Chapter 4

Fitness Landscapes

[1] As we have seen from Chapter 2, there has been a lot of interest in evolving controllers for both physically simulated creatures as well as for real physical robots. However, a range of different ANN architectures are used for controller evolution and in the majority of the work conducted, the choice of the architecture used is made arbitrarily.

There have been some preliminary experiments that compared the performance of evolved feed-forward versus dynamically recurrent ANNs for controlling Khepera robots in a simulated space-constrained box-pushing experiment (Spronck, Sprinkhuizen-Kuyper, and Postma 2001). It was found that feed-forward architectures were sufficient for successfully completing the task although the recurrent architecture provided much more stability for the controller's behavior, particularly in maneuvering out of problematic positions. However, no fitness landscape analysis was provided for the underlying fitness landscape of the controller's search space.

As such, the literature remains largely inconclusive as to which ANN architecture provides the most efficient and effective space for searching the range of possible controllers through evolutionary methods. This represents the motivation for this chapter where we compare the search space for four different types of ANN architecture for controller evolution through an analysis of the fitness landscape as-

[1] Some of the material presented in this chapter have been previously published in Teo and Abbass (2003).

sociated with each type of architecture. We intend to ascertain whether additional recurrent and input-output connections to a standard feed-forward ANN architecture yields any benefit in terms of providing a fitness landscape which is easier to search for locomotion controllers. The results from this chapter will provide a basis for selecting the appropriate ANN architecture to be used for controller evolution in later chapters of this thesis.

4.1 Introduction

The idea of fitness landscapes was first proposed by Wright (1932) and has since proven to be an invaluable tool for analyzing evolutionary theories (Kauffman 1993; Adami 1998). It serves as a powerful tool for visualizing the evolutionary process through its imagery of mountainous peaks, hills, valleys, ridges and plateaus that are encountered through the exploration and exploitation of genotype space. Evolution can thus be viewed as movements within a multi-dimensional search space. Although initially introduced by Wright as a non-mathematical tool for visualizing biological selection and variation, fitness landscapes have since become highly amenable to mathematical analysis. A discussion of the various metrics that have been proposed for mathematically characterizing fitness landscapes is given in Section 4.3.

In an evolutionary computation context, a fitness landscape comprises of three main elements: (1) the set of genotypes, (2) the fitness function that evaluates the genotypes, and (3) the genetic operators that define the neighborhood relationships between the set of genotypes (Vassilev, Fogarty, and Miller 2000). The fitness landscape is thus normally a high-dimensional space with $n + m$ dimensions, where n is the genotype length and the extra m dimensions representing the fitness values associated with the genotype when evaluated using the m fitness functions. In traditional evolutionary computation, m is usually 1 since the evolutionary optimization process is conducted on one objective function only. In this case, a genotype with length two can be visualized as a 3D landscape where genetic operations carried out

on the set of genotypes will produce small movements on the landscape during the evolutionary search process. On the other hand, in an evolutionary multi-objective search process, m would be ≥ 2 since the solution is being optimized along two or more objective functions.

4.2 Search Space Difficulty

The performance of an EA is thus tied intimately to the structure of its fitness landscape. In attempting to identify the difficulties presented by a particular landscape, the evolutionary search is typically characterized in terms of the degree of epistasis and modality (Smith, Husbands, Layzell, and O'Shea 2002). Epistasis refers to the situation where the fitness of a genotype is dependent on multiple gene interactions. Modality refers to the situation where the search space has large numbers of optima. Both high epistasis and modality will lead to a rugged fitness landscape (Vassilev, Fogarty, and Miller 2000). Such rugged search spaces can be visualized as a landscape with many hill-tops that are separated by deep valleys. In other words, there is no steady or smooth progression of fitness values from one point to another neighboring point and thus increases the difficulty for the evolutionary search to move to higher areas of fitness during the optimization process. Therefore, highly epistatic and multi-modal problems will lead to a rugged landscape that is more difficult to search compared to a smoother landscape with low epistasis and modality.

4.3 Analyzing Fitness Landscapes

A number of fitness landscape analysis techniques have been proposed for measuring the degree of ruggedness of the underlying search space. These mathematical treatments of the fitness landscape comprise of two main streams: (1) statistical measures, and (2) information measures.

4.3.1 Statistical Measures

Weinberger (1990) used the autocorrelation function to measure the ruggedness of the landscape. A sequence of fitness values is generated using a random walk through the search space. The autocorrelation ρ between sets of fitness points separated by a distance of Γ is then approximated by

$$\rho(\Gamma) \approx \frac{E(f_t f_{t+s}) - E(f_t)E(f_{t+s})}{V(f_t)} \tag{4.1}$$

where $E(f_t)$ represents the expectation and $V(f_t)$ the variance of the sequence of N fitness values $\{f_t\}_{t=1}^{N}$. A high correlation indicates a smooth landscape since neighboring points have highly similar fitness values. On the other hand, a low correlation indicates a rugged landscape since neighboring points have highly dissimilar fitness values.

Weinberger also proposed another correlation measure called the correlation length to define landscape ruggedness. It is simply the distance beyond which the sets of fitness points become uncorrelated. The correlation length τ between sets of fitness points separated by a distance of Γ is calculated as

$$\tau(\Gamma) = -\frac{1}{\ln(\rho(1))} \tag{4.2}$$

where $\rho(1)$ is the autocorrelation of neighboring points. The magnitude of this length indicates the smoothness of the landscape. A longer correlation length would thus indicate a very smooth fitness landscape whereas a shorter length would indicate a more rugged landscape. A number of other correlational metrics have also been proposed for characterizing fitness landscapes (Manderick, de Weger, and Spiessens 1991; Lipsitch 1991; Hordijk 1996; Vassilev, Fogarty, and Miller 2000).

4.3.2 Information Measures

Apart from statistical analysis, a distinctly different methodology based on classical information theory (Shannon 1948) known as information content has been proposed for characterizing fitness landscapes (Vassilev, Fogarty, and Miller 2000). It also approximates the ruggedness of the underlying search space through

analysis of a sequence of fitness values $\{f_t\}_{t=1}^{N}$ obtained through a random walk of N steps over the landscape but instead measures the entropy or amount of fitness change encountered during the walk. Four measures were proposed by Vassilev in conjunction with this information analysis of fitness landscapes (Vassilev, Fogarty, and Miller 2000):

1. *Information Content* ($H(\epsilon)$): indicates the ruggedness of landscape path
2. *Partial Information Content* ($M(\epsilon)$): indicates the modality of landscape path
3. *Information Stability* ($\epsilon^{\star}$): indicates the magnitude of landscape path's optima
4. *Density-Basin Information* ($h(\epsilon)$): characterizes the landscape structure around optima

The information content characterizes the amount of ruggedness with respect to the flat areas of the landscape. The degree of flatness depends on a sensitivity parameter ϵ which is explained in the following paragraph. The information content is given by

$$H(\epsilon) = -\sum_{p \neq q} P_{[pq]} \log_6 P_{[pq]} \tag{4.3}$$

where $H(\epsilon)$ represents the entropy of the system. The probabilities $P_{[pq]}$ represent the frequencies of possible sub-blocks pq of elements from the string $S(\epsilon) = s_1 s_2 s_3 ... s_N$ where $s_i \in \{\bar{1}, 0, 1\}$. The string $S(\epsilon)$ is enumerated using the following function

$$s_i = \Psi_{f_t}(i, \epsilon) \tag{4.4}$$

where

$$\Psi_{f_t}(i, \epsilon) = \begin{cases} \bar{1} & \text{if } f_i - f_{i-1} < -\epsilon \\ 0 & \text{if } |f_i - f_{i-1}| \leq \epsilon \\ 1 & \text{if } f_i - f_{i-1} > \epsilon \end{cases} \tag{4.5}$$

for a particular value of the parameter ϵ. This parameter ϵ controls the sensitivity for measuring the entropy and is a real-valued number chosen from the range $[0, L]$ where L represents the maximum fitness difference of the sequence $\{f_t\}_{t=1}^{N}$.

