The GOLEM Project: Evolving Hardware Bodies and Brains[1]

Jordan B. Pollack Hod Lipson

Computer Science Department, Volen Center for Complex Systems
Brandeis University, Waltham MA 02454-9110, USA
lipson@cs.brandeis.edu

Abstract

The GOLEM project is an attempt to extend evolutionary techniques into the physical world by evolving diverse electro-mechanical machines (robots) that can be fabricated automatically. In this work we go beyond evolution of hardware controllers and demonstrate for the first time a path that allows transfer of virtual diversity of ***morphology*** *into reality. Our approach is based on the use of only elementary building blocks in both the design and embodiment. We describe a set of preliminary experiments evolving electromechanical systems composed of thermoplastic, linear actuators and neurons for the task of locomotion, first in simulation then in reality. Using 3D solid printing, these creatures then replicate automatically into reality where they faithfully reproduce the performance of their virtual ancestors.*

1 Introduction

The field of evolutionary robotics is suffering a dichotomy. Objects of study in this field are either digital creatures that are diverse and dynamic but remain virtual, or robots that are physical but have a predominantly fixed architecture, and evolutionary techniques are applied only to their control level but not their morphology. Indeed, studies in the field of evolutionary robotics reported to date involve either entirely virtual worlds (Sims 1994; Komosinski 1999), or, when applied in reality, adaptation of only the control level (either in FPGA hardware on in software) of manually designed and constructed robots with a predominantly fixed architecture (Husbands and Meyer, 1998). Other works involving real robots make use of high-level building blocks, like wheeled platforms and C-subroutines comprising significant pre-programmed knowledge (Leger, 1999)

Yet one of the prevailing characteristics of natural life is full autonomy, not only in power and behavior, but also in the design and fabrication of the entire robot. In nature we see no distinction between body and brain – this distinction has been introduced artificially by our own disciplinary culture. We hypothesize that only when we extend hardware evolution to encompass both body and brain, can we expect synthetic creatures to bootstrap and sustain their own evolution.

We thus seek automatically designed and constructed physical robots that are functional in the real world, diverse in architecture (possibly each slightly different), and producible in their entirety at low cost and large quantities. So far these requisites have not been met.

1.1 Structure of this paper

This paper is structured as follows: First, we outline the GOLEM project and our approach for autonomy, based on the use of elementary building blocks, and contrast it with current approaches in evolutionary robotics. We then describe a set of preliminary experiments examining the ability to achieve fully automated design and transfer into reality, both with as little human intervention as possible. We make no claims as to the evolutionary computation itself: we use a simple evolutionary algorithm and hence do not elaborate on the evolutionary program itself beyond providing sufficient details for replication. We then present results of both virtual and real machines evolved for the task of locomotion, and compare their performance. In performing this comparison we essentially complete a physical synthetic evolution cycle. We then conclude with some remarks on the significance of these results.

2 Elementary Building Blocks

The approach we propose is based on the use of only elementary constituents in both the design and fabrication process. As building blocks become more elementary, external knowledge associated with them is minimized, and

[1] Extracts from H. Lipson, J. B. Pollack, "Evolving Physical Creatures", *Proceedings of Artificial Life 7, Portland Oregon, 2000.*

0-7695-0762-X/00 $10.00 © 2000 IEEE

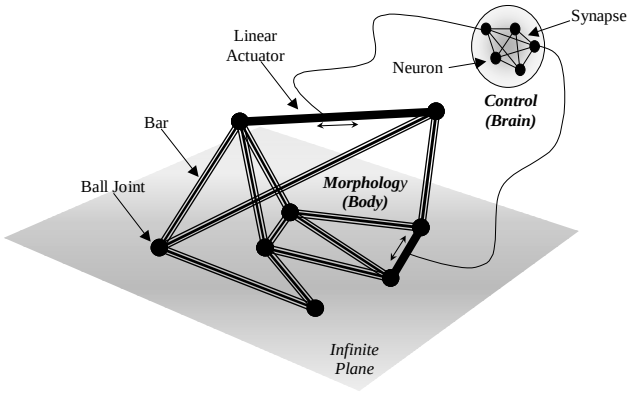


Figure 1. Schematic illustration of an evolvable robot, containing only linear bar/actuators and control neurons. Some bar architectures form rigid substructures

at the same time architectural flexibility is maximized. Similarly, use of elementary building blocks in the fabrication process allows it to be more simple and systematic. The use of elementary building blocks also minimizes potential inductive bias that might be introduced inadvertently into the evolutionary substrate. In theory, if we could use only atoms as building blocks, only laws of physics as constraints and only atom-manipulation as a fabrication process, then this principle would be maximized. Earlier reported work on evolution complete-creatures used higher level knowledge and consequently limited architectures (like only tree structures, Sims 1994; Komosinski 1999) and resulted in expedited convergence to working solutions, but at the expense of exploration power and reduced claims of innovation that might be attributed to the algorithm.

