Analysing the Evolvability of Neural Network Agents Through Structural Mutations

Ehud Schlessinger[1], Peter J. Bentley[2], and R. Beau Lotto[1]

[1] Institute of Ophthalmology, University College London, 11-43 Bath Street, London EC1V 9EL
{e.schlessinger, lotto}@ucl.ac.uk
[2] Department of Computer Science, University College London, Malet Place, London WC1E 6BT
P.Bentley@cs.ucl.ac.uk

Abstract. This paper investigates evolvability of artificial neural networks within an artificial life environment. Five different structural mutations are investigated, including adaptive evolution, structure duplication, and incremental changes. The total evolvability indicator, E_{total}, and the evolvability function through time, are calculated in each instance, in addition to other functional attributes of the system. The results indicate that incremental modifications to networks, and incorporating an adaptive element into the evolution process itself, significantly increases neural network evolvability within open-ended artificial life simulations.

1 Introduction

Understanding the causal relationship between the functional structure of adaptive neural networks and ecological history is a fundamental objective in neuroscience research. And an important method for addressing this challenge is to model artificial neural networks within open-ended, ecologically relevant Artificial Life environments. The problem, however, is that standard neural network training algorithms (such as back-propagation) cannot be used in this paradigm, as such they do not offer the complete set of input and output values needed for training. An alternative is to therefore use the same mechanism nature does – evolution through 'natural selection'.

There are, however, many issues to consider when evolving neural networks. Fundamental to any of these is the evolution of network topology (i.e., modifications to its underlying node and connection structure). Here we address the question of the *process* by which network elements are to be added (and removed) by focusing, not on evolved network solutions as such, but on the evolvability of the systems itself.

Evolvability is the ability of a population to produce offspring fitter than any yet in existence [1], and not to produce less fit variants [13], and is therefore fundamental to the process of evolution itself. Evolvability is also known as evolutionary adaptability [8] and as such, a major element of evolvability is the capacity to adapt to changing environments by learning to exploit commonalities over time in those environments. By understanding evolvability and how to promote it, not only will it be possible to

M. Capcarrere et al. (Eds.): ECAL 2005, LNAI 3630, pp. 312 – 321, 2005.
© Springer-Verlag Berlin Heidelberg 2005

solve increasingly complex problems, but one may also better understand evolution of network systems generally.

The key properties required to generate systems exhibiting high evolvability are not well understood, particularly in the context of ecologically relevant artificial life simulations. Nonetheless, several factors are thought to be correlated with high evolvability.

(1) The mapping of genetic variation onto phenotypic variation [4, 15], and the selection of search operators used, determine the distribution of local optima in the search space, and affect search difficulty [1, 7]. More specifically, a many-to-one genotype-to-phenotype mapping (a redundant mapping), is essential for evolvability. By enabling some mutations to be phenotypically irrelevant, it is possible to better explore the search space through neutral networks [5]. Evolution of neural networks, in our view, particularly those that are used for control and classification, qualifies for the complex mapping condition; Fogel [6] defined an evolved neural network's phenotype as its behaviour, and not its constituent weights. Using this definition, changing many aspects of a neural network would not necessarily change its phenotype (behaviour).
(2) Gradual effects of the search operators seem to play an important part [2, 9].
(3) Structural duplication and modularity are recognised as promoters of evolvability [17], as they enable evolution to 'reuse' structures within networks [10].
(4) Finally, the ability of evolution of adapting elements of itself can also promote evolvability, since it enables evolution to differentially tune search operators throughout evolution [4, 7].

This paper analyses the effect on the evolvability of an artificial life simulation, as measured by the evolvability indicator, E_{total}, using five different types of structural mutations. Each of the mutation types incorporates the various principles described above for increasing evolvability. Secondary effects on evolved traits were also measured, such as the number of successful 'runs', quality of evolved solutions, and the variability of the evolved forms.