The information analysis will be most sensitive when ϵ is 0 producing the maximal string of 1's and $\bar{1}$'s when enumerating $S(\epsilon)$ and hence provides the most detailed description of the landscape. Conversely, the analysis will be least sensitive when ϵ is L producing a string of 0's when enumerating $S(\epsilon)$ and hence provides the least detailed description of the landscape. In effect, ϵ acts as an accuracy setting for the information analysis and provides an idea of the landscape profile according to varying degrees of detail.

The partial information content is obtained by filtering out non-essential parts of $S(\epsilon)$ to obtain an indication of the modality encountered during the walk. It is given by

$$M(\epsilon) = \frac{\mu}{n} \tag{4.6}$$

where μ is the length of the derived string $S'(\epsilon)$ and n is the length of the original string $S(\epsilon)$. μ is calculated using a recursive function $\Phi_S(1, 0, 0)$ defined as

$$\Phi_S(i, j, k) = \begin{cases} k & \text{if } i > n \\ \Phi_S(i+1, i, k+1) & \text{if } j = 0 \text{ and } s_i \neq 0 \\ \Phi_S(i+1, i, k+1) & \text{if } j > 0, s_i \neq 0 \text{ and } s_i \neq s_j \\ \Phi_S(i+1, j, k) & \text{otherwise} \end{cases} \tag{4.7}$$

where k will return the value of μ upon completion of the evaluation. When $M(\epsilon)$ is 0, this is an indication that no slopes were present in the path. However when $M(\epsilon)$ is 1, this indicates that the path is maximally multi-modal. Furthermore, the expected number of optima can be calculated from the partial information content as $\lfloor \frac{nM(\epsilon)}{2} \rfloor$.

The information stability ($\epsilon^\star$) is defined to be the smallest value of ϵ corresponding to $H(\epsilon) = 0$. A high information stability indicates that the largest possible difference between two neighboring points is similarly high. Thus, it provides an idea of the magnitude of the landscape path's optima encountered during the walk.

In order to characterize the landscape structure around optima, it was also suggested that the *density-basin information* ($h(\epsilon)$) be calculated (Vassilev, Fogarty,

and Miller 2000). This measure gives an indication of the flat and smooth areas of the landscape and gives an indication of the density as well isolation of peaks in the landscape. The formula for calculating this measure is given by

$$h(\epsilon) = -\sum_{p \in \{\bar{1},0,1\}} P_{[pp]} \log_3 P_{[pp]} \tag{4.8}$$

where pp represent sub-blocks 00,11 and $\bar{1}\bar{1}$. A high number of peaks existing within a small area of the landscape would thus give a high value for $(h(\epsilon))$. On the other hand, an isolated optimum would give a low value for $(h(\epsilon))$. As such, this provides an idea of the size and nature of the basins of attractions of optima. Landscapes with high density-basin information should thus be easier for an evolutionary search process to become "attracted" to a fitter solution space and the converse for landscapes with lower density-basin information.

In summary, higher values of information content, partial information content and information stability suggest higher degrees of epistasis and modality, which leads to a more rugged landscape that is harder to search. At the same time, the density-basin information should further assist in determining the search space difficulty by characterizing the landscape around optima. Therefore, using these information-theoretic measures to compare between different artificial evolutionary systems should provide useful characterizations of the search difficulty associated with the different fitness landscapes.

4.4 Experimental Setup

Three series of experiments were performed to provide an insight into the search space difficulty associated with each type of ANN architecture. The fitness of each genotype in these experiments was evaluated according to the f_1 objective function only, which is the locomotion distance achieved by the controller as defined in Section 3.4.1.

In the first series of experiments, random sampling of solutions was conducted for all four architectures. Since random search is used, each genotype is

generated independently from all other genotypes. Furthermore, the variable number of hidden units for each network specified by the genotype is initialized randomly ranging between 0 to 15 hidden units according to a uniform distribution. From prior experiments, it was found that the best controllers only required between 2–4 hidden units (Teo and Abbass 2002a; Teo and Abbass 2002b; Teo and Abbass 2002c). As such, we have chosen to set the upper bound for this parameter at 15 hidden units. In each run, a total of 30,000 genotypes were sampled. This is equivalent to an evolutionary run with a population size of 30 over 1000 generations, which is the intended setup for later experiments involving artificial evolution of ANNs, to ensure that a fair comparison of the search space can be made. Sampling of the search space using random search is replicated for 10 different seeds giving 10 independent runs with a total of 300,000 fitness evaluations (although a single run that directly generates the required 300,000 fitness evaluations would be equivalent, the setup using 10 runs initialized from 10 seeds is used to maintain consistency across all experimental setups).

In the second series of experiments, trial solutions were obtained using a hill-climbing algorithm for all four ANN architectures. New genotypes were generated from the currently accepted genotype using a mutation of 0.1. The mutation operator changes both the values of the connection weights and number of active hidden units in the network. A move from the best solution found so far to a trial solution is accepted only if the trial solution has a higher fitness than the best solution found so far. Otherwise, another trial solution is generated. The fitness for all solutions generated during the hill-climb were recorded and analyzed. Each run was again allowed to sample a maximum of 30,000 genotypes over 10 independent runs as in the random search experiments. In order to reduce the amount of bias on the number of hidden units present in each ANN during initialization of the genotype, each of the 10 independent runs was started using networks initialized with increasing probabilities of having more hidden units ranging from 0 to 15 hidden units. This guarantees that the genotype space is sampled uniformly in terms of the variable hidden layer.

For the last series of experiments, a random walk was performed using all four ANN architectures. The fitness value obtained for every genotype at each step of the walk was recorded. A new point in the search space was generated through a 0.1 mutation of the previous genotype reached during the walk. Again to ensure fair comparisons, each walk was allowed a maximum of 30,000 steps and 10 separate walks were carried out staring from different points in the search space. Also, as with the hill-climbing experiments, each of the 10 independent runs were started using networks initialized with increasing probabilities of having more hidden units to ensure that the genotype space is sampled uniformly in terms of the variable hidden layer.

The information content analysis was carried out only for the search spaces sampled using random walk. This is due to the neighborhood definition of this landscape measure, which is only meaningful when all subsequent fitness points of genotypes sampled in the search space are related through a walk obtained using the genetic operators (see Section 4.1). Furthermore, from the definition of the autocorrelation function, to conduct such an analysis requires sampling of fitness points that are separated by a distance of Γ. However, an autocorrelation analysis was not possible in our particular problem as the algorithm used in our random walk cannot guarantee that fitness points of step length > 1 are unique points in the landscape. This was due to the fact that the mutation operator used to generate subsequent neighborhood points is non-directional in the sense that later mutations may return a particular walk to a previously encountered landscape point. A directional approach would also not provide an accurate picture of the actual fitness step lengths because of two problems: (1) network symmetry, and (2) hidden unit activation. Firstly, mutations arising from distinctly different trajectories can lead to identical ANNs by virtue of architectural symmetries that can arise in the network. Secondly, the flipping of a single bit in the genotype which turns a particular hidden unit on or off will cause an entire set of connection weights to either become active or inactive in the ANN and hence, the mutation of a single hidden unit can cause a very large change in the phenotype. These are known problems in measuring diversity

when evolving neural networks (Yao 1999). As such, an autocorrelation analysis conducted in this case will not be reliable since we cannot guarantee that fitness points after s steps are actually at a distance $\Gamma = s$ away from each other. Hence, we focus our fitness landscape analysis of neighborhood points sampled by random walk using only the informational measures.

This initial investigation of the underlying fitness landscapes should provide some indication on which of the four proposed types of ANN architecture would allow for better controller evolution. Although there are limitations associated with fitness landscape analysis methods, which we discuss in Section 4.6, the results from this initial investigation will at least provide some basis for deciding which type of ANN architecture is to be used for the remainder of the artificial evolution experiments.

