TRADING-OFF MIND COMPLEXITY AND LOCOMOTION IN VIRTUALLY EMBODIED QUADRUPED ROBOT

Jason Teo, Hussein A. Abbass

Artificial Life and Adaptive Robotics (A.L.A.R.) Lab,
School of Computer Science, University of New South Wales,
Australian Defence Force Academy Campus, Canberra, Australia.
{j.teo,h.abbass@adfa.edu.au}

ABSTRACT

This paper investigates the use of a multi-objective approach for evolving artificial neural networks that act as controllers for the legged locomotion of a 3-dimensional, artificial quadruped creature simulated in a physics-based environment. The Pareto-frontier Differential Evolution (PDE) algorithm is used to generate a pareto optimal set of artificial neural networks that optimizes the conflicting objectives of maximizing locomotion behavior and minimizing neural network complexity. Here we also analyze the evolutionary dynamics of controller evolution.

Keywords: artificial evolution, artificial life, embodied cognitive science, evolutionary robotics.

1. INTRODUCTION

There has been a strong resurgence of research into the evolution of morphology and controller of physically simulated creatures. The pioneering and captivating work of Sims [19] in 1994 has not been parallelled until very recently. Further work in this area was limited by the complexity of programming a realistic physics-based environment and the steep computational resources required to run the artificial evolution. These physically realistic simulations of evolving artificial minds and bodies have become more accessible to the wider research community as a result of the recent convergence in the maturation of physics-based simulation packages and increase of raw computing power of personal computers [21].

Research in this area generally falls into two categories: (1) the evolution of controllers for creatures with fixed [10, 18] or parameterized morphologies [15, 17], and (2) the evolution of both the creatures' morphologies and controllers simultaneously [4, 9, 12, 14]. Some work has also been carried out in evolving morphology alone [6] and evolving morphology with a fixed controller [13]. Related work using mobile robots have also shown promising results in robustness and the ability to cope with changing environments by evolving plastic individuals that are able to adapt both through evolution and lifetime learning [7, 8, 16]. However, the artificial evolution conducted in these experiments focused on a single objective, for example walking, swimming, light-following, block pushing or obstacle avoidance. A better understanding of controller complexity and the behavior of evolved controllers should pave the way towards the emergence of more complex artificial creatures with more complex morphologies and behaviors.

In this paper, we investigate the use of a multi-objective approach in evolving controllers for a fixed morphology artificial creature. A multi-objective approach for evolving the controller of the creature allows for an investigation into the relationship between the capability of the evolved locomotion behavior and the size of the brain required for generating the desired behavior. By generating a pareto-frontier consisting of multiple ANNs with differing locomotion capabilities and varying architecture complexities, a comparison of controller size against behavior fitness can be made. A further advantage of using a multi-objective approach for artificial evolution is that genetic diversity is maintained naturally during the course of the evolutionary process. A common problem with evolutionary optimization algorithms is premature convergence due to loss of genetic diversity and this phenomenon has been observed to cause problems as well in the artificial evolution of virtual creatures [11]. An evolutionary multi-objective algorithm promotes reproductive diversity by allowing the evolutionary process to optimize along distinct goals.

This study will hopefully provide some insights into the architectural complexity of controllers required for generating walking behaviors in 3D, physically simulated creatures. In addition, it also provides a new paradigm for evolving controllers as a set of pareto optimal ANNs can be generated in a single run. This allows the user the option to choose from a variety of controllers with varying architectural complexities and behavioral competencies to suit the eventual simulation environment, constraints and purposes.

The artificial evolutionary system proceeds along two separate goals: to (1) maximize horizontal locomotion and, (2) minimize the complexity of the controller. In the current study, controller complexity is measured using the number of hidden nodes that are used in the ANN. In future work, we intend to define more rigorous measures of controller complexity by taking into consideration other ANN architectural features such as number of connection weights as well as number of nodes in the input and output layers.

2. METHODS

2.1. Evolving Artificial Neural Networks

Traditionally ANNs are trained using learning algorithms such as *backpropagation* (BP) to determine the connection weights between nodes. However such methods are gradient-based techniques which usually suffer from the inability to escape from local minima when attempting to optimize the connection weights. To overcome this problem, evolutionary approaches have been proposed as an alternative method for optimizing the connection weights. ANNs evolved via this method is thus referred to as evolutionary artificial neural networks (EANNs). In the literature, research into (EANNs) usually involves one of three approaches; evolving the weights of the network, evolving the architecture, or evolving both simultaneously. For a thorough review of EANNs, refer to [22]. In this paper, we are evolving both the weights and architecture of the ANN.

