DOI: http://dx.doi.org/10.7551/978-0-262-31709-2-ch107

Emergence of Autonomous Behaviors of Virtual Characters through Simulated Reproduction

Yuri L. B. Nogueira[1], Carlos Eduardo F. de Brito[1], Creto A. Vidal[1] and Joaquim B. Cavalcante-Neto[1]

[1]Department of Computing, Federal University of Ceará, Brazil
yuri@lia.ufc.br

Abstract

This paper addresses the problem of autonomous behaviors of virtual characters. We postulate that a behavior is regarded as autonomous when the actions performed by the agent result from the interaction between its internal dynamics and the environment, rather than being externally controlled. In this work, we argue that an autonomous behavior is an agent's solution to a given problem, which is obtained through a process of self-organization of the dynamics of a system that is composed of the agent's controller, its body and the environment. That process allows the emergence of complex behaviors without any description of actions or objectives. We show a technique capable of adapting an artificial neural network to consistently control virtual Khepera-like robots by means of simulated reproduction, with no measure of the robots' fitness. All the robots are either male or female, and they are capable of evolving different kinds of behaviors according to their own characteristics, guided solely by the environment's dynamics.

Introduction

Contextualization

In this paper, we address the problem of autonomous behaviors of virtual characters (Shao and Terzopoulos (2007); Whiting et al. (2010)). A behavior is considered *autonomous* when the actions performed by the agent result from a close interaction between its internal dynamics and the circumstantial events in the environment, rather than from external control or specification dictated by a pre-defined plan.

That definition of autonomous behavior seems to entail an apparent contradiction to the process of creating virtual characters. Given that true autonomy implies no predefined behaviors, how is it possible to *design* the internal dynamics of an agent that is supposed to interact autonomously with its environment? The attempt to answer that question led us to investigate ways of obtaining behaviors by emergence.

Emergence can be described as the appearance of a system's global characteristic that cannot be found in any of its parts (Klaus and Mainzer (2009)). For example, although a portion of water at normal temperature is in the liquid state, we cannot say that a single water molecule can display this property. In general, the emergent properties are associated with dynamical patterns that get established through the interactions among the component parts of the system. In our particular case, the system should be considered as composed of the agent itself, defined by a virtual body and a controller, together with every aspect of the environment, both the objects and the lawful regularities that hold in the virtual world. In this setting, we define the notion of *emergent behavior* as follows: the behavior of a virtual character is called emergent when it is not explicitly described in any of the components of the system, and arises as a result of the dynamical interaction of the components, and their specific individual properties.

It is useful to think of the emergent behavior as an agent's solution to a given problem, which is obtained through a process of *self-organization*. Indeed, the real world's biological agents constantly come up with new behaviors to overcome challenges and to adapt to a changing environment. The emergence of a new behavior reflects the process of reorganization of the internal structures of the agent. In nature, this self-organization process is controlled mainly by Darwinian evolution dynamics: generation of diversity and natural selection.

These ideas have inspired many researchers to attempt to evolve neural controllers for virtual characters using Genetic Algorithms (GA) (Sims (1994); Nogueira et al. (2008); Pilat and Jacob (2010); Palmer and Chou (2012)). So, instead of anticipating and modelling all the ways in which the agent could possibly behave, the idea is to describe a task to be achieved (that is, to create a virtual environment with challenges for survival), and let the evolutionary process shape the virtual agent's control dynamics. That is expected to lead to the emergence of behaviors, which not only solve the task, but also are coherent with the capabilities of the agent's body and with the environment's characteristics.

We argue, however, that that approach leads only to a weak form of autonomy, because the GA guides the self-organization process using a *predefined* objective function. So, in that sense, behavior is still externally described. In nature, the quality or fitness of an agent depends on its internal constitution and the way it couples with the environment. Hence, the (natural) selection criteria also constitute an emergent characteristic of the system. In this work, in order to achieve a higher level of autonomy, we present a technique for obtaining emergent autonomous behaviors of virtual characters without an externally specified objective.

Proposed Solution

In this paper, we study the emergence of autonomous behaviors of virtual Khepera-like robots with:

- Non-interpreted simplified "vision" sensors;
- Controller consisting of an Artificial Neural Network (ANN);
- Adaptation through simulated sexual reproduction.

