

The Role of Nearly Neutral Mutations in the Evolution of Dynamical Neural Networks

Eduardo Izquierdo-Torres

Department of Informatics
University of Sussex, BN1 9QH. UK.
e.j.izquierdo-torres@sussex.ac.uk

Abstract

The evolution of continuous time recurrent neural networks is increasingly being employed to evolve nervous systems for autonomous agents. Nonetheless, the picture of populations engaged in hill-climbing rugged fitness landscapes poses a problem of becoming trapped on a local hilltop. Developments in evolutionary theory and molecular biology have pointed to the importance of selective neutrality. The neutral theory claims that the great majority of evolutionary changes are caused not by Darwinian selection but by random drift of selectively neutral or nearly neutral mutants. However, with a few exceptions neutrality has generally been ignored in artificial evolution. This paper addresses the distribution of fitness effects of new mutations when evolving dynamical systems and provides evidence of an improved evolutionary search process when incorporating nearly-neutral drift. This is one of the most fundamental problems in artificial evolution, because it lies at the heart of maintaining a constant-innovative property.

Introduction

Continuous time recurrent neural networks (CTRNNs) are increasingly being employed as simple model nervous systems for robot controllers. For this, the approach has been an evolutionary one, evolving increasingly complex neural controllers (Beer & Gallagher, 1992; Harvey, Husbands, Cliff, Thompson & Jakobi, 1997). However, in attempting to address the features of the fitness landscapes as an abstraction of the dynamics of evolution a picture has become ingrained of a landscape as a rugged terrain. This poses the problem of populations becoming trapped on a local hilltop (Barnett, 1997). Yet, developments in evolutionary theory and molecular biology point towards the importance of selective neutrality (Kimura, 1983). Kimura introduces this phenomenon to question the preeminence of selection as the sole mediator of the dynamics of biological evolution. More recently, the identification of neutral networks – connected networks of genotypes mapping to common phenotypes (and therefore equal fitness) in RNA secondary structure folding (Schuster, Fontana, Stadler & Hofacker 1994) and protein structure (Babajide, Hofacker, Sippl & Stadler 1997) has revived the interest of selective neutrality.

Our interest stems from the growing evidence that such neutrality and indeed neutral networks may be a feature of fitness landscapes which arise in evolving neural

controllers for autonomous agents (Cliff, Husbands & Harvey, 1993; Smith, Husbands & O’Shea, 2001; Smith, Phillippides, Husbands & O’Shea, 2002), and from the evidence that the dynamics of evolutionary processes on such landscapes is qualitatively different from the traditional rugged landscapes (Harvey & Thompson, 1996; Barnett, 1997, 1998, 2001).

Within the last few years there has been an increasing amount of research into the possible benefits and applications of neutral networks in evolution. However, nearly all of this work has been primarily concerned with discrete alphabet genotypes. Conversely, the most common approaches to evolving dynamical neural networks as neural controllers for autonomous agents have been based on real-value encodings. It will be argued herein that when dealing with real-valued landscapes emphasis needs to be made on nearly-neutral mutations (as opposed to exactly neutral), given that in real-valued landscapes it is unlikely for two points to have the same fitness.

This has led us to ask: What proportions of mutations are deleterious, neutral, and advantageous? What is the strength of selection that acts on non-neutral mutations? In other words, what is the distribution of fitness effects of new mutations? This question is central to our understanding of near neutrality and the maintenance of a constant-innovative property at both the genotypic and phenotypic level. Unfortunately, even in biology relatively little is known about the distribution of fitness effects, despite its importance. Recently, analysis of mutation accumulation experiments have shown that the distribution of fitness effects for deleterious mutations is highly leptokurtic (Piganeau & Eyre-Walker, 2003); that is with a few mutations having large effects, and the vast majority having mild effects. A similar distribution of effects is obtained when evolving dynamical neural networks, placing great importance on neutral drift during its evolution. In this paper we consider the usefulness of such neutrality.