Abbass et al. first introduced the Pareto-frontier Differential Evolution (PDE) algorithm for vector optimization problems [3]. PDE is an adaptation of the original *Differential Evolution* (DE) algorithm introduced by Storn and Price [20] for optimization problems over continuous domains. In this initial investigation, the PDE algorithm outperformed all previous methods on five benchmark problems. PDE combined with local search was later introduced for evolving ANNs in the MPANN algorithm [1]. MPANN was found to be highly effective for knowledge discovery in databases. In subsequent work, the MPANN algorithm was empirically shown to possess better generalization in medical diagnosis of breast cancer whilst incurring a much lower computational cost [2].

2.2. Representation

Similar to [1, 2], our chromosome is a class that contains one matrix Ω of real numbers representing the weights of the artificial neural network and one vector ρ of binary numbers (one value for each hidden unit) to indicate if a hidden unit exists in the network or not; that is, it works as a switch to turn a hidden unit on or off. The sum of all values in this vector represents the actual number of hidden units in a network. This representation allows simultaneous training of the weights in the network and selecting a subset of hidden units. The morphogenesis of the chromosome into the ANN is depicted in Figure 1.

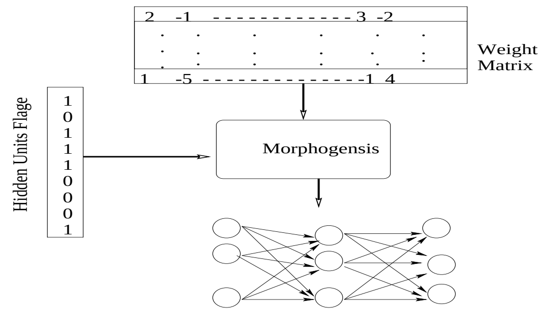

Figure 1: The representation used for the chromosome.

2.3. The PDE algorithm

We have a multi-objective problem with two objectives in this study: (1) one is to maximize the horizontal distance travelled by the creature from its initial starting position, and (2) to minimize the number of hidden units. The pareto-frontier of the tradeoff between the two objectives will have a set of networks with different number of hidden units and different locomotion behaviors. An entire set of controllers is generated in each evolutionary run without requiring any further modification of parameters by the user. The PDE algorithm for evolving ANNs consists of the following steps:

1. Create a random initial population of potential solutions. The elements of the weight matrix Ω are assigned random values according to a Gaussian distribution $N(0,1)$. The elements of the binary vector ρ are assigned the value 1 with probability 0.5 based on a randomly generated number according to a uniform distribution between $[0,1]$; otherwise 0.
2. Repeat
   (a) Evaluate the individuals in the population and label those who are non-dominated.
   (b) If the number of non-dominated individuals is less than 3 repeat the following until the number of non-dominated individuals is greater than or equal to 3:
      i. Find a non-dominated solution among those who are not labelled.
      ii. Label the solution as non-dominated.
   (c) Delete all dominated solutions from the population.
   (d) Repeat
      i. Select at random an individual as the main parent α_1, and two individuals, α_2, α_3 as supporting parents.
      ii. **Crossover:** with some probability $Uniform(0,1)$, do

$$\omega_{ih}^{child} \leftarrow \omega_{ih}^{\alpha_1} + N(0,1)(\omega_{ih}^{\alpha_2} - \omega_{ih}^{\alpha_3}) \quad (1)$$

$$\rho_h^{child} \leftarrow \begin{cases} 1 & if(\rho_h^{\alpha_1} + N(0,1)(\rho_h^{\alpha_2} - \rho_h^{\alpha_3})) \geq 0.5 \\ 0 & otherwise \end{cases} \quad (2)$$

otherwise

$$\omega_{ih}^{child} \leftarrow \omega_{ih}^{\alpha_1} \quad (3)$$

$$\rho_h^{child} \leftarrow \rho_h^{\alpha_1} \quad (4)$$

and with some probability $Uniform(0,1)$, do

$$\omega_{ho}^{child} \leftarrow \omega_{ho}^{\alpha_1} + N(0,1)(\omega_{ho}^{\alpha_2} - \omega_{ho}^{\alpha_3}) \quad (5)$$

otherwise

$$\omega_{ho}^{child} \leftarrow \omega_{ho}^{\alpha_1} \quad (6)$$

where each weight in the main parent is perturbed by adding to it a ratio, $F \in N(0,1)$, of the difference between the two values of this variable in the two supporting parents. At least one variable must be changed.