We show that our technique is capable of generating multiple behaviors in a population of robots: foraging, mating and obstacle avoidance. In our experiments, we could also observe different behaviors according to the gender of the robot and a complex use of the sensors for navigation.

In "Related Works" Section, we discuss the attempts of the community in obtaining autonomous behaviors of artificial agents. One notice a research trend that seeks to reduce the amount of external information provided to the system, moving from the traditional objective-driven GA to a completely environment-driven evolution, using ideas based on reproduction dynamics, similar to our work.

In "Controller" Section we describe the controller used in our virtual robots and the genetic encoding we developed to evolve it, since our simulated reproduction technique is based on the exchange of genetic material between a pair of robots of opposite genders. In "The Experiment" Section, we report the experiments, explaining the constitution of the robots and of the environment, and analyzing the dynamics of the whole system. The obtained results presented in "Behaviors" Section and final discussions are made in conclusion.

Related Works

Evolutionary computation has long been used as a tool to develop autonomous behaviors in artificial agents (Sims (1994); Palmer and Chou (2012)). Most works address behavior as a domain specific problem, and, traditionally, have proposed solutions, which, *a priori*, fix the objectives the agents need to achieve and the metric to evaluate how well the agents perform the task of meeting the objectives. However, there are also efforts along the line of creating techniques that incorporate additional aspects of natural selection in order to obtain greater complexity and autonomy of behaviors. In this section, we will briefly discuss the research path from the explicitly objective-driven canonical genetic algorithm to environment-driven open-ended evolution (Bredeche and Montanier (2012)).

Objective-driven evolution

The Virtual Reality community has extensively applied GAs in order to create virtual worlds automatically, in which autonomous characters present convincing behaviors. The proposed techniques are usually problem-oriented, with the evolutionary processes guided by fitness functions designed according to the expected behaviors of the characters. Some examples are the distance-based fitness of the walkers from Sims (1994) and Nogueira et al. (2008), or the speed-based fitness of the light followers from Pilat and Jacob (2010). Palmer and Chou (2012) went one step further by proposing a distributed GA that coevolves an interacting population of virtual hunter robots, instead of evolving a single individual at a time without taking into account possible interactions among them. However, the agents reproduce according to their relative fitness, based on a harvest score. The main characteristic of these works is the generation of behaviors that solve the problem in a way that is implicitly designed in the objective function.

Indeed, addressing the problem of autonomous behaviors through a problem-oriented technique, such as the canonical GA, leads to the evolution of agents capable of solving a single problem at a time. In order to achieve behavioral diversity, Schrum and Miikkulainen (2010) studied fitness-based shaping of behaviors to multiobjective domains, by dividing problems into a set of goals, i.e., a group of multiple fitness measures. A battle domain involving a scripted virtual fighter and a group of virtual monsters is used to illustrate the technique. The monsters had to maximize the inflicted damage, to minimize the received damage, and to maximize their life span. Another study that focuses on behavioral diversity is presented by Lehman and Stanley (2011), and suggests that one should abandon specific objectives and guide the search towards the novelty of solutions. These works attempt to overcome GA's lack of behavioral diversity by proposing ways of evolving several objectives simultaneously. However, each problem an agent should solve has to be properly predefined, because it is selected through some type of performance measure.

The effects of sexual gender discrimination were also investigated through evolutionary computation. Zhang et al. (2009) proposed a GA that uses a population consisting of male and female individuals, and a fitness function based on a model of the Baldwin effect. The work is concerned with the sexual reproduction in GA, and presents numerical simulation benchmarks in order to show improvements regarding convergence speeds, prevention of premature convergence and ability to solving high dimension problems. That work incorporates another feature of natural selection to GA: the gender differentiation. However, it does not specifically analyze the effects of this new feature on the generated behavior.

Da Rold et al. (2011) studied the effects of gender determination on behavior through a simulation with male and female robots in a virtual world containing energy resources.

The reported results show that the robots acquired different patterns of behaviors according to the gender and to the pregnancy status of the females. However, the sexual dynamics was not incorporated into the evolutionary algorithm itself, since a simple GA was used, with a fitness function based on the number of matings. Mating consists in a contact between two robots of opposite genders in which the female robot gets a psychological pregnancy (i.e., it does not generate offspring), remaining in that state for a specific amount of time, during which it cannot take part in another mating. For the GA, each gender constitutes a different population, which are evolved separately, although the evaluation of the individuals depends on the interaction between the two types of robots. The exhibited behavioral diversity is related to the fact that agents with different characteristics have to solve the same problem in different ways, suggesting that gender determination is an important aspect of natural selection. However, the way this feature is exploited in that work still shows the convergence of solutions to a predefined problem.