4.5 Results and Discussion

The results from the experiments described above are presented in three sections. The first section provides a characterization of the search space using random sampling, followed by hill-climbing and finally using random walk. For each sampling technique, a 3D graph was first plotted to show the frequency distribution of sampled genotypes in terms of their solution fitness as well as the number of hidden units present in the ANN. As the objective is a continuous function, genotypes were grouped into 5000 discrete intervals to calculate the frequency distribution. These 3D frequency graphs were rotated along the X-Y axis in order to provide a clearer depiction of the distribution characteristics as it was found that the default orientation in the original plots often resulted in some features becoming obscured by large peaks in the distribution. This convention is adopted throughout the thesis whenever such graphs are depicted. Additional 2D smoothed contour graphs were plotted to provide a clearer depiction of the relationship between number of hidden units present in the ANN and quality of solutions. A smoothed probability density function was also plotted for each architecture by grouping solutions according to their fitness irrespective of the number of hidden units present in the ANN. Also,

the best solution found by each of these algorithms over the 10 independent runs was plotted over time. This will give an indication of how the search proceeded over time. Finally, a comparison of the best overall solution found using the different algorithms is discussed along with the average and standard deviation of the best solutions found.

4.5.1 Random Search

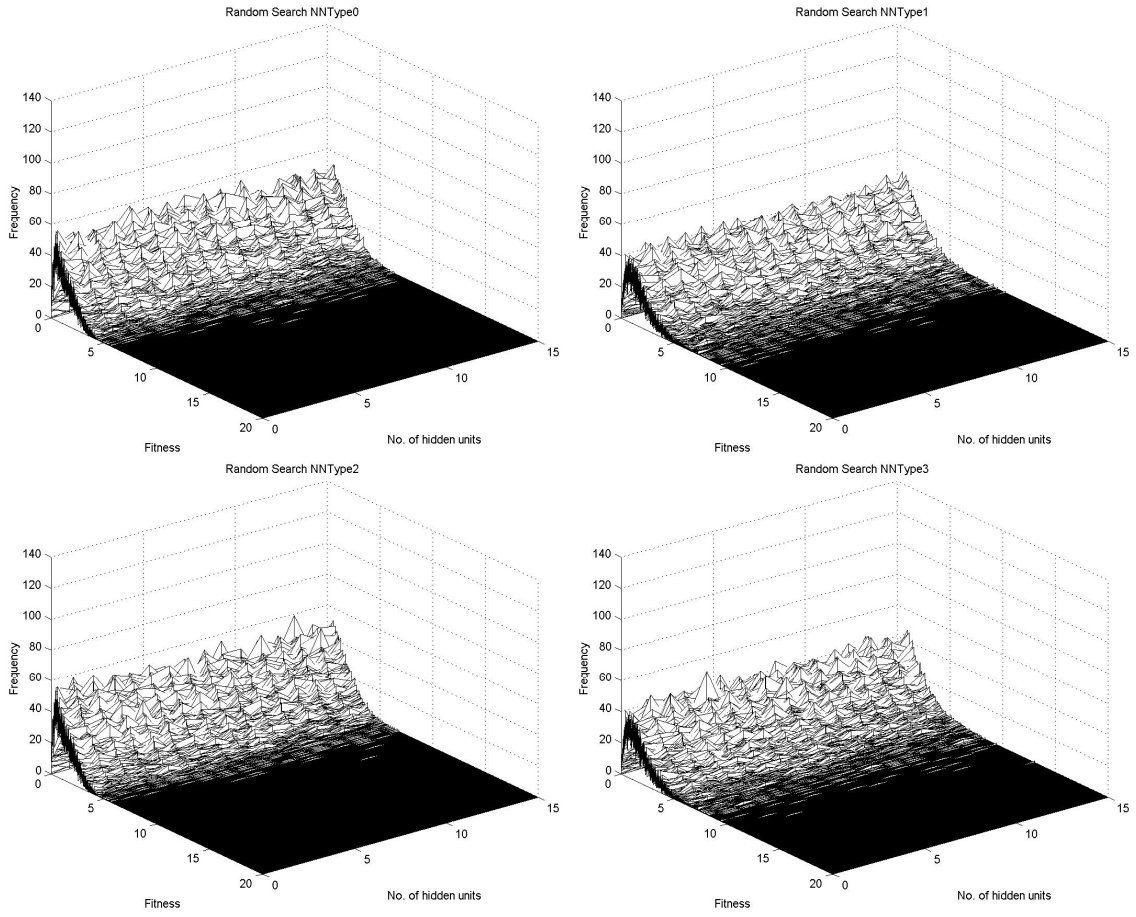

Figure 4.1: Frequency distribution of solutions using random search for ANN architecture 1. NNType0 (top left), 2. NNType1 (top right), 3. NNType2 (bottom left), 4. NNType3 (bottom right). X-axis: Fitness, Y-axis: No. of hidden units, Z-axis: Frequency.

Figure 4.1 is a frequency histogram of the distribution of solutions in terms

of their fitness and number of hidden units. From these graphs, it is apparent that the number of hidden units does not affect the quality of the solutions sampled at random. An analysis conducted on the mean fitness across ANNs with different numbers of hidden units showed that there was no correlation between the number of units present in the hidden layer and the locomotion capability of the ANN. What this means is that from random sampling, controller size in terms of number of hidden units does not appear to affect the locomotion capability of the creature. This may be due to the fact that solutions with good locomotion abilities may be extremely rare and isolated in the search space, which makes this problem of automatic generation of artificial creature controllers a particularly hard problem.

2D contour graphs of frequency distribution of solutions obtained from random search in terms of fitness and number of hidden units present in the ANN are given in Figure 4.2. No obvious concentration of solutions can be seen from these graphs, which confirms the earlier observation that the number of hidden units does not appear to affect the solutions found by random search. However, it is interesting to note that the final contour line extends further by more than 1 unit distance in NNType1 (Figure 4.2.2) and NNType3 (Figure 4.2.4) compared to NNType0 (Figure 4.2.1) and NNType2 (Figure 4.2.3). This suggests that it was slightly easier to reach fitter regions of the controller's objective space using the NNType1 and NNType3 architectures compared to using the NNType0 and NNType2 architectures.

Figure 4.3 shows the probability density function of solutions obtained using random search for all four types of ANN architecture. It is clear from these graphs that a random search of the genotype space yields a very high percentage of low fitness solutions. For all four types of ANN, the most commonly sampled genotype only yields a fitness of around 1. This is a clear indication that a uniform sampling of the genotype space yields a highly skewed distribution of solutions in the objective space. The NNType1 (Figure 4.3.2) and NNType3 (Figure 4.3.4) architectures appear to provide slightly more solutions with higher fitness although the difference is not very obvious. This can be seen from the small shift to the right of the probability curve for these networks. Also, the probability of generating