3 Experiments

In a set of experiments we used bars as building blocks of structure, neurons as building blocks of control, and additive fabrication (Demos et al, 1998) as a production process. Bars connected with free joints can form trusses that represent arbitrary rigid, flexible and articulated structures as well as multiple detached structures, with revolute, linear and planar joints at various levels of hierarchy. Similarly, sigmoidal neurons can connect to create arbitrary control architectures such as feed-forward and recurrent nets, state machines and multiple independent brains. Additive fabrication allows automatic generation of arbitrarily complex physical structures and series of physically different bodies. A schematic illustration of a possible architecture is shown in Figure 1. The bars connect to each other through ball-and-socket joints, neurons can connect to other neurons through synaptic connections, and neurons can connect to bars. In the latter case, the length of the bar becomes governed by the output of the neuron, essentially making it a linear actuator. No sensors were used at this stage.

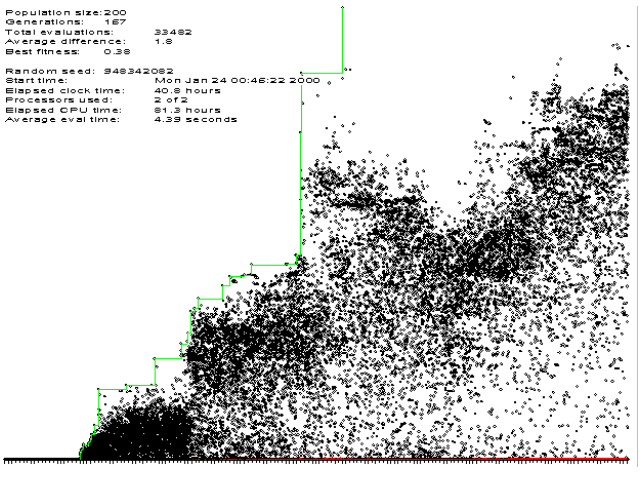


Figure 2. A typical plot of solution fitness (0-0.38) as function of generation (167).

Starting with a population of 200-1000 machines that were initially comprised of zero bars and neurons, we conducted evolution in simulation. The fitness of a machine was determined by its locomotion ability: the net distance its center of mass moved on an infinite plane in a fixed

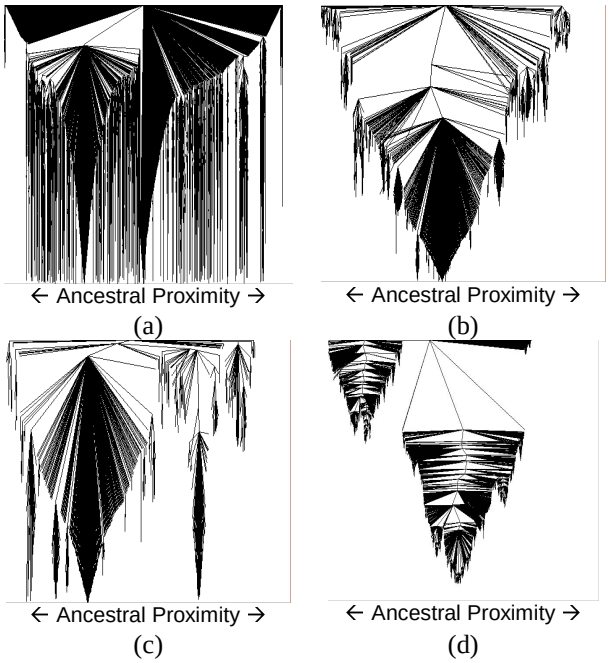


Figure 3. Phylogenic trees of several different evolutionary runs. Vertical axis represents generations and horizontal axis represents ancestral proximity. Trees exhibit various degrees of divergence and speciation: (a) extreme divergence, (b) extreme convergence, (c) intermediate level, and (d) massive extinction. The trees were thinned and depict several hundred generations each.

