-free method, we introduce the stochastic behaviors in the following section.

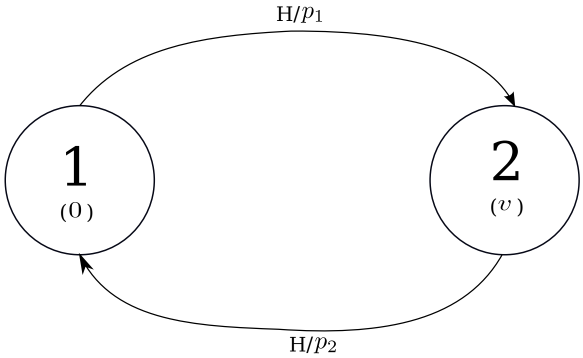
1.4 Case Study Two

1.4.1 Stochastic Behaviors

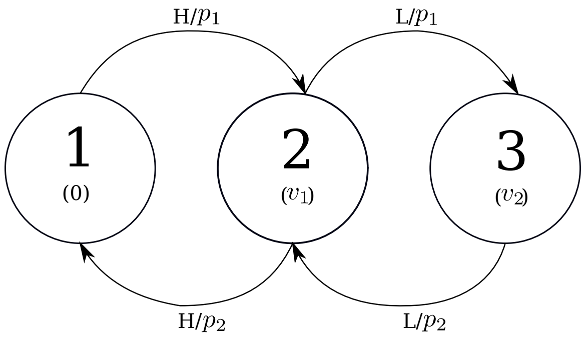
Stochastic behaviors are widely observed in the animal kingdom. Given the same stimuli, the animal may behave differently. In order to make the animal under investigation reveal all its behavioral repertoire, ethologists sometimes need to interact with the animals in real time and make action according to its stochastic response. Based on this motivation, in this section we apply the metric-free method to learn stochastic behaviors to see whether a machine could also be as 'intelligent' as humans to some extent. Our hypotheses are: 1) through learning by interaction, the machine learns the agent behavior better or faster; 2) the metric-free method is superior to metric-based system identification method in learning stochastic behaviors.

In order to verify these two hypotheses mentioned above, we designed a stochastic behavior shown in Fig. 1.13. This behavior is represented as a finite state machine in a general case. The agent's behavior under investigation is described as follows.

The agent's behavior depends on the level of the light intensity in the environment. When the agent is in initial state, 1, if the light intensity is in low level (L), the agent always stays in state 1; if the light intensity is in high state (H), the agent has the probability



(a) (a) 2 states



(b) (b) 3 states

Figure 1.14: The stochastic behaviors with (a) 2 states and (b) 3 states under invesitgation.

of p_1 to move forward to state 2; otherwise it keep staying in state 1. When the agent is in the middle state, $s \in 2, 3, \cdots, n-1$, if the level of light intensity is switched (from H to L or from L to H) and keeps in that level, the agent has the probability of $p1$ to move forward to another state with a higher number; if the light intensity is not switched and keeps in that level, the agent would move back to a previous state with a probability of $p2$. When the agent is in the final state, n, if the level of light intensity is switched and keeps in that level, the agent would always keep in that state; for the other case, it moves like it is in the middle state, that is, with a probability of $p2$ to move back to the previous state. Fig. 1.14(a) and Fig. 1.14(b) show the agent behavior with 2 and 3 states, respectively. These are the two behaviors to be investigated in the thesis.

In the agent's behavior, p_1 is selected to a low value and p_2 is set to a high value. This means the agent has a low chance of moving forward to a state with a higher number and thus the higher state has lower observability. The classifiers need to learn how to interact with the agent to learn its behavior as fast as possible.

Suppose that the agent move in one-dimensional space and moves in a consistent speed in each state. For the sake of simplicity, we assume that the agent stays static and the initial state is known. The system identification task is to identify the parameters of the agent's other states $(v_1, v_2, v_3, \cdots, v_n)$ and the two probabilities, p_1 and p_2. The speed of the agent in each state is chosen arbitrarily. p_1 and p_2 are chosen to have a value of 0.1 and 1.0. In this case, a good strategy for the classifiers to interact with the agent is capturing the point when the agent starts to jump from a state with a lower number to a state with a higher number and switches the light intensity immediately and then keeps in that level. By doing this, the agent would keep staying in the state with a higher number, and this makes it easier for the system to learn the behavior. If the classifiers fail to do that, the agent would immediately move to its previous state.

For the 2-state finite state machine, v is selected to be 0.5; for the 3-state finite state machine, v_1 and v_2 are selected to be 0.5 and 1.0, respectively. Therefore, the parameters to be identified for the 2-state and 3-state agent behavior are:

$$\mathbf{q}_1 = (v, p_1, p_2) = (0.5, 0.1, 1.0). \tag{1.6}$$

$$\mathbf{q}_2 = (v_1, v_2, p_1, p_2) = (0.5, 1.0, 0.1, 1.0). \tag{1.7}$$

1.4.2 Simulation Setup

The simulation setup used for learning the stochastic behaviors in Fig. 1.14 is the same with that in Section 1.3.2. We the metric-free methods, we still used three setups: "Interactive", "Passive 1" and "Passive 2", as discussed in Section 1.3.2. We also compared the metric-free method with the two metric-based methods (SPEA and CoEATM) discussed in Section 1.3.3.5 and Section 1.3.3.6, respectively. Note that for each setup, we added a certain amount of noise into the measure of speed. This is realized by multiplying the real speed with a random value in $[0.95, 1.05]$ in each time step.

Table 1.3: This table shows a comparison of all approaches for learning the 2-state stochastic behavior shown in Fig. 1.14(a). The numbers show means of the AEs in the parameters with respect to the parameters of the agent.

	v_1	p_1	p_2
Interactive (metric-free)	0.003	0.03	1.6×10^{-5}
Passive 1 (metric-free)	0.01	0.03	1.0×10^{-5}
Passive 2 (metric-free)	0.02	0.02	1.0×10^{-5}
Passive 1 (SPEA)	0.47	0.9	1.0
Passive 2 (SPEA)	0.47	0.9	1.0
CoEATM	0.47	0.9	1.0

1.4.3 Results: Two States

1.4.3.1 Analysis of Evolved Models

Fig. 1.15 shows a box plot with the parameters of the evolved models with the highest subjective fitness in the 1000$^{\text{th}}$ generation for (a) metric-free method, (b) SPEA, and (c) CoEATM. Using the metric-free method, the system identified the parameters of the agent with good accuracy. For the other two metric-based methods, all the three parameters are not learned well. Instead, the three evolved parameters converge into three different values: $v_1 \to 0.0$, $p_1 \to 1.0$, $p_2 \to 0.0$. The means (standard deviations) of the AEs in the parameters for each method were shown in Table 1.3. Clearly, the metric-free method learns the stochastic behavior significantly better than the metric-based methods. There is no significant difference among "Interactive", "passive 1" and "passive 2" setups of the metric-free method in terms of AEs of the evolved parameters.

In order to show the advantage of "Interactive" setup of the metric-free method, we investigate the convergence of model parameters during the evolutionary process. This reflects how faster each method learns the agent behavior. Fig. 1.16 shows the convergence of the model parameters over generations for the three setups of the metric-free method. As we can see, the evolved models in the "Interactive" setup converge much faster than those in the two passive setups. The convergence time of the passive setups is almost twice as much as that in the "Interactive" setup. For the "Interactive" setup, after about 100 generations, all the three parameters converge into their true values. In terms of v, there is much smaller disturbance in the "Interactive" setup than that in the other two setups.

