Local Evolvability of Statistically Neutral GasNet Robot Controllers

Tom Smith[*1,2], **Phil Husbands**[1,3] **and Michael O'Shea**[1,2]

[1]Centre for Computational Neuroscience and Robotics (CCNR)

[2]School of Biological Sciences

[3]School of Cognitive and Computing Sciences

University of Sussex, Brighton, UK

[*]toms@cogs.susx.ac.uk

Abstract

In this paper we introduce and apply the concept of *local evolvability* to investigate the behaviour of populations during evolutionary search. We focus on the evolution of *GasNet* neural network controllers for a robotic visual discrimination problem, showing that the evolutionary process undergoes long neutral fitness epochs. We show that the local evolvability properties of the search space surrounding a group of *statistically neutral* solutions do vary across the course of an evolutionary run, especially during periods of population takeover. However, once takeover is complete there is no evidence for further increase in local evolvability across fitness epochs. We also see no evidence for the neutral evolution of increased solution robustness, but show that this may be due to the ability of evolutionary algorithms to focus search on volumes of the fitness landscape with above average robustness.

Keywords: Local Evolvability, Neutral Evolution, Fitness Landscape, Evolutionary Algorithm, Evolutionary Robotics, Solution Robustness.

1 Introduction

In this paper, we investigate the dynamics of evolutionary search on the large, noisy, heterogenous fitness landscape underlying an evolutionary robotics experiment. We show that the evolution of successful controllers can be characterised by long fitness epochs during which the best

individuals in the population move along statistically neutral networks, interspersed by short periods of fitness increase. By defining *local evolvability* measures of the search space surrounding solutions, we investigate whether the properties of a set of statistically neutral solutions change during the course of an evolutionary run. We also investigate the difference between the statistically neutral solutions generated during the evolutionary process, and statistically neutral solutions encountered over a non-adaptive neutral walk.

1.1 Local evolvability and the transmission function

Evolvability is loosely defined as the capacity to evolve, alternatively the ability of an individual or population to generate fit variants (Altenberg, 1994; Marrow, 1999; Wagner and Altenberg, 1996). Evolvability is therefore more closely allied with the potential for fitness than with fitness itself; two equal fitness individuals or populations may have very different evolvabilities (Turney, 1999). Closely related definitions of evolvability explicitly make the link with increase in solution complexity over time (Nehaniv, 2000), and the evolution of the genotype-to-phenotype mapping (McMullin, 2000).

It has also been argued that there may be trends for evolvability to increase during evolution (Altenberg, 1994; Wagner and Altenberg, 1996; Dawkins, 1989; Turney, 1999). However, as evolvability is more directly related to fitness potential than fitness itself, long-term change cannot be due to straight fitness selection. Thus any trend towards change in evolvability can only be understood through some higher order selection mechanism, by which evolution tends to retain solutions, or their descendants, with more evolvable genetic systems (Dawkins, 1989; Kirschner and Gerhart, 1998, argue for such mechanisms under which evolvability may be selected for).

Researchers in both biology and evolutionary computation often link evolvability with the properties of the search space. For example, Burch and Chao (2000) show that the evolvability of RNA viruses can be understood in terms of their mutational neighbourhood, while many evolutionary computation researchers (see e.g. Ebner et al., 2001; Marrow, 1999) argue that changing the properties of the search space (through such mechanisms as adding neutrality) can affect evolvability as evidenced by the speed of evolution. The interest in evolvability for evolutionary computation practitioners is thus tied closely to work on properties of the search space which influence the ease of finding good solutions in the space (Weinberger, 1990; Hordijk, 1996; Jones and Forrest, 1995; Naudts and Kallel, 2000).

In this paper, we identify the *local evolvability* of solution(s) as some measure based on the fitness of offspring from those solution(s). We define a set of four metrics of local evolvability (see Smith et al., 2002a, for further details):

$$E_a = P(F_o \geq F_p) \quad (1)$$

$$E_b = < F_o > \quad (2)$$

$$E_c = < F_o >^{75,100} \quad (3)$$

$$E_d = < F_o >^{0,25} \quad (4)$$

where F_p, F_o are the fitnesses of the parent and offspring solutions, and $< F_o >^{75,100}, < F_o >^{0,25}$ the expected fitnesses over the top and bottom quartiles of the offspring solutions. Thus E_a is the probability of offspring solutions being of greater or equal fitness to the parent, E_b the expected offspring fitness, E_c the expected fitness over the top quartile of offspring fitnesses, and E_d the expected fitness over the bottom quartile of offspring fitnesses.

The four metrics can be similarly defined over some population, e.g. a population of solutions at some time-point during evolution, as the local evolvability calculated over offspring from all individuals in that population. In Smith et al. (2002a) we have used the metrics calculated over samples of equal fitness solutions to derive *fitness evolvability portraits* for the set of NK (Kauffman, 1993) and terraced NK landscapes (Newman and Engelhardt, 1998). The metrics are plotted against fitness, and used to describe the ruggedness, local modality and neutrality of the spaces.

This tight definition of local evolvability in terms of the search space surrounding solutions (or populations of solutions) allows us to track the behaviour of populations during evolution. In particular, we focus on a population of solutions that are indistinguishable from each other in terms of their fitness, investigating whether they move to areas of the search space that are distinguishable in terms of their local evolvability properties. See section 6 for experimental details on calculating the evolvability metrics.

1.2 Adaptive evolution on neutral networks

In the neutral theory of molecular evolution, Kimura (1983) argues that the majority of genotypic mutations may be selectively neutral. There is increasing evidence for such neutral evolution

in a number of fields, including RNA secondary structure (Grüner et al., 1996), bacteria (Elena et al., 1996), evolvable hardware (Vassilev and Miller, 2000), and digital organisms (Adami, 1995). Evolution on fitness landscapes with high levels of neutrality is typically characterised by periods during which fitness does not increase, or *fitness epochs*, interspersed by short periods of rapid fitness increase, i.e. *epochal evolution* (van Nimwegen et al., 1997; Crutchfield and van Nimwegen, 1999) or *punctuated equilibrium* (Eldredge and Gould, 1972; Gould and Eldredge, 1977).

During fitness epochs, genotype structure may not be conserved, as the evolving population moves through networks of connected equal fitness solutions, or *neutral networks*. Eventually, a *portal* genotype of higher fitness may be discovered, and the population moves up to the higher fitness neutral network; the time spent in neutral evolution is related to the size of the network and the number of portal genotypes (argued by van Nimwegen and Crutchfield, 2000, to be an *entropy barrier*, very different to the *fitness barrier* of low fitness genotypes that must be passed to escape from local optima). Despite the undirected nature of the neutral movement, genotypic change may lay down structure required for the final portal genotype to be discovered.

Recent work describing the population dynamics of evolving populations on neutral networks has argued that with large enough populations and mutation rates, the population will tend to move towards volumes of the neutral network where solutions have more neutral neighbours than on average across the neutral network, i.e. the neutral evolution of robustness (van Nimwegen et al., 1999). Similarly, Wilke (2001) shows epochal evolution in the *average fitness* of the population, despite the highest fitness in the population remaining constant; the fitter genotypes move neutrally to *flatter* areas of the search space containing more neutral neighbours. Thus the average population fitness increases despite the highest fitnesses remaining fixed; evolution of populations can be adaptive even during neutral fitness epochs. As in the evolution of evolvability, any adaptation during neutral movement can be explained through higher-level selection, in which solutions (or their descendants) of greater robustness are likely to be retained in the evolutionary population.

Having defined our metrics of local evolvability over the fitness distribution of solution offspring, we are now in a position to investigate the behaviour of populations evolving along neutral networks in an extremely noisy genotype-to-fitness mapping space, in which the initial genotype translates to an intermediate neural network phenotype, with the final fitness measuring how well this network performs over time in controlling a robot engaged in solving a visual

shape discrimination task.

2 An evolutionary robotics search space

One of the new styles of Artificial Intelligence to have emerged recently is evolutionary robotics (Cliff et al., 1993; Nolfi and Floreano, 2000; Floreano and Mondada, 1994; Husbands and Meyer, 1998). The evolutionary process involves evaluating, over many generations, whole populations of robot control systems specified by artificial genotypes. These are interbred using a Darwinian scheme in which the fittest individuals are most likely to produce offspring. Fitness is measured in terms of how good a robot's behaviour is according to some evaluation criterion.