0-7695-0762-X/00 $10.00 © 2000 IEEE

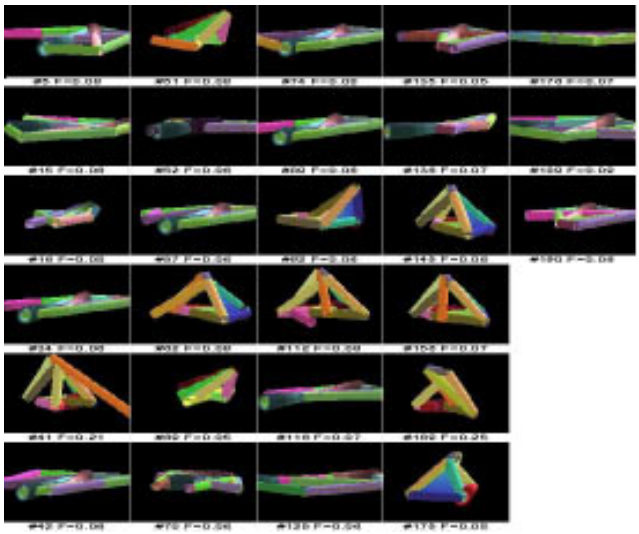
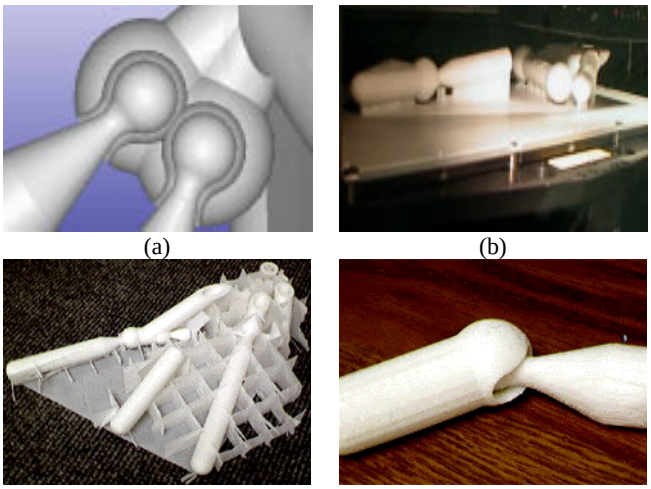


Figure 4. A sample instance of an entire generation, thinned down to show only different individuals. Note the two prevailing species.

duration. The process iteratively selected fitter machines, created offspring by adding, modifying and removing building blocks, and replaced them into the population. Figure 2 shows a typical progress of fitness of creatures as function of generations. (See Appendix A for details of the experiments).

We use only elementary operators of mutation that introduce least external knowledge. This process continued for several hundred generations. Both body (morphology) and brain (control) were co-evolved simultaneously. Although it is common practice in the field to separate and evolve a control for a fixed morphology and vice versa, in nature there is no such distinction – like a chicken and egg – neither came first. Coevolution has been successfully used to solve problems such as sorting networks (Hillis, 1990), and cellular automata (Juillé and Pollack, 1998).

The simulator we used for evaluating fitness supported quasi-static motion in which each frame is statically stable . This kind of motion is simpler to transfer reliably into reality, yet is rich enough to support low momentum locomotion (See Appendix B for details of the simulation). Typically, several tens of generations passed before the first movement occurred. For example, at a minimum, a neural network generating varying output must assemble and connect to an actuator for any motion. Various patterns of evolution dynamics emerged, some of which are reminiscent of natural phylogenic trees. Figure 3 presents examples of extreme cases of convergence, speciation, and massive extinction. A sample instance of an entire generation, thinned down to only unique individuals is shown in Figure 4.



(a) (b) (c) (d)

Figure 5. (a) Automatically fleshed joints, (b) replication process, (c) pre-assembled robot in mid print, (d) joint close-up

4 Fabrication

Selected robots out of those with winning performance were then *automatically* replicated into reality: their bodies, which exist only as points and lines, are first converted into a solid model with ball-joints and accommodations for linear motors according to the evolved design (Figure 5a). The solidifying stage was automatic but used a hand-coded procedure describing a generic bar, joint, and actuator. The virtual solid bodies were then materialized using commercial rapid prototyping technology (Figure 5b). This machine uses a temperature-controlled head to extrude thermoplastic material layer by layer, so that the arbitrarily evolved morphology emerges as a solid three-dimensional structure (Figure 5c) without tooling or human intervention. The entire pre-assembled machine is printed as a single unit, with fine plastic supports connecting between moving parts; these supports break away at first motion (Figure 5d).

