

# Effect of Encoding on the Evolvability of an Embodied Neural Network

Chad W. Seys  
Dept of Electrical Engineering and Computer  
Science  
Case Western Reserve University  
Cleveland OH, 44106  
chad.seys@cwru.edu

Randall D. Beer  
Dept of Electrical Engineering and Computer  
Science  
Dept of Biology  
Dept of Cognitive Science  
Case Western Reserve University  
Cleveland OH, 44106  
beer@eecs.cwru.edu

## ABSTRACT

The evolvability of a neural network controller for a hexapod agent encoded directly and symmetrically is examined. The symmetric encoding imposes a structural regularity on the neural network and decreases the size of genotype space relative to the direct encoding. The symmetrically encoded neural networks are found to be more evolvable than the directly encoded neural networks, but it is unknown whether structural regularity or decreased size of the genotype is more important. To test whether structural regularity is more important than genotype size, the architecture of the neural network is manipulated to increase the genotype size of the symmetric encodings so that they are larger than the directly encoded genotypes. These symmetric encodings are still found to be more evolvable than the direct encodings despite having a larger genotype. In these experiments it is the encoding which determines evolvability more than size of genotype space.

## Categories and Subject Descriptors

I.2 [Artificial Intelligence]: Evolutionary computing and genetic algorithms

## General Terms

Algorithms, Performance, Theory

## Keywords

Structural Regularity, Symmetry, Genotype Size, Evolvability, Adaptability, Encoding, Representation, Genotype to Phenotype Mapping

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.  
Copyright 2006 ACM 1-59593-186-4/06/0007 ...\$5.00.

## 1. INTRODUCTION

Evolutionary algorithms (EAs) such as genetic algorithms (GAs) are widely used as a means to solve problems from many areas of interest in ways unexpected by their programmers. As the number of parameters describing the problems to which EAs are applied grow larger, the size of the parameter space grows exponentially [17] and leads to a scalability problem.

Noticing that nature has evolved systems with many parts, researchers have begun to extract the essential attributes of nature's genotype to phenotype mapping which they hope will ameliorate the scalability problem. Some of the attributes which the genotype to phenotype mapping should facilitate are functional modularity, hierarchy, and structural regularity [11]. Structural regularity is a super-set including symmetry, repetition, and self-similarity which, because of their existence, allow a description of the system to be encoded with fewer parameters [10].

Experiments have been performed comparing the relative ability of different encodings which facilitate these "essential attributes" in the phenotypes to evolve complex systems. Gruau [5] makes a comparison between a single cellular encoding of neural networks (controlling a hexapod robot) with and without a mechanism for the automatic definition of sub-neural networks (ADSN). It was found that the cellular encoding used in conjunction with ADSN was able to find solutions that cellular encoding alone could not. Bentley and Kumar [3] define three different "embryogenies", some of whose genotype to phenotype mappings could be evolved and others which could not. These embryogenies were compared against a direct mapping on their ability to create tessellations of tiles at a number of scales. Komosinski and Rotaru-Varga [9] use two direct encodings and one developmental encoding of 3D agents. The encodings were compared on their relative ability to discover high fitness agents performing one of three tasks: active or passive height achievement or locomotion speed. Most recently, Hornby [7] compares five different encodings of simulated physical tables which differ in the degree of the attributes of combination, control-flow, and abstraction. These encoding are compared on their ability to discover complete tables with little excess material at different scales. All of these comparisons find that those encodings which facilitate more of the essential attributes find higher fitness solutions, and where examined, also find scalability improvements over those en-

codings which facilitate fewer of the essential attributes.

However, in most of these experiments the comparisons made were between very different encodings, so that although facilitation of the essential attributes as a group was shown to be important, it is difficult to determine the relative importance of individual attributes. Moreover, the encodings which facilitate the essential attributes used by each author are (sometimes greatly) different from each other and it is difficult to discriminate which encoding is more suited for what task. By understanding the relative importance of attributes individually it may be possible to more fully understand and improve existing encodings or create new ones for the evolution of complex systems.

An increasing degree of structural regularity, a decrease in genotype size, and an increase in fitness of solutions discovered is a repeating theme in the experiments mentioned above. Some of the papers [3, 9, 5] explicitly credit performance increases to both structural regularity and a decrease in genotype size. The question of what extent the size of genotype space and structural regularity contribute to the success of encodings is widely applicable, yet remains unexplored.

This paper compares the relative effect of the attribute of symmetry and number of genotype parameters on evolvability of a simple system.<sup>1</sup> The system consists of a neural network controlling a simulated hexapod agent in a simple walking task. We sometimes make a distinction between two types of evolvability, “discoverability” and “adaptability”. Discoverability is defined as the fraction of GA searches, beginning with a random initial population, which result in a high fitness individual being found. Adaptability is the ability of a population containing high fitness individuals to maintain a high fitness during a GA search while an environmental change occurs.