2 System

Mosaic World is an A-Life system designed for exploring the computational principles by which vision can overcome stimulus ambiguity [12]. Mosaic World offers a virtual environment made up of a 2D grid of 'coloured' surfaces under multiple simulated light sources. This environment emulates key characteristics of natural scenes. The space is inhabited by virtual agents, 'critters', that survive by consuming positive resources and avoiding negative resources. Every surface's value is determined from its reflectance – its colour. Consumed resources slowly regenerate.

The critter population is maintained by the critter reproduction. Critters can reproduce both sexually and asexually. In the event that all critters perish, a new population is created, where 80% are random critters and the rest are mutated clones of critters that showed general promising survival skills earlier in the run. Every critter starts out with a certain amount of energy, and dies if it runs out of energy. Critters slowly lose

energy over time, or due to moving, turning, resource consumption and reproducing. A critter dies if it steps over the edge of the world, or into a hole.

Critter behaviour (such as mating, eating, and moving) is determined by the output of a modified 3D feed-forward neural network. Network topology is determined by the critter's genome, though its behaviour is an emergent property of the interaction between the nodes within this topology. The input layer contains receptors (which are input units modified to enable evolution of vision) and a health monitor unit, which receives the percentage of the critter's remaining health. The hidden layer contains standard hidden units. The output layer contains standard output units, which determine the critter's behaviour. Every unit in the network has an [x,y] coordinate relative to the critter's centre, which defines its location in its layer. Using the layer and the [x,y] coordinate, networks of different architectures can be crossed over during sexual reproduction, as each network possesses the same coordinate reference frame.

The units of the network communicate through connections that can extend between units from higher layers to lower layers, and can also connect units to empty coordinates in the network (unconnected connections). Connections can be active or inactive. Only active connections participate in the feed-forward process.

2.1 Mutation Operators

In this work, we focus on an investigation of evolvability using several types of structural mutations. A full description of the non-structural mutations is in [12].

For a mutation type to be useable, it must have the ability to completely alter a neural network's structure by adding and deleting elements. In order that we are able to test the effects of suggested principles thought to increase evolvability, every mutation type used in our experiments incorporated some of these principles. The three principles tested are: *incremental changes to network topology*, where every change done to the network structure is very small, *adaptive evolution,* where evolution can modify some aspects of itself, and *structural duplication*, where existing substructures of the network are copied and can be reused.

Deletion and addition of units (receptors, hidden) are performed using *Delete Unit* (0.5% per unit) and *Add Unit* mutations (2%). Deletion and addition of connection weights are performed using *Delete Connection* (0.1% per connection) and *Add Connection* mutations (1%). When a unit is added, it is randomly placed in the appropriate layer with a bias towards the centre and forms connections with units in the adjacent layers. All new connections are initialised with random values. When a unit is removed, all its outgoing connections are removed. If, as a result of a unit being deleted, a connection now has no end destination, it remains in the network. These connections are termed 'unconnected connections', and as such are not used in feed-forward processing, only becoming functional again if the old unit is replaced.

Receptors in the input layer can change locations through *Drift* mutation (0.3% per receptor). *Switch* mutations (0.3% per element) can cause a connection or a receptor to become active or inactive. An inactive element is not used in the feed-forward process, and is deleted from the genome when a number of generations have not activated it again.

The above probabilities were empirically determined to be suitable during the course of 15 experiments (roughly 750 runs).