The resulting structures contained complex joints that would be difficult to design or manufacture using traditional methods (see Figure 6b). Standard motors are then snapped in, and the evolved neural network is executed

Table I: Comparison of performance of physical creatures versus their virtual origin. Values are net distance [cm] center of mass traveled over 12 cycles of neural network. Distances in physical column are compensated for scale reduction (actual distance in parentheses).

Distance traveled [cm]	Virtual	Physical
Tetrahedron (Figure 6a)	38.5	38.4 (35)
Arrow (Figure 6b)	59.6	22.5 (18)
Pusher (Figure 6c)	85.1	23.4 (15)

0-7695-0762-X/00 $10.00 © 2000 IEEE

to activate the motors.

5 Results

In spite of the relatively simple task and environment (locomotion over an infinite horizontal plane), surprisingly different and elaborate solutions were evolved. Machines typically contained around 20 building blocks, sometimes with significant redundancy (perhaps as means against mutation, Lenski et al, 1999). Not less surprising is the fact that some exhibited symmetry, which was neither specified nor rewarded for anywhere in the code; a possible explanation is that symmetric machines are more likely to move in a straight line, consequently covering a greater net distance and acquiring more fitness. Similarly, successful designs appear to be robust in the sense that changes to bar lengths would not significantly hamper their mobility. The corresponding physical machines (3 to date) then faithfully reproduced their virtual ancestors' behavior in reality. Table 1 compares the performance of the physical creatures versus their virtual origin. While the distances traveled in differ significantly between simulation and reality, it should be noted that the actual mode of locomotion was faithfully replicated. The difference stems from slipping of the limbs on the floor surface, implying that the simple Newtonian friction model we used was insufficiently accurate. Three samples are shown and described in Figure 6, exploiting principles of ratcheting (6a), anti-phase synchronization (6b) and dragging (6c). Others (not shown here) used a sort of bi-pedalism, where left and right "limbs" are advanced in alternating thrusts. Some mechanisms moved articulated components to produce crab-like sideways motion. Other machines used a balancing mechanism to shift friction point from side to side and advance by oscillatory motion.

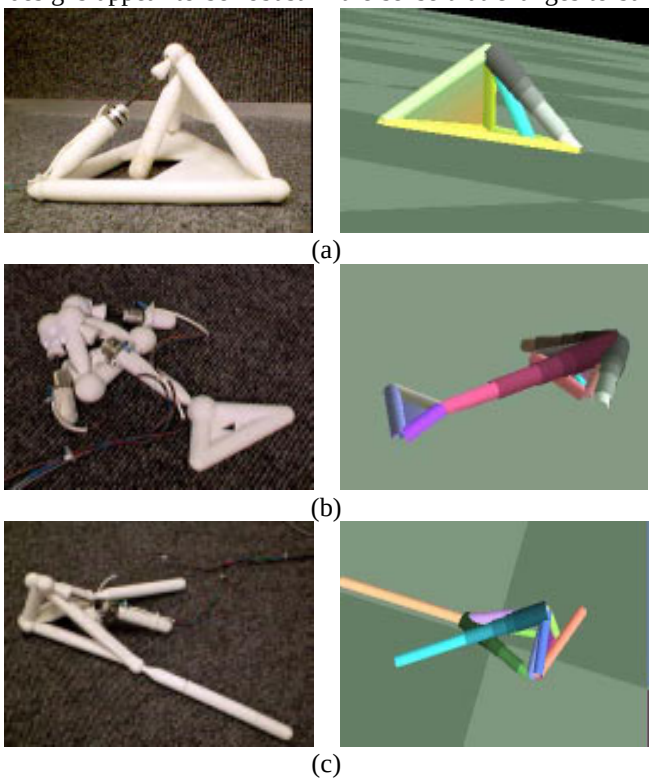


Figure 6. (a) A tetrahedral mechanism that produces hinge-like motion and advances by pushing the central bar against the floor. (b) This surprisingly symmetric machine uses a 7-neuron network to drive the center actuator in perfect anti-phase with the two synchronized side limb actuators. While the upper two limbs push, the central body is retracted, and vice versa. (c) This mechanism has an elevated body, from which it pushes an actuator down directly onto the floor to create ratcheting motion. It has a few redundant bars dragged on the floor, which might be contributing to its stability. These machines perform in reality is the same way they perform in simulation. Motion videos of these robots and others can be viewed at http://www.demo.cs.brandeis.edu/golem