1.4.3.2 Analysis of Evolved Classifiers

As the metric-based methods failed to learn the stochastic behavior, in this section we only analyzed the classifiers evolved in the metric-free method. In order to investigate why "Interactive" setup learns the agent behavior much faster than the two passive setups. We post-evaluated how the classifiers interact with the agent during the experimental process.

Fig. 1.17 shows how the classifiers from different generations (in a particular coevolution run) interact with the agent in a trial. As shown in the top left of Fig. 1.17, the classifier outputs H level and then waits until the agent moves from state 1 to state 2, and it immediately switches the light intensity to L level in order to make the agent stay in state 2 as long as possible. In this case, the agent's behavior in state 2 can be fully revealed. That means the classifier has learned the good strategy to interact with the agent at the very beginning of the coevolution run (before 50 generations). After 500 generations, the classifier changed the strategy a little bit. Instead of always outputting H level, it keeps switching between H and L level until it observes the agent moves from state 1 to state 2, and after that it immediately keeps outputting L level. This strategy is learned by the classifiers during the evolutionary process. When the light intensity is changed randomly, it is very unlikely to generate such sequence. Also, there is no optimal/best fixed sequence of inputs to extract the full information of the stochastic agent behavior, as given the same input, the agent would probably exhibit different behavior. Using metrics (such as square error) to quantitatively measure the difference between the models and agents are not appropriate to learn such stochastic behaviors.

1.4.4 Results: Three States

1.4.4.1 Analysis of Evolved Models

This section discusses the results of learning the 3-state stochastic agent behavior shown in Fig. 1.14(b). Fig. 1.18 shows the distribution of the evolved models in the 1000^{th} generation for all setups. The "Interactive" setup of the metric-free method is the only one that learn all the parameters of the agent with good accuracy. This highlights the benefit of interaction. For the other setups using pre-defined metrics, all the parameters are not learned well. Table 1.4 compares the model accuracy of all approaches.

Table 1.4: This table shows a comparison of all approaches for learning 3-state stochastic behavior shown in Fig. 1.14(b). The numbers show means of the AEs in the parameters with respect to the parameters of the agent.

	v	v_2	p_1	p_2
Interactive (metric-free)	0.005	0.03	0.02	2.0×10^{-6}
Passive 1 (metric-free)	0.02	0.23	0.05	0.03
Passive 2 (metric-free)	0.02	0.47	0.08	0.13
Passive 1 (SPEA)	0.47	0.97	0.9	1.0
Passive 2 (SPEA)	0.47	0.98	0.9	1.0
CoEATM	0.46	0.96	0.67	0.89

Fig. 1.19 shows the evolutionary process of the models for the metric-free method. In both passive setups, v_2 disturbed dramatically, while in the “Interactive” setup this parameter converged to its true value smoothly within about 200 generations.

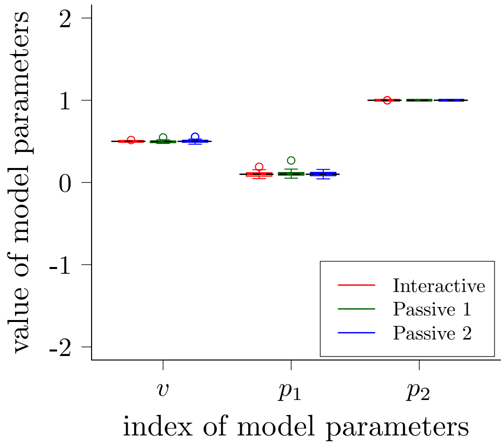
1.4.4.2 Analysis of Evolved Classifiers

Fig. 1.20 shows an example of how the classifiers evolved to interact with the agent in order to extract all of its information. The strategy learned by the classifiers shown in 3 of the 4 sub-figures (corresponding to 100$^{\text{th}}$, 200$^{\text{th}}$, 1000$^{\text{th}}$) is as follows. The classifier outputs H first, and once the agent moves forward from state 1 to state 2, the classifier switches the light intensity into level L and keeps that level. As long as the agent moves forward from state 2 to state 3, the classifier switches the light intensity from L to H and keeps in that level. Note that the classifier lost its ability to interact with the agent in the 500$^{\text{th}}$ generation shown in bottom left of Fig. 1.20. However, this does not influence the learning process.

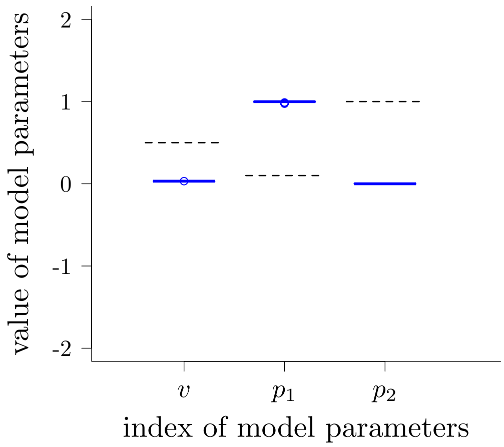
1.5 Summary

This chapter presented the results of learning deterministic and stochastic agent behaviors using the metric-free method with interaction. In both behaviors, the results show that learning through interaction is better or faster than only through passive observation. When learning deterministic behaviors, through coevolution of inputs (tests) and models, the system can still perform well using pre-defined models. However, the metric-free method perform significantly better than the metric-based methods in learning stochastic behaviors.

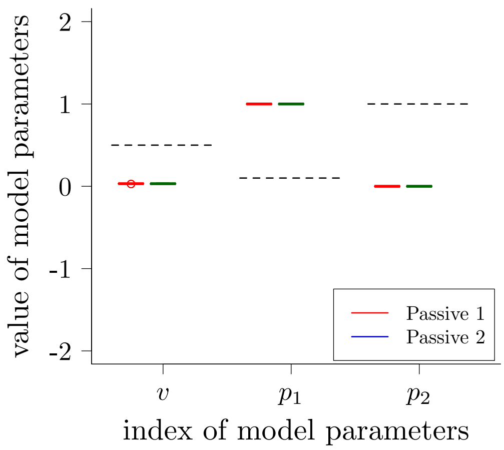
In the interactive metric-free approach, the classifiers were shown to learn good strategies to interact with the agents. In the deterministic behavior, the post-evaluation over 100 runs through performing a grid search shows that the overall best classifier can distinguish between the agent and potential models very well, even if the latter differs only slightly in one of its parameters from the agent. The metric-free method can still perform well when the system is subjective to noise, which shows the robustness of our method.