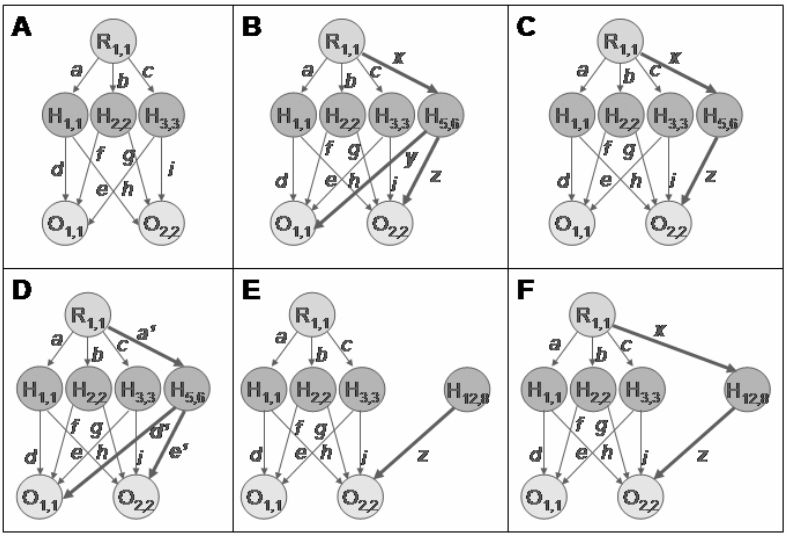

Fig. 1. Illustrating addition of a hidden unit using the five types of mutations. [A] The original neural network with 1 receptor, 3 hidden units, and 2 output units. [B] Using mutation type (i), new unit (H5,6) is fully connected through 3 random connections. [C] Using mutation type (ii), new unit (H5,6) connects to (R1,1) and (O2,2). [D] Using mutation type (iii), new unit (H5,6) is a clone of (H1,1). [E] Using mutation type (iv) new unit (H12,8) only connects to (O2,2) as the distance parameter is very high. [F] Using mutation type (v) new unit (H12,8) connects to the closest receptor (R1,1) and closest output unit (O2,2).

The following types of structural mutations were used in the experiments (see fig. 1). The probabilities of these mutations occurring are identical for all types. The tested principle appears in parenthesis.

Type 1 - fully connected (non-gradual changes): New units connect to **all** units in adjacent layers. Using this mechanism, every mutation makes a large change to the networks.

Type 2 - single connection (gradual changes): New units connect to a **single**, randomly chosen, unit in every adjacent layer. The *Delete Unit* mechanism is disabled – units are automatically removed when they have no outgoing or no incoming connections. Using this mechanism, every mutation makes a small change to the network.

Type 3 – reuse of structures (structural duplication): Added units are cloned from a random unit in the same layer. The new unit possesses a copy of every incoming and outgoing connection of the original.

Type 4 – distance dependent (adaptive evolution, gradual changes): Added units connect to **all** units in adjacent layers within a given distance. The distance parameter is an evolvable gene of a critter. By evolving a low distance parameter, the change to the network can be very small or very large.

Type 5 – shortest connection (adaptive evolution, gradual changes): Added units connect to the **closest** unit in every adjacent layer. The *Delete Unit* mechanism is disabled – units are automatically removed when they have no outgoing or no incom-

ing connections. Using this mechanism, every mutation makes a small change to the network. Additionally, evolution can now utilise the 3D coordinate system to create modules, which adds an adaptive element (albeit weaker than type 2).

2.2 Measuring Evolvability in Mosaic World

Mosaic World is more than just a population of individual critters – it is a dynamic ecosystem in which critters survive if their genomes enable them to interact with each other and their current environment effectively enough to gather resources [12].

Previously suggested measurements of evolvability [1, 13] do not take into account conditions specific to the ecologically relevant conditions of Mosaic World, and as a result they could not be used. These methods require accurately measuring fitness, which is not feasible for three reasons: First, no one statistic encapsulates all the required behaviours a critter must know to be termed fit. Second, the fitness of all critters is linked, as critters compete against each other on resources; a fit critter, effectively, decreases the fitness of other critters. Third, although reproduction does not directly contribute to a critter's fitness, controlling reproduction is crucial to the species' collective fitness: The population, as a whole, must replenish itself at a rate that is sustainable by the available resources of the world. Thus, a critter must share some of this collective fitness.

Therefore, the evolvability measurement we use here is based on the evolvability used in the Avida ALife environment [11]. This measurement was expanded by factoring environment difficulty. We believe that evolvability can either be expressed by demonstrating that a population gradually improves over time, or alternatively, by showing a population adapting to an environment that gradually becomes more challenging. By quantifying these aspects, we define the total evolvability indicator in Mosaic World, E_{total}, using equation (1) – its range of possible values is 0 to 1, and the *evolvability function through time*, using equation (2). Both measures incorporate four different elements: survivability, population success, environment difficulty and time variance.