6 Conclusions

In summary, while both the machines and task we describe in this work are fairly simple from the perspective of what teams of human engineers can produce, and what biological evolution has produced, we have demonstrated for the first time a robotic bootstrap, where robotically designed electromechanical systems can be robotically manufactured. We have carefully minimized human intervention both in the design and in the fabrication stages. Besides snapping in the motors, the only human work was in informing the simulation about the universe it could expect to be manufacturable.

Without reference to specific organic chemistry, life is an autonomous design process that is in control of a complex set of chemical factories allowing the generation and testing of physical entities which exploit the properties of the medium of their own construction. Using a different medium, namely off-the-shelf rapid manufacturing, and evolutionary design in simulation, we have replicated this autonomy of design and manufacturing. Our claimed advance, namely the ability to move artificial evolution from simulation to the real world is not a mere curiosity; rather, some claim that if indeed artificial systems are to ultimately interact and integrate with reality, they must learn, evolve and be studied in it (Beer, 1990). Technological advances in multi-material rapid prototyping, MEMS and nano-fabrication, higher fidelity of physical simulation and increased understanding of evolutionary computational processes thus open exciting opportunities in more fully automating this path towards artificial life.

0-7695-0762-X/00 $10.00 © 2000 IEEE

Acknowledgments

This research was supported in part by the Defense Advanced Research Projects Administration (DARPA) Grant No. DASG60-99-1-0004. Hod Lipson acknowledges the generous support of the Fischbach Postdoctoral Fellowship.

Appendix A

Details of the evolutionary simulation

Experiments were performed using version 1.2 of GOLEM (Genetically Organized Lifelike Electro Mechanics), which can be obtained from http://www.demo.cs.brandeis.edu/golem. We carried out a simulated evolutionary process (Holland, 1975; Koza, 1992): The fitness function was defined as the net Euclidean distance that the center-of-mass of an individual has moved over a fixed number (12-24) of cycles of its neural control. We started with a population of 200-1000 null (empty) individuals. Random seed was randomized. Individuals were then selected, mutated, and replaced into the population in steady-state as follows: The selection function can be either random, fitness proportionate or rank proportionate. The mutation operators used to generate an offspring can be any of the following (with probability): Small mutation in length of bar or neuron synaptic weight (0.1), removal/addition of a small dangling bar or unconnected neuron (0.01), split vertex into two and add a small bar or split bar into two and add vertex (0.03), attach/detach neuron to bar (0.03). The dice is rolled until at least one mutation is applied. After mutation, a new fitness is assigned to the individual by means of a simulation of the mechanics and the control (see details below). The offspring is inserted into the population by replacing an existing individual either chosen at random, chosen in inverse-proportion to its fitness, or using similarity proportionate elitism (deterministic crowding, Mafoud, 1995). Various permutation of selection-replacement methods are possible; we typically used fitness-proportionate or rank selection with random replacement, or random selection with similarity proportionate replacement where similarity was approximated by the distance in the ancestral tree. The process continued for 500-5000 generations (approx 10^5 to 10^6 evaluations overall). The process was carried both serially and in parallel (on a 16-processor computer). On parallel computers we noticed an inherent bias towards simplicity: Simpler machines could complete their evaluation sooner and consequently reproduce more quickly than complex machines.

Appendix B

Details of the simulation

Both the mechanics and the neural control of a machine were simulated concurrently. The mechanics were simulated using quasi-static motion, where each frame of the motion is assumed to be statically stable. This kind of motion is simple to simulate and easy to induce in reality, yet is rich enough to support various kinds of low-momentum motion like crawling and walking (but not jumping). The model consisted of ball-joined cylindrical bars with true diameters. Each frame was solved by relaxation: An energy term was defined, taking into account elasticity of the bars, potential gravitational energy, and penetration energy of collision and contact. The degrees of freedom of the model (vertex coordinates) were then iteratively adjusted according to their derivatives to minimize the energy term, and the energy was recalculated. Static friction was also modeled. The use of relaxation permitted handling singularities (e.g. snap-through buckling) and under-constrained cases (like dangling bar). Noise was added to ensure the system does not converge to unstable equilibrium points, and to cover simulation-reality gap. Material properties modeled correspond to the properties of the rapid prototyping material (E=0.896GPa, ρ=1000Kg/m^3 σ_{yield}=19MPa). The neural network was simulated in discrete cycles. In each cycle, actuator lengths were modified in small increments not larger than 1 cm.