(a) (a) Metric-free

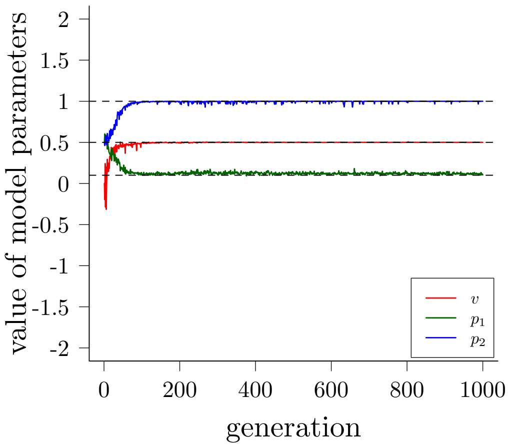


(b) (b) SPEA

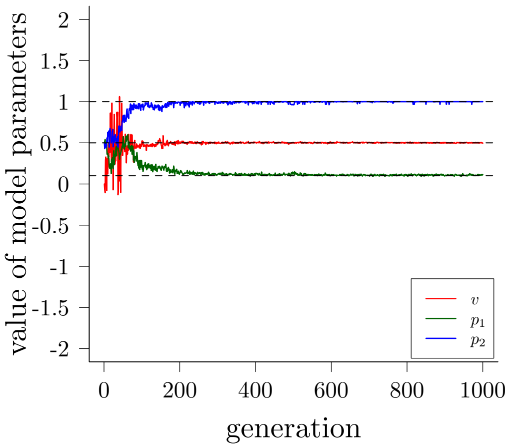


(c) (c) CoEATM

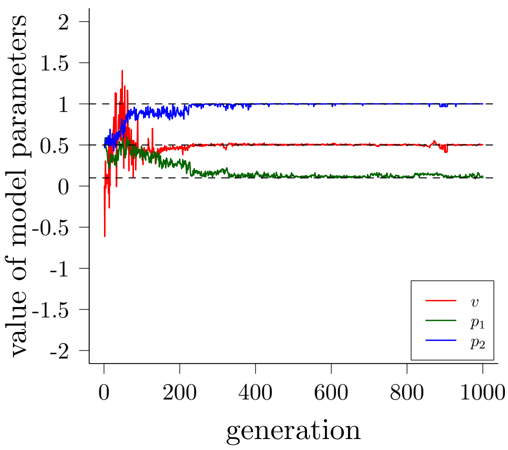
Figure 1.15: This plot shows the distributions of the evolved models with the highest subjective fitness in the 1000$^{\text{th}}$ generation in the coevolutions. Each box corresponds to 30 coevolution runs. The dotted lines correspond to the values of the three parameters that the system is expected to learn (i.e. those of the agent with 2 states). See texts for details.



(a) (a) Interactive



(b) (b) Passive 1



(c) (c) Passive 2

Figure 1.16: Evolutionary process of the evolved model parameters for (a) "Interactive", (b) "Passive 1" and (c) "Passive 2" setups of the metric-free method when learning the 2-state agent behavior. Curves represent mean values across 30 coevolution runs. Dotted black lines indicate true values.

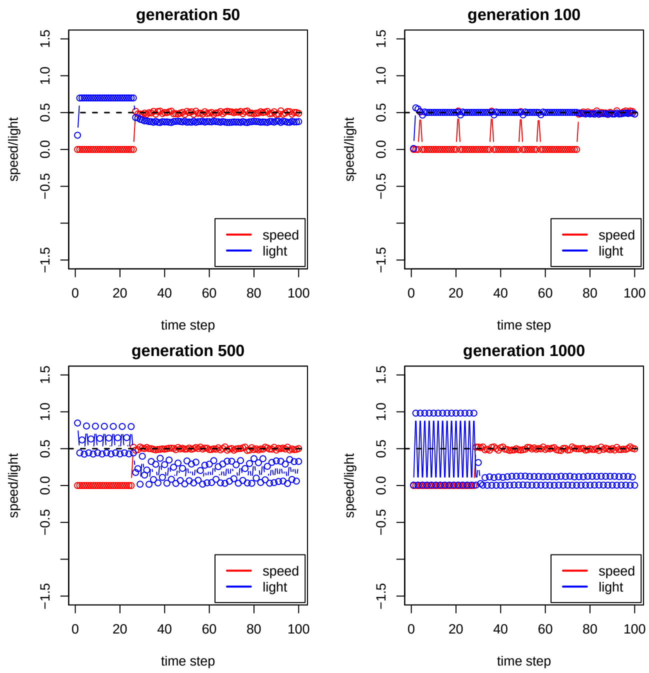
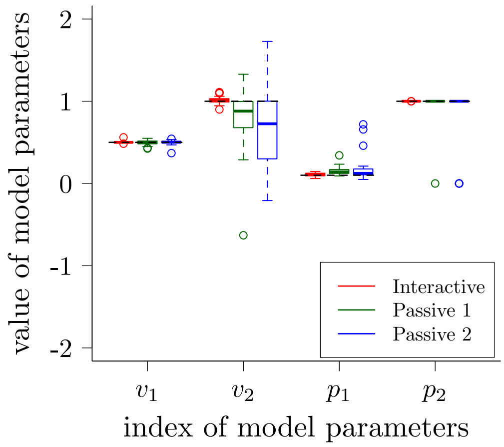
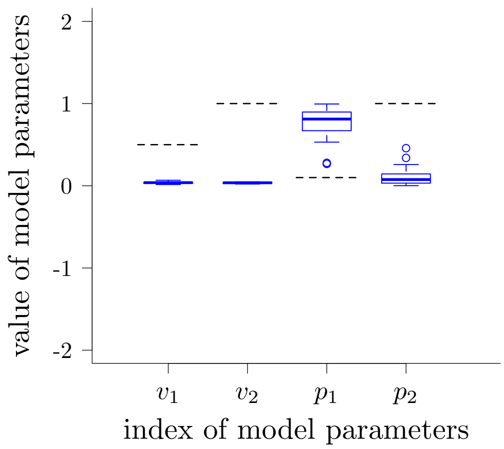


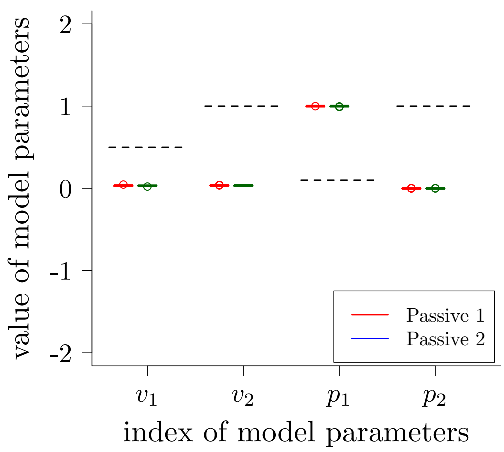
Figure 1.17: This plot shows an example of how the classifiers evolved (in a particular coevolution run) to change the light intensity to interact with the agent during a trial.