Neutrality in Artificial Evolution

Neutral theory was first introduced by Kimura (1983) as genetic change without selection pressure in evolution. The importance of this theory is that genetic code can be constantly altered with no fitness disadvantage. Thus,

neutrality provides an explanation for a crucial aspect of natural evolution – the constant-innovative property.

In the abstraction of a search space as a fitness landscape, a neutral network has been proposed in the form of a plateau connecting two peaks (Barnett, 1997; Smith, Husbands & O’Shea, 2001). This leads to a population being able to achieve any possible phenotype, given enough time.

Barnett (1997) and Shackleton et al (2000) have applied neutralist ideas to binary search spaces, namely, adding tunable neutrality to Kauffman’s NK landscapes and evaluating various redundant mappings. Many benefits have been supported by these studies in which drift allows for the discovery of better adapted phenotypes. Smith (2001; 2002) has suggested benefits in evolvability and the speed of evolution and Jakobi (1995) has suggested benefits in producing increasingly complex behaviors.

Currently, little or no research has been carried out on real-valued search spaces and converting the genotype to a binary representation requires certain level of quantization. Zendric-Ahsmore (2003) explores several different encoding schemes when evolving a fully connected recurrent neural network. In his work, exploitation of neutrality is studied via redundant mappings and achieved only when binary dimensions are added to the real-valued search space. In the case of evolving hardware (Harvey & Thompson, 1993), results encourage the application of potentially useful junk genotype.

This work takes a different approach to exploiting neutrality based on the hypothesis that when evolving dynamical neural networks as model nervous systems neutrality comes for free. The intuitive reason behind this idea is the implicit redundant genotype-phenotype mapping generated by the infinite possibilities of real-valued genotypes and the finite number of significantly different phase portraits.

Finally, in the rarer cases when artificial evolution practitioners consider neutrality, often neutral and non-neutral mutations are treated as ‘all or nothing’ with respect to selection. However, when exploring neutrality in real-value search spaces exact neutrality will be practically impossible. For this reason, in this work we consider mutations whose behavior is influenced by both selection and random drift.

Experimental Design

In order to measure fitness effects from mutations and to take advantage of nearly neutral networks in the evolution of CTRNNs, we experiment with the commonly used real-value encoding on a very simple optimization problem using a slightly modified version of a population of 1+1 evolutionary technique.

Real-Valued Direct Encoding CTRNN

The property of neutrality in the dynamics of evolution is greatly determined by the encoding scheme and the genotype-phenotype mapping of the model being evolved.

Our model represents the common encoding and mapping scheme used in the evolutionary robotics literature, where the genotype is real-valued and maps almost directly to the CTRNN. The genotype is a vector of n^2 real numbers between $[0, 1]$, where n is the number of neurons in the network. These values are mapped to $[-5, 5]$ to represent the connection weights w_{ji} in the CTRNN, which mediate the interaction between the neurons, positive (excitatory), negative (inhibitory). This model encodes a fully connected real-valued CTRNN. For simplicity, the biases in the transfer functions and time-constant parameters were kept fixed, and are thus not included in the genotype. This model allows for no structural neutrality, that is, there is no redundancy in the genotype \rightarrow CTRNN-architecture mapping, on the basis that any mutation, as small as it may be, will cause a change in the strength of the connections of the network. However, will mutations inevitably affect the overall behavior of the resulting dynamical system? Small changes in the parameters of the system may not change its attractors and overall behavior significantly. Therefore, it is likely that the genotype \rightarrow phenotype mapping is redundant (considering the phenotype as the long-term behavior of the system). As has been discussed, when evolving CTRNNs various levels of phenotype can be characterized: phenotype as architecture of the CTRNN and phenotype as the long-term behavior of the resulting dynamical system. For the reasons herein described we will treat the phenotype as the latter.