iii. **Mutation:** with some probability $Uniform(0,1)$, do

$$\omega_{ih}^{child} \leftarrow \omega_{ih}^{child} + N(0, mutation\_rate) \quad (7)$$

$$\omega_{ho}^{child} \leftarrow \omega_{ho}^{child} + N(0, mutation\_rate) \quad (8)$$

$$\rho_h^{child} \leftarrow \begin{cases} 1 & if \rho_h^{child} = 0 \\ 0 & otherwise \end{cases} \quad (9)$$

(e) Until the population size is M

3. Until maximum number of generations is reached.

3. EXPERIMENTS

3.1. The Simulation Model

The simulation is carried out in a physically realistic environment which allows for rich dynamical interactions to occur between the creature and its environment. This in turn enables complex walking behaviors to emerge as the creature evolves the use of its sensors to control the actuators in its limbs through dynamical interactions with the environment. Furthermore, the accurate modelling of the simulation environment plays a crucial part in producing artificial creatures that move and behave realistically in 3D [21]. A dynamic rather than kinematic approach is paramount in allowing for effective artificial evolution to occur. Physical properties such as forces, torques, inertia, friction, restitution and damping need to be incorporated into the artificial evolutionary system. To this end, the Vortex physics engine [5] was employed to generate the physically realistic artificial creature and its simulation environment. A screen capture of the quadruped moving in its environment is shown in Figure 2.

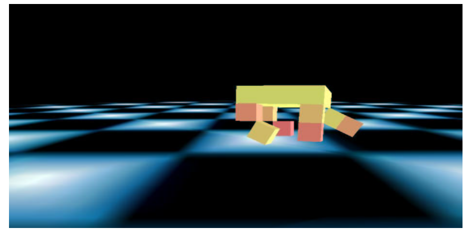

Figure 2: Screen capture of quadruped in the simulation environment.

The artificial creature is a basic quadruped with 4 short legs. Each leg consists of an upper limb connected to a lower limb via a hinge (one degree-of-freedom) joint and is in turn connected to the torso via another hinge joint. The mass of the torso is 1kg and each of the limbs is 0.5kg. The torso has dimensions of 4 x 1 x 4m and each of the limbs has dimensions of 1 x 1 x 1m. The hinge joints are allowed to rotate between -1.57 to 0 radians for limbs that move counter-clockwise and 0 to 1.57 radians for limbs that move clockwise from their original starting positions. Each of the hinge joints are actuated by a motor that generates a torque producing rotation of the connected body parts about that hinge joint. Correspondingly, the artificial creature has 12 sensors and 8 actuators. The 12 sensors consist of 8 joint angle sensors $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$ corresponding to each of the hinge joints and 4 touch sensors $(x_9, x_{10}, x_{11}, x_{12})$ corresponding to each of the 4 lower limbs of each leg. The 8 actuators $(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8)$ represent the motors that control each of the 8 articulated joints of the creature. These motors are controlled via outputs generated from the ANN controller which is then used to set the desired velocity of rotation of the connected body parts about that joint.

3.2. Experimental Setup

A total of 480 evolutionary runs were conducted with varying population sizes, crossover rates, and mutation rates while fixing the fitness evaluation window to 500 timesteps. The crossover rate used were 0, 0.1, 0.2, 0.5 and 1 and the mutation rates used were also 0, 0.1, 0.2, 0.5 and 1 (the evolutionary setup with a crossover rate of 0 and a mutation rate of 0 was omitted since this setup does not generate any variability at all in the population). The maximum number of hidden units permitted in evolving the artificial neural network was fixed at 15 nodes. Each experimental setup was repeated using 10 different seeds to allow the artificial evolution to commence from different starting points in the search space. The number of generations and population size were fixed at 20 and 30 respectively for the first set of runs. In the second set of runs, these parameter values were reversed to 30 for number of generations and 20 for population size to enable a fair comparison between the effect of the two population sizes (the total number of genotypes over the entire span of the evolutionary process was kept constant at 600 genotypes in both these setups).