Simulator physics

Static solution of each frame is achieved by defining a global system energy term H and then slightly modifying each of the systems degrees of freedom so as to lower H according to its partial derivatives. This process continues until relaxation is reached or instability is determined. The terms included in H define the richness of the simulation.
For example, a basic model including bar flexion and gravitational energy would be:

$$H = \sum_1^n k\delta^2 + \sum_1^n mgh$$

where n is the number of bars, and for each bar i, m represents the mass, h represents the average height, and the term k represents the stiffness,

$$k_i = \frac{E_i A_i}{l_i}$$

where E is the material's module of elasticity, A is the cross section, and l is the length of the bar, and δ represents the difference between the bar's current length and its original length, given by

$$\delta_i = \sqrt{(v_x - w_x)^2 + (v_y - w_y)^2 + (v_z - w_z)^2} - l_i$$

0-7695-0762-X/00 $10.00 © 2000 IEEE

and *v*,*w* represents the two current endpoints of the bar. Differentiating the total energy *H* with respect to each of the degrees of freedom (coordinates of the endpoints, in this case), produces direction of adjustment (second derivatives would produce a more accurate adjustment, etc.).

$$\Delta w_x = \frac{\partial H}{\partial w_x} \cdot \Delta s$$

where Δs (the relaxation factor) represents the adjustment magnitude for each iteration. Each degree of freedom is then updated by its adjustment value, and *H* is recalculated. Small Δs produce a stable but slower convergence, whereas large Δs can solve static frames faster but may run into stability problems. This process is repeated until adjustments go below a certain threshold.

Although this solution method is slower that simultaneous solutions, (e.g. finite elements), it is capable of solving correct highly-nonlinear cases (e.g. contact collision and snap-through buckling) and, more importantly, under-constrained states: for example, a truss in mid fall which is not yet fully supported (by six rigid-body equations), as it falls into position. Indeed as machines move and function within the simulator they often pass through such stages.

As additional real-world energy terms are added to *H*, it becomes more accurate. We have included terms for friction, collision between bars, external forces and noise.

References

Beer R. D., (1990) *Intelligence as Adaptive Behavior*, Academic Press

Dimos. D, Danforth S.C., Cima M.J., (1998), *Solid Freeform and Additive Fabrication*, MRS Symposium, Boston Massachusetts

Hillis, W. D. (1990). Co-evolving parasites improve simulated evolution as an optimizing procedure. *Physica.*

Holland J., (1975) *Adaptation in natural and artificial systems*, University of Michigan Press

Husbands P., Meyer J. A., (1998), *Evolutionary Robotics*, Springer Verlag

Juillé, H. and Pollack, J. B. (1998). Coevolving the "Ideal" Trainer: Application to the Discovery of Cellular Automata Rules. *Proceedings of the Third Annual Genetic Programming Conference* , Madison, Wisconsin, July 22 - 25, 1998

Komosinski M., Ulatowski S., (1999) Framstics: Towrds a simulation of a nature-like world, creatures and evolution, ECAL '99, pp. 261-265

Koza J., (1992) *Genetic Programming*, MIT Press

Leger C., (1999), "Automated Synthesis and Optimization of Robot Configurations: An Evolutionary Approach", *Ph.D. Thesis*, Carnegie Mellon University

Lenski R. E., Charles O., Collier T., Adami C., (1999) "Genome Complexity, robustness and genetic interactions in digital organisms", *Nature* 400, pp. 661-664

Mahfoud S. W., (1995), "Niching methods for genetic algorithms", *Ph.D. Thesis*, University of Illinois at Urbana-Champaign

Sims K., (1994) Evolving Virtual Creatures, Computer Graphics, SIGGRAPH '94, pp. 15-22

0-7695-0762-X/00 $10.00 © 2000 IEEE