The symmetry (or potential for asymmetry) of the neural network was a property the genotype to phenotype mapping. The six legs of the hexapod agent provided a six-fold symmetry around which the “symmetrical” encoding of the neural network was manually created. The symmetrical encoding used a genotype which contained one parameter for every parameter in the phenotype of a subnetwork  $\frac{1}{6}^{th}$  the neural network. This subnetwork was copied six times to form the neural network. Every genotype parameter in the symmetrical encoding was reused six times in the phenotype. The “direct” encoding of the neural network contained one genotype parameter for every neural network parameter. Directly encoding the neural network allows every parameter to take on a different value and potentially create asymmetrical networks.

The number of genomic parameters was adjusted by using two neural network architectures. In order to compare the relative effects of symmetry and number of genotype parameters, it was necessary to create symmetric neural networks with the same number of genotype parameters as the asymmetric neural networks. This was accomplished by increasing the number of genomic parameters in the symmetric encoding by adding connections to the neural network architec-

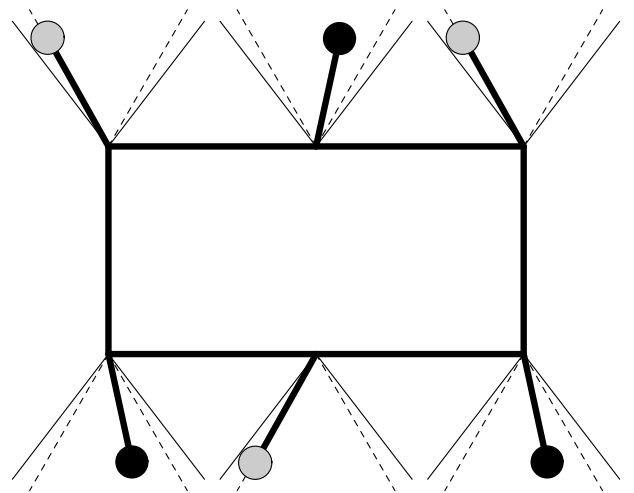
<sup>1</sup>“Evolvability” is a term that has many meanings in the literature. Two definitions, both of which apply to the experiments being performed in this paper are 1) ability of random variations to sometimes produce improvement [15] and 2) ability to respond to a selective challenge [6]. Reisinger [12] discusses evolvability and its definition more fully.

ture. The two architecture types were “locally-connected” and “fully-connected” and will be described in the Methods section.

This paper will show that the symmetric encodings are more evolvable than the direct encodings. Furthermore, the symmetric encodings with more genomic parameters are more evolvable than smaller direct encodings.

## 2. METHODS

The body (Figure 1) is based on prior work by Beer & Gallagher [2] and similar to [8]. Pairs of legs are attached on either side at the extreme ends of the length of the body as well as a pair in the middle. Effectors apply clockwise and counter-clockwise torques to each leg which sum to produce a resultant force which can move the leg through an angle of  $\pm\frac{5\pi}{24}$  radians, with 0 radians being perpendicular to the body’s long axis. In the central  $\pm\frac{\pi}{6}$  region the torque applied could be 10 times larger than in the extreme  $\pm\frac{\pi}{24}$  regions. These “weak regions” are designed to degrade fitness gradually and amplify the effects of a neural network oscillating in a way not tuned to the legs’ length. While the foot is raised the resultant torque moves the leg through an angle relative to the body. When legs move past the outer angle limits their feet are disengaged from the ground and the leg angle reset to just inside the outer angle limit. If at least three legs are in a position such that their low-



**Figure 1: Hexapod agent modeling a tripod gait. Feet on the ground are black while feet off the ground are grey. Leg effectors become weak at the extremes of the leg’s range of motion (between solid and dashed lines).**

ered feet form a base below the body’s center of gravity, the agent is considered “supported” and a sum of forces from all legs with feet on the ground are applied to the body. If the body is unsupported, then a large amount of friction is applied in a direction opposite its velocity. There is an enforced maximum velocity of 1 which prevents the body from “outrunning” the legs. (Because there is no sensory feedback, body velocity and leg oscillation frequency are not coupled. The body “outruns” the legs without a maximum velocity.) A small amount of friction opposes velocity even when the agent is supported so that the legs must continue

to accelerate the agent rather than merely have a supporting arrangement. The body is initialized with all legs at the forward-most limit of the non-weak region and all feet down so that the body is supported. A high fitness individual, defined as an individual with fitness greater than or equal to 0.75, begins walking by pushing backwards with all six legs and then one set of three legs decouples from the other three, and reaches forward and begins a tripod gait.