All the works discussed so far have the common characteristic of a centralized evaluation of the agents' fitness. A paradigm shift is presented in Embodied Evolution (Watson et al. (2002)), a distributed evolutionary algorithm embodied in physical robots. In that work, the agents have a reproduction function explicitly defined in terms of their energy level, in such a way that the genes that control robots with higher energy levels have greater probability of spreading out, while those that control robots with lower energy levels have greater probability of being replaced. The environment is endowed with energy resources, and the robots that are capable of benefiting the most from them are the ones that will spread their genes. Notice that, although the robots develop a behavior that is not directly selected, one can still say that the probability associated with the reproduction function plays the role of a fitness function, because it is explicitly designed to select individuals according to the preconception that those with higher levels of energy are the fittest ones.

Environment-driven evolution

In the environment-driven evolution approach, no fitness function is described, and the evolution is carried out by environmental pressures. That is, there is no explicit evaluation of an individual in order to select it or not, but the better performing individuals will naturally spread out according to the dynamics of the whole system.

Bredeche et al. (Bredeche and Montanier (2010); Bredeche et al. (2012)) applied this idea to evolve a population of autonomous real robots. They developed the Environment-driven Distributed Evolutionary Adaptation algorithm (EDEA), and showed that their algorithm is robust to the so called reality gap: a swarm of real robots is able to evolve efficient survival behavior strategies, with no fitness function being ever formulated. Although their work is presented mainly from an engineering point of view, many interesting conceptual discussions arise in this context, most of them independent of particular implementations.

The authors observe that the key to EDEA is the implicit nature of fitness function, that may be seen as a result of two motivations (Bredeche and Montanier (2010)):

- ***extrinsic motivation****: agent must cope with environmental constraints in order to maximize survival, which results solely from the interaction between the agent and the environment around (...);*
- ***intrinsic motivation****: set of parameters (ie. "genome") must spread across the population to survive, which is imposed by the algorithmic nature of the evolutionary process. Therefore, genomes are naturally biased towards producing efficient mating behaviors (...).*

A low correlation between the two motivations can increase the problem's complexity, since it will possibly imply conflicting objectives. Thus, an efficient environment-driven algorithm must address a "trade-off between extrinsic and intrinsic motivations as the optimal genome should reach the point of equilibrium where genome spread is maximum (e.g. looking for mating opportunities) with regards to survival efficiency (e.g. ensuring energetic autonomy)" (Bredeche and Montanier (2010)).

The idea of environment-driven evolution fits well with our analysis of emergent autonomous behavior, since both claim that the evolution of the system should be guided by the dynamics of the interactions among its component parts. In this sense, to be precise, we can say that the evolution is not only environmentally driven, but also population-driven, or better, system-wise driven. We note that every aspect of the system may offer an opportunity for improving adaptation, in ways that cannot be foreseen *a priori*. Individual characteristics of the agents and specific behaviors cannot be judged 'good' or 'bad' in isolation, but depend on the behavior of the rest of the population, and on the current dynamics of the system. The experiment in Bredeche et al. (2012) illustrates this point well, where one can see that the individual behavior of going towards the 'sun' is 'good' (i.e., favors reproduction) because a large number of robots in the population also tend to do so. From this perspective, we can say that there is not even an implicit fixed fitness function, since the dynamics of the system may change, and so the conditions for adaptation also may change. In other words, the implicit fitness function may be considered as another emergent aspect of the system.

In this paper, we study the emergence of autonomous behavior of virtual agents, using environment-driven evolution. Simulations gave us greater flexibility and allowed us to implement robotic sexual reproduction, a feature that is still impractical to obtain in real world experiments. Therefore, we could explore additional aspects of the emergence of autonomous behaviors, investigating, for example, the effects of population size and resources fluctuation on competitive behavior, the system's ability to follow alternative evolutionary paths, and the impact of gender differentiation on the generation of behavioral diversity.