(a) (a) Metric-free

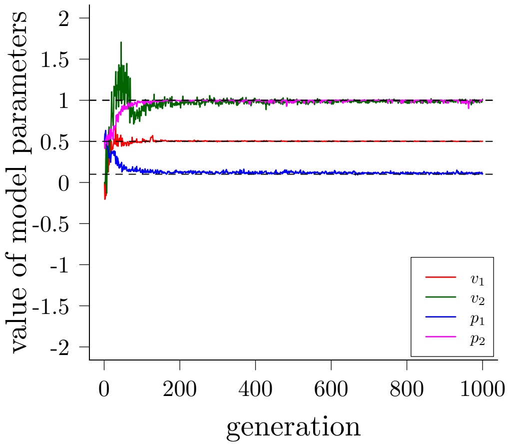


(b) (b) SPEA

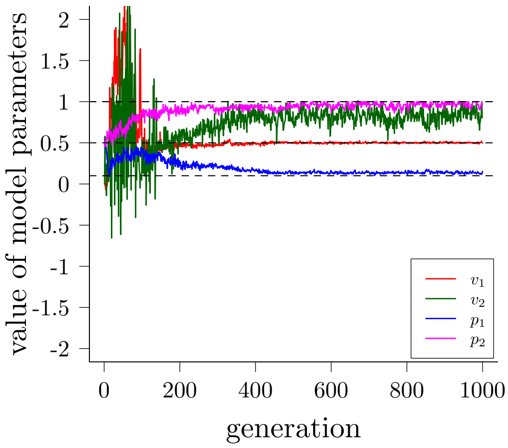


(c) (c) CoEATM

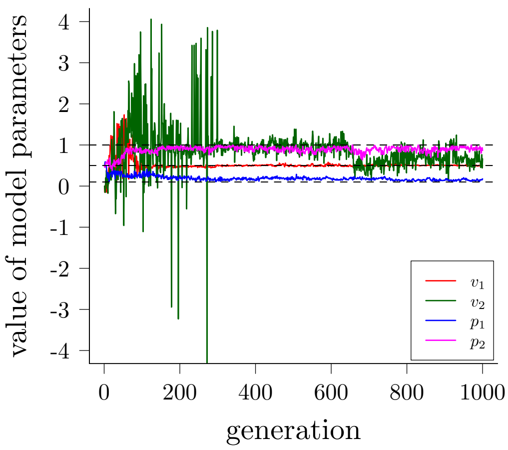
Figure 1.18: This plot shows the distributions of the evolved models with the highest subjective fitness in the 1000$^{\text{th}}$ generation in the coevolutions. Each box corresponds to 30 coevolution runs. The dotted lines correspond to the values of the three parameters that the system is expected to learn (i.e. those of the agent with 3 states). See texts for details.



(a) (a) Interactive



(b) (b) Passive 1



(c) (c) Passive 2

Figure 1.19: Evolutionary process of the evolved model parameters for (a) “Interactive”, (b) “Passive 1” and (c) “Passive 2” setups of the metric-free method when learning the 3-state agent behavior. Curves represent mean values across 30 coevolution runs. Dotted black lines indicate true values.

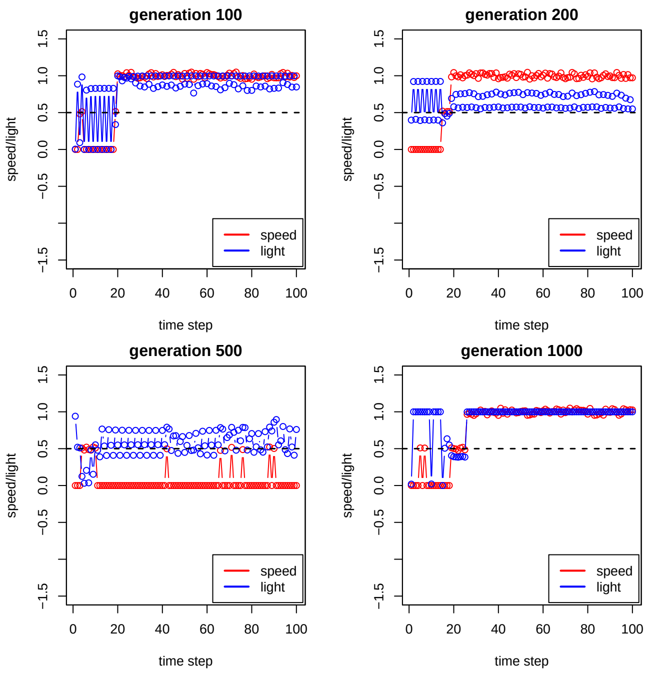


Figure 1.20: This plot shows an example of how the classifiers evolved (in a particular coevolution run) to change the light intensity to interact with the agent during a trial.

Bibliography

[1] J. P. Grotzinger, "Habitability, taphonomy, and the search for organic carbon on mars," *Science*, vol. 343, no. 6169, pp. 386–387, 2014. [Online]. Available: http://www.sciencemag.org/content/343/6169/386.short

[2] R. P. Hertzberg and A. J. Pope, "High-throughput screening: new technology for the 21st century," *Current Opinion in Chemical Biology*, vol. 4, no. 4, pp. 445 – 451, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1367593100001101

[3] J. Bolhuis and L. Giraldeau, *The behavior of animals: mechanisms, function, and evolution.* USA: Wiley-Blackwell, 2004.

[4] W. J. Sutherland, "The importance of behavioural studies in conservation biology," *Animal Behaviour*, vol. 56, no. 4, pp. 801–809, 1998.

[5] D. Floreano and C. Mattiussi, *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies.* Cambridge, MA: MIT Press, 2008.

[6] J.-A. Meyer and A. Guillot, "Biologically inspired robots," in *Springer Handbook of Robot.*, ser. Springer Handbooks, B. Siciliano and O. Khatib, Eds. Berlin, Heidelberg, Germany: Springer, 2008, pp. 1395–1422.

[7] R. King, J. Rowland, S. G. Oliver, and M. Young, "The automation of science," *Science*, vol. 324, no. 5923, pp. 85–89, 2009. [Online]. Available: http://www.sciencemag.org/content/324/5923/85.abstract

[8] J. Evans and A. Rzhetsky, "Machine science," *Science*, vol. 329, no. 5990, pp. 399–400, 2010. [Online]. Available: http://www.sciencemag.org/content/329/5990/399.short

[9] D. Waltz and B. G. Buchanan, "Automating science," *Sci.*, vol. 324, no. 5923, pp. 43–44, 2009.

[10] L. Ljung, "Perspectives on system identification," *Annu. Reviews in Control*, vol. 34, no. 1, pp. 1–12, 2010.

[11] S. A. Billings, *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains.* Hoboken, NJ, USA: Wiley, 2013.

[12] S. M. Henson and J. L. Hayward, "The mathematics of animal behavior: An interdisciplinary dialogue," *Notices of the AMS*, vol. 57, no. 10, pp. 1248–1258, 2010.

[13] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *PNAS*, vol. 99, no. 10, pp. 7280–7287, 2002.

[14] J. Bongard and H. Lipson, "Nonlinear system identification using coevolution of models and tests," *IEEE Trans. Evol. Computation*, vol. 9, no. 4, pp. 361–384, 2005.

[15] ——, "Automated reverse engineering of nonlinear dynamical systems," *PNAS*, vol. 104, no. 24, pp. 9943–9948, 2007.

[16] G. D. Ruxton and G. Beauchamp, "The application of genetic algorithms in behavioural ecology, illustrated with a model of anti-predator vigilance," *Journal of Theoretical Biology*, vol. 250, no. 3, pp. 435–448, 2008.

[17] S. Camazine, J.-L. Deneubourg, N. R. Franks, *et al.*, *Self-Organization in Biological Systems.* Princeton, NJ: Princeton University Press, 2001.

[18] D. Helbing and A. Johansson, "Pedestrian, crowd and evacuation dynamics," in *Extreme Environmental Events*, R. A. Meyers, Ed. Springer, 2011, pp. 697–716.

[19] J. Harvey, K. Merrick, and H. A. Abbass, "Application of chaos measures to a simplified boids flocking model," *Swarm Intell.*, vol. 9, no. 1, pp. 23–41, 2015.

[20] W. S, B. S, F. R, *et al.*, "Modeling collective animal behavior with a cognitive perspective: a methodological framework," *PLoS ONE*, vol. 7, no. 6, 2012, e38588.

[21] M. Gauci, J. Chen, W. Li, T. J. Dodd, and R. Groß, "Self-organized aggregation without computation," *The Int. J. of Robot. Research*, vol. 33, no. 8, pp. 1145–1161, 2014.

[22] ——, "Clustering objects with robots that do not compute," in *Proc. 2014 Int. Conf. Autonomous Agents and Multi-Agent Syst.*, IFAAMAS Press, Paris, France, 2014, pp. 421–428.