Artificial neural networks (ANNs) have been successfully used in a large number of evolutionary robotics experiments (for an overview see Nolfi and Floreano, 2000). Typically, external sensory data is used for the network input, and the network output is used to control the robot motors. Other styles of control architectures have also been used for evolutionary robotics experiments, notably genetic programming (Koza, 1992), classifier systems (Holland, 1975) and evolvable hardware (Thompson, 1998). In previous work, we have investigated the use of non-standard neural network architectures, focusing on developing control structures that produce successful solutions in fewer evaluations using artificial evolution (Husbands et al., 1998). In experiments on a variety of robotics tasks, we have shown that a particular style of network, the "GasNet", is particularly amenable to evolutionary search.

2.1 The GasNet architecture

The GasNet is an arbitrarily recurrent ANN augmented with a model of diffusing gaseous modulation, in which the instantaneous activation of a node is a function of both the inputs from connected nodes and the current concentration of gas(es) at the node. Thus in addition to the standard electrical activity 'flowing' between nodes, an abstract process analogous to the diffusion of gaseous modulators such as Nitric Oxide is at work (Philippides et al., 2000). In this process, the virtual gases do not alter the electrical activity in the network directly but rather act by changing the gain of transfer function mapping between node input and output in a concentration dependent manner.

The network underlying the GasNet model is a discrete time-step, recurrent neural network with a variable number of sigmoid transfer function nodes. These nodes are connected by either

excitatory (with a weight of +1) or inhibitory (with a weight of -1) links with the output O_i^n, of node i at time-step n determined by a continuous mapping from the sum of its inputs, as described by the following equation:

$$O_i^n = \tanh\left[k_i^n\left(\sum_{j\in C_i} w_{ji}O_j^{n-1} + I_i^n\right) + b_i\right] \tag{5}$$

where C_i is the set of nodes with connections to node i and $w_{ji} = \pm 1$ depending on whether the link is excitatory or inhibitory and multiplies the input from node j (which is the output from node j from the previous time-step). I_i^n is the external (sensory) input to node i at time n, and b_i is a genetically set bias. Each node has a genetically set default transfer function parameter k_i^0, which can be altered at each time-step by the concentration of the diffusing virtual gas at node i to give k_i^n.

2.2 Gas diffusion in the networks

In order to incorporate the gas concentration model, the network is placed in a 2D plane, with node positions specified genetically. The GasNet diffusion model is controlled by two genetically specified parameters namely the radius of influence r around the emitting node, and the rate of build up and decay s. Spatially, the gas concentration varies as an inverse exponential of the distance from the emitting node with a spread governed by r, with the concentration set to zero for all distances greater than r (equation 6). This is loosely analogous to the length constant of the natural diffusion of Nitric Oxide, related to its rate of decay through chemical interaction. The maximum concentration at the emitting node is one and the concentration builds up and decays from this value linearly as defined by equations 7 and 8 at a rate determined by s. The governing equations are:

$$C(d,t) = \begin{cases} e^{-2d/r} \times T(t) & d < r \\ 0 & \text{else} \end{cases} \tag{6}$$

$$T(t) = \begin{cases} H\left(\frac{t-t_e}{s}\right) & \text{emitting} \\ H\left[H\left(\frac{t_s-t_e}{s}\right) - H\left(\frac{t-t_s}{s}\right)\right] & \text{not emitting} \end{cases} \tag{7}$$

$$H(x) = \begin{cases} 0 & x \le 0 \\ x & 0 < x < 1 \\ 1 & \text{else} \end{cases} \tag{8}$$

where C(d,t) is the concentration at a distance d from the emitting node at time t. t_e is the time at which emission was last turned on, t_s is the time at which emission was last turned off, and s (controlling the slope of the function T) is genetically determined for each node. To summarise, within a radius of r from the node, gas builds up (and decays) linearly to a maximum of $e^{-2d/r}$ in s time-steps. The total concentration at a node is then determined by summing the concentrations from all other emitting nodes (nodes are not affected by their own concentration, to avoid runaway positive feedback).

2.3 Modulation by the gases

There are two virtual gases in the network, gas 1 and gas 2, which increase and decrease k_i^n (see equation 5) respectively in a concentration dependent fashion. Both the type of gas emitted by a node and the conditions under which it emits are specified genetically. Nodes emit either (a) gas 1, (b) gas 2 or (c) no gas, and emission occurs when either (a) the node activity increases beyond the electrical threshold 0.5, or (b) the local concentration of gas 1 increases beyond the threshold 0.1, or (c) the local concentration of gas 2 increases beyond the threshold 0.1. The concentration-dependent modulation is described by equations 9 to 12, with transfer parameters updated on every time-step as the network runs. Thus we have:

$$k_i^n = \mathbf{P}[ind_i^n] \tag{9}$$

$$\mathbf{P} = \{-4.0, -2.0, -1.0, -0.5, -0.25, 0.0, 0.25, 0.5, 1.0, 2.0, 4.0\} \tag{10}$$

$$ind_i^n = f\left(ind_i^0 + C_1^n(N - ind_i^0) - C_2^n ind_i^0\right) \tag{11}$$

$$f(x) = \begin{cases} 0 & x \leq 0 \\ \lfloor x \rfloor & 0 < x < N \\ N & \text{else} \end{cases} \tag{12}$$

where $\mathbf{P}[i]$ refers to the ith element of set $\mathbf{P}$, ind_i^n is node i's index into the set $\mathbf{P}$ of possible discrete values k_i^n can assume, N is the number of elements in $\mathbf{P}$, ind_i^0 is the genetically set default value for ind_i, C_1^n is the concentration of gas 1 at node i on time-step n and C_2^n is the concentration of gas 2 at node i on time-step n. Both gas concentrations lie in the range $[0, 1]$.

Thus, the concentration of each gas is directly proportional to any change in ind_i^n, with a corresponding change in k_i^n. Although the change in k_i^n is non-linear these values represent a smooth change in the slope of the transfer function. Since the transfer functions can change

throughout the lifetime of the network, this system provides a form of network plasticity not seen in most other ANNs.

2.4 Visual shape discrimination

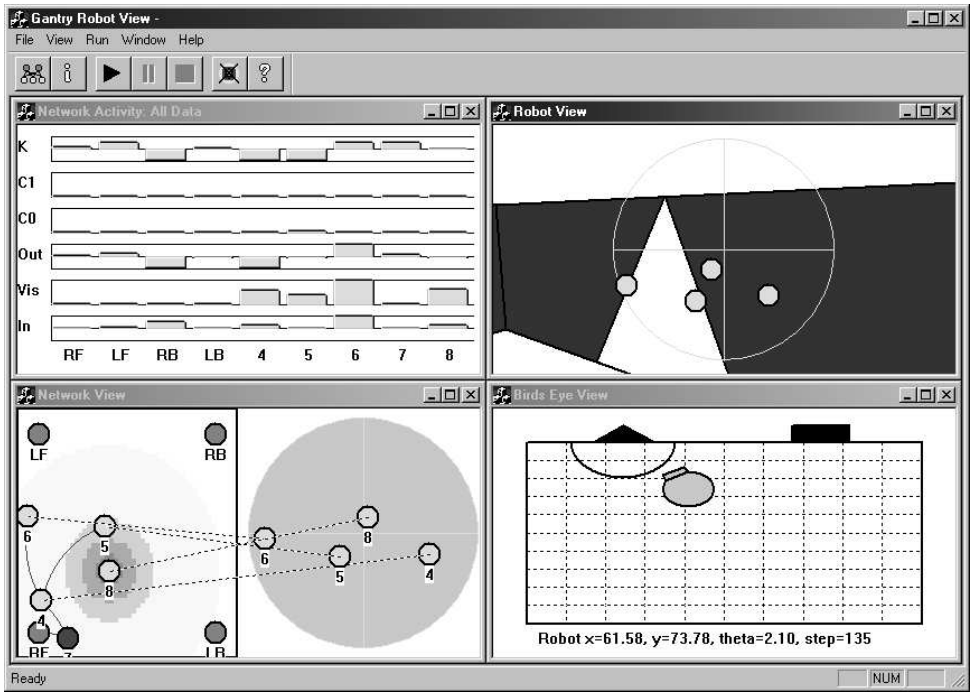


Figure 1: Screen shot of the simulated arena and robot. The bottom-right view shows the robot position in the arena with the triangle and square. Fitness is evaluated on how close the robot approaches the triangle. The top-right view shows what the robot 'sees', along with the pixel positions selected by evolution for visual input. The top-left view shows the instantaneous activity of all nodes in the neural network. The bottom-left view shows the robot control neural network.

