Evolving Controllers for Virtual Creature Locomotion

Michael Sanders, Richard Lobb, Patricia Riddle
Department of Computer Science, University of Auckland, Auckland, New Zealand
msanders@ihug.co.nz, {richard, pat}@cs.auckland.ac.nz

ABSTRACT

We consider the application of Evolutionary Algorithms (EAs) to the problem of automating the locomotion of computer-simulated creatures. We introduce *niching* as a way of maintaining genetic diversity and show that it results in the generation of a range of locomotion controllers and increases the probability of finding difficult or rare locomotion modes.

Keywords

Evolutionary algorithms, niching, virtual creatures, mass-spring systems, locomotion controllers.

1. INTRODUCTION

In constructing virtual worlds, one particular challenge is to create convincing virtual animal life. This paper is concerned with the problem of getting physically-based virtual creatures to move plausibly with a range of possible styles or *gaits*.

Virtual creatures move by means of virtual muscles, which are activated by *control signals* from a *locomotion controller*. For the purposes of this paper we can regard the controller as a multi-function evaluator, which receives inputs from the creature regarding its state, such as the height or velocity of a particular point on the creature or whether a particular portion of the creature is in contact with the ground. The evaluator computes the control signals for the virtual muscles as a function of its inputs using only arithmetic operators, standard functions like *abs* and *max*, and a few special time-dependent functions like *sine-wave* and *time-delay*. The details of how controllers are represented, namely as data flow graphs, are given in [Sanders 2000].

Our virtual creatures (Fig. 1) are implemented by simple 2D mass-spring systems of the sort described in [Witkin and Baraff 1997]. The virtual muscles are just springs with a variable rest length. However, the evolutionary algorithms that we describe for generating controllers should be applicable to any physically-based virtual creature model.

2. CONTROLLER SYNTHESIS

The focus of this paper is on the search method used to find effective efficient locomotion controllers within the vast set of possible controllers. Since most creatures will have several

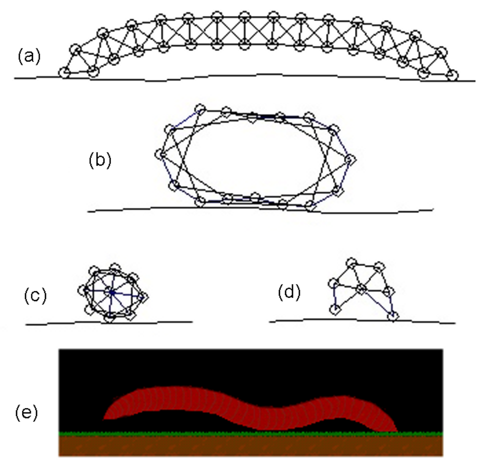

Figure 1. Some of our 2D mass-spring creatures in motion: (a) worm, (b) ring, (c) octagon, (d) biped, (e) texture-mapped worm.

different possible gaits, a general goal would be to find at least one controller for each possible gait. However, the concept of a gait is in general not well defined – for example, is a brisk walk a different gait from a slow amble? Hence a more realistic goal is to find a set of controllers that provides as wide a range of perceptually distinct locomotion styles as possible.

A variety of search methods such as simulated annealing [Tu and Terzopoulos 1994; Grzeszczuk and Terzopoulos 1995], stochastic gradient ascent [van de Panne and Fiume 1993] and evolutionary algorithms [Sims 1994] have been used to search for controllers for virtual creatures. None of that work has focused on the problem of generating a range of gaits. We chose to use evolutionary algorithms (EAs), adapted to find a set of distinct controllers rather than just a single controller.

3. SIMPLE EVOLUTIONARY ALGORITHMS

An evolutionary algorithm (EA) is a search method modelled on natural evolutionary processes. The search is formulated as an optimisation problem – we define a real-valued *fitness function* the value of which increases with the optimality of the solution candidate. In our case, a solution candidate is a description of a particular locomotion controller and the fitness function is a measure of how far the virtual creature moves over a fixed time during a physically-based simulation. We also include in the fitness function a factor that favours energy-efficient controllers over inefficient ones. The EA maintains a fixed-sized population of solution candidates, the optimality of which is improved over time through reproduction operations involving cloning, mutation, crossover and grafting; for details see [Sanders 2000]. Usually only the fittest members of a particular population are selected for reproduction to create the next generation; we call an EA in which the selection probability is based only on fitness a *Simple Evolutionary Algorithm* (SEA).

4. NICHING

A problem with SEAs is loss of genetic diversity. After several generations all members of the population become fairly similar – loosely, we can say they are all members of the same dominant breeding family. In recent years, *niching* has been proposed as a solution [Mahfoud 1995; Petrowski 1997]; it explicitly maintains genetic diversity by having a set of distinct *niches* within the population.

We implemented both an SEA and a niching-based evolutionary algorithm or *NEA*. Our NEA uses the clearing-based selection of [Petrowski 1997]: when selecting the set of "breeders" to form the new generation we select only individuals that are "sufficiently dissimilar" to those already selected. This requires that we have a measure of genetic distance between two individuals – we use a heuristic measure of the similarity of the two controller graphs – see [Sanders 2000].