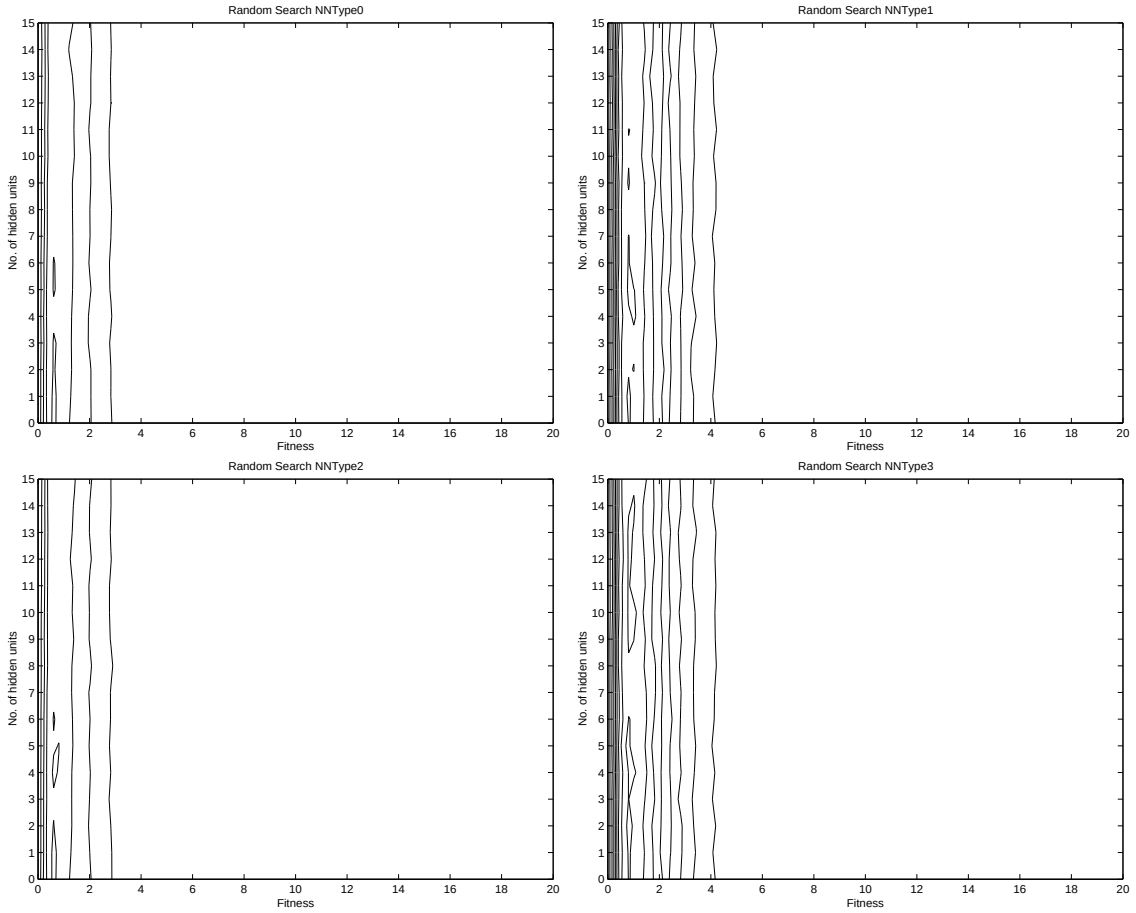

Figure 4.2: Contour graphs of frequency distribution of solutions obtained using random search for ANN architecture 1. NNType0 (top left), 2. NNType1 (top right), 3. NNType2 (bottom left), 4. NNType3 (bottom right). X-axis: Fitness, Y-axis: No. of hidden units.

controllers begins to approach 0 for NNType0 (Figure 4.3.1) and NNType2 (Figure 4.3.3) beyond a fitness 6 whereas for NNType1 and NNType3, this only occurs at a fitness of beyond 8. This observation gives a weak indication that ANNs with direct connections from input to output (NNType1 & NNType3) may be easier to search whereas recurrent connections only (NNType2) do not provide any significant advantage over a standard feed-forward architecture (NNType0).

The best solution obtained over the 30,000 iterations of random search for 10 independent runs is depicted in Figure 4.4. The final best solutions clustered

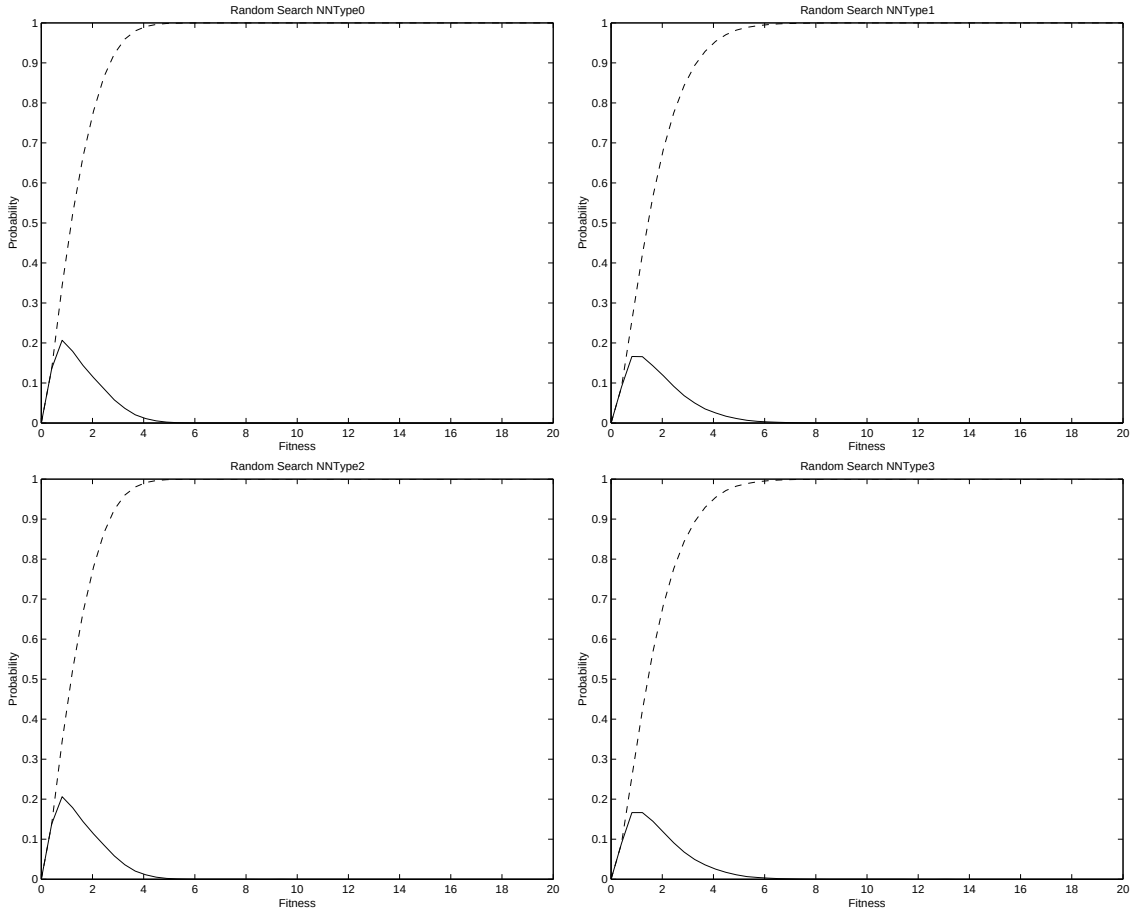

Figure 4.3: Density (solid) and cumulative (dashed) probability distribution of solutions obtained using random search for ANN architecture 1. NNType0 (top left), 2. NNType1 (top right), 3. NNType2 (bottom left), 4. NNType3 (bottom right). X-axis: Fitness, Y-axis: Probability.

between a fitness of 6 to 9 for NNType0 (Figure 4.4.1) and NNType2 (Figure 4.4.3) whereas for NNType1, the final best solutions clustered between 9 and 11 (Figure 4.4.2). There was a larger spread of best final solutions in NNType3 ranging between 9 and 13 (Figure 4.4.4). The NNType3 architecture also had more runs in which significant fitness improvements still occurred in the latter parts of the random search compared to the other three architectures where in most runs, the major improvements in fitness occurred before the 10,000th iteration resulting in large plateau areas in the end regions of these graphs.