[23] H. Schildt, *Artificial intelligence using C.* New York, NY, USA: McGraw-Hill, 1987.

[24] E. Charniak, *Introduction to artificial intelligence.* Reading, MA, USA: Addison-Wesley, 1985.

[25] D. B. Fogel, *Evolutionary computation: toward a new philosophy of machine intelligence.* Street Hoboken, NJ, USA: Wiley-IEEE Press, 1995.

[26] M. L. Minsky, "Logical versus analogical or symbolic versus connectionist or neat versus scruffy," *AI magazine*, vol. 12, no. 2, pp. 34–51, 1991.

[27] A. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, no. 236, pp. 433–460, 1950.

[28] P. Jackson, *Introduction to expert system.* Boston, MA, USA: Addison-Wesley, 1998.

[29] M. Newborn, *Kasparov versus Deep Blue: Computer Chess Comes of Age.* New York, NY, USA: Springer-Verlag, 1997.

[30] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[31] N. J. Nilsson, "Shakey the robot," SRI International Technical Note, Tech. Rep., 1984.

[32] R. Brooks, "A robust layered control system for a mobile robot," *Robotics and Automation, IEEE Journal of*, vol. 2, no. 1, pp. 14–23, Mar 1986.

[33] M. Sasaki, T. Kageoka, K. Ogura, H. Kataoka, T. Ueta, and S. Sugihara, "Total laboratory automation in japan: Past, present and the future," *Clinica Chimica Acta*, vol. 278, no. 2, pp. 217 – 227, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S000989819800148X

[34] A. Persidis, "High-throughput screening," *Nature biotechnology*, vol. 16, no. 5, pp. 488–493, 1998.

[35] K. E. Whelan and R. D. King, "Intelligent software for laboratory automation," *Trends in Biotechnology*, vol. 22, no. 9, pp. 440–445, 2004.

[36] N. Gauld and Gaston., "Driving miss daisy: The performance of an automated insect idenfitication system," *Hymenoptera: evolution, biodiversity and biological-control*, pp. 303–311, 2000.

[37] N. MacLeod, M. Benfield, and P. Culverhouse, "Time to automate identification," *Nature*, vol. 467, no. 7312, pp. 154–55, 2010. [Online]. Available: http://www.nature.com/nature/journal/v467/n7312/full/467154a.html?type=access_denied

[38] C. Darwin, *On the Origin of Species.* England: Dover Publications, 1859.

[39] J. H. Holland, *Adaptation in Natural and Artificial Systems.* Boston, Massachusetts: MIT Press, 1992.

[40] M. J. W. Lawrence J. Fogel, Alvin J. Owens, *Artificial Intelligence through Simulated Evolution.* Chichester, UK: Wiley, 1966.

[41] I. Rechenberg, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Stuttgart: Fromman-Hozlboog Verlag, 1994.

[42] J. Koza, *Genetic Programming.* Cambridge MA: MIT Press, 1992.

[43] C. Rosin and R. Belew, "New methods for competitive coevolution," *Evolutionary Computation*, vol. 5, no. 10, pp. 1–29, 1997.

[44] R. Dawkins and J. R. Krebs, "Arms races between and within species," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 205, no. 1161, pp. 489–511, 1979. [Online]. Available: http://rspb.royalsocietypublishing.org/content/205/1161/489.abstract

[45] J. Cartlidge and S. Bullock, "Combating coevolutionary disengagement by reducing parasite virulence," *Evolutionary Computation*, vol. 12, no. 2, pp. 193–222, 2004.

[46] P. J. Angeline and J. B. Pollack, "Competitive environments evolve better solutions for complex tasks," *Bibliometrics*, vol. 155, no. 18, pp. 1–5, 1993.

[47] L. Panait and S. Luke, "A comparative study of two competitive fitness functions," in *Proceedings of the Genetic and Evolutionary Computation Conference*. Boston, Massachusetts: MIT Press, 2002, pp. 567–573.

[48] T. Tan and J. Teo, "Competitive coevolution with k-random opponents for pareto multiobjective optimization," in *Natural Computation, Third International Conference on*, 2007, pp. 63 – 67. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.38.3029

[49] O. E. David, H. J. van den Herik, M. Koppel, and N. S. Netanyahu, "Genetic algorithms for evolving computer chess programs," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 5, pp. 779–789, 2014.

[50] G. Gutin, A. Yeo, and A. Zverovich, "Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the {TSP}," *Discrete Applied Mathematics*, vol. 117, no. 13, pp. 81 – 86, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166218X01001950

[51] C. Wang, S. Yu, W. Chen, and C. Sun, "Highly efficient light-trapping structure design inspired by natural evolution," *Sci. Rep.*, vol. 3, no. 1, pp. 1–7, 2013.

[52] H. GS, L. JD, and L. DS, "Computer-automated evolution of an x-band antenna for nasa's space technology 5 mission," *Evolutionary Computation*, vol. 19, no. 1, pp. 1–23, 2011.

[53] R. Bellman, *Dynamic Progamming and Lagrange Multipliers.* Princeton, NJ, USA: Princeton University Press, 1957.

[54] J. J. E. Dennis and J. J. Mor, "Quasi-newton methods, motivation and theory," *SIAM Review*, vol. 19, no. 1, pp. 46–89, 1977.

[55] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Pittsburgh, PA, USA, Tech. Rep., 1994.

[56] N. Hansen, S. Muller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es)," *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, March 2003.

[57] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, vol. 3, no. 1, pp. 1–16, 1995.

[58] K. J. R. and J. P. Rice, "Automatic programming of robots using genetic programming," in *AAAI*. MIT Press, 1992, pp. 1–6.

[59] R. A. Brooks, "Artificial life and real robots," in *Proceedings of the First European Conference on Artificial Life*. MIT Press, 1992, pp. 3–10.

[60] D. Floreano and S. Nolfi, "Adaptive behavior in competing co-evolving species," in *The 4th European Conference on Artificial Life*. MIT Press, 1997, pp. 378–387.

[61] D. Floreano, P. Drr, and C. Mattiussi, "Neuroevolution: from architectures to learning," *Evolutionary Intelligence*, vol. 1, no. 1, pp. 47–62, 2008. [Online]. Available: http://dx.doi.org/10.1007/s12065-007-0002-4

[62] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002. [Online]. Available: http://nn.cs.utexas.edu/?stanley:ec02

[63] B. D.M. and O. C., "Understanding evolutionary potential in virtual cpu instruction set architectures," *PLoS ONE*, vol. 8, no. 12, p. e83242, 2013. [Online]. Available: http://nn.cs.utexas.edu/?stanley:ec02

[64] B. Batut, D. P. Parsons, S. Fischer, G. Beslon, and C. Knibbe, "In silico experimental evolution: a tool to test evolutionary scenarios," in *Proceedings of the Eleventh Annual Research in Computational Molecular Biology (RECOMB) Satellite Workshop on Comparative Genomics*. BioMed Central Ltd, 2013, pp. 1–6.

[65] J.-M. Montanier and N. Bredeche, "Surviving the Tragedy of Commons: Emergence of Altruism in a Population of Evolving Autonomous Agents," in *European Conference on Artificial Life*, Paris, France, Aug. 2011. [Online]. Available: https://hal.inria.fr/inria-00601776

[66] W. M, F. D, and K. L, "A quantitative test of hamilton's rule for the evolution of altruism," *PLoS Biology*, vol. 9, no. 5, p. e1000615, 2011.