5. RESULTS

We used an SEA to generate controllers for a number of 2D mass-spring creatures, including all those shown in Fig. 1. Two problems occurred. Firstly, the evolution process would sometimes become trapped in an inefficient local fitness maximum – inspection of the population showed that all individuals were similar and the local maximum prevented further improvement. Secondly, controllers were always of the "open loop" variety, i.e. they used signal generators to provide the time varying control signals rather than using sensor data to synchronize the signals with the actual motion of the animal. Only by removing signal generators from the set of allowable controller functions were we able to obtain such "closed loop" controllers. We attribute both these problems to the fact that an SEA rapidly loses genetic diversity.

When we moved to an NEA, both these problems disappeared: evolution never became trapped in low-quality local maxima and each evolution usually produced a population containing both open loop and closed controllers. The result is that qualitatively different locomotion styles or gaits were exhibited within the output population. For example, the octagon creature in Fig. 1 has a highly efficient closed-loop rolling gait that was found only by an NEA – the SEA could find only a range of "shuffling" gaits in which the creature moves by sliding sideways. Examples of the locomotion styles of our virtual creatures are on the web site [Sanders 2000].

Figure 2 contains two visualizations of the "birth" of a new generation for an SEA and an NEA after the evolutions have progressed for 100 generations. The lines connect children to their parents in a (blue, white, green) 3-space of (#neurons, fitness, #sensors) respectively. The visualizations show clearly that the SEA evolution has become homogenous. All children are being created from a group of near-identical parents near the top-left corner. The NEA on the other hand, exhibits considerable diversity. Note the explosions of outgoing child-links from the fittest candidates of each niche, and also the considerable variation in fitness of the niches. Subjectively, we observe that this diversity of controllers corresponds to a diversity of locomotion styles.

6. CONCLUSIONS

The locomotion controllers found by our evolutionary algorithms (EAs) produce smooth flowing movement. However, a simple EA sometimes fails to find *any* good controllers and is not well suited to the generation of a range of gaits.

The use of niching substantially improves the EA's performance. A single run of a niched EA generates a range of gaits and is able to locate difficult gaits that the simple EA never finds. Given that even our simple 2D creatures have significantly multimodal controller search spaces, we believe that a niching method should be used with *any* evolutionary approach to controller synthesis.

A full discussion of the ideas and results presented in this paper can be found in [Sanders 2000]. Initial results from this work were published in [Sanders et al. 2000].

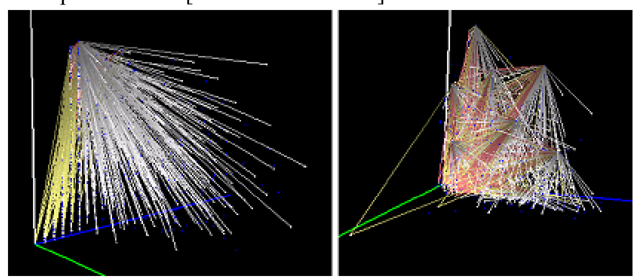

Figure 2. Visualizations of the birth of the one hundred and first generation with an SEA (left) and an NEA (right).

REFERENCES

GRZESZCZUK, R. and TERZOPOULOS, D. 1995. Automated Learning of Muscle-Actuated Locomotion through Control Abstraction. In *Proc. ACM SIGGRAPH 95*, pp. 63-70.

MAHFOUD, S. 1995. Niching Methods for Genetic Algorithms. PhD thesis, Illinois Genetic Algorithms Laboratory (IlliGAL), Dept of General Engineering, University of Illinois: Urbana, IL.

PETROWSKI, A. 1997. A New Selection Operator Dedicated to Speciation. In Th. Baeck, ed., *Proc. Seventh International Conference on Genetic Algorithms*, (ICGA97), Morgan Kaufmann, San Francisco, CA, pp 144-151.

SANDERS, M. 2000. Evolving Locomotion Controllers for Virtual Creatures. MSc thesis, Dept of Computer Science, University of Auckland, Auckland, New Zealand. http://www.cs.auckland.ac.nz/GG/michael_sanders.html.

SANDERS, M., LOBB, R., RIDDLE, P. 2000. Locomotion Controllers for Mass-Spring Creatures. In *Proc. Advances in Intelligent Systems, Theory and Applications*, Canberra, Australia, February 2000, pp 189-194.

SIMS, K. 1994. Evolving Virtual Creatures. In *Proc. ACM SIGGRAPH 94*, pp. 15-22.

TU, X. and D. TERZOPOULOS, 1994, Artificial Fishes: Physics, Locomotion, Perception, Behavior. In *Proc. ACM SIGGRAPH 94, pp. 43-50.*

VAN DE PANNE, M. and E. FIUME, E. 1993. Sensor-actuator Networks. In *Proc. ACM SIGGRAPH 93*, pp 335-342.

WITKIN, A. and D. BARAFF, 1997, Physically Based Modeling: Principles and Practice. ACM SIGGRAPH 1997 Course Notes. http://www-2.cs.cmu.edu/~baraff/sigcourse/.