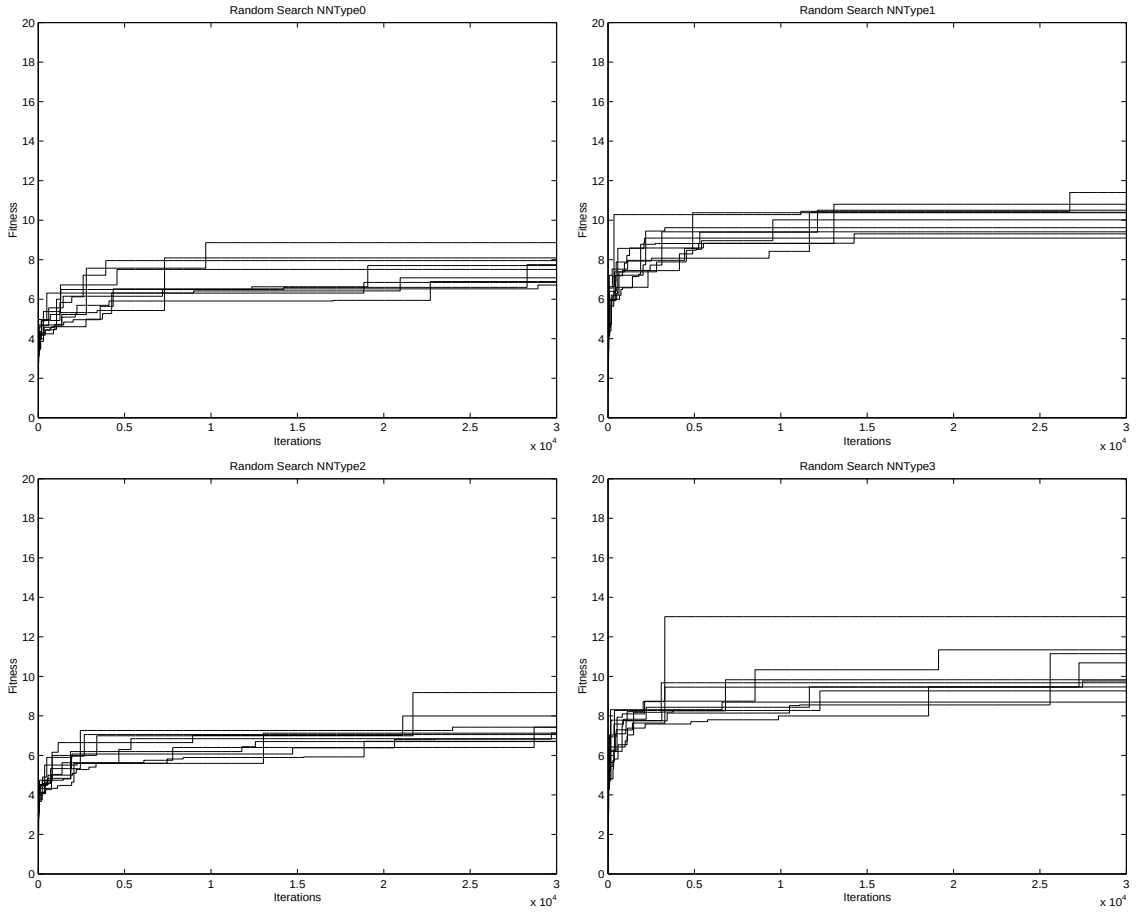

Figure 4.4: Best fitness for solutions obtained over time for 10 runs using random search for ANN architecture 1. NNType0 (top left), 2. NNType1 (top right), 3. NNType2 (bottom left), 4. NNType3 (bottom right). X-axis: Iterations, Y-axis: Fitness.

NNType	Overall Best Fitness	Average Best Fitness ± Standard Deviation	t-statistic (against NNType0)	No. of Hidden Units
0	8.8637	7.5406 ± 0.6733	-	10.0 ± 2.3
1	11.3962	10.0931 ± 0.7348	8.60	7.4 ± 4.0
2	9.1804	7.3609 ± 0.7490	(0.47)	10.9 ± 2.5
3	13.0225	10.2878 ± 1.2747	5.58	9.2 ± 4.2

Table 4.1: Comparison of best solutions found using random search over 10 independent runs.

Table 4.1 shows the overall best f_1 fitness obtained from 10 independent runs of random search along with the average best fitness and standard deviations for all four ANN architecture types. The overall best fitness was obtained using NNType3 followed by NNType1. The next best overall fitness was given by NNType2 and the worst was NNType0. In terms of the average best fitness, NNType3 had the highest value, followed by NNType1, NNType0 and NNType2 respectively. The differences between means of NNType1 and NNType3 against NNType0 were statistically significant at both $\alpha = 0.05$ and $\alpha = 0.01$. This indicates that in terms of the best controllers found, additional input-output connections in NNType1 architectures were able to yield better controllers on average when searched randomly as was the case with NNType3, which had both additional input-output as well as recurrent connections. However, recurrent-only architecture in NNType2 did not show any significant advantages over the standard feed-forward architecture in NNType0.

In terms of the number of hidden units used in the best controllers, the solutions found by random search used an average of between 7.4 and 10.9 hidden units. Surprisingly, the best solutions found using the NNType1 and NNType3 architecture types, which had higher locomotion fitness than NNType0 and NNType2, required on average less number of hidden units compared to these latter two architectures. However, the standard deviations were also much higher in NNType1 and NNType3 suggesting that the apparent inverse relationship between controller size and locomotion distance could have been due to chance encounters with small-sized networks with better locomotion capabilities.

4.5.2 Hill-Climbing

Figure 4.5 plots the frequency distribution of solutions in terms of fitness and number of hidden units. In this case, the number of hidden units does affect the controller's locomotion capabilities as evidenced by the non-uniform distribution of solutions across the objective space. The hill-climbing algorithm appeared to favor genotypes that had between 4 and 10 units in the variable hidden layer as evidenced

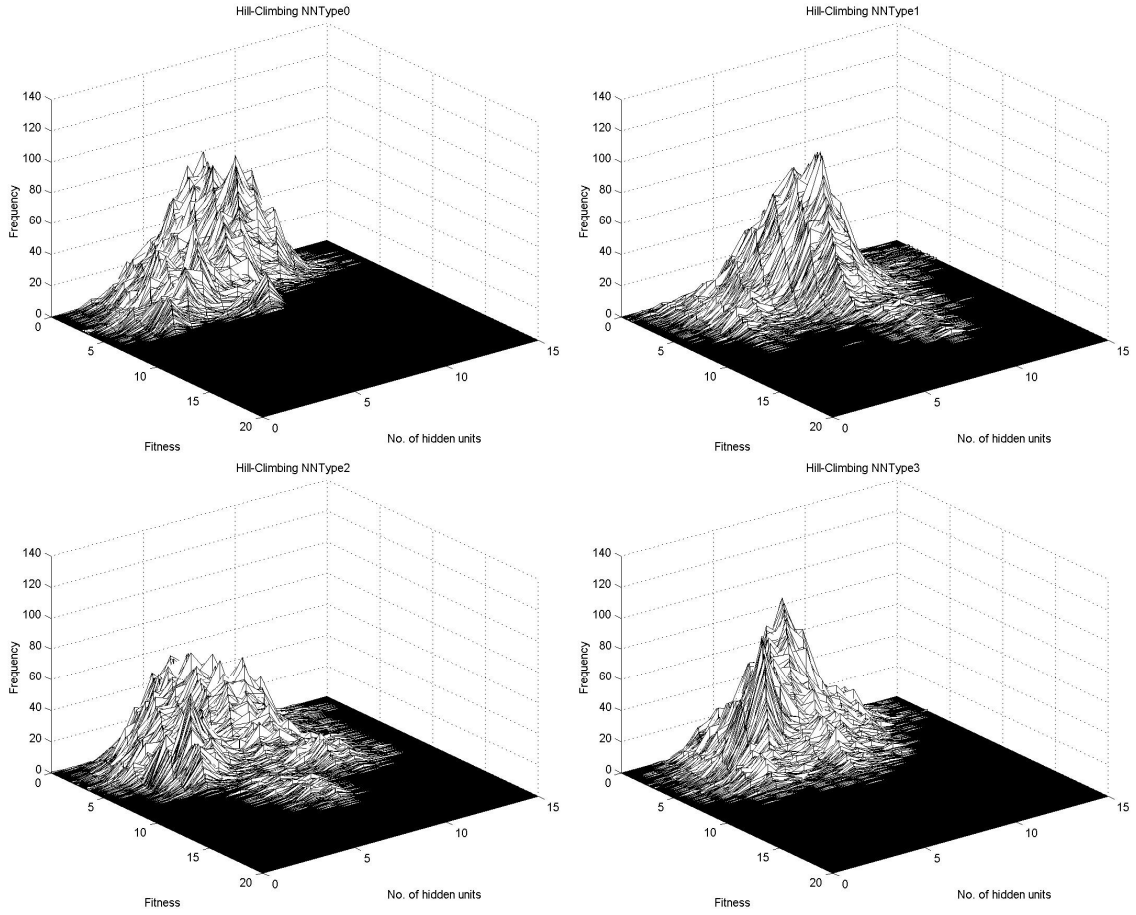

Figure 4.5: Frequency distribution of solutions using hill-climbing for ANN architecture 1. NNType0 (top left), 2. NNType1 (top right), 3. NNType2 (bottom left), 4. NNType3 (bottom right). X-axis: Fitness, Y-axis: No. of hidden units, Z-axis: Frequency.

by the significantly higher frequencies of samples appearing in this region of the objective space.