[67] D. Floreano, S. Mitri, S. Magnenat, and L. Keller, "Evolutionary conditions for the emergence of communication in robots," *Current Biology*, vol. 17, no. 6,

pp. 514 – 519, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0960982207009281

[68] A. JE and B. JC, "Environmental influence on the evolution of morphological complexity in machines," *PLoS Computational Biology*, vol. 10, no. 1, p. e1003399, 2014.

[69] D. Cliff and G. F. Miller, "Co-evolution of pursuit and evasion ii: Simulation methods and results," *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, vol. 92, no. 2, pp. 101–106, 1995.

[70] D. Floreano, "Evolutionary robotics in behavior engineering and artificial life," in *Evolutionary Robotics: From Intelligent Robots to Artificial Life. Applied AI Systems, 1998. Evolutionary Robotics Symposium.* AAI Books, 1998.

[71] S. Koos, J.-B. Mouret, and S. Doncieux, "The transferability approach: Crossing the reality gap in evolutionary robotics," *Evolutionary Computation, IEEE Transactions on*, vol. 17, no. 1, pp. 122–145, Feb 2013.

[72] S. Koos, A. Cully, and J. Mouret, "Fast damage recovery in robotics with the t-resilience algorithm," *CoRR*, vol. abs/1302.0386, 2013. [Online]. Available: http://arxiv.org/abs/1302.0386

[73] D. Floreano and F. Mondada, "Evolution of homing navigation in a real mobile robot," *IEEE Trans. Syst., Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 3, pp. 396–407, 1996.

[74] R. Watson, S. Ficiei, and J. Pollack, "Embodied evolution: embodying an evolutionary algorithm in a population of robots," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 1, 1999, pp. –342 Vol. 1.

[75] A. Eiben, E. Haasdijk, and N. Bredeche, "Embodied, On-line, On-board Evolution for Autonomous Robotics," in *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution.*, ser. Series: Cognitive Systems Monographs, S. K. E. P. Levi, Ed. Springer, 2010, vol. 7, pp. 361–382. [Online]. Available: https://hal.inria.fr/inria-00531455

[76] A. Eiben, S. Kernbach, and E. Haasdijk, "Embodied artificial evolution," *Evolutionary Intelligence*, vol. 5, no. 4, pp. 261–272, 2012. [Online]. Available: http://dx.doi.org/10.1007/s12065-012-0071-x

[77] A. E. Eiben and J. Smith, "From evolutionary computation to the evolution of things," *Nature*, vol. 521, no. 7553, pp. 467–482, 2015.

[78] J. R. Tumbleston, D. Shirvanyants, N. Ermoshkin, R. Janusziewicz, A. R. Johnson, D. Kelly, K. Chen, R. Pinschmidt, J. P. Rolland, A. Ermoshkin, E. T. Samulski, and J. M. DeSimone, "Continuous liquid interface production of 3d objects," *Science*, vol. 347, no. 6228, pp. 1349–1352, 2015. [Online]. Available: http://www.sciencemag.org/content/347/6228/1349.abstract

[79] L. Ljung, "System identification: Theory for the user," *Englewood Cliffs, NJ: Prentice-Hall*, 1999.

[80] D. B. Fogel, *System identification through simulated evolution: a machine learning approach to modeling.* Needham, MA, USA: Ginn Press, 1991.

[81] E. J. Vladislavleva, G. F. Smits, and D. Den Hertog, "Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming," *Trans. Evol. Comp*, vol. 13, no. 2, pp. 333–349, Apr. 2009. [Online]. Available: http://dx.doi.org/10.1109/TEVC.2008.926486

[82] J. Bongard and H. Lipson, "Nonlinear system identification using coevolution of models and tests," *IEEE Trans. Evol. Comput.*, vol. 9, no. 4, pp. 361–384, 2005.

[83] ——, "Automated damage diagnosis and recovery for remote robotics," in *Proc. 2004 IEEE Int. Conf. Robot. and Autom.* IEEE Computer Society Press, New Orleans, LA, 2004, pp. 3545–3550.

[84] ——, "Automated robot function recovery after unanticipated failure or environmental change using a minimum of hardware trials," in *Proc. 2004 NASA/DoD Conf. Evolvable Hardware.* IEEE Computer Society Press, Los Alamitos, CA, 2004, pp. 169–176.

[85] S. Koos, J. Mouret, and S. Doncieux, "Automatic system identification based on coevolution of models and tests," in *Proc. 2009 IEEE Congr. Evol. Computation.* IEEE Press, Trondheim, Norway, 2009, pp. 560–567.

[86] M. Mirmomeni and W. Punch, "Co-evolving data driven models and test data sets with the application to forecast chaotic time series," in *Proc. 2011 IEEE Congr. Evol. Comput.* IEEE Press, New Orleans, LA, USA, 2011, pp. 14–20.

[87] D. Le Ly and H. Lipson, "Optimal experiment design for coevolutionary active learning," *IEEE Trans. Evol. Computation*, vol. 18, no. 3, pp. 394–404, 2014.

[88] B. Kouchmeshky, W. Aquino, J. C. Bongard, and H. Lipson, "Co-evolutionary algorithm for structural damage identification using minimal physical testing," *International Journal for Numerical Methods in Engineering*, vol. 69, no. 5, pp. 1085–1107, 2007. [Online]. Available: http://dx.doi.org/10.1002/nme.1803

[89] M. Mirmomeni and W. Punch, "Co-evolving data driven models and test data sets with the application to forecast chaotic time series," in *2011 IEEE Congress on Evolutionary Computation.* Auburn University, New Orleans, LA, 2011, pp. 14 –20.

[90] J. Bongard, V. Zykov, and H. Lipson, "Resilient machines through continuous self-modeling," *Sci.*, vol. 314, no. 5802, pp. 1118–1121, 2006.

[91] S. Koos, J. B. Mouret, and S. Doncieux, "The transferability approach: Crossing the reality gap in evolutionary robotics," *IEEE Trans. Evol. Computation*, vol. 17, no. 1, pp. 122–145, Feb 2013.

[92] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, "Robots that can adapt like animals," *Nature*, vol. 521, no. 7553, pp. 503–507, 2015.

[93] P. J. O'Dowd, M. Studley, and A. F. T. Winfield, "The distributed co-evolution of an on-board simulator and controller for swarm robot behaviours," *Evol. Intell.*, vol. 7, no. 2, pp. 95–106, 2014.

[94] N. Jakobi, P. Husbands, and I. Harvey, "Noise and the reality gap: the use of simulation in evolutionary robotics," in *Advances in Artificial Life: Proc. 3rd European Conf. Artificial Life.* Springer-Verlag, 1995, pp. 704–720.

[95] B. Hedwig and J. F. A. Poulet, "Complex auditory behaviour emerges from simple reactive steering," *Nature*, vol. 430, no. 7001, pp. 781–785, 2004.

[96] E. Baird, M. J. Byrne, J. Smolka, E. J. Warrant, and M. Dacke, "The dung beetle dance: An orientation behaviour?" *PLoS ONE*, vol. 7, no. 1, p. e30211, 01 2012. [Online]. Available: http://dx.doi.org/10.1371%2Fjournal.pone.0030211

[97] M. D. M. Byrne, "Visual cues used by ball-rolling dung beetles for orientation," *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, vol. 189, no. 6, pp. 411–418, 2003.

[98] E. G. Matthews, "Observations on the ball-rolling behavior of canthon pilularius," *Psyche*, pp. 75–93, 1963.