The evolutionary task at hand is a visual shape discrimination task; starting from an arbitrary position and orientation in a black-walled arena, the robot must navigate under extremely variable lighting conditions to one shape (a white triangle) while ignoring the second shape (a white square). Fitness over a single trial was taken as the fraction of the starting distance moved towards the triangle by the end of the trial period, and the evaluated fitness was returned as the average over 16 trials of the controller from different initial conditions:

$$F = \frac{1}{136} \sum_{i=1}^{i=16} i(1 - \frac{D_i^F}{D_i^S}) \tag{13}$$

where D_i^F is the distance to the triangle at the end of the ith trial, and D_i^S the distance to

the triangle at the start of the trial, and the i trials are sorted in descending order of $1 - \frac{D^F}{D^S}$. Thus good trials, in which the controller moves some way towards the triangle, receive a smaller weighting than bad trials, encouraging robust behaviour on all 16 trials.

Success in the task was taken when an evaluated fitness of 1.0 was obtained over thirty successive generations of the evolutionary algorithm. In the work reported here, fitness evaluations are carried out in a verified *minimal simulation* (Jakobi, 1998), see figure 1 for screen-shot of a fitness evaluation in simulation. Evolved controllers have been successfully transferred to the real robot (Husbands et al., 1998). As in many problems requiring controllers to provide sensor-to-motor mappings over time, fitnesses are extremely time consuming to evaluate (in the work presented here, evaluating a sample of 10^6 fitnesses takes around 24 hours on a Pentium II 700MHz machine) and inherently extremely noisy.

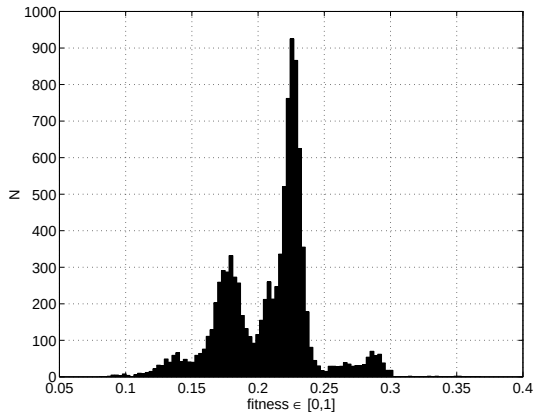


Figure 2: The fitness distribution of a single genotype evaluated $10,000$ times in the minimal simulation evaluation environment. 95% of the fitnesses lie in the range $[0.1343, 0.2856]$, with possible controller fitness $\in [0, 1]$.

Figure 2 shows the distribution of fitnesses from a single controller over $10,000$ evaluations. It should be emphasised that the environmental noise for the robot controllers is not simply variation in the received fitness score, but is a crucial feature of the robot minimal simulation model. Controllers must evolve to be robust to such noise, so as to successfully transfer to the real world; two controllers may be of equal fitness when evaluated in a noiseless environment, but may be of very different fitnesses in the full noise model. Although the level of noise in the model is higher than that found in the real world, to successfully evolve robot controllers in simulation able to operate successfully in the real world this noise is necessary (Jakobi, 1998). Since we are interested in the properties of landscapes defined by genotype-to-fitness mappings that are used in real problems, we need to develop ways of understanding these kinds of landscape.

2.5 The solution representation

The neural network robot controllers were encoded as variable length strings of integers, with each integer allowed to lie in the range $[0, 99]$. Each node in the network was coded for by nineteen parameters, controlling such properties as node connections, sensor input, node bias, and all the variables controlling gas diffusion as described in sections 2.2 and 2.3. Both the robot control network, an arbitrarily recurrent ANN, and the robot sensor input morphology, i.e. the position of the input pixels on the visual array, were under evolutionary control. Thus mutation of solutions (section 2.6) is able to produce offspring with varying network architecture, network node properties and sensor morphology. In all experiments, the GA population were initially seeded with networks containing ten neurons. For further details see Husbands et al. (1998); Smith and Philippides (2000).

2.6 The evolutionary algorithm and mutation operator

A distributed asynchronously updating evolutionary algorithm was used, with a population of 100 solutions arranged on a 10×10 grid. Fitness was awarded on the fraction of the distance moved towards the triangle over a series of 16 runs with different initial conditions, see equation 13. Parents were chosen with probability proportional to their fitness ranked in ascending order over the mating 'pool' consisting of a randomly chosen grid-point plus its eight nearest neighbours. The parent solution was mutated to create the offspring solution which was placed back in the mating pool, replacing a solution chosen with probability proportional to their fitness ranked in descending order over the mating pool. One generation was specified as 100 such breeding events. Figure 3 shows the pseudo-code for the evolutionary algorithm.

Three mutation operators were applied to solutions with probability $\mu\%$ during evolution (for the experiments detailed here, $\mu = 4$). First, each integer in the string had a $\mu\%$ probability of mutation in a Gaussian distribution $N(0, 10)$ centred on its current value (20% of these mutations completely randomised the integer). Second, there was a $\mu\%$ chance per *genotype* of adding one neuron to the network, i.e. increasing the genotype length by 19. Third, there was a $\mu\%$ chance per genotype of deleting one randomly chosen neuron from the network, i.e. decreasing the genotype length by 19. It should be noted that the value of $\mu = 4$ used in these experiments is a much larger level of mutation than typically used in artificial evolution optimisation (and certainly much larger than in biological evolution). However, lower levels

```
- Initialise population of 100 solutions on 10x10 grid.
- Evaluate each solution fitness.
- Repeat until success criterion met, or MaxGenerations reached:
  - Repeat 100 times for 1 generation:
    - Select solution at random.
    - Create mating pool of solution plus 8 nearest grid neighbours.
    - Pick parent P through ascending rank-based selection on mating pool.
    - Create offspring O through mutation of P, and evaluate fitness.
    - Place O in solution grid, replacing mating pool solution picked
            through descending rank-based selection.
```

Figure 3: Pseudo-code for the asynchronously updating evolutionary algorithm.

of mutation produce extremely slow evolution of successful solutions (Smith et al., 2002b); in section 6.1 we see that the number of neutral mutations at this mutation rate is still significant.

3 Statistical neutrality

In section 1.2 we discussed the possible adaptation of populations during neutral movement fitness epochs. However, as we have seen in the previous section, the fitness landscape defined by the GasNet controllers is inherently extremely noisy; neutrality in such a landscape is not trivial to define. Even the repeated evaluation of the same controller produces a wide range of possible scores (figure 2), so how can we identify different solutions of equal fitness?

In this paper we use the concept of statistical neutrality, based on solution fitness distributions. Two solutions are defined to be statistically neutral if it is likely that the two distributions of fitnesses are drawn from the same distribution. More precisely, two fitness distributions are defined to be statistically neutral if we fail to reject the null hypothesis that they are not significantly different. Such a probabilistic definition of neutrality seems practical for the noisy fitness landscape we are investigating, and owes much in spirit to the "nearly neutral" theory (Ohta, 1992), where small fitness differences may be swamped by finite population sampling effects. Even if real underlying fitness differences exist between our statistically neutral solutions, they are unlikely to be observed by the evolutionary process.

In practice, we generate small distributions of fitnesses for each sample in the population, and

compare with the fitness distribution from some test solution. If the Student t-test (Press et al., 1992) probability for the sample and test distributions being drawn from the same distribution is not smaller than some value ν_S, we argue that the sample and test solutions can be considered to be statistically neutral. Although this statistical definition of neutrality will not identify every solution correctly - some solutions of equal fitness may be rejected while others of different fitness may be included, we argue that it has merit when considering the neutrality of solutions in noisy fitness landscapes. In the experiments carried out in the next section we take $\nu_S = 0.1$, i.e. two solutions are said to have different fitnesses if there is less than 10% probability of their fitness distributions being drawn from a single distribution. The two solutions are statistically neutral otherwise.

4 Fitness epochs in GasNet evolution

In previous work, we have shown the amenability of the GasNet architecture to evolutionary search for good robot controllers (Husbands et al., 1998; Smith and Philippides, 2000). An ongoing project aims to explain the observed evolutionary speed difference in terms of the properties of the underlying fitness landscapes of the GasNet and other neural network architectures.