The body was controlled by a continuous-time recurrent neural network (CTRNN):

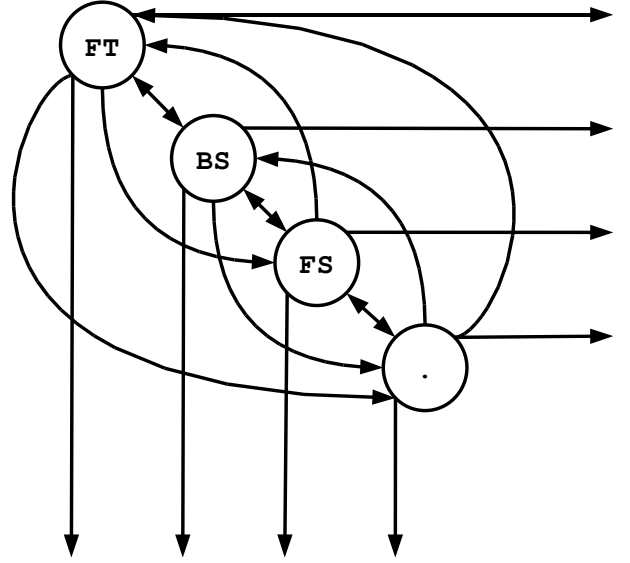
$$\tau_c \dot{y}_c = -y_c + \sum_{j \in \text{IN}} w_{jc} \sigma(y_c + \theta_c),$$

$$\text{IN} = \{\text{self and other incoming connections}\}$$

where  $y_c$  is the state of the “current” neuron  $c$ ,  $\dot{y}_c$  denotes the time rate of change of this state,  $\tau_c$  is the neuron’s membrane time constant,  $w_{jc}$  is the strength of the connection from neuron  $j$  to the current neuron  $c$ ,  $\theta_c$  is a bias term,  $\sigma(x) = 1/(1 + e^{-x})$  is the standard logistic function, and IN is the set of all neurons with connections to the current neuron, including the current neuron because of a self-connection. Connection weights and biases were constrained to the range 16 and time constants to the range [0.5, 10]. The state of each neuron was initialized randomly to a value between  $\pm 0.1$  and their outputs adjusted to match. This random noise helped break symmetry in what otherwise could be a completely symmetrical neural network architecture and initial body arrangement. The body and neural network were integrated using the forward Euler method with a step size of 0.1.

A real-valued genetic algorithm (GA) is used to evolve the CTRNN parameters. For every CTRNN parameter there is a GA parameter linearly rescaled to lie between [-1, 1]. A population of 100 individuals is maintained and parents are selected for reproduction using a linear rank-based method with the most fit individual producing an average of 1.1 offspring. Children are generated by mutation of selected parents by adding the parent’s vector to a random displacement vector with uniformly distributed direction and normally distributed magnitude [1]. The mutation magnitude has a zero mean and variance is set to different values depending on the stage of the experiment, as explained below. Use of Elitism (preserving the best individual found by the search through all generations) and the number of generations of search also vary by stage of experiment, also explained below.