Survivability: The critter's survival ability is best expressed by its age. A critter that can survive for long obviously managed to learn important skills required to survive in the world. Furthermore, by surviving longer, a critter may get more opportunities to reproduce and as a result spread fit genetic material to its offspring.

Population success: A population's 'fitness' is best expressed by its size at a given time. A population that managed to maintain itself through time, collectively learned how to balance resource consumption and reproduction through its constituent critters. Also, a larger population has more individuals that pass on traits to offspring, and is more likely to survive a 'catastrophe' purely because of its greater size.

World difficulty: In many experiments the environment is altered over time to make it more challenging for a critter to survive. A population that manages to survive under conditions in which the selection pressure continuously grows, shows an indication of adaptability, and thus, evolvability. This aspect of the equation is controllable by the researcher and must be directly tied in, from a numerical point of view, to the difficulty of the world in order to measure evolvability, i.e. if survival in the world at time t is twice as hard as the initial conditions, the difficulty factor at time t is 2.

Time: Only by looking at the relative changes of survivability, population success and world difficulty over time, we can precisely obtain the total evolvability measure.

In conclusion, these four elements measure the capacity of Mosaic World's population to evolve. A population that maintains large numbers, where each agent survives for long, in an increasingly difficult environment, consistently through time – can be said to be a population with a great capacity to evolve. Therefore, this function measures the capacity of a population to generate fit offspring through time.

$$E(t) = \frac{D(t)}{D_{\max}} \frac{\sum_{i=0}^{P_t} \left(\frac{A_{t,i}}{A_{\max}} \right)}{P_{\max}} \quad (1)$$

$$E_{total} = \frac{\sum_{i=0}^{t} E(t)}{t_{\max}} \quad (2)$$

$$Resilience = \frac{\sum_{i=0}^{t} iE(i) - n\overline{E(t)}\bar{t}}{\sum_{i=0}^{t} i^2 - n(\bar{t})^2} \quad (3)$$

$$Stamina = \overline{E(t)} - \text{Resilience}\bar{t} \quad (4)$$

Where: E_{total} is a population's evolvability indicator, E(t) is the evolvability at time t, D(t) is the difficulty factor at time t, D_{max} is the maximal difficulty of D(t), P_t is the size of the population at time t, $A_{t,p}$ is the age of a member of population p at time t, A_{max} is the critter maximum life span, P_{max} is the maximal population the environment can support, t_{max} is the total length of time of the experiment, n is the number of data values.

Example: With a population size *P* of 400 at time 10000, all critter ages *A* are 1500, the difficulty factor *D* at time 10000 is 100, using maximum difficulty D_{max} of 350, maximum population size P_{max} of 10000, and maximum age A_{max} of 15000, evolvability at time 10000 is E(10000)=100/350*(400*1500/15000)/10000=0.00114.

By extracting the height and the slope of a linear trendline of the evolvability function through time (using equations (3) and (4)), we gain two extra indicators: (i) *Resilience (slope)*: Defines the resilience of the population to change. Lower values indicate populations more tolerant to change. (ii) *Stamina (height)*: Defines the population's ability to thrive when conditions are easy.

3 Experiments

The main objective of the experiments was to measure the evolvability function through time, E(t), and the total evolvability, E_{total}. A secondary objective was to obtain additional statistics examining effects other than evolvability of the structural mutations used: Variability of evolved forms (average structure), quality of critter solutions and the percentage of successful runs (a run failed when no population of critters evolved without the need for a restart).