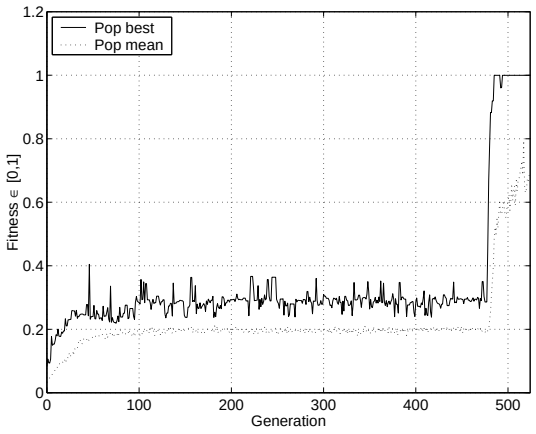
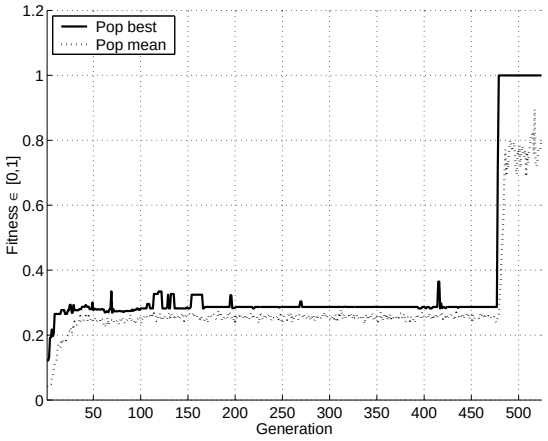


Figure 4: Best and mean fitness of the population over generations for a single GasNet evolutionary run.

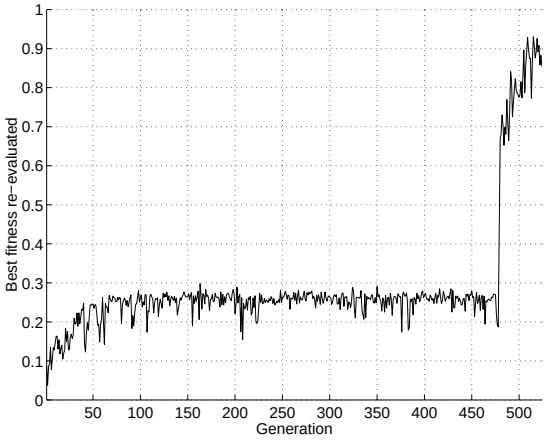
Figure 4 shows a typical GasNet evolutionary run, with the population best and mean evaluated fitnesses plotted over time. Both fitnesses climb quickly from an initial near-zero random performance, with a long period of apparent stasis (albeit a noisy stasis), before the best fitness reaches 100% around generation 480. In terms of robot controller behaviour, the period of apparent stasis corresponds to approaching the first white object seen in the arena, thus hitting the triangle on approximately 50% of evaluations (the selective fitness shown in figure

4 is below 50% due to the weighting on evaluations, see section 2.4). The high fitnesses seen after generation 480 correspond to approaching the triangle on every single evaluation, while the lower fitnesses seen in the first few generations correspond to ballistic behaviours such as "move straight forward". Other GasNet evolutionary runs show similar behaviour; long periods of apparent stasis interspersed with short periods of fitness increase, although in many cases there is more than one period of apparent stasis.

4.1 Apparently neutral fitness epochs



(a) Best and average fitnesses evaluated in a noiseless environment



(b) Best fitness found in the population, with fitness set as the mean evaluated fitness over ten evaluations in a noisy environment

Figure 5: **(a)** Best and mean fitness of the population over generations from the same evolutionary run shown in figure 4, with fitness evaluated in a simulated environment without sensor and motor noise. **b** The best fitness found in the population from the same evolutionary run, with fitness set as the mean evaluated fitness over ten evaluations in a noisy environment.

In figure 5(a) we show the best and mean fitnesses over the same evolutionary run shown in figure 4, with fitness evaluated in a noiseless environment. It appears to be the case that fitness reaches a static level very quickly, around generation 10, then stays constant until generation 477 with a few short-lived higher fitness 'blips'. However, although the noiseless environment fitness evaluation scenario lends support to the hypothesis that the apparent stasis period is indeed a fitness epoch, it is not nearly the full story. In figure 5(b), we show the best fitness found in the population over time, with the fitness of every individual calculated as the mean

fitness over ten evaluations in a noisy environment. Two points need to be made. First, the best fitness in the population in a noisy environment is clearly increasing after the generation at which the noiseless environment reached apparent fitness stasis, up to at least generation 60. Second, the best fitness over the period from generations 100 to 477 appears to remain constant, and there is also some evidence for the neutral period to start earlier, around generation 60.

Although figure 5 strengthens the hypothesis that the fitness of the best solutions really does not change over long periods of the evolutionary run, we need to apply the notion of statistical neutrality to see whether this really holds up.

4.2 Statistically neutral fitness epochs

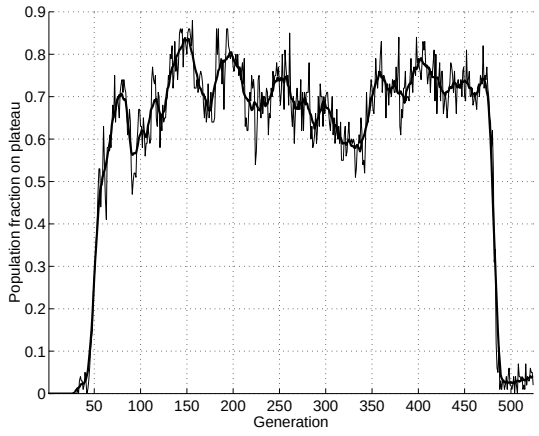


Figure 6: The fraction of the evolutionary population over generations that was statistically neutral with respect to a single test solution picked at random from the best individuals between generations 200 and 400. Solutions were considered statistically neutral if 15 fitness evaluations were not significantly different to 15 evaluations from the test solution (Student t-test, $P > 0.1$).

Figure 6 shows the fraction of the population at each generation that is statistically neutral with respect to a single test solution picked at random from the best genotypes between generations 200 and 400. Solutions were considered statistically neutral if 15 fitness evaluations were not significantly different to 15 evaluations from the test solution (Student t-test, $P > 0.1$). Solutions statistically fitter and less-fit than this test solution were identified as those solutions not statistically neutral, with higher or lower mean fitnesses respectively. Table 1 shows the total number of statistically neutral (and statistically fitter and less-fit) solutions over the population run. Three points can be made. First, from generation 32 onwards, a large number of fit solutions arose that were not significantly distinguishable from each other in terms of their fitnesses. Second, the proportion of these solutions in the population increased rapidly over generations 32 to 75, comprising approximately 70% of the population until around generation 480. Third, the numbers of solutions distinguishably fitter than this group was extremely small (32 out of

47,800 genotypes over the first 478 generations), which is plausibly due to the sampling error in evaluating the inherently noisy fitness of the solutions.

	N (fraction)
Number of solutions over evolutionary run	52,500
Number statistically neutral w.r.t test solution	30,235 (0.576)
Number statistically fitter than test solution	3,328 (0.063)
Number statistically fitter than test solution before generation 478	32 (0.001)
Number statistically less-fit than test solution	18,937 (0.361)

Table 1: The number of the evolutionary population that was statistically neutral, fitter or less-fit with respect to a single test solution picked at random from the best individuals between generations 200 and 400 over the evolutionary run shown in figure 4. Solutions were considered statistically neutral if 15 fitness evaluations were not significantly different to 15 evaluations from the test solution (Student t-test, $P > 0.1$). Solutions statistically fitter and less-fit than this test solution were identified as those solutions not statistically neutral, with higher or lower mean fitnesses respectively

A final point remains to be cleared up. Why does the best fitness of the population increase markedly up until around generation 60 (figure 5(b))? This again is plausibly due to sampling error; as the number of solutions on this statistically neutral fitness band increases (figure 6), the expected best fitness increases monotonically. Figure 7 shows the expected highest fitness drawn from the fitness distribution of a single genotype (figure 2), against the number of drawn samples. As the number of statistically neutral fitter genotypes increases from 1 to 70, the expected best fitness increases from 0.21 to 0.29; consistent with the fitness increase seen in the real evolutionary population.