Accompanying 2D contour graphs of frequency distribution of solutions in terms of fitness and number of hidden units present in the ANN are given in Figure 4.6. The peaks illustrated in these contour graphs provide a clearer picture of where the concentration of sampled genotypes occurred. For NNType0 and NNType1, the most commonly sampled genotypes had hidden layers of 8 units peaking at a fitness of just under 4 (Figure 4.6.1) and 5 (Figure 4.6.2) respectively. NNType2

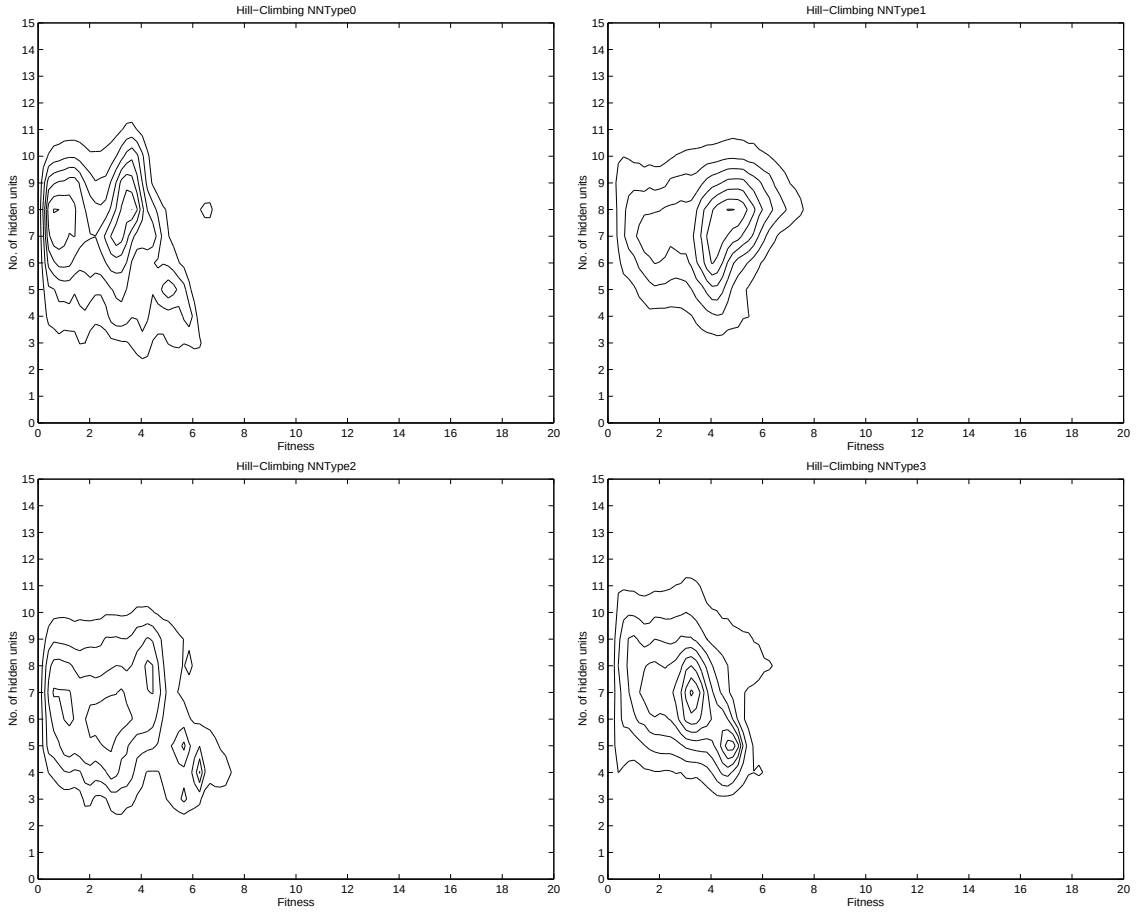

Figure 4.6: Contour graphs of frequency distribution of solutions obtained using hill-climbing for ANN architecture 1. NNType0 (top left), 2. NNType1 (top right), 3. NNType2 (bottom left), 4. NNType3 (bottom right). X-axis: Fitness, Y-axis: No. of hidden units.

had multiple but lower peaks at 4,6,7 and 8 hidden units with fitness ranging from just under 1 to just over 6 (Figure 4.6.3). Finally, the highest concentration of genotypes encountered during hill-climbing in NNType3 had hidden layers of 7 units and fitness of approximately 3.5 (Figure 4.6.4). A slightly lower but still very high peak could also be seen in NNType3 with 5 hidden units and a fitness of around 5. These observations suggest that for all four ANN architectures, hill-climbing is very susceptible to becoming stuck in local optima that are apparently very difficult to break out of.

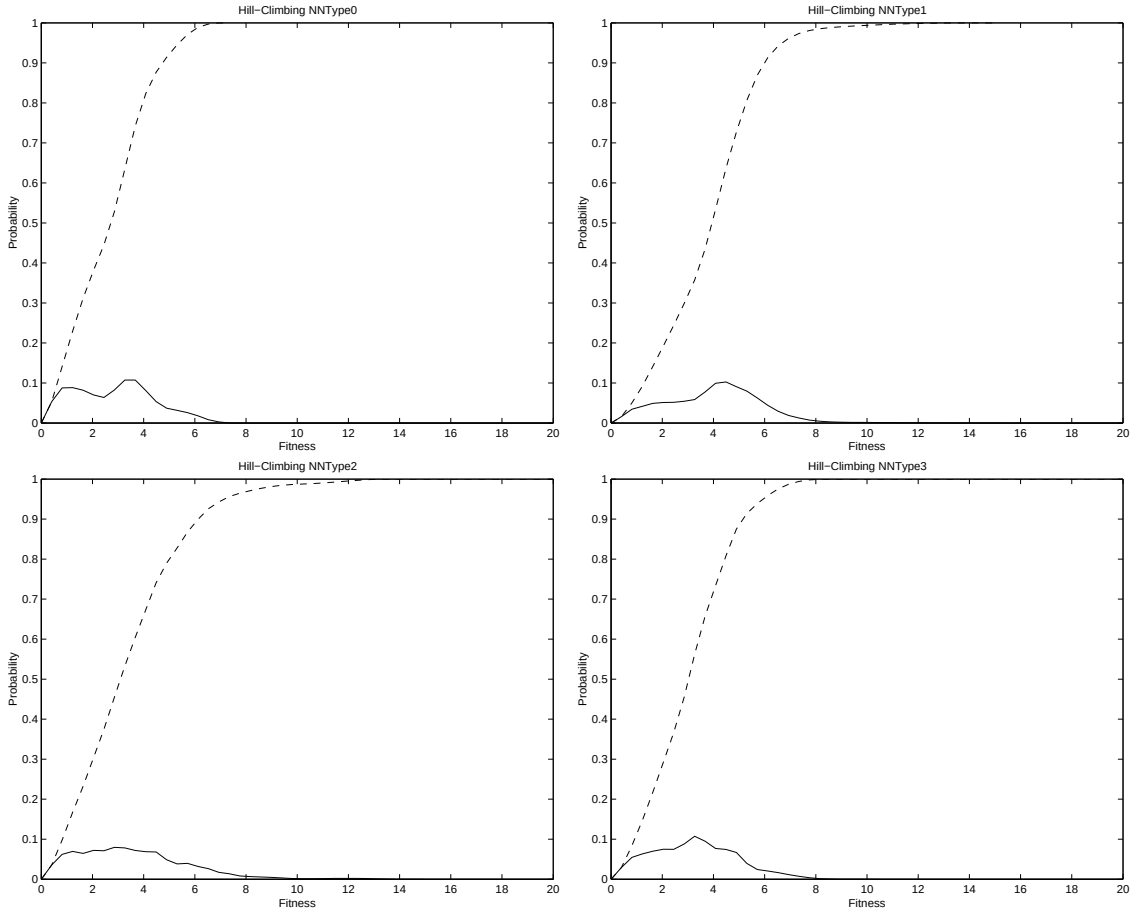

Figure 4.7: Density (solid) and cumulative (dashed) probability distribution of solutions obtained using hill-climbing for ANN architecture 1. NNType0 (top left), 2. NNType1 (top right), 3. NNType2 (bottom left), 4. NNType3 (bottom right). X-axis: Fitness, Y-axis: Probability.