The Controller

The Neural Network

The controller is essentially a Continuous Time Recurrent Neural Network (CTRNN), whose neurons are modeled in the following general form:

$$\frac{dy_i}{dt} = \frac{1}{\tau_i}(-y_i + \sum_{j=1}^{n} w_{ji} f(s_j) + I) \qquad (1)$$

where t is time, y_i and τ_i are, respectively, the internal state and the time constant for each neuron i, w_{ji} is the weight of the jth input synapse of neuron i, s_j is the state of the neuron linked to the jth input synapse, $f()$ is the activation function of a neuron and I represents a constant input to neurons.

Furthermore, we also use two types of neurons that do not have internal dynamics: the afferent and the efferent neurons. An afferent neuron, whose internal state is the value of one of the network's input, cannot receive input from another neuron. The afferent neurons constitute the network's input layer. An efferent neuron, on the other hand, is part of the network's output layer, and its internal state is the average of the internal states of all the neurons connected to it.

The Genetic Encoding

The controller of each robot is encoded into two chromosomes. The first chromosome encodes the stimulus I (Equation 1), and the second, which we call the Network Chromosome (NC), holds the gender of the robot and the description of the ANN itself. This grouping was chosen so that a "male brain" could evolve together with a "male gene", and a "female brain" could do so with a "female gene", while the same constant I could be tested with different networks.

The NC is defined according to a simplified version of Matiussi's Analog Genetic Encoding (AGE) (Mattiussi and Floreano (2007)) focused at evolving a CTRNN for the control of virtual characters. To create the synapses, AGE defines an alignment score: a network-specific interaction map that leads to a complex chromosomal representation. Our proposal keeps the idea of an implicit interaction between genes that encode the synapses. However, we specify a simpler similarity function, which not only makes it possible to describe the chromosome as a simple binary array, that encodes the parameters of the network in a more straightforward way, but also maintains the advantageous properties of AGE's interaction maps for ANNs evolution.

In implicit interaction, while the neurons are explicitly described in the chromosome, the synapses are implicitly defined, since they are formed by the interaction between genes, and not by a gene itself. To decode the ANN, we basically follow a two-step process:

1. Read the chromosome and extract the neurons and their respective input and output "ports", which we call "Neuronic Terminals" (TR);
2. Create the synapses from the interaction between the TRs.

This encoding scheme allows us to easily search augmenting topologies of neural networks.

In our work, a chromosome is an array of bits in which a single bit defines a "gender gene", and each group of 32 bits afterwards defines a regular gene. The single bit "gender gene" was introduced in order to enable sexual reproduction. The other genes are defined by a tuple $< id, v >$, where id identifies the encoded element, i.e., whether it is a neuron or a TR, and v is a value that indicates a property of the encoded element.

To decode the NC, we read the first bit to determine the gender of the robot, and then we read each subsequent gene (group of 32 bits) isolating its identifier from its value. A gene identified as a neuron creates a neuron element in the network. In the decoding sequence, any TR gene that appears before the first neuron gene is ignored; and after each new neuron gene, only the first two TR genes are considered. The first of those valid TR genes determines its input terminal, while the second TR determines its output terminal. The value of a neuron specifies its time constant τ_i (Equation 1), and the values of the TRs are used to calculate the synapses' weights between the neurons.

The first eight bits of the gene hold the id and are decoded according to Table 1. Note that the probability $P(TR)$ is greater than $P(N)$, since we expect more synapses than neurons.

Table 1: Genes' identifiers

Id value	Meaning
$0 \leq id \leq 51$	Neuron (N)
$52 \leq id \leq 255$	Neuronic Terminal (TR)

The last 24 bits of the gene encode the value v, which is linearly mapped into a floating-point number in the range $[-1, 1]$. If the value is related to a neuron gene, the result is directly attributed to the time constant of the neuron. If related to a TR, it is further used to calculate a synapse weight according to the equation:

$$w(i, o) = \frac{eb \cdot (i + o)}{nb \cdot 2}, \qquad (2)$$

where w is the weight of a synapse that links an output terminal of value o with an input terminal of value i. The symbol nb indicates the total number of bits that represent the value

(24 bits), and eb is the number of equal bits at the same position between the binary representations of i and o. We also defined an existence condition empirically to increase topological diversity: if $\lfloor eb/4 \rfloor \ mod \ 3 = 0$ then $w(i, o) = 0$. The whole process of network decoding is shown in Figure 1.