4. RESULTS AND DISCUSSION

4.1. Evolutionary Parameters

First we analyzed the effect of population size on the evolved locomotion behaviors. Overall, there did not appear to be any obvious differences in the range and quality of the evolved controllers between population sizes of 20 and 30. Both produced a considerably similar quality of locomotion behaviors although a larger population size did seem to produce controllers that were slightly better in terms of average locomotion fitness. There were 12 different combinations of crossover and mutation rates with a population size of 30 in which the best average locomotion fitness exceeded 2.5m as compared to only 8 with a population size of 20. Both also generated a relatively similar spread of locomotion behaviors although again a larger population size did seem to produce more varied genotypes in terms of the number of hidden units that were used in the ANN. There were 12 different combinations of crossover and mutation rates with a population size of 30 that produced 11 or more different ANN architectures compared to only 10 with a population size of 20. As such, there is a very slight advantage in using a larger population size in terms of quality and spread of the locomotion behaviors.

Different combinations of crossover and mutation rates did appear to produce results that varied across two broad spectrums. With both population sizes of 20 and 30, two distinct groups of controllers were generated through the evolutionary process: (1) runs that produced high quality solutions but with a low spread of genotypes, and (2) runs that produced mediocre solutions with a high spread of genotypes. Again, the quality of the solutions refers to the average locomotion fitness and the spread of genotypes refers to the number of ANNs with different sizes in terms of hidden units. The first group of pareto optimal solutions with high quality and low spread were observed when fairly low mutation rates of 0.1 and 0.2 were used in combination with a low to medium crossover rate of between 0.1 to 0.5. The second group of pareto optimal solutions with lower quality but with a much wider spread of controller sizes were observed when a high mutation rate of 1 was used. In this latter case, crossover did not seem to affect the spread very much as all rates between 0 and 1 generated between 13 and 15 different genotypes. One exception to this observation was noticed in the run with a population size of 20 that had a mutation rate of 0 but still produced a wide spread of 14 controllers with different sizes. This may be explained by the very high

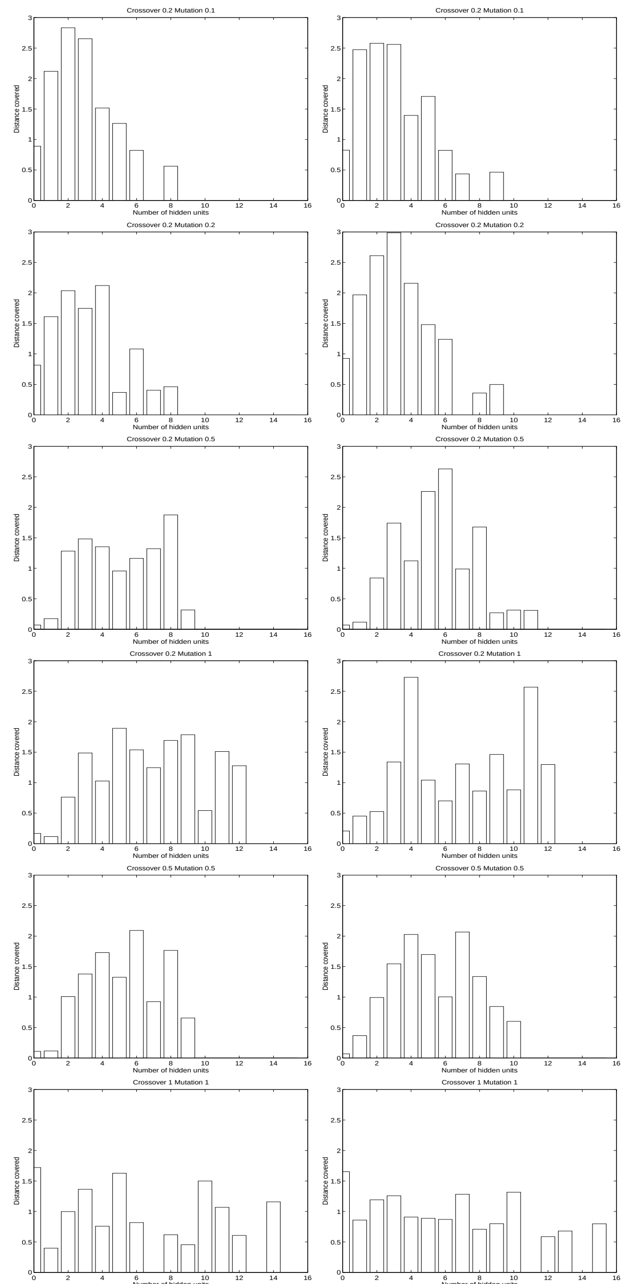

Figure 3: The average fitness for each number of hidden units with population sizes 20 (on left) and 30 (on right). Crossover and mutation combinations plotted from top to bottom are (0.2,0.1), (0.2,0.2), (0.2,0.5), (0.2,1.0), (0.5,0.5), and (1.0,1.0)

crossover rate of 1 which compensated for the non-existence of mutation. Another interesting phenomenon that could be observed only in the runs which used a population size of 30 was the ability of the evolutionary process to generate both high quality solutions (where the best average locomotion fitness was above 2.5m) as well as a large spread of 13 different controller sizes. This was achieved with a mutation rate of 1 and crossover rates of 0.2 and 0.5.