The Task

Long-term behaviors of dynamical systems are of particular interest to agent-environment interaction (Beer, 1995). For this reason, the optimization problem – evolution of a logic gate by a CTRNN – is intended to represent a “generic” scenario where, from a dynamical systems perspective, a successful solution depends essentially on the phase portrait of the system; that is, a successful solution depends on the attractors – the long term behavior of the system. This problem has been adapted from a very similar problem tackled by Barnett (2002).

The activity of each ‘neuron’ in the CTRNN is given by the general form (see for detail Beer 1995a):

$$y_i' = \frac{1}{\tau_i} \left(-y_i + \sum_{j=1}^N w_{ji} \sigma(y_j + \theta_j) + I_i \right) \quad (1)$$

where y_i is the state of neuron i ; y_i' denotes the time rate of change of this state (i.e. dy/dt); τ_i is its time-constant ($\tau > 0$); I_i represents an external (sensory) input; w_{ji} is the strength of the connection from unit j to unit i and $\sigma(x)$ is a sigmoid:

$$\sigma(x) = \tanh(x) \quad (2)$$

with bias term (threshold) θ_j . In simulation, node activations are calculated forward through time by straight-forward time-slicing using Euler integration.

The experiments were performed for two logic gates: AND and XOR. The CTRNN has to solve the logic gate as follows: two nodes (node 1 and 2) are designated as inputs, and the sigmoided output of a third node as output

(hyperbolic tangent so it is guaranteed to lie between -1 and 1). A network trial consists of four runs of the network, corresponding to the particular logic task (see table 1).

Inputs were chosen as $low=-0.5$, $high=+0.5$ and target outputs were chosen as: $low=-\lambda$, $high=+\lambda$ with $\lambda=0.6$. Node activations are initialized for each of the four runs to zero, with the exception of the output node which is initialized with -0.2. The off-centre initial activation was found to aid finding solutions in the very similar problem attempted by Barnett (2002).

Input 1	Input 2	AND output	XOR output
<i>High</i>	<i>High</i>	<i>High</i>	<i>Low</i>
<i>High</i>	<i>Low</i>	<i>Low</i>	<i>High</i>
<i>Low</i>	<i>High</i>	<i>Low</i>	<i>High</i>
<i>Low</i>	<i>Low</i>	<i>Low</i>	<i>Low</i>

Table 1. AND and XOR Logic

All time-constants are set to 5 and the step size for Euler integration is 0.02. During a run the appropriate (constant) inputs are fed continuously to the input nodes. Networks are run for a stabilization period $T_s=500$ time steps to allow for the network to fall into an attractor state, followed by an evaluation period of a further $T_e=500$ time steps; the idea is that a network be evaluated on its steady state network. Networks receive a fitness value f based on the mean distance between output and target (i.e. distance integrated over the evaluation period) averaged over the four runs; specifically:

$$f = \left[1 + \frac{k}{4(1+\lambda)} \sum_{r=1}^4 \Delta^{(r)} \right]^{-1} \quad (3)$$

with mean distance between target and output for the r -th run defined by:

$$\Delta^{(r)} = \frac{1}{T_e} \sum_{i=T_s}^{T_s+T_e} \left| \sigma(x_3^{(r)}{}_i) - \lambda^{(r)} \right| \quad (4)$$

where $x_3^{(r)}{}_i$ is the activation of the output node and $\lambda^{(r)}$ the appropriate target for the r -th run. The constant ‘‘stretch factor’’ k may be deployed to control selection pressure. We used $k=9$ for our experiments. Note that maximum possible fitness is 1 and minimum possible fitness is $1/(1+k)=0.1$ since the maximum mean distance $\Delta^{(r)}$ for each of the four runs is $1+\lambda$.

Artificial Evolution: Netcrawler

The evolutionary technique used consists of a very simple population of 1+1 with a random mutation strategy, or netcrawler (Barnett, 2000). This technique has been originally designed as an optimum way to drift along fitness landscapes of high neutrality. It is here used as an intuitive measure for the usefulness of the neutrality taking place in the dynamics of evolving dynamical systems.