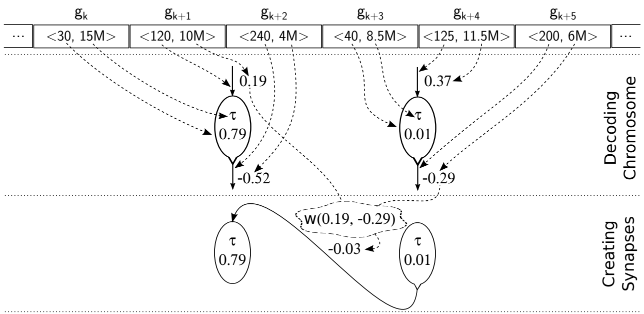

Figure 1: Building the network: First we decode the neurons and their respective terminals, then we apply Equation 2 to each pair of terminals to create the synapses. Only one synapse was created due to the existence condition (see text).

Regarding the input and output neurons, suppose that the robot has s sensors and m motors. In order to keep some structure of the ANN in the chromosome, we fix the first s genes that encode neurons to be afferent neurons, while we set the last m genes that encode neurons to be efferent neurons, that is, the inputs of the network are described in the beginning of the chromosome, the internal neurons are defined in the middle of the chromosome and the outputs are placed in the final part of the chromosome.

The Experiment

System description

Our simulation was developed with the *Irrlicht 3D Engine*[1], with physics provided by the *Bullet Physics Engine*[2]. The environment is populated by simulated "male" and "female" robots that live in a square room, bounded by walls, and filled with randomly distributed fruits (Figure 2), from which the robots can get energy to live.

The robots have cylindrical bodies. A black box in the cylindrical surface represents, at the same time, the eye, the mouth and the genitals of the robot and determines its front part. The robots guide themselves through the environment using their vision, obtain energy by eating fruits and reproduce through mating. Each of these functions are better described next.

The robot's vision is determined by three sensors positioned in the black box (the eye), as shown in Figure 3. Each sensor is able to catch the normalized distance ($[0, 1]$) to the nearest object inside its "Field Of Sense" (FOS) with respect

[1]http://irrlicht.sourceforge.net/

[2]http://bulletphysics.org/

Figure 2: The environment.

to its reach (the maximum detection distance of a sensor). The sensor that is located at the center of the eye is specialized to detect walls only, and has a FOS of $120°$ and a reach of approximately $4 * r$, where r is the radius of the robot's body. The other two sensors, placed at each side of the eye, are able to sense male robots, female robots and fruits, and have a FOS of $10°$ and reach of approximately $14 * r$. Those values were empirically chosen.

The wall sensor generates only one floating-point value indicating the normalized distance to the wall. In addition to the distance value, each one of the other two sensors also generate three bits that indicate the type of object sensed, i.e., a male robot, a female robot or a fruit. That means that the whole vision apparatus generates nine values.

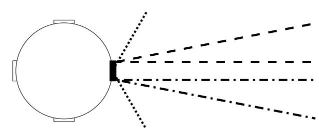

Figure 3: The distribution of the three vision sensors. The dotted lines represent the FOS of the wall sensor. The dashed lines and the dashed-dotted lines represent, respectively, the left sensor and the right sensor of robots/fruits.

Furthermore, there are proprioceptive senses of fertility and energy. The sense of fertility enables a male robot to know when it is infertile (1 if infertile, 0 otherwise) and a female robot to know when it is pregnant (1 if pregnant, 0 otherwise). The sense of energy enables a robot to know its level of energy, which ranges from 0 (the robot is fully energized) to 1 (the robot is totally exhausted). Therefore, the strength of the signal allows the robot to perceive when its energy is finishing. Thus, there are nine signals of vision and two signals of proprioceptive senses, leading to an ANN with eleven afferent neurons.

A robot has two motors, which are controlled by two efferent neurons respectively. When the first motor receives a signal from its efferent neuron, it moves the robot forward in case the value of the signal is positive, and moves the robot backward if the signal is negative. Likewise, the other motor makes the robot turn right in the event of a positive signal, and makes the robot turn left if the received signal is negative. The actions of the motors are simplified and are not physically accurate. The amplitude of the signals generated by the efferent neurons are increased 100 times, before they are applied as the robot's speed.