We are now in a position to give a preliminary description of the evolutionary run: From the initial random population we quickly evolve fixed ballistic solutions that approach the triangle on some of the evaluations. Around generation 32 we see the first emergence of controllers able to approach bright objects, so finding the triangle on 50% of evaluations. Controllers showing this behaviour rapidly take over the population, leading to a long fitness epoch. Finally, on generation 477 we see the evolution of controllers able to approach the triangle on significantly more than 50% of evaluations. This innovation leads rapidly to 100% fitness.

So we see that the evolutionary run does indeed show fitness epochs, during which the

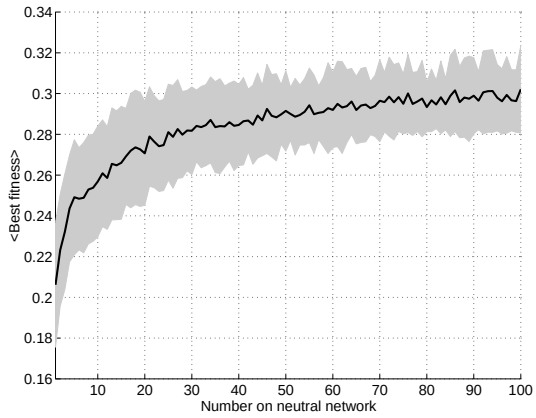


Figure 7: The expected highest fitness drawn from the fitness distribution of a single genotype (figure 2), against the number of drawn samples. With low numbers of fit solutions in the population, the expected best fitness remains near the mean of the noisy fitness distribution. However as the number of fitter genotypes increases, the apparent best fitness also increases; this leads to the apparent rise in the population best fitness seen in figure 5(b) before generation 60.

population is mainly comprised of solutions which are not distinguishable from each other in terms of fitness. But what is the population doing during this period - is it stuck in some local fitness optimum or moving in the search space along neutral networks?

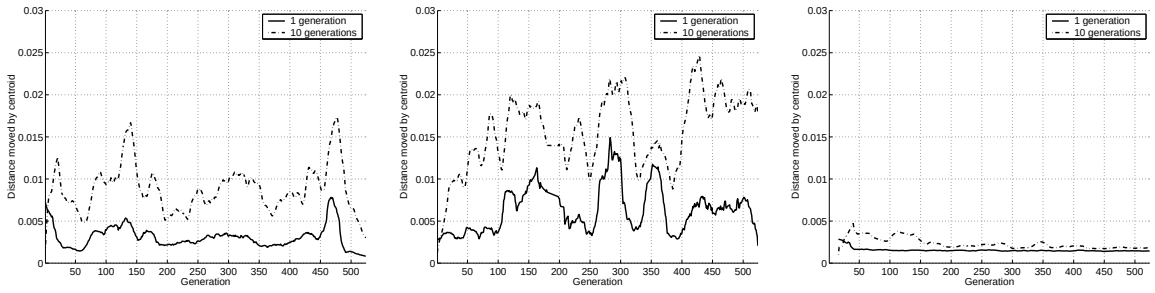
5 Genotypic change during a GasNet fitness epoch

The key feature distinguishing a population diffusing along a neutral network from that stuck at a local optimum is the movement of the population in genotype space, typically measured through the Euclidean distance moved by the population centre-of-mass, or centroid, over time. This approach is complicated by the variable length encoding scheme used in this paper (see section 2.5 for details of the solution representation), in which the neural network controllers can add and delete nodes in the network.

The approach taken here is to analyse the centroid movement over only those genotypes in the population of a certain length; in this section we focus only on those genotypes encoding solutions of nine nodes, i.e. genotypes of length 171 (applying the same results to other length solutions produces comparable results). In particular we can compare the observed centroid movement with the population movement over landscapes of known fitness.

5.1 Movement of the population centroid

Figure 8 shows the Euclidean distance moved over time by the centre-of-mass of all genotypes of length 171 in the population for three different fitness landscapes. Each subplot shows the distance moved by the centroid over one generation (solid line), and the distance moved over ten generations (dotted line), scaled by both the genotype length and possible range of each genotype locus (remember from section 2.5 that each locus on the genotype is an integer in the range $[0, 99]$). Figure 8(a) shows the distance moved by the centroid for the same evolutionary run analysed in the previous section, while for comparison figures 8(b) and 8(c) show the distance moved by the population under the same evolutionary algorithm, but evolving in a flat landscape and in the neighbourhood of a local optimum respectively (for the local optimum movement analysis, the evolutionary population was seeded with mutated copies of the optimal solution in a landscape where fitness was simply evaluated as the mean of the genotype locus values).



(a) GasNet landscape (b) Flat landscape (c) Local-optimum landscape

Figure 8: The Euclidean distance moved over time by the centre-of-mass of all genotypes of length 171 in the population for three different fitness scenarios **(a)** The GasNet evolutionary run analysed in the previous section; **(b)** Evolution on a flat landscape where each solution receives the same evaluated fitness; **(c)** Evolution in the neighbourhood of a locally-optimal peak. The distance moved over the last generation and over the last ten generations are plotted.

The main point to note from figure 8 is that the centroid movements for the GasNet evolution (figure 8(a)) are roughly 50% of the flat landscape evolution (figure 8(b)), but significantly greater than the movement seen in the locally-optimal landscape. This is backed up by the average movements shown in figure 9. It is likely that the sharp jumps in the centroid movements seen in figures 8(a) and 8(b) are likely to be due to the varying length nature of the mutation operator; evolution in fixed length flat landscapes (figure not shown) does not show such sharp

jumps.

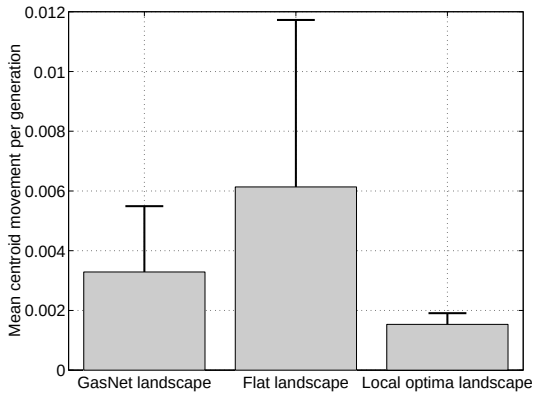


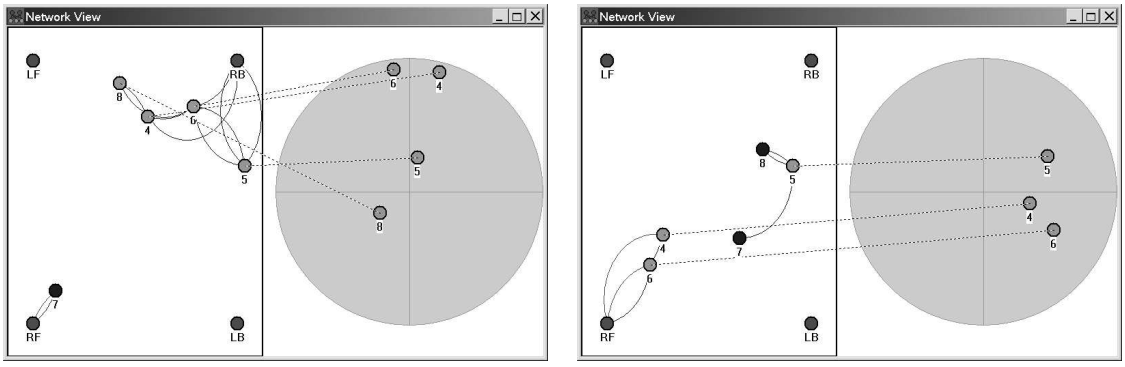
Figure 9: The average movements of the population centroids per generation for evolution over the GasNet landscape, flat landscape, and locally-optimal landscape (error bars show the standard deviations).

From this we argue that the distance moved by the length 171 population centroid (and indeed all other length centroids analysed) is consistent with the hypothesis that the population is moving significantly during the statistically neutral fitness epoch, and not stuck in a local optimum.