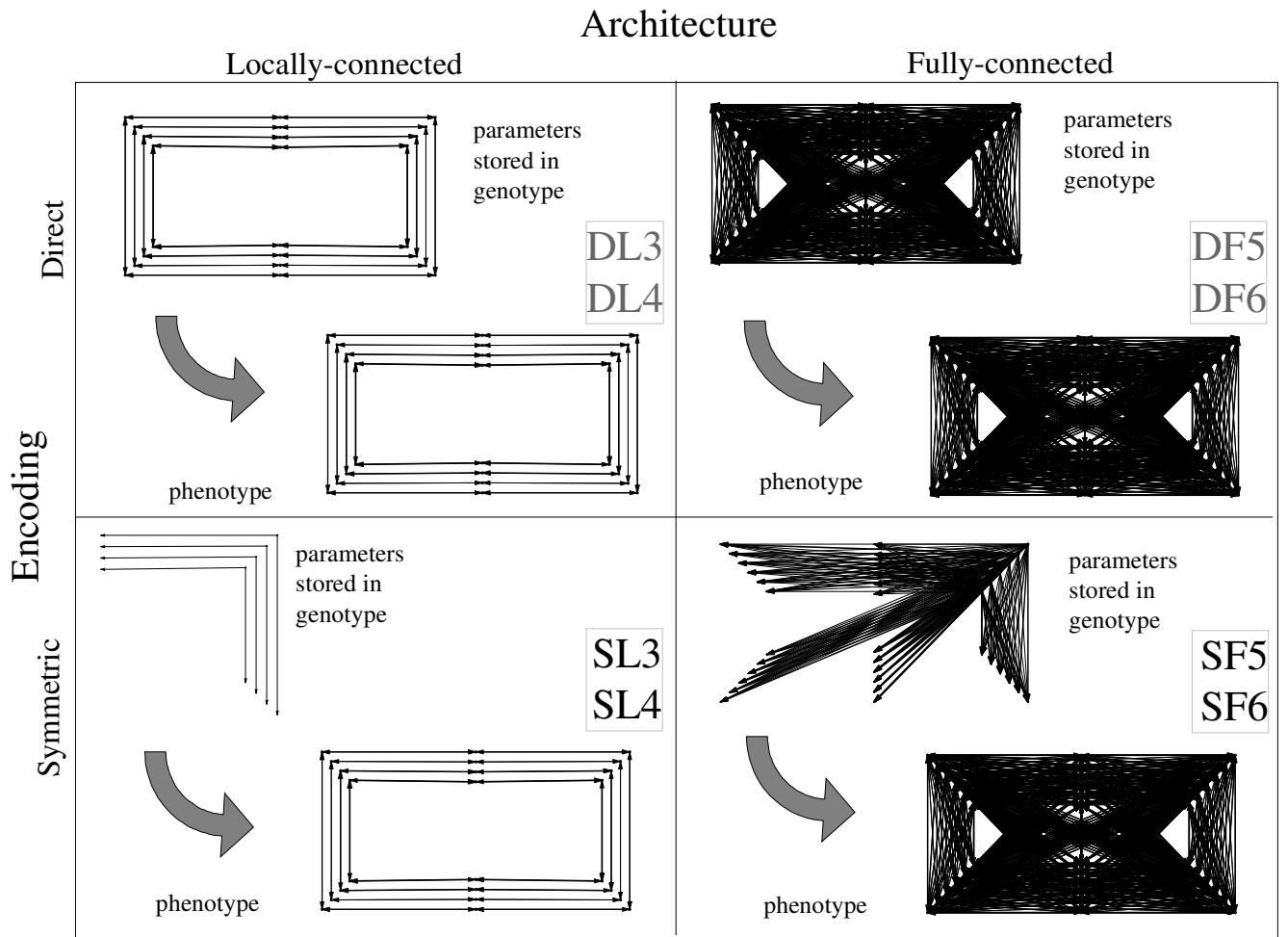
Two types of neural network architecture, called “locally-connected” and “fully-connected” are used to control the six legs of the body. Examples of these two architectures are marked “phenotype” in each of the two columns of Figure 3. The left column shows locally-connected (with an L as the center letter of the label ·L·, where “·” is anything) and the right fully-connected (·F·) neural network phenotypes. In both types of architectures there are always three neurons per leg which serve as motor neurons and control the effectors of the leg. Two of the neurons oppose each other and are named after their function: backward swing (BS) and forward swing (FS). The third neuron raises and lowers the foot (FT). Additional interneurons with no direct connections to the effectors can be added to the network in multiples of six. In the locally-connected architecture, each leg is associated with a fully connected subnet of neurons,



**Figure 2: Connections within a leg’s subnetwork (·L4). Foot (FT), BackSwing (BS), ForwardSwing (FS), and an interneuron (·). The number of interneurons varied in these experiments from 0 to 3. Outward pointing arrows are connections to neighboring leg subnetworks.**

which is composed of the leg’s motor neurons and  $\frac{1}{6}^{th}$  the interneurons. (These connections are shown in Figure 2). Each neuron in a subnet is connected only to its analogous neighbor in the subnet immediately to the left and right around the perimeter of the body. For example a BS neuron in any subnet will have one outgoing connection and one incoming connection from each of its two BS neurons in its left and right neighboring subnets (for a total of four connections from the neighboring subnets). These connections are shown in Figure 3 as arrows pointing from the source neuron (a circle) to the destination neuron. What appears to be a double-headed line is actually two arrows pointing in opposite directions lying on top of each other indicating incoming and outgoing connections between a pair of neurons. In the fully-connected architecture, every neuron is connected to every other neuron.

Two encoding schemes, direct and symmetric, are used to represent the neural network parameters. The top row of Figure 3 shows schematics of direct encodings (and have a D as the first letter of their label D·) and the bottom symmetric encodings (S·). In the direct encoding scheme, every parameter of every neuron was present in the genome and the number of parameters stored in the genotype are the same as those in the phenotype. Additionally, one subnet of motor neurons, interneurons (if any), and their connections are present in the genotype. The symmetric encoding genotype is mapped to a neural network by duplicating the parameters six times, once for each leg. The outgoing connections of the subnetwork are specified as a relative offset to a subnet arranged around a perimeter. E.g. “From BS (in genomic parameters) to BS in 2nd subnet clockwise around the perimeter.” Note that the direct encoding can encode



**Figure 3: Genomic encoding and architectures of neural networks: Top and bottom rows contain graphics symbolizing direct and symmetric encodings (respectively). The arrow is the mapping from genotype to phenotype parameters. Left and right columns show the neural network connectivity patterns of locally- and fully-connected architectures, respectively. The connections internal to each leg subnet are not shown in this Figure, but are shown in Figure 2.**

all neural networks that the symmetric encoding can, but not visa versa.