In summary, high quality and a low spread of solutions were obtained with low mutation and low to medium crossover whereas mediocre solutions with a wider variety of controller sizes were obtained with high mutation and low to high crossover Population size did not appear to significantly affect the quality and spread of pareto optimal solutions in these experiments although a very slight advantage in terms of quality and variety of controller sizes was observed with the larger population size of 30.

4.2. Evolutionary Dynamics

The best evolved controller in terms of the maximum horizontal distance moved from its initial position had a comparatively simple architecture with only 4 hidden units. This result was achieved with an evolutionary run that had similarly low crossover and mutation rates of 0.2 with a population size of 30 over 20 generations. To enable an analysis of the evolutionary dynamics that generated the best controller, the pareto-frontier of this particular setup is reported at each generation and is depicted graphically in Figure 4.

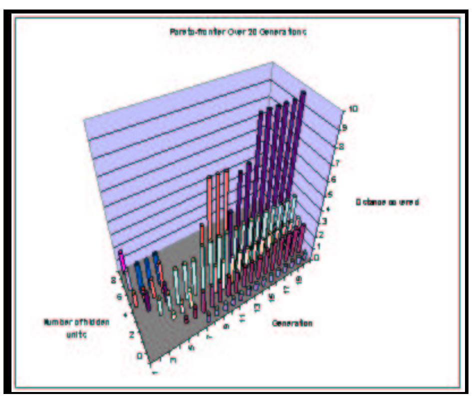

Figure 4: Pareto-frontier over 20 generations

Looking at the 1st generation, we observe a fairly even spread across different controller complexities ranging from 5 to 9 hidden units and which had similarly low locomotion capabilities. By the 2nd generation, we begin to see the effects of evolutionary pressure attempting to minimize the controller's complexity where the range of hidden units is reduced to between 4 and 6. An increase in genetic diversity is noticed in the 4th generation where five pareto optimal solutions were found. A sharp increase in locomotion capability and decrease in controller complexity is observed in the 5th generation. As a result of the strong evolutionary pressure to decrease the size of the ANN, a random controller with no hidden units appears in the 7th generation but does not achieve very much in terms of movement. Again genetic diversity emerges in the ninth generation with the reappearance of genotypes with 2 and 5 hidden units from previous generations that were lost during the reproduction process. The evolutionary process jumps to a higher fitness value in the 10th generation and this is where we start to see the optimization process begin to converge. There is no improvement at all in the 11th generation and the 12th generation only sees the addition of a single new genotype to the pareto-frontier. The only significant improvement between the 13th and 15th generations is in the ANN with 4 hidden units which increases its distance travelled by approximately 2m. The last significant though relatively small improvement comes in the last generation where the locomotion fitness of the ANN with 4 hidden units approaches 10.

Overall we see from the evolutionary dynamics of controller evolution that it is generally very hard for larger controllers with more hidden units to survive due to the strong evolutionary pressure of minimizing ANN complexity. This observation may also be attributed to the fact that a larger controller does not easily lead to locomotion behaviors that can't be achieved with a smaller controller. As a result, larger controllers find it hard to compete with smaller controllers in trying to maximize the horizontal distance travelled by the quadruped.

5. CONCLUSION

We have demonstrated a multi-objective approach to evolving artificial neural networks for controlling the locomotion of a 3D, physically simulated artificial creature. The pareto-frontier that resulted from each single evolutionary run provided a set of ANNs which maximized the locomotion capabilities of the creature and at the same time minimized the size of the controller. The evolutionary dynamics for controller synthesis were analyzed to provide a high-level view of the progression of the artificial evolution. For future work, we intend to investigate the effects of controller complexity when both the morphology and controller are co-evolved simultaneously.

6. REFERENCES

[1] Hussein A. Abbass. A memetic pareto evolutionary approach to artificial neural networks. In M. Stumptner, D. Corbett, and M. Brooks, editors, *Proceedings of the 14th Australian Joint Conference on Artificial Intelligence (AI'01)*, pages 1–12, Berlin, 2001. Springer-Verlag.