The original netcrawler consists of a hill-climber which is initially dropped into weight space according to a uniformly distributed random vector between $[0, 1]$. The individual is then mutated, and replaced if the mutated

individual is better or equally fit than the original. Given that the search space is real-valued it is unlikely that an equally fit individual be found. Therefore, in order to explore random drift, we experiment and compare slight variations to this technique. The variation consists in allowing individuals with certain slightly deleterious mutational loads to replace its ancestor. As the range of fitness is from 0.1 to 1, a 1% nearly-neutral netcrawler will be one were the current individual may be replaced by an individual who is at maximum 0.009 worst fit. A walk in a nearly-neutral network is considered when a slightly neutral replacement occurs. We conduct experiments with 0% (original netcrawler), 1%, 5%, 10%, 15% and 20% nearly neutral netcrawlers. The mutation operator used is a one locus per genotype variation taken from a uniformly random distribution in $[0, 1]$.

Neutrality Measurements

Despite the lack of an exact measurement for neutrality, Barnett has introduced several indicators which provide a meaningful idea of the presence of neutrality in fitness landscapes of discrete binary spaces. This work makes use of some of these measurements with certain adaptations for the real-valued scenarios. Due to space constraints only a brief account of the measurements will be provided here (for more detail see Barnett, 1997).

Neutral and Nearly Neutral Mutations: a mutation is neutral if the fitness of a genotype g and the fitness of a mutated version g' are equal or nearly equal.

The *Diffusion Coefficient* is an estimate of the velocity at which the population is moving in genotype space, as given by the squared Euclidean distance of the centroid of the population in adjacent generations. For our experiments, the centroid of the population is the genotype of the netcrawler as such. The squared Euclidean distance between centroids is defined by:

$$d_t = \sum_{i=1}^D (c_i^t - c_i^{t-h})^2 \quad (5)$$

where d_t is the distance traveled from generation $t-k$ to t , D is the size of the genome or the dimensionality of the genotypic search space, c_i^t represents the value of the centroid along the i -th axis at time t .

The *Lagged Diffusion Coefficient* is an estimate of the distance traveled by the population’s centroid in a certain time lapse. This is given by a lagged squared Euclidean distance between the centroid of the population at generation i and k generations prior, as defined by equation 5. In our experiments a time lag of $k=500$ was used.

Results

Results taken from several evolutionary runs using the experimental design described above are given in this section. The role of nearly-neutral mutations are interpreted and discussed with attention paid to its possible usefulness.

Firstly, the distribution of effects caused in fitness by one-locus mutations for 120 evolutionary runs using the

netcrawler to evolve the 3-neuron CTRNNs for the duration of 3000 generations was studied - a total of 360,000 mutations considered (see figure 1). From this we obtained that 78.8% were deleterious, 17.9% were advantageous and 3.3% were very close to exactly neutral¹.

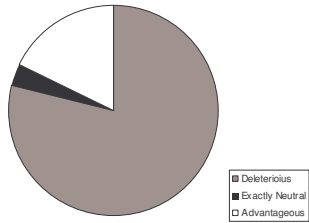


Figure 1. Proportion of fitness effects of a one-locus mutation: advantageous (white), deleterious (gray), and neutral (black).

Unsurprisingly, deleterious mutations are most common, given that in the face of random change fitness will most likely deteriorate. Despite the high proportion of deleterious mutations, the distribution of fitness effects of mutations shows that a great part of them cause only very small effects (figure 2). In fact, half the mutations cause an effect of less than 15% in fitness.