By choosing a particular combination of architecture and number of neurons, it was possible to create symmetrically encoded neural networks which had slightly more genomic parameters than directly encoded neural networks. The genome of an symmetrically encoded, fully-connected, neural network with 5 neurons (SF5) has 160 parameters, slightly more than the 126 parameters a directly encoded, locally-connected neural network with 3 neurons (DL3) has. Similarly, SF6 contains 228 parameters, which creates a larger genomic parameter space than DL4. If parameter space size alone were the sole determinant of discoverability, then one would predict the order, in increasing discoverability or adaptability to be DL3, SF5, DL4, SF6, but as the Results will show, this is not the case.

The adaptability experiments are conducted in three stages, with the overall goal being the creation of a population with high overall fitness. 1) A high fitness seed individual is discovered through GA search beginning with a random initial

population; 2) then the seed individual is used to create an “equilibrium GA search”, and finally; 3) the equilibrium GA search is continued while the population’s legs are incrementally shortened at a rate determined by the adaptation of the population to the new leg length.

First, high fitness seed individuals were discovered by creating a random initial population and performing a GA search with Elitism and a mutation variance of 0.05 for 500 generations. If the best individual had a fitness of greater than 0.75 it was used as a seed individual for the creation of equilibrium populations. The fitness of 0.75 was chosen as a cutoff by subjective observation. The behavior of the individuals was a fairly coordinated tripod gait with legs swinging through the full range of the angles except the extreme “weak” angles of the leg’s range of motion. This behavior was judged to indicate the neural network had optimized its output to the body with the nominal leg length.

Second, these high fitness individuals then serve as seed individuals for the creation of an “equilibrium GA search”. An equilibrium GA search is a situation where the popu-

lation’s location in parameter space and GA settings (such mutation variance and selection pressure) are balanced to allow high fitness individuals to exist for an indefinite number of generations without the use of Elitism. This can be achieved if a zero mutation variance is allowed (because no change in population occurs), but in addition, no adaptation can occur. A mutation variance somewhere above zero mutation variance (no adaptation) and a high mutation variance (destruction of equilibrium GA search) had to be found. Because the layout of genotype/fitness space is unknown, an ideal mutation variance could not be determined *a priori*. Instead, an attempt was made to create an equilibrium GA search at mutation variances at each power of 10 between  $1 \times 10^{-6}$  and  $1 \times 10^{-1}$  for each seed individual. The initial population of each attempt consisted of 100 copies of the seed individual. If the population was able to maintain a high median fitness ( $\geq 0.75$ ) over 500 generations of GA search (without Elitism) it was deemed to be an equilibrium GA search and the adaptation stage of the experiment was performed on it.

Third, the adaptability of the equilibrium GA searches was determined. The adaptability of equilibrium GA searches is the ability of an extra-genomic parameter to be changed while a target fitness is maintained. This is analogous to asking the question “How fast can the organism adapt to environmental change?”. Unlike in nature, the environment (politely) waits for the population’s fitness to rebound before changing further. In these adaptability experiments, the median population fitness was the target fitness and the leg length of the agent was the extra-genomic parameter that was changed.<sup>2</sup> First the median population fitness was recorded as the target fitness, then the length of the legs (starting at 15) of the individuals in the population were shortened (by 0.1). The equilibrium GA search continued until the median population fitness was greater than or equal to the target fitness and the legs were shortened incrementally again. This was repeated until either 5000 generations had passed, or rarely, a leg length of 10 was reached, and the change in leg length<sup>3</sup> was recorded. The change in leg length of the equilibrium GA searches sharing a common seed individual but run with different mutation variances are compared and the largest change used as the final adaptability number.

The significance of the differences in the adaptability numbers were determined by comparing the confidence intervals of the median created by bootstrapping. Bootstrapping is a method for generating distributions of a statistic such as the mean or median from raw data by repeatedly sampling the data. [4, 14] The confidence interval for the statistic can then be determined from the distribution of that statistic in the standard way. The 95% confidence interval of the median for each combination of encoding, architecture, and number of neurons was determined. If the 95% confidence intervals do not overlap, then there is a 5% or less chance that the medians are the same, i.e. a  $p < 0.05$ .

<sup>2</sup>The leg length is not part of the genome, so it should be considered part of the environment.

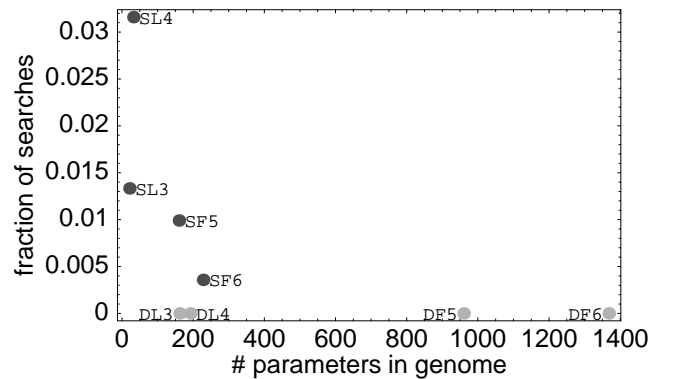
<sup>3</sup>Abs(final leg length – initial leg length)

### 3. RESULTS

#### 3.1 Symmetric encoding more discoverable than direct encoding

Before the adaptability of neural networks in various encodings and architectures was measured, high fitness individuals were evolved to be the seeds of an equilibrium GA search. This was done by running a GA search with high mutation variance (relative to the adaptability experiments) and Elitism, beginning with a randomly generated population.