To this end, two sets of experiments were performed. Each of the experiments required multiple populations that were evolved using the five structural mutations. Therefore, at least eight successfully evolved populations were collected for each of the mutation types (using the same randomly generated world). Each run started with identical population characteristics (all critters possessing fully connected networks: 3 receptors, 3 hidden units and 8 output units, 33 connections), and was stopped after 550,000 time steps. During each run, the regeneration rate of consumed surfaces was slowly reduced to increase challenge and force critter populations to adapt. Initially, consumed surfaces regenerated every 13 time steps 3% of their maximal value. Every 3,500 time steps regeneration slowed down by one unit, until the regeneration rate of 99 was reached. To analyse the effects of the mutation operators only, crossover was disabled during all runs and experiments.

Experiment 1 - Measuring Evolvability through Adaptation: This experiment attempted to test the maximum difficulty that a population can adapt to. Using the collected data and equations (1) and (2), E(t) was charted and E_{total} was calculated. Since the regeneration rate has a direct effect on the difficulty of the world, the rate was used as the difficulty factor in equation (1). Therefore, five copies of the five longest-lived critters of every evolved population were placed in an identical test world. The starting regeneration rate was set to 99, and every 1,000 time steps the it slowed down by one unit, indefinitely. A run was finished when all critters died.

Experiment 2 - Measuring the Quality of Evolved Solutions. This experiment attempted to measure the quality of evolved solutions, the critters. The criterion used was critter survivability, which was measured by averaging the critter survival ages across runs. To do this accurately, the effect of the critters on each other was negated by prohibiting reproduction, and by placing a very small number of critters in every world. Furthermore, the difficulty of the world was made static by fixing the regeneration rate (to 99). Therefore, five copies of the five longest lived critters of every run were placed in an identical test world. Critters were expected to survive as long as they could. All runs were stopped after 10,000 time steps, and were repeated 3 times. Critters that survived until the end of the run were assumed to have died then.

4 Results

In table 1, we see E_{total} for each type (as a percentage of the maximum E_{total} of type 4), and the resilience and stamina for each type (using equations (3) and (4) and divided by type 4's resilience for comparison purposes). In fig. 2, we see the evolvability function (weighted average) through time with E_{total} appearing in the legends for every type. Table 2 shows the minimum, maximum and average of the maximum regeneration rate a population could adapt to and of critter average survival age, as well as the percentage of successful runs and the average critter structure per type.

When comparing the E_{total} of all types, it is clear that adaptive evolution and gradual changes to networks increase E_{total}, whereas non-gradual changes, and structural duplication decrease it. Types 4 and 5, both utilising adaptive evolution and gradual

Table 1. The evolvability elements incorporated, the obtained E_{total} (as a percentage of E_{total} of type 4) and the extracted resilience and stamina values using a linear trendline of E(t) for every type (divided by type 4's resilience for comparison purposes)

Mutation type	Element Incorporated	E_{total} (%)	Resilience	Stamina
4	Adaptive evolution, Gradual changes	100.00%	-1	5.68
5	Adaptive evolution, Gradual changes	98.12%	-1.13	6.39
2	Gradual changes	78.50%	-0.98	5.53
1	Non-gradual changes	71.47%	-0.94	5.29
3	Structural duplication	41.58%	-0.41	2.34

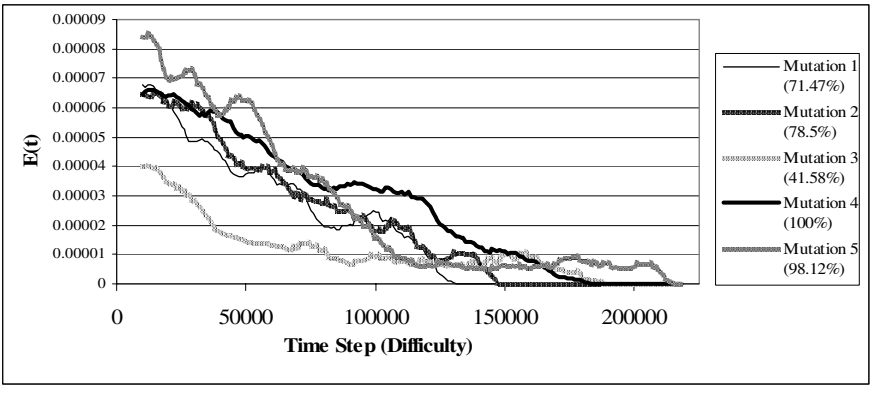

Fig. 2. The evolvability function (weighted average) for the five types of structural mutations and their relative evolvability indicator (of E_{total} for mutation type 4)