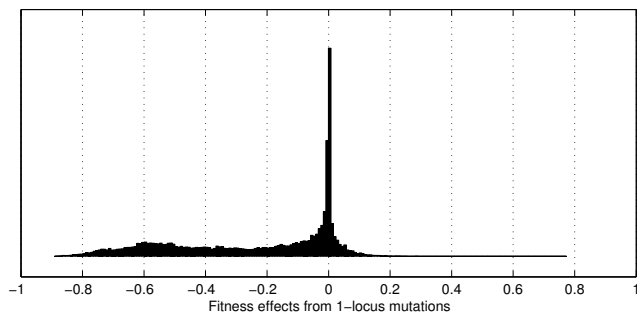


Figure 2. Distribution of fitness effects of mutations. Few mutations have large effects while most mutations have only slight effects on fitness.

As can be seen from the distribution, most changes in the real-valued genotype cause only slight changes in the fitness – as the fitness is given by the long-term behavior of the resulting dynamical system. Then this result hinges towards the potential for implicit redundancy in the genotype-phenotype mapping when evolving CTRNNs. So, what would happen if selection could not perceive a certain degree of too slight changes in the phenotype? Would the population engage in useful genetic drift? Or would individuals accumulate small deleterious mutations?

The neutral theory of molecular evolution has proven that slightly deleterious mutations are not necessarily part of adaptive evolution but can actually become part of random drift. To study the role of nearly-neutral mutations in the dynamics of evolution we allowed certain variations of effects on fitness not to be perceptible to selection. We

¹ The main factor causing the ‘exactly’ neutral effect is the limit of precision with which was experimented. Differences smaller than 10^{-5} are unperceived.

experimented and compared evolutionary processes with various ranges of nearly-neutral mutations for 20 runs each, starting at random using the netcrawler for 3000 generations on both problems: AND and XOR.

The mean fitness of the evolutionary runs using no-near neutrality, 1% and 5% near-neutrality are presented in Figure 3. As depicted, when slightly deleterious mutations are not perceived by the selection forces an improved evolutionary process is obtained. This implies that random drift plays a significant role in the evolutionary adaptation of the CTRNN.

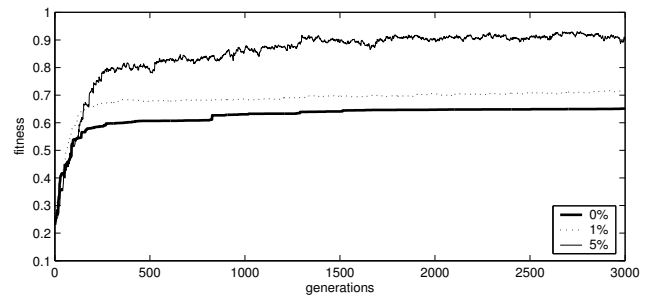


Figure 3. Comparison of averages fitness during evolution of original netcrawler solving the logic AND task against the modified versions of the algorithm which permit nearly neutral mutations.

However, increments in the percentage of selectively neutral mutations will cause an undesired evolutionary effect. This is due to the gradual accumulation of slightly deleterious mutations which in the long run has a detrimental effect on the fitness of the individuals. Figure 4 shows the mean fitness of evolutionary runs using 10%, 15% and 20% nearly neutral mutations.

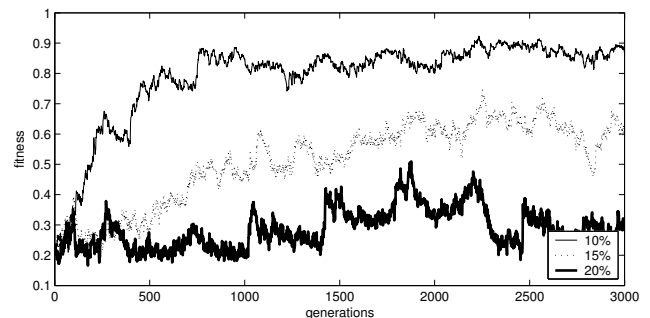


Figure 4. Average fitness of netcrawler solving logic AND task when bigger ranges of nearly neutral mutations are allowed.