Figure 4 gives the fraction of searches which successfully found a high fitness individual. Every encoding/architecture type was given at least 1000 attempts to find one high fitness individual. The searches are arranged in order of number of parameters and are labeled with their abbreviated name.



**Figure 4: Discoverability of the various neural networks types. Fraction of searches beginning with random initial population resulting in discovery of high fitness individual plotted by size of genotype. The directly encoded neural networks found no high fitness individuals after 1000 attempts.**

Only GA searches using symmetrically encoded architectures successfully discovered high fitness individuals. After 1000 attempts, no GA search performed on a directly encoded architecture discovered a high fitness individual. The symmetric encoding GA searches did discover high fitness individuals and enough searches were run so that 20 high fitness symmetrically encoded seeds of each architecture were found. The least discoverable symmetrically encoded architecture is SF6 with approximately 1 high fitness individual found per 280 searches and the most discoverable is SL4 with a ratio of  $\sim 1:30$ .

Interestingly, symmetric encodings were discoverable despite sometimes having more parameters than the directly encoded architectures. From Figure 4 SF5 and SF6 have more genomic parameters (160, 228 respectively) than DL3 and DL4 (126, 192 respectively), yet no high fitness individuals were found for either of the direct encodings. This pattern is echoed below in the adaptability findings in Section 3.3.

Because no directly encoded high fitness seeds were found, symmetrically encoded high fitness seeds were re-encoded using the direct encoding. This is possible because the genomic space of the direct encoding is a superset of the symmetric encoding. The generation of equilibrium GA searches

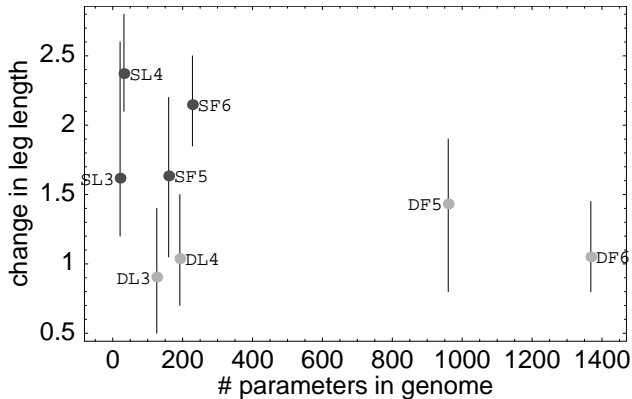
from the re-encoded seeds proceeded as described in the Methods.

### 3.2 Restriction of architecture by symmetric encoding increases adaptability

The symmetric encodings used in these experiments have the same network connectivity as the direct encodings, but the weights of the connections are restricted to a six-fold symmetry and therefore have a smaller genomic parameter space as described in the Methods section. The directly encoded neural networks have a wider range of phenotypes possible and include as a subset the symmetric encodings. In order to find what effect symmetrical restrictions have on this system, experiments were run in which the two architecture types were encoded both directly and symmetrically.

Two variants of the fully-connected and locally-connected neural networks were encoded both directly and symmetrically. The two fully-connected neural networks contained 5 and 6 neurons per leg, and the two locally-connected neural networks contained 3 and 4 neurons per leg.

Figure 5 plots the number of genomic parameters versus the change in leg length during the adaptation experiments. The median adaptability of the direct encoding neural networks is marked with a grey dot and symmetric encoding neural networks are marked with a black dot. The 95% confidence interval of the median are marked with lines extending vertically from each median marker. Next to each median marker is the abbreviated name of each neural network.



**Figure 5: Adaptability of all encoding / architecture combinations performed. The symmetric encoding of a specific architecture number of neurons is always more adaptable than the direct encoding. Also, if the encoding is the same, both architectures have approximately the same adaptability.**

Symmetrically encoding a neural network makes that architecture more adaptable when compared to the direct encoding of the same architecture. The comparisons to make are between neural networks which have different encoding type (first letter) but have the same architecture (second letter) and number of neurons. Specifically, comparisons should be made between DF5 & SF5, DF6 & SF6, DL3 & SL3, and DL4 & SL4. In every pairing the median adaptability of the symmetrically encoded version of the architecture is larger than the direct encoding. (In two out of the four pairs, DF6 & SF6 and DL4 & SL4 the 95% confi-

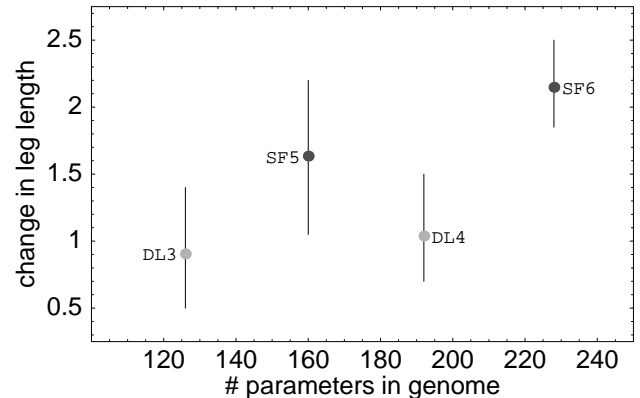
dence interval of the medians do not overlap and difference is significant.)