Table 2. Several statistics (average, min, max) describing the maximum regeneration rates the tested populations adapted to and the critter survivability, in addition to the average critter structure, and percentage of successful runs; broken down according to mutation types

Mutation type	Maximum adapted regeneration rate: Ave. (Min.-Max)	Survival age: Ave. (Min.-Max.)	Ave. critter structure: Receptors, Hidden (Connections)	Successful runs (%)
Random Critter		57.36 (56.08-59.48)	3, 3 (33)	
1	191.14 (119–222)	3182.37 (1277.23-4600.12)	4.03, 3.13 (29.47)	64%
2	197.12 (159–237)	3733.34 (2781.13-4801.6)	8.32, 10.74 (108.70)	73%
3	163.87 (109–277)	2388.49 (893.44-5339.6)	4.86, 4.51 (41.45)	50%
4	224.36 (171–272)	3905.31 (1625.16-5021.96)	4.98, 6.26 (55.48)	69%
5	202.62 (167–305)	3651.06 (2613.92-5321.28)	10.39, 12.21 (144.25)	62%

changes, had the highest E_{total} with type 4 the higher of the two. The difference in their evolvability functions were, however, significantly different: Type 5 had – on average – a higher stamina, but it was less resilient than type 4, and its populations quickly weakened as difficulty increases. Type 4 was more resilient, as evident in its average adaptation rate. Overall, the data suggests that the Type 4 structural mutation is slightly more evolvable [note that Type 4's average survival age was also the best of all runs; Type 5's was lower, but still very good]. It could be said, however, that Type 5, having a higher stamina, and lasting the longest in our adaptation experiment, is the most evolvable type. However, we believe the total area under the curve is the best indication of evolvability, since this measure takes into account both stamina and resilience.

Type 2, causing only small increments to the network, had a higher E_{total} than the Type 1's. It also had the best average survival age, and best rates of success. Despite its populations' decent performance, once the environment becomes too challenging, however, it its evolvability decreases significantly, causing its populations to perish.

Type 1, causing large changes to the network, had mediocre statistics and a low E_{total}. Generally, it seemed unable to utilise the structural mutations: on average, only one receptor, and no hidden units, were added at all. We believe this is another gauge of its low evolvability.

Type 3, utilising structural duplication, had the lowest E_{total} as well as the lowest scores on all other tests. It would be easy to dismiss this method of evolution as completely non evolvable, except for the fact that, despite having the low results of the vast majority of type 3's runs, some of its individual runs scored the *highest* average survival age and the near highest adaptation rates. The weakness of this approach is that cloning a fully connected hidden unit usually results in very large changes to the network (in some instances, 10+ connections being added at once), so it is possible this negative evolvability promoter far outweighs the positive evolvability gained by the structural duplication aspect. We can only deduce that this method has potential, but its weakness far outweighs its strength.

Looking at the evolved forms, it is obvious that all types utilised the structural mutations to increase their network's complexity, with some more than others. Some types in particular (types 2, 5) resulted in networks significantly larger than the starting networks. However, it does not seem as if the larger networks were inherently better or worse than the smaller ones. Interestingly, it seems as if these larger networks tended to provide the most consistent critters in terms of average survival age.

A possible criticism would suggest that highly evolvable populations would continue evolving forever, with E(t) values always above zero and E_{total} tending to infinity. However, in our system this is impossible. At the slowest rates of regeneration tested in our experiments, there are not enough resources left to support individuals, regardless of their genomes. Inevitably, evolvability must drop to zero at some point, for there will be no critters left in the population to evolve. Such eventual resource limitation leading to extinction is inevitable in all real and modeled systems (time will always be limited, if nothing else), so an infinite E_{total} may be impossible to achieve.