At 10% the evolutionary process maintains a high enough fitness, regardless of certain temporary fitness losses. However, when increased to 15% the process becomes highly erratic. Finally with 20% the process becomes completely unsustainable. Although these effects are particular to this problem, we can generalize the effect of slightly neutral mutations as requiring a sort of balance, where a little enables useful drifting, and too much generates undesired deleterious accumulations. These results place great importance in the interaction and balance

to be maintained between random genetic drift and positive selection.

The importance of nearly neutral mutations is further exemplified by its potential for vanishing the picture of local optima in the abstraction of fitness landscapes. Whereas conventional evolutionary techniques assume that all or nearly all of the variation depends on the initial random population and recombination operators - and hence evolution will cease when this initial variation is exhausted; for the netcrawler the reverse is true, mutation is considered the driving force behind evolution (no recombination at all). For this approach, if the fitness landscape comprises local optima there is a high probability that the netcrawler will get stuck on it. Nevertheless, when nearly-neutral mutations are considered, the population is never stuck, but constantly moves through genotype space; when it is not climbing a fitness slope it will be drifting along a ‘nearly’ neutral network.

The picture of individuals engaged in useful drifting along nearly-neutral networks is supported when comparing the lagged diffusion coefficient among the different ranges of nearly-neutral mutation effects. Figure 5 shows the average movement of the netcrawler in genotype space when no nearly-neutral mutations are considered, with 5% and 20% ranges for neutrality from 20 evolutionary runs.

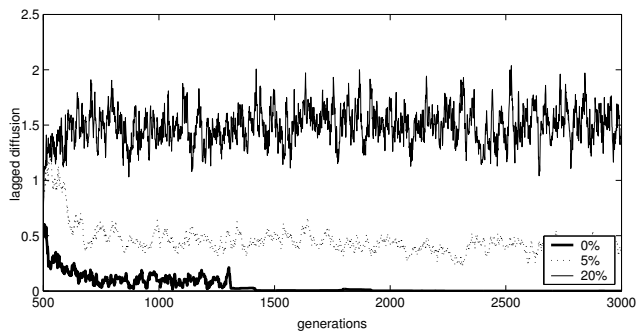


Figure 5. Lagged diffusion coefficient for different nearly-neutral ranges of mutations.

As can be appreciated, the exactly neutral netcrawler (0%) rapidly ceases to move or moves relatively very little after the first thousand generations, thus showing potential for local entrapment. The 5% nearly neutral keeps moving enough to reach higher fitness without accumulating too many deleterious mutations, thus improving the evolutionary search process. On the other extreme, the 20% nearly neutral shows too much random drift, which in turn doesn’t allow it to accumulate advantageous mutations (as shown by its poor evolutionary run in figure 4). Figure 6 depicts the drifting through nearly-neutral networks for a particular run using the 1% nearly neutral netcrawler. The dark dots represent the periods of selectively neutral drift and the thin vertical lines represent jumps across networks of higher fitness.

Figure 7 shows the average of the fitness of the last 500 generations on the experiments shown previously for each of the different ranges of nearly-neutral mutations. The actual data is represented in circles and the line is a simple

spline interpolant fitting of the data. An analogous result is shown for the evolutionary runs solving the XOR problem. Results for both problems indicate certain enhancement from the use of small amounts of drifting.

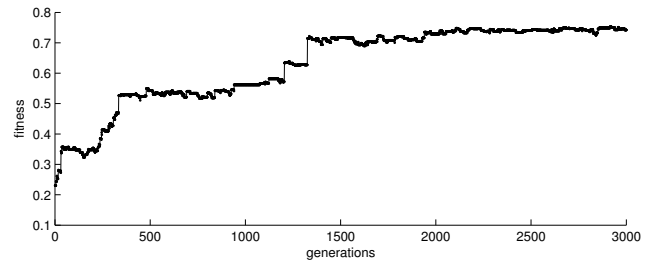


Figure 6. Evolutionary run from the 1% near neutrality netcrawler drifting through percolating nearly-neutral networks.