[2] Hussein A. Abbass. An evolutionary artificial neural network approach for breast cancer diagnosis. *Accepted to appear in Artificial Intelligence in Medicine*, 2002.

[3] Hussein A. Abbass, Ruhul Sarker, and Charles Newton. PDE: A pareto-frontier differential evolution approach for multi-objective optimization problems. In *Proceedings of the 2001 Congress on Evolutionary Computation (CEC2001)*, volume 2, pages 971–978, Piscataway, NJ, 2001. IEEE Press.

[4] Josh C. Bongard and Rolf Pfeifer. Repeated structure and dissociation of genotypic and phenotypic complexity in artificial ontogeny. In L. Spector, E. Goodman, A. Wu, W.B. Langdon, H.M Voight, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, and E. Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 829–836, San Francisco, 2001. Morgan Kauffman.

[5] Critical Mass Labs. Vortex. http://www.cm-labs.com, 2002.

[6] Peter Eggenberger. Evolving morphologies of simulated 3D organisms based on differential gene expression. In P. Husbands and I. Harvey, editors, *Proceedings of the 4th European Conference on Artificial Life (ECAL97)*, pages 205–213, Cambridge, MA, 1997. MIT Press.

[7] Dario Floreano and Francesco Mondada. Evolutionary neurocontrollers for autonomous mobile robots. *Neural Networks*, 11:1461–1478, 1998.

[8] Dario Floreano and Joseba Urzelai. Evolutionary robotics: The next generation. In T. Gomi, editor, *Proceedings of Evolutionary Robotics III*, pages 231–266, Ontario, 2000. AAI Books.

[9] Gregory S. Hornby and Jordan B. Pollack. Body-brain coevolution using L-systems as a generative encoding. In L. Spector, E. Goodman, A. Wu, W.B. Langdon, H.M Voight, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, and E. Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 868–875, San Francisco, 2001. Morgan Kauffman.

[10] Auke J. Ijspeert. A 3-D biomechanical model of the salamander. In J.-C. Heudin, editor, *Proceedings of the 2nd International Conference on Virtual Worlds (VW 2000)*, pages 225–234, Berlin, 2000. Springer-Verlag.

[11] Maciej Komosinski and Adam Rotaru-Varga. Comparison of different genotype encodings for simulated three-dimensional agents. *Artificial Life*, 7(4):395–418, 2001.

[12] Maciej Komosinski and Szymon Ulatowski. Framsticks: Towards a simulation of a nature-like world, creatures and evolution. In D. Floreano, J.-D. Nicoud, and F. Mondada, editors, *Advances in Artificial Life: Proceedings of the 5th European Conference on Artificial Life (ECAL99)*, pages 261–265, Berlin, 1999. Springer-Verlag.

[13] Lukas Lichtensteiger and Peter Eggenberger. Evolving the morphology of a compound eye on a robot. In *Proceedings of the Third European Workshop on Advanced Mobile Robots (Eurobot '99)*, pages 127–134, Piscataway, NJ, 1999. IEEE Press.

[14] Hod Lipson and Jordan B. Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406:974–978, 2000.

[15] Henrik H. Lund, John Hallam, and Wei-Po Lee. Evolving robot morphology. In *Proceedings of the 4th IEEE International Conference on Evolutionary Computation*, pages 197–202, Piscataway, NJ, 1997. IEEE Press.

[16] Stefano Nolfi and Dario Floreano. Learning and evolution. *Autonomous Robots*, 7(1):89–113, 1998.

[17] Chandana Paul and Josh C. Bongard. The road less travelled: Morphology in the optimization of biped robot locomotion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2001)*, volume 1, pages 226–232, Piscataway, NJ, 2001. IEEE Press.

[18] Richard Reeve. *Generating Walking Behaviors in Legged Robots*. Unpublished PhD thesis, University of Edinburgh, Scotland, 1999.

[19] Karl Sims. Evolving 3D morphology and behavior by competition. In R. Brooks and P. Maes, editors, *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, pages 28–39, Cambridge, MA, 1994. MIT Press.

[20] Rainer Storn and Kenneth Price. Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, International Computer Science Institute, Berkeley, 1995.

[21] Tim Taylor and Colm Massey. Recent developments in the evolution of morphologies and controllers for physically simulated creatures. *Artificial Life*, 7(1):77–87, 2001.

[22] Xin Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1426–1447, 1999.