5.2 Visualising the population centroid

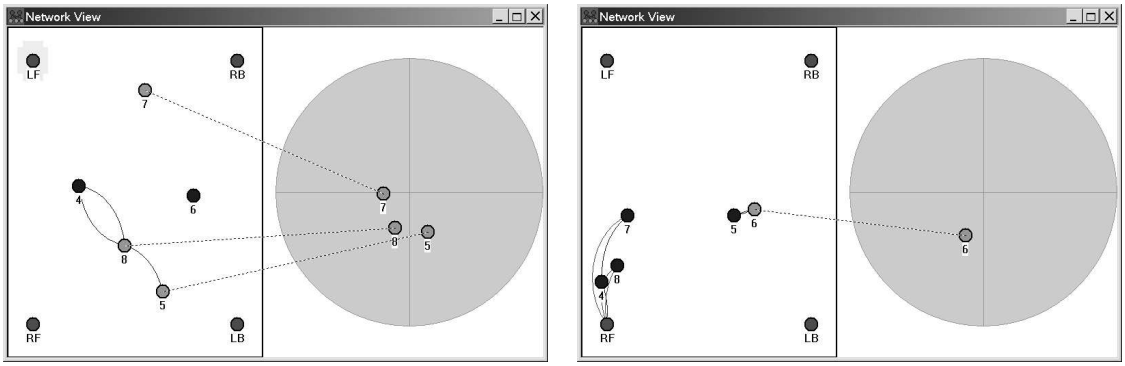
Although it is not possible to accurately visualise the population centre-of-mass over our high dimensional fitness landscape, we can plot the neural network robot control solution corresponding to this centroid. Figure 10 does just that for the length 171 population centroids on generations 100, 200, 300 and 400.

It should be stressed that the networks shown in figure 10 are not necessarily viable robot control solutions, rather they are the 'average' network defined by the mean centroid of all other network genotypes. The main point to note is that the network structures are very different; the population movement shown in the previous section does not merely alter connection weights and other node properties, but massively changes the entire network structure. The period during the fitness epoch is certainly not spent circling some local fitness optimum. The population is moving neutrally through significant volumes of the search space. In the next section we focus on whether there is adaptation of the population during this neutral movement, through application of the local evolvability measures to the statistically neutral population.



(a) Generation 100 centroid

(b) Generation 200 centroid



(c) Generation 300 centroid

(d) Generation 400 centroid

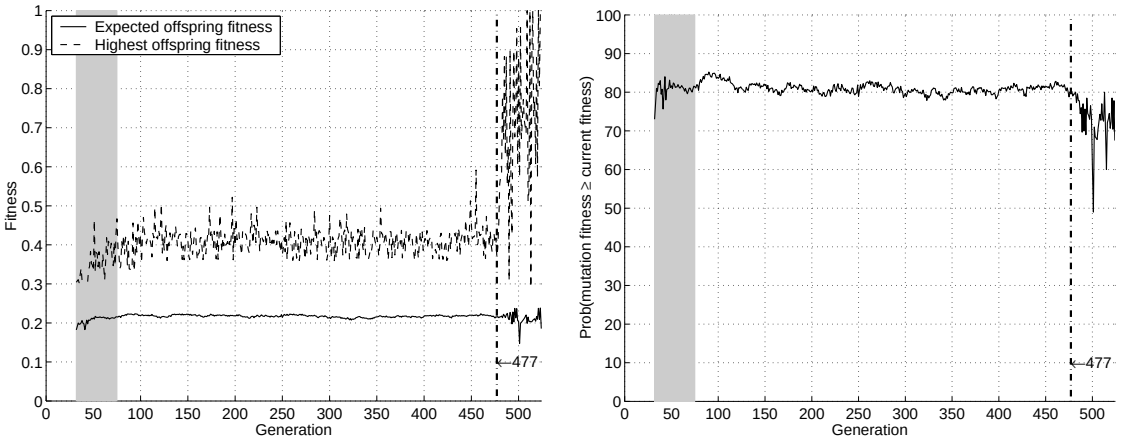
Figure 10: The centre-of-mass for the length 171 solutions plotted as individual networks over generations. See section 2.1 for details of the neural network control architecture.

6 Local evolvability over fitness epochs

In section 1.1 we defined four metrics of local evolvability based on the fitness distributions of offspring surrounding a solution, or population of solutions. In this section we calculate these metrics for the statistically neutral populations identified in section 4.2, plotting the metrics over generations. From this we can focus on the question of whether solutions that are indistinguishable in terms of fitness show differences in terms of the properties of the surrounding search space.

We calculate the local evolvability metrics as follows. Over each generation, each individual in the statistically neutral population was identified, and the fitness distribution calculated through saving the fitness of $1,000$ offspring created through $1,000$ applications of the mutation

operator (section 2.6) to that individual. Each of the offspring was also tested to determine whether they were statistically neutral (or statistically fitter or less-fit) with respect to the parent solution. The local evolvability metrics at each generation were calculated by combining the offspring fitness distributions from each individual in the statistically neutral sample at that generation. Thus we can determine the number of neutral or greater fitness offspring E_a (from the numbers of statistically neutral, fitter or less-fit offspring), the expected offspring fitness E_b, and the expected upper and lower quartiles of the offspring fitnesses, E_c and E_d (from the combined offspring fitness distributions).



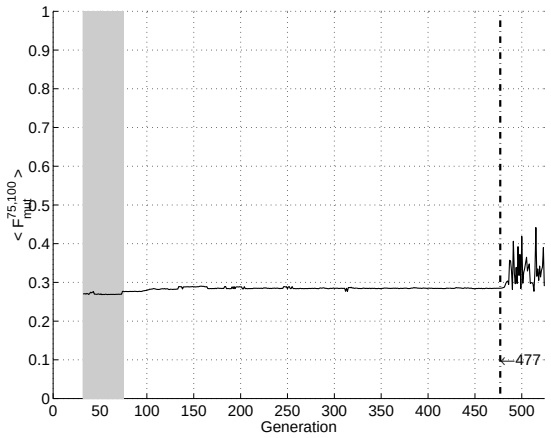
(a) Highest and expected offspring fitness, E_b

(b) Probability of offspring fitness equal or higher than current fitness, E_a

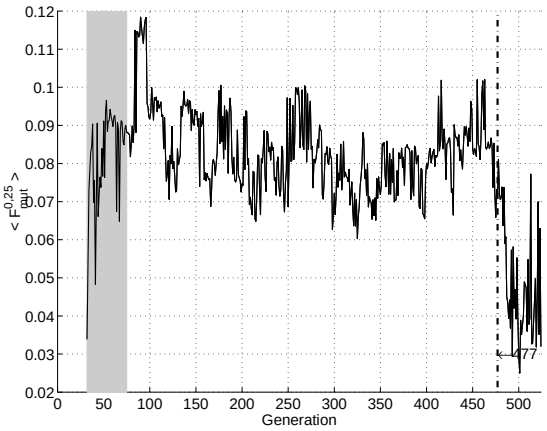
Figure 11: The local evolvability metrics plotted against generation, calculated over the statistically neutral populations identified in section 4.2. The grey band (generations 32 to 75) shows the period of takeover by the statistically neutral population, while the dot-dash line at generation 477 is the end of the fitness epoch, at this time-point fitter genotypes were evolved. **(a)** The highest and expected offspring fitness, E_b; **(b)** The probability of each offspring being of equal or higher fitness, i.e. a non-deleterious mutation E_a.

Figure 11 shows the highest and expected offspring fitnesses, and the probability of obtaining non-deleterious mutations over time. Figure 12 shows the expected fitnesses over the top and bottom quartile of offspring fitnesses. The first point to note is that there is clearly some variation in the local evolvability metrics over generations. This is a significant point; solutions that are indistinguishable from each other in terms of their evaluated fitnesses may be distinguishable from each other in terms of the local properties of the surrounding search space over the course

of an evolutionary run.



(a) Expected fitness over top quartile of offspring, E_c



(b) Expected fitness over bottom quartile of offspring, E_d

Figure 12: The local evolvability metrics plotted against generation, see figure 11 for details. **(a)** The expected offspring fitness over the top quartile of offspring fitnesses, E_c^{25}; **(b)** The expected offspring fitness over the bottom quartile of offspring fitnesses, E_d^{25}.