An important point that is made here is that big ranges of permissible drift results in individuals accumulating too many deleterious mutations. There lies a potentially dangerous use of neutrality which needs to be taken into account when evolving CTRNNs, despite ‘some’ random drift actually improving the evolutionary search.

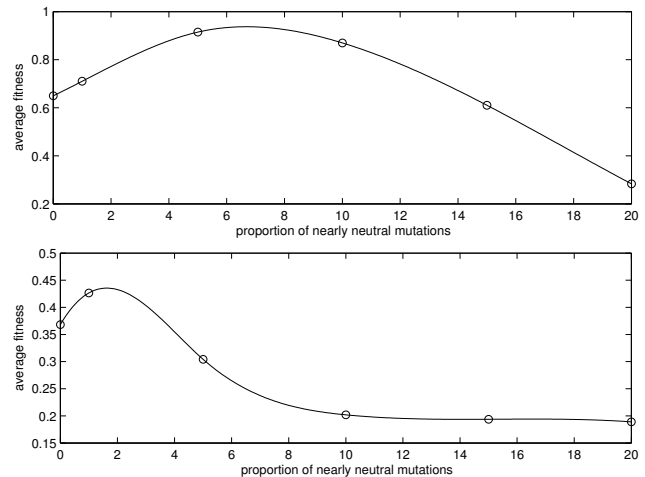


Figure 7. Average fitness on the last generations of evolved CTRNNs solving the logic AND (top) and XOR (bottom) problem using different ranges of nearly-neutral mutations.

Discussion

Although selection is the main force behind evolution, with this paper we wish to put emphasis on an approach to artificial evolution that takes into account other factors as well. In particular, we wish to point out that the approach to evolving dynamical neural networks may be greatly enhanced by random genetic drift. For this reason several issues have been addressed. Firstly, this paper has sketched the distribution of fitness effects of new mutations when evolving CTRNNs in real-valued search spaces. The results show that a big proportion of mutations cause little effect on fitness. Our results agree to a certain extent with recent theoretical work from molecular biology which states that the distribution of mutational effects on fitness is

leptokurtic. This has led us to explore the limits of nearly neutral mutations and how they affect this evolutionary process. From neutral theory, we know that small changes in fitness (less than $1/K$ where K is the size of the population) do not cause a selectively effective advantage or disadvantage. For simplicity we have experimented with an extreme evolutionary case where the population is of size $1+1$. Thus, unless explicitly stated any change in the population will fix, no matter how small or in fact neutral. This has allowed us to explore the ranges of selectively neutral effects from new mutations. We are, however, very careful of extrapolating our conclusions to (larger) populations. Our results show that ranges of effects of near neutrality from 1% to 10% actually improve the evolutionary process for the tasks treated here. We observed that the process of drifting through phenotypes of nearly-neutral fitness is reminiscent of biological neutral networks. Additionally, we observe that if the range of near neutrality is too big then accumulation of deleterious mutations increases and the population actually deteriorates significantly. The most important generalization drawn from this study is not the ranges which worked better for these problems but the fact that genetic drift plays an important role in the evolutionary search of dynamical systems. We put forward the hypothesis that this is due to an implicit redundant genotype \rightarrow phenotype mapping. Future work should be directed towards determining the exact relation between the genotype and the behavior when evolving dynamical systems.

Conclusions

In this paper we have done the following:

1. Emphasized the importance of nearly neutral networks and random genetic drift in artificial evolution.
2. Pointed out the implicit redundancy in the genotype-phenotype mapping when evolving dynamical neural network.
3. Considered the possibility that, for the case of evolving real-valued CTRNNs, useful neutrality comes for free.
4. Analyzed the importance of the distribution of fitness effects from mutations in the dynamics of evolution.
5. Highlighted the use of mutation as the driving force behind evolution in landscapes featuring neutrality.