5 Conclusions

The aim of this study was to investigate the evolvability of neural networks within an artificial life simulation. Specifically, we tested the efficacy of five different types of

structural mutations, which incorporate different general principles thought to be important for network evolvability. Two experiments were performed, and the resulting E_{total} and evolvability function over time were calculated and compared.The experiments conducted indicate that certain principles increase evolvability when used to evolve neural network artificial agents. The two most significant promoters of evolvability are adaptive evolution and gradual changes to the networks. Structural duplication, despite exhibiting on average very low evolvability, showed some potential by evolving some of the best individual runs. Non-gradual changes to the networks seemed to be detrimental to evolvability (or at least, did not seem to increase it).

To summarise: the method chosen to in evolving neural networks for artificial life simulations plays a significant factor in all elements of the evolved runs. Researchers attempting to evolve neural networks are encouraged to use these principles.

References

1. Altenberg, L. (1994). The Evolution of Evolvability in Genetic Programming. In: *Advances in Genetic Programming*, K. E. Kinnear Jr., ed. MIT Press.
2. Altenberg, L. (1995). Genome growth and the evolution of the genotype-phenotype map. in W. Banzhaf and F. H. Eeckman, eds, *Evolution and Biocomputation: Computational Models of Evolution*. 205-259. New York. Springer-Verlag, Berlin, Heidelberg, 1995.
3. Astor, J. and Adami, C. (1998). Development and evolution of neural networks in an artificial chemistry. Proc. *Third German Workshop on Artificial Life*, Verlag Deutsch, 15-30
4. Bedau, M. A. and Packard, N.H. (2003). Evolution of evolvability via adaptation of mutation rates. *Biosystems* 69(2-3): 143-162
5. Ebner, M., Shackleton, M. and Shipman, R. (2001). How neutral networks influence evolvability. *Complexity*, 7(2) 19-33, Wiley Periodicals, 2001.
6. Fogel, D.B. (1995). Phenotypes, Genotypes, and Operators in Evolutionary Computation. Proceedings of the *1995 IEEE International Conference on Evolutionary Computation, Perth*, Australia, IEEE Press, 193-198.
7. Glickman, M. and Sycara, K. (1999). Comparing Mechanisms for Evolving Evolvability. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 1999 Workshop Program*. Orlando, Florida, USA, July 13, 1999.
8. Kirschner, M. and Gerhart, J. (1998). Evolvability. *Proceedings of the National Academy of Sciences*, 95:8420-8427.
9. Kumar, S. and Bentley, P. J.(2000). Implicit Evolvability: An Investigation into the Evolvability of an Embryogeny. A late-breaking paper in the *Genetic and Evolutionary Computation Conference* (GECCO 2000), July 8-12, 2000, Las Vegas, Nevada, USA.
10. Nolfi, S. and Parisi, D. (2002). Evolution of artificial neural networks. In *Handbook of brain theory and neural networks*, Second Edition. Cambridge, MA: MIT Press, 418-421.
11. Ofria, C., Adami, C., and Collier, T. C. (2002) Design of Evolvable Computer Languages. IEEE Transactions on Evolutionary Computation, 6:420-424.
12. Schlessinger, E., Bentley P.J. and Lotto R.B. (2005) Evolving Visually Guided Agents in an Ambiguous Virtual World. To appear in the *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2005*. Washington, DC, USA, June 25-29, 2005.
13. Smith, T.M.C., Husbands, P., Layzell, P. and O'Shea, M. (2002). Fitness Landscapes and Evolvability. *Evolutionary Computation*, 10(1):1-34.
14. Stanley, K.O. and Miikkulainen, R. (2002). Evolving Neural Networks Through Augmenting Topologies. *Evolutionary Computation* 10(2):99-127.
15. Wagner, P. W. and Altenberg, L. (1996). Complex adaptations and the evolution of evolvability. *Evolution* 50, 967-976.