[99] S. Garnier, J. Gautrais, and G. Theraulaz, "The biological principles of swarm intelligence," *Swarm Intelligence*, vol. 1, no. 1, pp. 3–31, 2007. [Online]. Available: http://dx.doi.org/10.1007/s11721-007-0004-y

[100] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," *Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.

[101] R. Jeanson, C. Rivault, J.-L. Deneubourg, S. Blanco, R. Fournier, C. Jost, and G. Theraulaz, "Self-organized aggregation in cockroaches," *Animal Behaviour*, vol. 69, no. 1, pp. 169 – 180, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0003347204002428

[102] C. R. Carroll and D. H. Janzen, "Ecology of foraging by ants," *Annu. Review of Ecology and Systematics*, vol. 4, pp. 231–257, 1973.

[103] J. E. Lloyd, "Bioluminescent communication in insects," *Annual Review of Entomology*, vol. 16, pp. 97–122, 1971.

[104] O. H. Bruinsma, "An analysis of building behaviour of the termite macrotermes subhyalinus (rambur)," Ph.D. dissertation, Wageningen University, Wageningen, The Netherlands, 1979.

[105] M. Dorigo and L. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *Evolutionary Computation, IEEE Transactions on*, vol. 1, no. 1, pp. 53–66, 1997.

[106] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.

[107] O. Holland and C. Melhuish, "Stigmergy, self-organization, and sorting in collective robotics," *Artificial Life*, vol. 5, no. 2, pp. 173–202, 1999.

[108] G. Di Caro and M. Dorigo, "Antnet: Distributed stigmergetic control for communications networks," *J. Artif. Int. Res.*, vol. 9, no. 1, pp. 317–365, Dec. 1998. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622797.1622806

[109] K. Socha, "Aco for continuous and mixed-variable optimization," in *Ant Colony Optimization and Swarm Intelligence*, ser. Lecture Notes in Computer Science, M. Dorigo, M. Birattari, C. Blum, L. Gambardella, F. Mondada, and T. Sttzle, Eds. Springer Berlin Heidelberg, 2004, vol. 3172, pp. 25–36. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-28646-2_3

[110] J. Bjerknes and A. T. Winfield, "On fault tolerance and scalability of swarm robotic systems," in *Distributed Autonomous Robotic Systems*, ser. Springer Tracts in Advanced Robotics. Springer, Berlin, Heidelberg, 2013, vol. 83, pp. 431–444.

[111] J. Chen, M. Gauci, W. Li, A. Kolling, and R. Gros, "Occlusion-based cooperative transport with a swarm of miniature mobile robots," *Robotics, IEEE Transactions on*, vol. 31, no. 2, pp. 307–321, April 2015.

[112] M. Gauci, J. Chen, T. Dodd, and R. Groß, "Evolving aggregation behaviors in multi-robot systems with binary sensors," in *Distributed Autonomous Robotic Systems*, ser. Springer Tracts in Advanced Robotics. Springer, Berlin, Heidelberg, 2014, vol. 104, pp. 355–367.

[113] E. ahin, "Swarm robotics: From sources of inspiration to domains of application," in *Swarm Robotics*, ser. Lecture Notes in Computer Science, E. ahin and W. Spears, Eds. Springer Berlin Heidelberg, 2005, vol. 3342, pp. 10–20. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-30552-1_2

[114] B. Gerkey and M. Mataric, "Sold!: auction methods for multirobot coordination," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 758–768, Oct 2002.

[115] A. F. T. Winfield, "Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots," in *Distributed Autonomous Robotic Systems 4*, L. Parker, G. Bekey, and J. Barhen, Eds. Springer Japan, 2000, pp. 273–282. [Online]. Available: http://dx.doi.org/10.1007/978-4-431-67919-6_26

[116] V. Trianni, R. Gro, T. Labella, E. ahin, and M. Dorigo, "Evolving aggregation behaviors in a swarm of robots," in *Advances in Artificial Life*, ser. Lecture Notes in Computer Science, W. Banzhaf, J. Ziegler, T. Christaller, P. Dittrich, and J. Kim, Eds. Springer Berlin Heidelberg, 2003, vol. 2801, pp. 865–874. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-39432-7_93

[117] S. Garnier, C. Jost, J. Gautrais, M. Asadpour, G. Caprari, R. Jeanson, A. Grimal, and G. Theraulaz, "The embodiment of cockroach aggregation behavior in a group of micro-robots," *Artificial Life*, vol. 14, no. 4, pp. 387–408, Oct. 2008. [Online]. Available: http://dx.doi.org/10.1162/artl.2008.14.4.14400

[118] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Distributed Autonomous Robotic Systems 5*. Springer, 2002, pp. 299–308.

[119] J. McLurkin and J. Smith, "Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots," in *in 7th International Symposium on Distributed Autonomous Robotic Systems (DARS*. Citeseer, 2004.

[120] K. Fujibayashi, S. Murata, K. Sugawara, and M. Yamamura, "Self-organizing formation algorithm for active elements," in *Reliable Distributed Systems, 2002. Proceedings. 21st IEEE Symposium on*, 2002, pp. 416–421.

[121] J. Chen, M. Gauci, M. J. Price, and R. Groß, "Segregation in swarms of e-puck robots based on the brazil nut effect," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, ser. AAMAS '12. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 163–170. [Online]. Available: http://dl.acm.org/citation.cfm?id=2343576.2343599

[122] A. Turgut, H. elikkanat, F. Gke, and E. ahin, "Self-organized flocking in mobile robot swarms," *Swarm Intelligence*, vol. 2, no. 2-4, pp. 97–120, 2008. [Online]. Available: http://dx.doi.org/10.1007/s11721-008-0016-2

[123] E. B. C.R. Kube, "Collective robotics: from social insects to robots," *Adaptive Behavior*, vol. 2, no. 2, pp. 189–218, 1993.

[124] C. Kube and E. Bonabeau, "Cooperative transport by ants and robots," *Robotics and Autonomous Systems*, vol. 30, no. 12, pp. 85 – 101, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889099000664

[125] R. Gross and M. Dorigo, "Towards group transport by swarms of robots," *Int. J. Bio-Inspired Comput.*, vol. 1, no. 1/2, pp. 1–13, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1504/IJBIC.2009.022770

[126] J. Werfel, K. Petersen, and R. Nagpal, "Designing collective behavior in a termite-inspired robot construction team," *Science*, vol. 343, no. 6172, pp. 754–758, 2014.

[127] G. S. Fraenkel and D. L. Gunn, *The Orientation of Animals: Kineses, Taxes, and Compass Reactions.* New York: Dover Publications, 1961.

[128] S. D. Sulkin, "Larval orientation mechanisms: The power of controlled experiments," *Ophelia*, vol. 32, no. 1-2, pp. 49–62, 1990.

[129] I. Rano, "A steering taxis model and the qualitative analysis of its trajectories," *Adaptive Behaviour*, vol. 17, no. 3, pp. 197–211, 2009.

[130] S. Camazine, *Self-organization in biological systems.* Princeton University Press, 2003.

[131] B. Webb, "What does robotics offer animal behaviour?" *Animal Behaviour*, vol. 60, no. 5, pp. 545 – 558, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0003347200915148

[132] ——, "Using robots to model animals: a cricket test," *Robotics and Autonomous Systems*, vol. 16, no. 2134, pp. 117 – 134, 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0921889095000445

[133] A. Popov and V. Shuvalov, "Phonotactic behavior of crickets," *J. of Comparative Physiology*, vol. 119, no. 1, pp. 111–126, 1977.