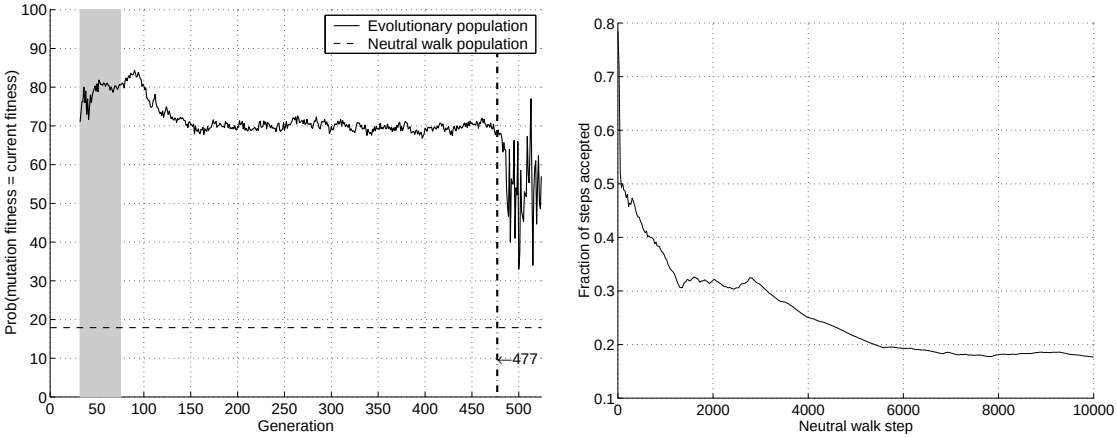
The second main point to note is that the change in local evolvability for the statistically neutral solutions is more marked at the beginning and end of the fitness epoch than during the epoch, i.e. both during takeover of the population at generations 32-75 (grey band), and after the discovery of fitter solutions at generation 477. In particular, we see that the expected offspring fitness and highest offspring fitness (figure 11(a)), and the expected fitness over both the top and bottom quartiles of offspring (figures 12(a) and 12(b)) show evidence of increase during the takeover period. It is possible that this is in part due to the same statistical sampling error described in section 4.2; as the statistically neutral solutions take over the population, the expected fitness will increase due to the increased number of samples. However, the expected fitness increases could not be due to this effect, so it is likely that the population has moved to a "better" area of space with higher local evolvability, i.e. solutions have greater fitness offspring on average. During the fitness epoch there is little evidence for increase in local evolvability, but once the high fitness solutions have been discovered on generation 477, there is a sharp increase in the highest offspring fitness and expected upper quartile offspring fitness, with a decrease in the probability of non-deleterious mutations and the expected fitness over the bottom quartile.

We can now give a much fuller description of the behaviour of the population during evolution than that given earlier. During the takeover of the evolutionary population on generations 32-75, there is some evidence for increase in local evolvability as shown by the increase in highest and expected offspring fitnesses. However, once takeover has occurred, there is no evidence for change in local evolvability - the population is not moving neutrally to better volumes of the search space. Once the fitter portal genotypes are discovered on generation 477, the number of solutions on the statistically neutral network drops dramatically, and there is evidence that the remainder are not as robust as during the fitness epoch. So overall we see little evidence for an increase in local evolvability during the neutral fitness epoch, except in the early stages during takeover of the evolutionary population. Following the arguments for adaptation on neutral networks given in section 1.2, we might find this result surprising: why do we not see neutral evolution of robustness?

6.1 Evolutionary algorithms and evolution of robustness

It is often argued that population evolutionary algorithms provide robustness "for free" (see e.g. Eigen, 1987; Huynen and Hogeweg, 1994; Thompson, 1997), in the sense that the evolutionary search process identifies solutions more insensitive to mutation than on average across the space. This might explain why we did not see evidence for the neutral evolution of robustness in the previous section; the evolutionary algorithm may already have discovered extremely robust areas of the search space.

Figure 13 shows the probability of obtaining equal offspring from the statistically neutral sample over the evolutionary run (figure 13(a)), and the fraction of attempted mutations which were accepted as statistically neutral over a neutral walk (figure 13(b)). The neutral walk was generated as follows. From a starting solution, the best individual in the population on generation 100, the mutation operator was applied successively, and only statistically neutral mutations accepted. The fraction of accepted mutations asymptotically approaches the degree of neutrality over the neutral network (van Nimwegen et al., 1999). Figure 13(a) shows this asymptotic neutrality plotted; as can be seen the neutrality over the fitness epoch is over three times the neutral walk asymptotic value, although after fitter solutions are discovered on generation 477 this ratio drops sharply. It appears that the evolutionary population shows far greater robustness than on average across the space.



(a) Probability of statistically neutral mutation over the evolutionary run

(b) Fraction of mutations accepted as statistically neutral along a neutral walk

Figure 13: The fraction of statistically neutral mutations over **(a)** the evolutionary population, and **(b)** the population sampled over neutral walk. The neutral walk was started from the best individual in the evolutionary population on generation 100, with successive mutations being accepted if the mutated solution was statistically neutral with respect to the solution at the start of the walk. A number of walks were carried out, and all showed similar behaviour. Asymptotically, the fraction of accepted steps converges to the degree of neutrality across the neutral network (van Nimwegen et al., 1999); plotted across **(a)** we see that this average neutrality is far lower than the neutrality seen in the evolutionary population.

7 Discussion

The literature dealing with the dynamics of artificial evolutionary search on "real" problem spaces, in the sense of problems not primarily defined for analysis, is not well developed. Major exceptions include the work on RNA folding landscapes (see e.g. Reidys et al., 2001; Grüner et al., 1996), evolvable hardware experiments (see e.g. Thompson et al., 1999; Vassilev and Miller, 2000) and the evolution of digital organisms such as Avida (see e.g. Wilke et al., 2001). In this paper we have investigated in detail a single evolutionary run on an evolutionary robotics fitness landscape, evolving neural network controllers for a robotic visual discrimination problem.

Through the use of *statistical neutrality* of solutions, i.e. solutions that are indistinguishable in terms of evaluated fitness, we have investigated the behaviour of a set of selectively neutral solutions over the course of an evolutionary run. We have shown that the evolutionary run

does show a long fitness epoch during which the majority of solutions in the population are statistically neutral with respect to each other. During this period the population moves significantly in genotype space; fitness is conserved but genotype structure is not. By defining a set of local evolvability metrics (Smith et al., 2002a) we have investigated the regions of search space surrounding the statistically neutral sample, showing that although they are indistinguishable in terms of their fitnesses, there is variation in the search space through which they move. In particular, we saw variation at the boundaries of the fitness epoch, both during takeover of the population by the statistically neutral population and once fitter genotypes are discovered, however we saw no evidence for change in local evolvability during the bulk of the fitness epoch. Finally, through statistically neutral walks, we showed that the evolutionary population occupies volumes of the search space that are far more robust than on average across the space.

The evolution of robustness may turn out to be a fundamental organisational principle in understanding the dynamics of evolutionary search. The pressure on solutions to produce more and more viable offspring may be important in all phases of evolution, during both neutral fitness epochs and hill-climbing episodes, and population-based evolutionary search is extremely good at exploiting this pressure in order to drive the population towards robust areas of the fitness space. The argument that such evolution of robustness can occur during fitness epochs is an attractive idea, however it may well be the case that in many noisy real world problems we do not see further neutral evolution of robustness from a population already produced through evolutionary search.

Our concluding remarks are on the relationship between local evolvability and evolvability. Although evolvability is typically discussed in terms of change over time, in some sense that change must start from the current location of the solution in the search space. In the absence of a rigorous definition of evolvability, the properties of the local search space seem as good a place as any to start trying to identify solution evolvability.

Acknowledgements

The authors would like to thank the three anonymous reviewers, Andy Philippides, Inman Harvey, Lionel Barnett and all the members of the Centre for Computational Neuroscience and Robotics (http://www.cogs.sussex.ac.uk/ccnr/) for constructive discussion. We would also like to thank the Sussex High Performance Computing Initiative

(http://www.hpc.sussex.ac.uk/) for computing support. TS is funded by a BTexaCT Future Technologies Group sponsored Biotechnology and Biology Science Research Council Case award (http://www.btexact.com/projects/ftg/).

References

Adami, C. (1995). Self-organized criticality in living systems. *Physics Letters A*, 203:29–32.