The energy level of a robot increases 7,500 energy units (eu) whenever a fruit is eaten, i.e., when the robot touches a fruit with its mouth. The maximum energy value is 100,000eu and continuously decreases in direct proportion to the applied motor signals plus a value proportional to the robot's age. If the energy is exhausted, the robot dies and is removed from the environment. Thus, for example, suppose that o_1 and o_2 are the amplified output of the two efferent neurons of the ANN and that t is the robot's age. Then, the energy consumption C is calculated according to the equation:

$$C = (|o_1| + |o_2|)^2 + \frac{t}{75.0}. \tag{3}$$

Regarding the dynamics of fruits replacement, at each time step, a new fruit is randomly placed in the room, provided that:

- The number of fruits does not exceed 30 fruits; and
- The total number of objects, i.e., the number of fruits plus the number of robots, does not exceed the limit of 45 objects.

Simulated Reproduction

The simulation starts with 15 robots. If the population becomes smaller than 6 individuals, we place 15 new random individuals in random positions. Each of these robots has its energy randomly initialized with a value between 10,000eu and 20,000eu. The genetic information is also randomly generated. Since we have just one bit to express the robot's gender, 50% of the population consists of female robots.

Mating is consummated whenever a male robot's genitals (the black box) touch the body (any part of the cylinder) of a fertile female robot. A female is fertile if its energy is greater than 25,000eu. Since any robot has an initial energy of 20,000eu at most, every female is infertile at first, and needs to eat some fruits in order to reproduce.

If mating occurs, the male robot gets infertile during 250 simulation steps and the female robot gets pregnant. The new robot is placed adjacent to its mother, so the female robots remains pregnant until she goes to a free place where its child can be positioned. The child's energy is initialized with a value between 15,000eu and 25,000eu, which is taken from its mother. Therefore, when, at the moment it gets pregnant, a female robot has low energy (a value close to 25,000eu), there is a higher probability that it will die sooner. After giving birth, the female gets infertile during 250 simulation steps to avoid a pregnancy right after the other.

The chromosomes of the new individual are generated by crossing over the parents' chromosomes. Since a chromosome is simply an array of bits, the process of crossover sets two breakpoints randomly, and exchange the bits between the pair of chromosomes at the defined range. Note that this method can generate mutation by breaking a gene, since it is defined by a group of 32 bits. This is expected to create variability. We also apply an explicit mutation, randomly changing bits in the chromosomes with a probability of 0.1%. Since after the crossover we still have a pair of chromosomes, we simply discard one randomly.

The genetic information of an individual encodes its ANN directly. Therefore, when crossover takes place between a pair of chromosomes of different individuals, the process can be viewed as if pieces of each individual's brains were being exchanged. Consequently, the newly generated brain can lead to a robot with behavioral traits inherited from both parents.

Life Dynamics

Note that, according to the reproduction dynamics described, if there is a high density of fruits and robots at the environment, mating is relatively easy to occur and can occasionally happen in a random way. In fact, this is necessary to avoid endless resumptions of the population and to bring some line of evolution. In so far as that the population evolves, those individuals who present some type of strategy are at a greater advantage and will impose new conditions to the system, causing random behaviors to decrease.

Another important point to comment is the balance between the number of individuals in the population and the amount of energy resources. According to the fruits' replacement dynamics, if the population grows, the number of fruits available reduces. Therefore, with scarce energy resources in the environment, the robots with worse performance will die. Note that, if the population size grows above 45, no fruit will appear and, thus, the robots that are less efficient will die before new fruits appear. That dynamics prevents population explosion automatically and provides some selective pressure, which guides evolution.

Behaviors

We ran the simulation several times and obtained mixed interesting behaviors. Due to space constraints, we cannot present the results of all the runs[3]. Therefore, we will focus the discussion in a common recurring result: females tend to seek food and males tend to look for mating. Furthermore, the robots learned how to deviate from the wall and how to use their simple vision to guide themselves through the environment efficiently and meet their needs. In Figure 4, a sequence of frames shows two robots presenting the mentioned behaviors, which are detailed in the respective labels.

[3]Watch the video with multiple runs of our experiments: https://www.youtube.com/watch?v=zyDdjD6d5CE

Other interesting strategies also emerged from the population in different simulation runs, always leading to the population's survival and stability.