[134] A. M. Farah and T. Duckett, "Reactive localisation of an odour source by a learning mobile robot," in *In Proceedings of the Second Swedish Workshop on Autonomous Robotics.* SWAR Stockholm, Sweden, 2002, pp. 29–38.

[135] A. Lilienthal and T. Duckett, "Experimental analysis of smelling braitenberg vehicles," in *In Proceedings of the ieee international conference on advanced robotics.* Coimbra, Portugal, 2003, pp. 58–63.

[136] T. Balch, F. Dellaert, A. Feldman, A. Guillory, C. Isbell, Z. Khan, S. Pratt, A. Stein, and H. Wilde, "How multirobot systems research will accelerate our understanding of social animal behavior," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1445 –1463, 2006.

[137] J. Chappell and S. Thorpe, "Ai-inspired biology: Does ai have something to contribute to biology?" *Proceedings of the International Symposium on AI Inspired Biology: A Symposium at the AISB 2010 Convention, Leicester, UK*, 2010.

[138] J. Faria, J. Dyer, R. Clément, *et al.*, "A novel method for investigating the collective behaviour of fish: Introducing 'robofish'," *Behavioral Ecology and Sociobiology*, vol. 64, no. 8, pp. 1211–1218, 2010.

[139] J. Halloy, F. Mondada, S. Kernbach, *et al.*, "Towards bio-hybrid systems made of social animals and robots," in *Biomimetic and Biohybrid Systems*, ser. Lecture Notes in Comput. Sci. Springer, Berlin, Heidelberg, Germany, 2013, vol. 8064, pp. 384–386.

[140] J. Halloy, G. Sempo1, G. Caprari, *et al.*, "Social integration of robots into groups of cockroaches to control self-organized choices," *Sci.*, vol. 318, no. 5853, pp. 1155–1158, 2007.

[141] T. Schmickl, S. Bogdan, L. Correia, *et al.*, "Assisi: Mixing animals with robots in a hybrid society," in *Biomimetic and Biohybrid Systems*, ser. Lecture Notes in Comput. Sci. Springer, Berlin, Heidelberg, Germany, 2013, vol. 8064, pp. 441–443.

[142] R. Vaughan, N. Sumpter, J. Henderson, *et al.*, "Experiments in automatic flock control," *Robot. and Autonomous Syst.*, vol. 31, no. 1, pp. 109–117, 2000.

[143] J. Krause, A. F. Winfield, and J.-L. Deneubourg, "Interactive robots in experimental biology," *Trends in Ecology and Evolution*, vol. 26, no. 7, pp. 369 –375, 2011.

[144] S. G. Halloy J., "Social integration of robots into groups of cockroaches to control self-organized choices," *Science*, vol. 318, no. 5853, pp. 1155–1158, 2007. [Online]. Available: http://www.sciencemag.org/cgi/content/abstract/sci;318/5853/1155

[145] J. Krause, A. F. Winfield, and J.-L. Deneubourg, "Interactive robots in experimental biology," *Trends in Ecology and Evolution*, vol. 26, no. 7, pp. 369 – 375, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169534711000851

[146] R. Vaughan, N. Sumpter, A. Frost, and S. Cameron, "Robot sheepdog project achieves automatic flock control," *The fourth international conference*

on Autonomous agents, pp. 489–493, 1998. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.38.3029

[147] A. Gribovskiy, J. Halloy, J.-L. Deneubourg, H. Bleuler, and F. Mondada, "Towards mixed societies of chickens and robots," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. Boston, Massachusetts: MIT press, 2010, pp. 4722 –4728.

[148] V. Kopman, J. Laut, G. Polverino, *et al.*, "Closed-loop control of zebrafish response using a bioinspired robotic-fish in a preference test," *J. of The Roy. Soc. Interface*, vol. 10, no. 78, pp. 1–8, 2013.

[149] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, no. 236, pp. 433–460, 1950.

[150] S. Harnad, "Minds, machines and turing: The indistinguishability of indistinguishables," *J. Logic, Language and Inform.*, vol. 9, no. 4, pp. 425–445, 2000.

[151] J. L. Elman, "Finding structure in time," *Cognitive Sci.*, vol. 14, no. 2, pp. 179–211, 1990.

[152] H.-G. Beyer, *The Theory of Evolution Strategies.* Berlin, Heidelberg, Germany: Springer, 2001.

[153] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies - a comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.

[154] A. Eiben and J. E. Smith, *Introduction to evolutionary computing.* Berlin, Heidelberg: Springer-Verlag, 2003.

[155] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. on Evol. Comput.*, vol. 3, no. 2, pp. 82–102, 1999.

[156] F. Mondada, M. Bonani, X. Raemy, *et al.*, "The e-puck, a robot designed for education in engineering," in *Proc. 9th Conf. on Autonomous Robot Systems and Competitions*, vol. 1. IPCB: Instituto Politécnico de Castelo Branco, 2009, pp. 59–65.

[157] W. Li, M. Gauci, and R. Groß, "Coevolutionary learning of swarm behaviors without metrics," in *Proceedings of the 2014 Genetic and Evolutionary Computation Conference.* ACM Press, Vancuver, Canada, 2014, pp. 201–208.

[158] S. Magnenat, M. Waibel, and A. Beyeler, "Enki: The fast 2D robot simulator," http://home.gna.org/enki/, 2011.

[159] R. L. Graham and N. J. A. Sloane, "Penny-packing and two-dimensional codes," *Discrete and Computational Geometry*, vol. 5, no. 1, pp. 1–11, Jan. 1990.

[160] P. Levi and S. Kernbach, *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution.* Berlin, Heidelberg: Springer-Verlag, 2010.

[161] B. Eldridge and A. Maciejewski, "Limited bandwidth recognition of collective behaviors in bio-inspired swarms," in *Proc. 2014 Int. Conf. Autonomous Agents and Multi-Agent Syst.* IFAAMAS Press, Paris, France, 5 2014, pp. 405–412.

[162] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library.* Sebastopol, CA: O'Reilly Media, 2008.

[163] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[164] W. Li, M. Gauci, J. Chen, and R. Groß, "Online supplementary material," http://naturalrobotics.group.shef.ac.uk/supp/2014-006/, 2014.

[165] R. D. King, J. Rowland, *et al.*, "The automation of science," *Sci.*, vol. 324, no. 5923, pp. 85–89, 2009.

[166] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *Sci.*, vol. 324, no. 5923, pp. 81–85, 2009.

[167] E. Martin, *Macmillan Dictionary of Life Sciences (2nd ed.).* London: Macmillan Press, 1983.

[168] M. Dacke, M. J. Byrne, C. H. Scholtz, and E. J. Warrant, "Lunar orientation in a beetle," *Proc. of the Roy. Soc. of London. Series B: Biological Sci.*, vol. 271, no. 1537, pp. 361–365, 2004.

[169] E. Baird, M. J. Byrne, J. Smolka, E. J. Warrant, and M. Dacke, "The dung beetle dance: an orientation behaviour?" *PLoS ONE*, vol. 7, no. 1, p. e30211, 2012.

[170] H. Beyer, *The theory of evolution stratigies.* Berlin: Springer, 2011.

[171] L. Grossman, "Computer literacy tests: Are you human?" June 2008. [Online]. Available: http://www.time.com/time/magazine/article/0,9171,1812084,00.html