Figure 4.7 shows the probability density function of solutions obtained using hill-climbing for all four types of ANN architecture. The genotypes that were sampled using hill-climbing yielded a set of solutions with much higher fitness than those obtained using random sampling. The architecture that generated the lowest fitness was NNType0 (Figure 4.7.1). Here, the probability of generating a controller approached 0 beyond a fitness of 7. This is expected since the NNType0 architecture has the least number of available connections between layers. Surprisingly, the next best architecture was NNType3 (Figure 4.7.4) which had the most number of

available connections between layers. The probability of generating a controller in this case approached 0 beyond a fitness of only 10. For NNType1 (Figure 4.7.2) and NNType2 (Figure 4.7.3), the probability only approached 0 beyond a much higher fitness of 14. This may be due to a chance encounter with a much fitter solution which caused the sampling process to cluster around a local optimum with a higher fitness, thereby biasing the distribution of solutions towards this area of the objective space (the presence of outliers in NNType1 and NNType2 is discussed again later in this section). It should be noticed though that the majority of the solutions were still sampled around the low quality areas of the search space yielding controllers with locomotion distances of between 0 and 6. This is again an indication that the landscape might be quite rugged and thus very easy for a hill-climbing algorithm to become stuck in a local optimum.

The best solution obtained over the 30,000 iterations of hill-climbing for 10 independent runs is depicted in Figure 4.8. The final solutions appeared to cluster between a fitness of 4 to 6 and is most apparent in NNType0 (Figure 4.8.1). The best solutions obtained with NNType1 (Figure 4.8.2) and NNType2 (Figure 4.8.3) had a larger spread of fitness values compared NNType3 (Figure 4.8.4). What is noticeably clear is that most of the improvement in the quality of solutions occurred within an extremely short window at the start of the search process and subsequent improvements were minimal except only in a single run each with NNType1 and NNType2 causing large plateau areas in the graphs. This supports the earlier hypothesis that the landscape may be quite rugged and that a hill-climbing algorithm may get stuck very easily in a local optimum and find it difficult to obtain fitter solutions that will enable it to move away from the local optimum.

Table 4.2 shows the overall best f_1 fitness obtained from 10 independent runs of hill-climbing along with the average best fitness and standard deviations for all four ANN architecture types. The overall best fitness was obtained using NNType1 although the overall best fitness from NNType2 was only less by 0.47. This was followed by NNType3 and the worst was NNType0. This is consistent with the mean of the best fitness which also indicates that NNType1 and NNType2

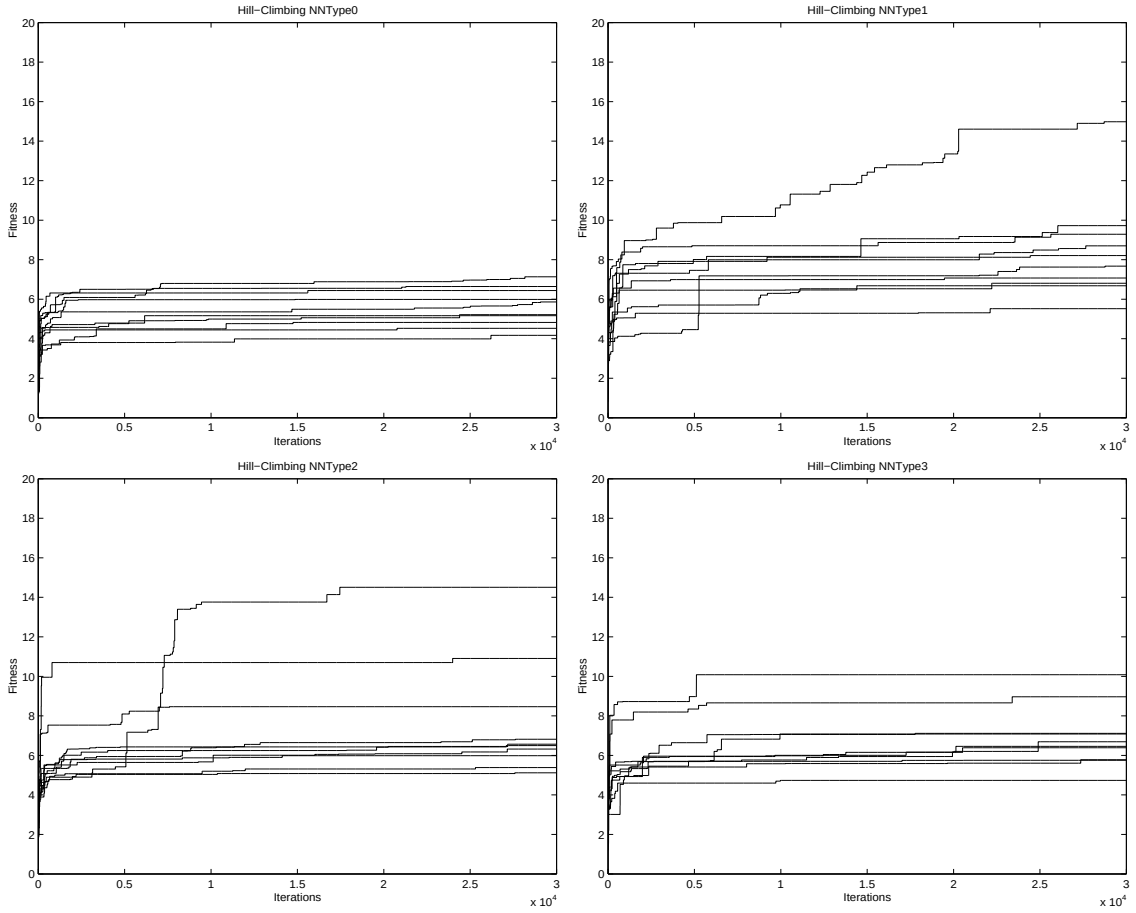

Figure 4.8: Best fitness for solutions obtained over time for 10 runs using hill-climbing for ANN architecture 1. NNType0 (top left), 2. NNType1 (top right), 3. NNType2 (bottom left), 4. NNType3 (bottom right). X-axis: Iterations, Y-axis: Fitness.

NNType	Overall Best Fitness	Average Best Fitness ± Standard Deviation	t-statistic (against NNType0)	No. of Hidden Units
0	7.1333	5.5969 ± 0.9714	-	7.0 ± 2.3
1	14.9792	8.4652 ± 2.6246	3.41	7.4 ± 2.0
2	14.5086	7.6568 ± 2.9365	2.06	6.4 ± 2.5
3	10.0832	6.9057 ± 1.5719	2.18	6.8 ± 2.2

Table 4.2: Comparison of best solutions found using hill-climbing over 10 independent runs.

were the easiest architectures to search using hill-climbing in generating efficient controllers for the creature. Correspondingly, the worst architecture was NNType0 which had the least possible number of connections allowable between layers of the network. However, it should be noted that the standard deviations for NNType1 and NNType2 were higher than NNType3 or NNType0 and may indicate the presence of outliers that were chanced upon during the search. A t-test showed that the only significant difference was between NNType0 and NNType1. NNType2 and NNType3 did not show any significant differences in terms of their average best solution compared to NNType0 at both $\alpha = 0.05$ and $\alpha = 0.01$. This is an indication that in terms of the best solutions found using a hill-climbing algorithm, only additional input-output connections (NNType1) were advantageous in yielding higher quality locomotion controllers and that neither additional recurrent-only (NNType2) nor additional recurrent plus input-output connections (NNType3) provided any significant advantages over the standard feed-forward architecture (NNType0).

There was very little difference in the number of hidden units used by the best controllers found using hill-climbing. On average, the best solutions found using NNType2 required the least number of hidden units at 6.4 while NNType1 required the most at 7.4. The standard deviation among the best controllers was also very similar across the different architectures ranging between 2.0 and 2.5 hidden units.

4.5.3 Random Walk

The frequency distribution of solutions obtained from a random walk of the fitness landscape is presented in Figure 4.9. All four architectures yielded a fairly similar but again highly skewed distribution over the objective space. The majority of genotypes sampled by a random walk again clustered around the very low quality areas of the search space and around ANNs with hidden layers of between 3 and 12 units. As such, a very high percentage of low fitness solutions again appeared to dominate the random walk. As with hill-climbing, there are indications that the size of the hidden layer affects the locomotion capabilities of the controller. This is more evident from the contour graphs that follow.