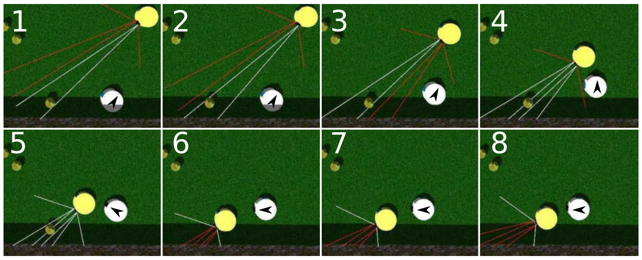

Figure 4: Commonly observed behaviors. Note that the male robot (the one with the arrow indicating his direction) follows the female robot that seeks the fruit. Another interesting point to emphasize, is the complex use of the simple vision: the female senses the fruit with her left sensor (Frame 1), and then turns left (Frames 2 and 3) in order to use the right sensor to determine the direction to follow, correcting the movements and maintaining the object between the two sensors (Frames 4 and 5). Frames 6, 7 and 8 show the female robot turning right in order to deviate from the wall after catching the fruit.

Since we do not have to define any objective, there is no variable to watch in order to follow convergence toward an expected behavior. However, some values indicate the evolution of the whole population, and the analysis of the relations among those values allows us to objectively demonstrate the emergence of the described behaviors.

The mean lifespan of the population over a period of time is a good parameter to see the emergence of some strategy of the population in order to survive. Another aspect that can indicate characteristics of the behaviors is the correlation between the male and female lifespans. Figure 5 shows the mean lifespans of the population in the simulation run described in Figure 4. Note that the female robots have converged to a mean lifespan greater than that of male robots. This is related to the fact that the female robots were always searching for food actively, while the male robots caught a fruit occasionally. However, both gender increased their efficiencies.

The average number of collected fruits and the average number of matings are good parameters to analyze the behavioral characteristics of each gender. In Figure 6a we can see that, in the analyzed simulation run, the female robots converged to collect about 10 fruit on average during their lifespan, while the average number of fruits collected by the male robots is less than one, demonstrating the preference of female robots for collecting fruits. In Figure 6b, we can note the preference of the male robots for the mating behavior. However, it is important to mention that other runs presented different strategies, such as, for example, the formation of clusters of robots, which increase the probability of matings, and the presence of robots with both foraging and mating behaviors, regardless of the gender.

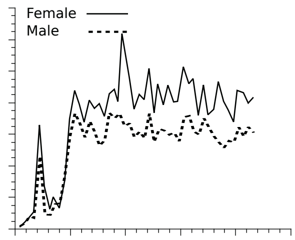

Figure 5: Mean lifespan of the population every 900 seconds of simulation.

The observation of the population size at a given time along with the average number of collected fruits and the mean lifespan, shows some aspects of the general behavior of the whole population. Note that there is a peak in the graph of Figure 6a before convergence around a certain smaller value. Analyzing Figures 5 and 6a at approximately the same time (about 4000 seconds and 1 hour, respectively), we can see that as the robots learn how to catch fruits, there is an increase in the mean lifespan. As described, the number of fruits placed in the environment depends on the number of robots. So, an increase in the mean lifespan leads to population growth and, consequently, to reduction of the available resources, hence reducing the average number of collected fruits per robot. That leads to a balance of the population size, preventing population explosion, as shown in Figure 7.

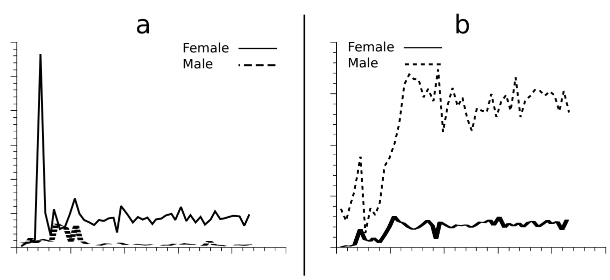

Figure 6: Average number of collected fruits and matings every 900 seconds of simulation. (a) The greater number of fruits collected by the female robots indicates their behavioral tendency to foraging. (b) Note the increase of matings with time. This shows a male preference to such behavior.

Conclusions

We described an artificial life system where virtual Khepera-like robots developed multiple autonomous behaviors, without any description of their objectives. The observed behaviors emerged solely from the self-organization of the dynam-

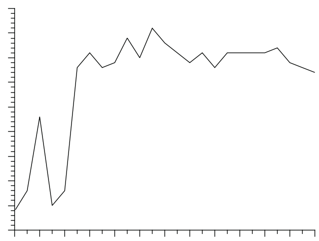

Figure 7: Population size every 30 minutes. Note that with 2.5 hours of simulation, the population size increases to about 45 individuals and then is balanced around this number due to shortage of resources caused by the large amount of robots that developed the foraging behavior.

ics of the system. The robots were divided into genders and were controlled by an ANN. We presented a genetic encoding for the ANN, which allowed the adaptation of controllers through simulated reproduction, providing an implementation of environment-driven evolution.