Furthermore, the two architectures (fully- and locally-connected) are roughly equally adaptable using the same encoding type. The median adaptabilities of each symmetrically encoded architecture are all greater than the median adaptability of all directly encoded architecture. However, the clustering is not tight and many of the 95% confidence intervals of the median overlap between the two encoding types.

The greater adaptability of symmetrically encoded neural networks cannot be caused by the location of the seed individual in genotype space. The directly encoded seed individuals had the same non-zero neural network weights as the symmetrically encoded seeds and the equilibrium GA search began in the same region of genotype and phenotype space. Additionally, because the direct encoding's genotype space is a superset of the symmetric's genotype space, in principle the directly encoded population can continue to mirror the symmetrically encoded population during equilibrium GA search. Instead the lower adaptability of directly encoded neural networks is probably due to the existence of additional lower fitness neural networks possible in the larger genotype space.

### 3.3 Encoding more important to adaptability than genome size

One reasonable hypothesis is that fewer genomic parameters leads to more discoverable, and by extension, adaptable, phenotypes. In order to test this hypothesis, experiments were run to compare the adaptability of different encodings with the similar numbers of genomic parameters.



**Figure 6: Subset of all adaptation experiments. DL3, SF5, DL4, SF6. Adaptability does not decline with number of parameters in the genome. Instead, adaptability is greater when the neural network is symmetrically encoded even when the number of parameters is greater.**

Two symmetrically encoded, fully-connected architectures with slightly more parameters than two direct encodings of the locally-connected architectures were created. The symmetrically encoded, fully-connected neural networks have 5 and 6 neurons per leg (SF5, SF6) for a total of 160 and 228 genomic parameters, respectively. The directly encoded, locally-connected neural networks have 3 and 4 neurons per leg (DL3, DL4) for a total of 126 and 192 genomic pa-

rameters, respectively. Note that the number of genomic parameters in the symmetrically encoded, fully-connected networks are intermediate or greater than the number of genomic parameters in the directly encoded, locally-connected networks.

Figure 6 is a magnified view of Figure 5 and isolates the neural networks DL3, SF5, DL4, SF6. As in Figure 5 the x-axis is the number of genomic parameters and the y-axis is the change in leg length.

Symmetrically encoded neural networks are more adaptable even when the symmetric encoding has more genomic parameters than the direct encoding. Based on genomic parameter space size one would predict the order of increasing discoverability or adaptability to be DL3, SF5, DL4, SF6. This is not the case: SF6 (228 genomic parameters) is significantly more adaptable than DL4 (192) and though not significantly so, the median adaptability for SF5 (160) is higher than DL3 (126) median adaptability.

These results cannot be shown to be merely because the locally-connected neural architecture is less adaptable than the fully-connected architecture. As the results of Section 3.2 show, if the locally-connected neural networks DL3 & DL4 are symmetrically encoded, their adaptability is the same (if not more) than the symmetrically encoded fully-connected neural networks SF5 & SF6.

In these experiments, the size of genomic parameter space is less important to adaptability than encoding.

## 4. DISCUSSION

Most of the encodings cited in the introduction have been based on attributes of biological genotype to phenotype maps (GPmaps). While this approach has demonstrated results, the underlying reasons for the current state of biological GPmaps are less well understood. As a step towards crystallizing the understanding of some of these concepts, this paper studied the relationship between number of genomic parameters describing a neural network, encoding type, and discoverability and adaptability. It was shown that the number of genomic parameters is less important to discoverability than the encoding of the neural network. Symmetrically encoded neural networks containing more genomic parameters were more discoverable and adaptable than directly encoded neural networks.

One possible way to explain these results is that it is not the parameter space size which is most important, but the symmetrical way in which the symmetric encoding was unpacked to create the neural network. When there are fewer parameters in the genotype than in the phenotype, there is necessarily some reuse of genotype parameters upon creation of the phenotype. In the universe of GPmaps, this reuse is not necessarily in a structurally regular, or symmetric way.

Future work could study the possibility that describing a phenotype using an symmetric encoding in the “wrong” way could make a system less evolvable. For example, in this system the six-fold symmetrical unpacking of the genotype into the six leg’s neural subnetworks could be viewed as a special symmetric mapping contrived to make the system more evolvable than the direct encoding. Is it possible to create an symmetric encoding which is less evolvable than the direct encoding? The attempt to do so might clarify what types of “reuse” are important.