Altenberg, L. (1994). The evolution of evolvability in genetic programming. In Kinnear Jr, K., editor, *Advances in Genetic Programming*, chapter 3, pages 47–74. MIT Press, Cambridge, Massachusetts.

Burch, C. and Chao, L. (2000). Evolvability of an RNA virus is determined by its mutational neighbourhood. *Nature*, 406:625–628.

Cliff, D. T., Harvey, I., and Husbands, P. (1993). Explorations in evolutionary robotics. *Adaptive Behaviour*, 2(1):71–104.

Crutchfield, J. and van Nimwegen, E. (1999). The evolutionary unfolding of complexity. In Landweber, L., Winfree, E., Lipton, R., and Freeland, S., editors, *Evolution as Computation: Proceedings of a DIMACS Workshop*. Springer, Berlin.

Dawkins, R. (1989). The evolution of evolvability. In Langton, C., editor, *Artificial Life: Proceedings of the Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*, volume VI of *Santa Fe Institute Studies in the Sciences of Complexity*, pages 201–220. Addison-Wesley, Redwood, California.

Ebner, M., Langguth, P., Albert, J., Shackleton, M., and Shipman, R. (2001). On neutral networks and evolvability. In *Proceedings of the 2001 Congress on Evolutionary Computation: CEC2001*, pages 1–8. IEEE Press, Piscataway, New Jersey.

Eigen, M. (1987). New concepts for dealing with the evolution of nucleic acids. In *Cold Spring Harbor Symposia on Quantitative Biology*, volume LII.

Eldredge, N. and Gould, S. (1972). Punctuated equilibria: An alternative to phyletic gradualism. In Schopf, T., editor, *Models in Paleobiology*, pages 82–115. Freeman, San Francisco, California.

Elena, S., Cooper, V., and Lenski, R. (1996). Punctuated evolution caused by selection of rare beneficial mutations. *Science*, 272:1802–1804.

Floreano, D. and Mondada, F. (1994). Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot. In Cliff, D., Husbands, P., Meyer, J.-A., and Wilson, S., editors, *From*

Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behaviour, SAB94. MIT Press, Cambridge, Massachusetts.

Gould, S. and Eldredge, N. (1977). Punctuated equilibria: The tempo and mode of evolution reconsidered. *Paleobiology*, 3:115–151.

Grüner, W., Giegerich, R., Strothmann, D., Reidys, C., Weber, J., Hofacker, I., Stadler, P., and Schuster, P. (1996). Analysis of RNA sequence structure maps by exhaustive enumeration: I Neutral networks, II Structures of neutral networks and shape space covering. *Monathefte Chem.*, 127:355–374, 375–389.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, Michigan.

Hordijk, W. (1996). A measure of landscapes. *Evolutionary Computation*, 4(4):335–360.

Husbands, P. and Meyer, J.-A., editors (1998). *Evolutionary Robotics: First European Workshop, EvoRobot98*. Springer, Berlin.

Husbands, P., Smith, T., Jakobi, N., and O'Shea, M. (1998). Better living through chemistry: Evolving GasNets for robot control. *Connection Science*, 10(3-4):185–210.

Huynen, M. and Hogeweg, P. (1994). Pattern generation in molecular evolution: Exploitation of the variation in RNA landscapes. *Journal of Molecular Evolution*, 39:71–79.

Jakobi, N. (1998). Evolutionary robotics and the radical envelope of noise hypothesis. *Adaptive Behaviour*, 6:325–368.

Jones, T. and Forrest, S. (1995). Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In Eshelmann, L., editor, *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA95)*, pages 184–192. Morgan Kaufmann, San Mateo, California.

Kauffman, S. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, Oxford, UK.

Kimura, M. (1983). *The Neutral Theory of Molecular Evolution*. Cambridge University Press, Cambridge, UK.

Kirschner, M. and Gerhart, J. (1998). Evolvability. *Proceedings of the National Academy of Sciences, USA*, 95:8420–8427.

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, Massachusetts.

Marrow, P. (1999). Evolvability: Evolution, computation, biology. In Wu, A., editor, *Proceedings of the 1999 Genetic and Evolutionary Computation Conference Workshop Program (GECCO-99 Workshop on Evolvability)*, pages 30–33. Morgan Kaufmann, San Mateo, California.

McMullin, B. (2000). The Von Neumann self-reproducing architecture, genetic relativism and evolvability. In Maley, C. and Boudreau, E., editors, *Artificial Life 7 Workshop Proceedings*, pages 11–14.

Naudts, B. and Kallel, L. (2000). A comparison of predictive measures of problem difficulty in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 4(1):1–15.

Nehaniv, C. (2000). Measuring evolvability as the rate of complexity increase. In Maley, C. and Boudreau, E., editors, *Artificial Life 7 Workshop Proceedings*, pages 55–57.

Newman, M. and Engelhardt, R. (1998). Effects of selective neutrality on the evolution of molecular species. *Proceedings of the Royal Society of London, B*, 265:1333–1338.

Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence and Technology of Self-Organizing Machines*. MIT Press, Cambridge, Massachusetts.

Ohta, T. (1992). The nearly neutral theory of molecular evolution. *Annual Review of Ecology and Systematics*, 23:263–286.

Philippides, A., Husbands, P., and O'Shea, M. (2000). Four-dimensional neuronal signaling by nitric oxide: A computational analysis. *Journal of Neuroscience*, 20(3):1199–1207.

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1992). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK, 2nd edition.

Reidys, C., Forst, C., and Schuster, P. (2001). Replication and mutation on neutral networks. *Bulletin of Mathematical Biology*, 63(1):57–94.

Smith, T., Husbands, P., Layzell, P., and O'Shea, M. (2002a). Fitness landscapes and evolvability. *Evolutionary Computation*, 10(1):1–34. To appear.

Smith, T., Husbands, P., and O'Shea, M. (2002b). Adapting to a changing environment: Evolvability and analysis of robot control networks. In *From Animals to Animats 7: Proceedings of the Seventh International Conference on Simulation of Adaptive Behaviour, SAB2002*. Submitted.

Smith, T. and Philippides, A. (2000). Nitric oxide signalling in real and artificial neural networks. *BT Technology Journal*, 18(4):140–149.

Thompson, A. (1997). Evolving inherently fault-tolerant systems. *Proceedings Institution Mechanical Engineers, Part I*, 211:365–371.

Thompson, A. (1998). *Hardware Evolution: Automatic Design of Electronic Circuits in Reconfigurable Hardware by Artificial Evolution*. Distinguished Dissertation Series. Springer, Berlin.

Thompson, A., Layzell, P., and Zebulum, R. S. (1999). Explorations in design space: Unconventional electronics design through artificial evolution. *IEEE Transactions on Evolutionary Computation*, 3(3):167–196.

Turney, P. (1999). Increasing evolvability considered as a large-scale trend in evolution. In Wu, A., editor, *Proceedings of the 1999 Genetic and Evolutionary Computation Conference Workshop Program (GECCO-99 Workshop on Evolvability)*, pages 43–46. Morgan Kaufmann, San Mateo, California.

van Nimwegen, E. and Crutchfield, J. (2000). Metastable evolutionary dynamics: Crossing fitness barriers or escaping via neutral paths? *Bulletin of Mathematical Biology*, 62(5):799–848.

van Nimwegen, E., Crutchfield, J., and Huynen, M. (1999). Neutral evolution of mutational robustness. *Proceedings of the National Academy of Sciences, USA*, 96:9716–9720.

van Nimwegen, E., Crutchfield, J., and Mitchell, M. (1997). Finite populations induce metastability in evolutionary search. *Physics Letters A*, 229:144–150.

Vassilev, V. and Miller, J. (2000). The advantages of landscape neutrality in digital circuit evolution. In Miller, J., Thompson, A., Thomson, P., and T., F., editors, *Proceedings of the Third International Conference on Evolvable Systems: From Biology to Hardware (ICES'2000)*, volume 1801 of *Lecture Notes in Computer Science*, pages 252–263. Springer, Berlin.

Wagner, G. and Altenberg, L. (1996). Complex adaptations and the evolution of evolvability. *Evolution*, 50(3):967–976.

Weinberger, E. (1990). Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63:325–336.

Wilke, C. (2001). Adaptive evolution on neutral networks. *Bulletin of Mathematical Biology*, 63:715–730.

Wilke, C., Wang, J., Ofria, C., Lenski, R., and Adami, C. (2001). Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, 412:331–333.