The system was capable of exhibiting several types of behaviors, according to the robots' characteristics. A common situation observed, was the emergence of mating behavior in male robots and foraging behavior in female robots. A single individual was also able to show multiple behaviors, such as avoiding collisions with the walls and use of vision to pursue its own objectives. Although different behaviors have emerged from different simulation runs, the system was always able to show the evolution of robots presenting strategies that led to an increase in the mean lifespan and in the size of the population.

The results of our experiments show that the self-organization of the system is capable of producing an intimate coupling between agent and environment, producing complex and natural behaviors without any *a priori* description. This characteristic is clearly illustrated with the strategy developed by the virtual agent to compensate for its primitive visual sensory apparatus and to be able to find a direction to an object, an information that is not originally provided by the sensors.

Acknowledgements

This work was supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

References

Bredeche, N. and Montanier, J.-M. (2010). Environment-driven Embodied Evolution in a Population of Autonomous Agents. In *Parallel Problem Solving From Nature*, pages 290–299.

Bredeche, N. and Montanier, J.-M. (2012). Environment-driven Open-ended Evolution with a Population of Autonomous Robots. In *Evolving Physical Systems Workshop*, East Lansing, United States.

Bredeche, N., Montanier, J.-M., Liu, W., and Winfield, A. (2012). Environment-driven Distributed Evolutionary Adaptation in a Population of Autonomous Robotic Agents. *Mathematical and Computer Modelling of Dynamical Systems*, 18(1):101–129.

Da Rold, F., Petrosino, G., and Parisi, D. (2011). Male and female robots. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, 19(5):317–334.

Klaus and Mainzer (2009). From embodied mind to embodied robotics: Humanities and system theoretical aspects. *Journal of Physiology-Paris*, 103(3-5):296 – 304.

Lehman, J. and Stanley, K. O. (2011). Abandoning objectives: Evolution through the search for novelty alone. *Evol. Comput.*, 19(2):189–223.

Mattiussi, C. and Floreano, D. (2007). Analog Genetic Encoding for the Evolution of Circuits and Networks. *IEEE Trans. Evol. Comput.*, 11(5):596–607.

Nogueira, Y. L. B., Vidal, C. A., and Cavalcante-Neto, J. B. (2008). A nervous system model for direct dynamics animation control based on evolutionary computation. In *SAC '08: Proc. of the 2008 ACM symposium on Applied computing*, pages 1793–1800. ACM.

Palmer, M. E. and Chou, A. K. (2012). An artificial visual cortex drives behavioral evolution in co-evolved predator and prey robots. In *Proc. of the 14th intl. conf. on Genetic and evolutionary computation conference companion*, GECCO Companion '12, pages 361–364. ACM.

Pilat, M. L. and Jacob, C. (2010). Evolution of vision capabilities in embodied virtual creatures. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, GECCO '10, pages 95–102, New York, NY, USA. ACM.

Schrum, J. and Miikkulainen, R. (2010). Evolving agent behavior in multiobjective domains using fitness-based shaping. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, GECCO '10, pages 439–446, New York, NY, USA. ACM.

Shao, W. and Terzopoulos, D. (2007). Autonomous pedestrians. *Graph. Models*, 69(5-6):246–274.

Sims, K. (1994). Evolving virtual creatures. In *Proc. of the 21st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1994*, pages 15–22. ACM.

Watson, R. A., Ficici, S. G., and Pollack, J. B. (2002). Embodied evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems*, 39(1):1–18.

Whiting, J. S., Dinerstein, J., Egbert, P. K., and Ventura, D. (2010). Cognitive and behavioral model ensembles for autonomous virtual characters. *Comput. Intell.*, 26(2):142–159.

Zhang, M., Zhao, S., and Wang, X. (2009). A hybrid self-adaptive genetic algorithm based on sexual reproduction and baldwin effect for global optimization. In *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, CEC'09, pages 3087–3094, Piscataway, NJ, USA. IEEE Press.