The architecture of the neural network, whether fully-connected or locally-connected, appears to have little effect

on the adaptability of the neural network. This might be somewhat surprising given that the more connections there are, the wider the influence each individual neuron has on the neural network as a whole. One might expect that the locally-connected architectures have structural modularity that leads to a “structural separation of functions that reduces the amount of coupling between internal and external behavior” [11] and allows for greater adaptability. On the other hand the dynamical functioning of these neural networks has not been determined and others [16] show that structural topology does not imply functional modularity. It may be that the two architectures function in similar ways though they are structurally different.

Would a varying environment of evolution select for more adaptable neural networks later? It was somewhat surprising how non-adaptable all architectures studied were. Most of the adaptation experiments in which the legs were shortened incrementally from length 15 towards length 10 terminated because the maximum number of generations had passed rather than reaching the leg length 10. Plotting generation number versus leg length reveals a general pattern of initially fast rate of adaptation that quickly flattens out for the rest of the search. Since it is possible to evolve high fitness leg length 10 individuals using a GA search beginning on a random population, it may be that there is no high fitness connection between the two regions in genotype space. I.e. It may be that high fitness leg length 15 individuals are not located in a region of genotype space which has a high fitness path to a genotype which produces a high fitness leg length 10 individual. If the leg were incrementally changing length throughout the GA search, the result might be a more adaptable neural network, not through evolution of the GPmap (as in [13]), but by filtering out the genotypes which do not have neighbors in genotype space that produce high fitness phenotypes at a variety of leg lengths.

Currently natural genotype to phenotype mappings are imitated without fully understanding why they are successful, how they evolved, or their influence on the evolution of organisms. By studying genotype to phenotype mappings, we improve our ability to evolve artificial systems and understand the evolution of natural systems.

## 5. ACKNOWLEDGMENTS

We would like to thank the reviewers for their useful comments and references. Thanks also goes to the Beer Nuts who were tortured with incomplete drafts of this paper.

This research was supported in part by NSF grant EIA-0130773 and in part by an NSF IGERT fellowship.

## 6. REFERENCES

- [1] T. Bäck. *Evolutionary algorithms in theory and practice*. Oxford University Press, 1996.
- [2] R. D. Beer and J. C. Gallagher. Evolving dynamical neural networks for adaptive behavior. *Adaptive Behavior*, 1(1):92–122, 1992.
- [3] P. Bentley and S. Kumar. Three ways to grow designs: A comparison of ?embryogenies for an evolutionary design problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 1999.
- [4] B. Efron and R. J. Tibshirani. *An introduction to the bootstrap*. CRC Press, Boca Raton, FL, 1994.

- [5] F. Gruau. Automatic definition of modular neural networks. *Adaptive Behavior*, 3(2):151–183, 1995.
- [6] T. F. Hansen. Is modularity necessary for evolvability? remarks on the relationship between pleiotropy and evolvability. *BioSystems*, 69:83–94, 2002. "ability to respond to a selective challenge".
- [7] G. S. Hornby. Measuring, enabling and comparing modularity, regularity and hierarchy in evolutionary design. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2005.
- [8] J. Kodjabachian and J.-A. Meyer. Evolution and development of modular control architectures for 1-d locomotion in six-legged animats. *Connection Science*, 10(3-4):211–237, 1998.
- [9] M. Komosinski and A. Rotaru-Varga. Comparison of different genotype encodings for simulated three-dimensional agents. *Artificial Life*, 7(4):395–418, 2001.
- [10] H. Lipson. Principles of modularity, regularity, and hierarchy for scalable systems. In *GECCO Workshop on Modularity, Regularity, and Hierarchy in Evolutionary Computation*, 2004.
- [11] H. Lipson, E. de Jong, and J. R. Koza. Workshop introduction. In *GECCO Workshop on Modularity, Regularity, and Hierarchy in Evolutionary Computation*, 2004.
- [12] J. Reisinger, K. O. Stanley, and R. Miikkulainen. Towards an empirical measure of evolvability. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2005.
- [13] M. Toussaint. On the evolution of phenotypic exploration distributions. In *Foundations of Genetic Algorithms (FOGA VII)*, volume 7, pages 169–182, 2003.
- [14] H. Varian. Bootstrap tutorial. *The Mathematica Journal*, 9(4):768–775, 2005.
- [15] G. P. Wagner and L. Altenberg. Perspective: Complex adaptations and the evolution of evolvability. *Evolution*, 50(3):967–976, 1996.
- [16] R. A. Watson. *Compositional evolution: Interdisciplinary investigations in evolvability, modularity, and symbiosis*. PhD thesis, Brandeis University, Dept. of Computer Science, 2002.
- [17] X. Yao. Evolving artificial neural networks. In *Proceedings of the IEEE*, volume 87, pages 1423–1447, 1999.