All of these issues deserve further development. They have been presented here in full knowledge of this fact. Nonetheless, if this paper raises awareness of the importance of near-neutrality in the dynamics of artificial evolution of dynamical systems then its objectives will have been accomplished.

Acknowledgements

The author thanks Inman Harvey for insightful discussions. The author is supported by Programme Alβan, the European Union Programme of High Level Scholarships for Latin America, identification number E03M12883VE.

References

- Babajide, A., Hofacker, I., Sippl, M., and Stadler, P.F. 1997. Neutral networks in protein space: A computational study based on knowledge-based potentials of mean force. *Folding Design*, 2, 261-269.
- Barnett, L. 1997. Tangled webs: Evolutionary dynamics on fitness landscapes with neutrality. M.Sc. diss., University of Sussex. UK.
- Barnett, L. 2000. Netcrawling – optimal evolutionary search with neutral networks. In Proceedings of the 2001 Congress on Evolutionary Computation.
- Barnett, L. 2002. Explorations in evolutionary visualisations. In *Artificial life VIII Workshops. The 8th International Conference on the Simulation and Synthesis of Living Systems*.
- Beer, R.D., and Gallagher, J.G. 1992. Evolving dynamical neural networks for adaptive behavior. *Adaptive Behavior* 1:92-122.
- Beer, R.D. 1995a. On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior* 3(4):475-511.
- Beer, R.D. 1995b. A dynamical system perspective on agent-environment interaction. In *Artificial Intelligence* 72. 173-215.
- Cliff, D., Husbands, P., & Harvey, I. 1993. Evolving visually guided robots. In Meyer, J.-A., Roitblat, H., and Wilson, S. (Eds.), *From Animals to Animats 2: Proc. of the Sec. Intl. Conf. on Simulation of Adaptive Behaviour, (SAB92)*. 374-383.
- Harvey, I. and Thompson, A. 1996. Through the labyrinth evolution finds a way: A silicon ridge. In *Proc. 1st Internatl. Conf. Evol. Sys.: From Biology to Hardware (ICES 96)*. Springer-Verlag.
- Harvey, I., Husbands, P., Cliff, D., Thompson, A. and Jakobi, N. 1997. Evolutionary robotics: the Sussex approach. In *Robotics and Autonomous Systems*, v. 20: 205-224.
- Jakobi, N. 1995. Facing the facts: necessary requirements for the artificial evolution of complex behaviour. *Cognitive Science Research Paper 422*. University of Sussex. UK.
- Kimura, M. 1983. *The Neutral Theory of Molecular Evolution*. Cambridge, UK: Cambridge University Press.
- Piganeau, G. and Eyre-Walker A. 2003. Estimating the distribution of fitness effects from DNA sequence data: implications for the molecular clock. *Proc Natl Acad Sci U S A*; 100(18):10335-40.
- Shackleton, M., Shipman, R. and Ebner M. 2000. An investigation of redundant genotype-phenotype mappings and their role in evolutionary search. In *Proceedings of the International Congress on Evolutionary Computation*. IEEE Press. pp:493-500.
- Shuster, P., Fontana, W., Stadler, P.F., and Hofacker, I. 1994. From sequences to shapes and back: A case study in RNA secondary structures. *Proc. Roy. Soc (London) B*, 255:279-284.
- Smith, T., Husbands, P. and O'Shea, M. 2001. Neutral networks and evolvability with complex genotype-phenotype mapping. *European Conference on Artificial Life: ECAL2001*. 272-281. Springer
- Smith, T., Philippides, A., Husbands, P. and O'Shea, M. 2002. Neutrality and ruggedness in robot landscapes. *Congress on Evolutionary Computation: CEC2002*. pp. 1348-1353. IEEE Press.
- Zendric-Ashmore, L.A. 2003. An investigation into the benefits of neutrality within real-valued search landscapes. MSc. diss. University of Sussex. UK.