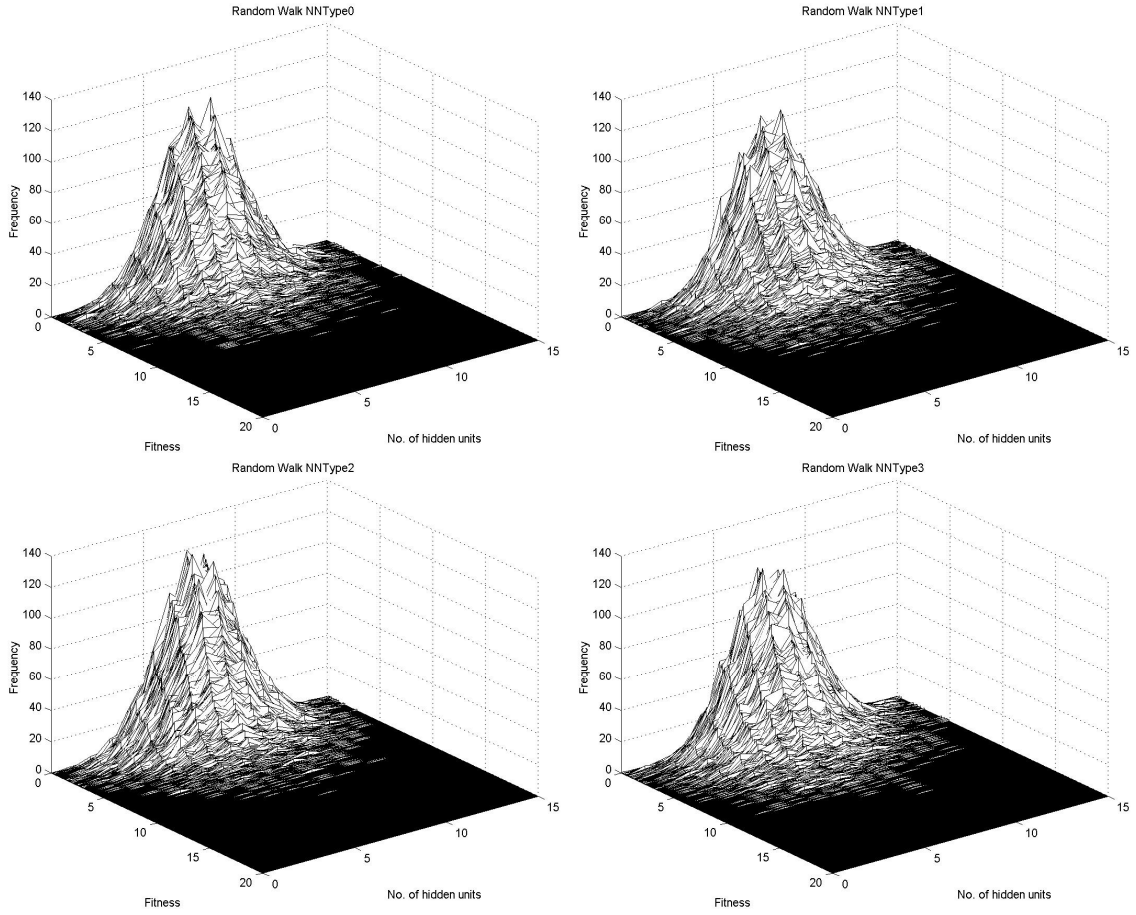

Figure 4.9: Frequency distribution of solutions using random walk for ANN architecture 1. NNType0 (top left), 2. NNType1 (top right), 3. NNType2 (bottom left), 4. NNType3 (bottom right). X-axis: Fitness, Y-axis: No. of hidden units, Z-axis: Frequency.

The effect of hidden units on the fitness of genotypes is very apparent in these accompanying 2D contour graphs depicted in Figure 4.10. Solutions with fitness above 4 had between 6 and 9 hidden units. It is also clear from the peaks on these graphs that the most frequently encountered genotype had between 7 and 8 hidden units. This may indicate that there is a large basin of attraction in this region of the search space. However, the fitness of solutions in this area is very low indeed (~ 1). Similar to the observations noted in the random search contour graphs, it was slightly easier to reach fitter regions of the controller's objective

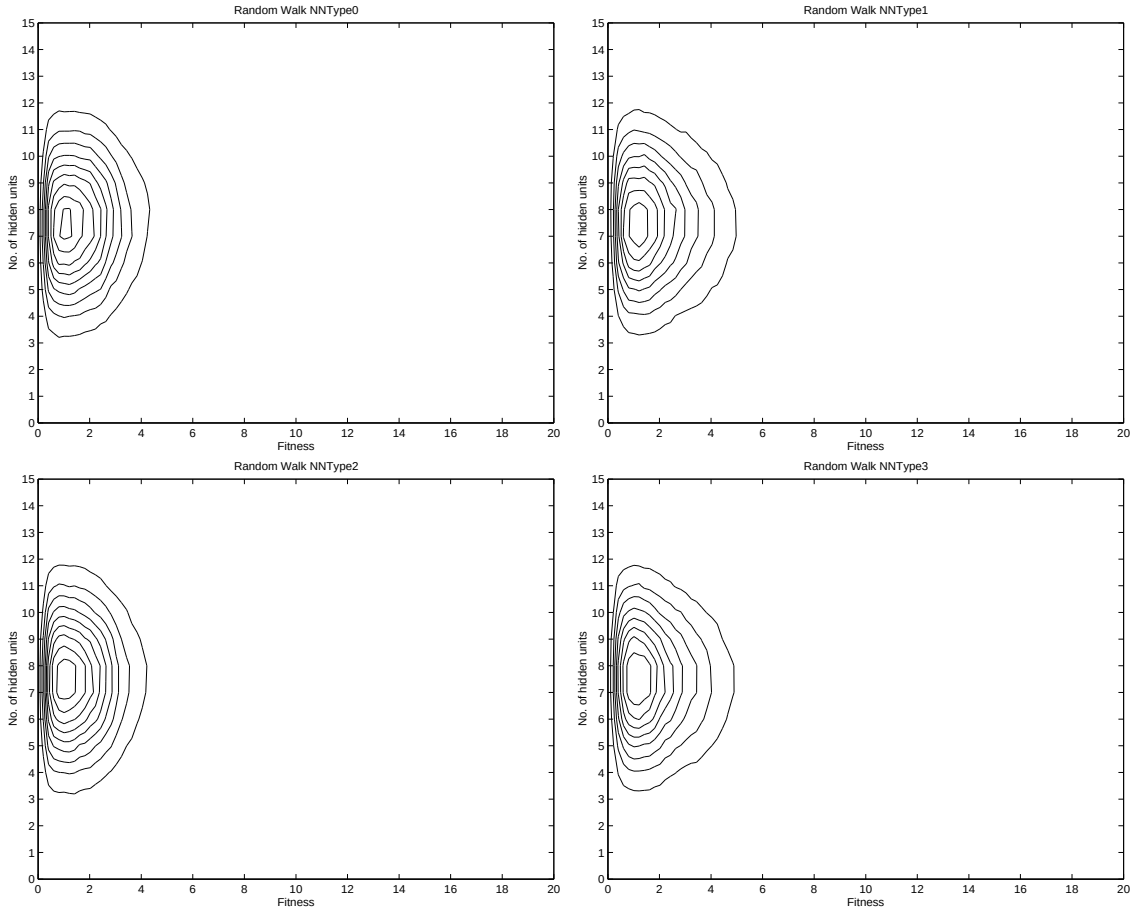

Figure 4.10: Contour graphs of frequency distribution of solutions obtained using random walk for ANN architecture 1. NNType0 (top left), 2. NNType1 (top right), 3. NNType2 (bottom left), 4. NNType3 (bottom right). X-axis: Fitness, Y-axis: No. of hidden units.

space when random walks were performed on the fitness landscape of ANNs with architecture NNType1 (Figure 4.10.2) and NNType3 (Figure 4.10.4) compared to NNType0 (Figure 4.10.1) and NNType2 (Figure 4.10.3). However, this effect is not very significant and thus only gives a weak indication that ANNs with direct connections from input to output (NNType1 & NNType3) may be easier to search than those with recurrent connections only (NNType2) or a standard feed-forward architecture (NNType0).

The probability density function of solutions obtained using random walk

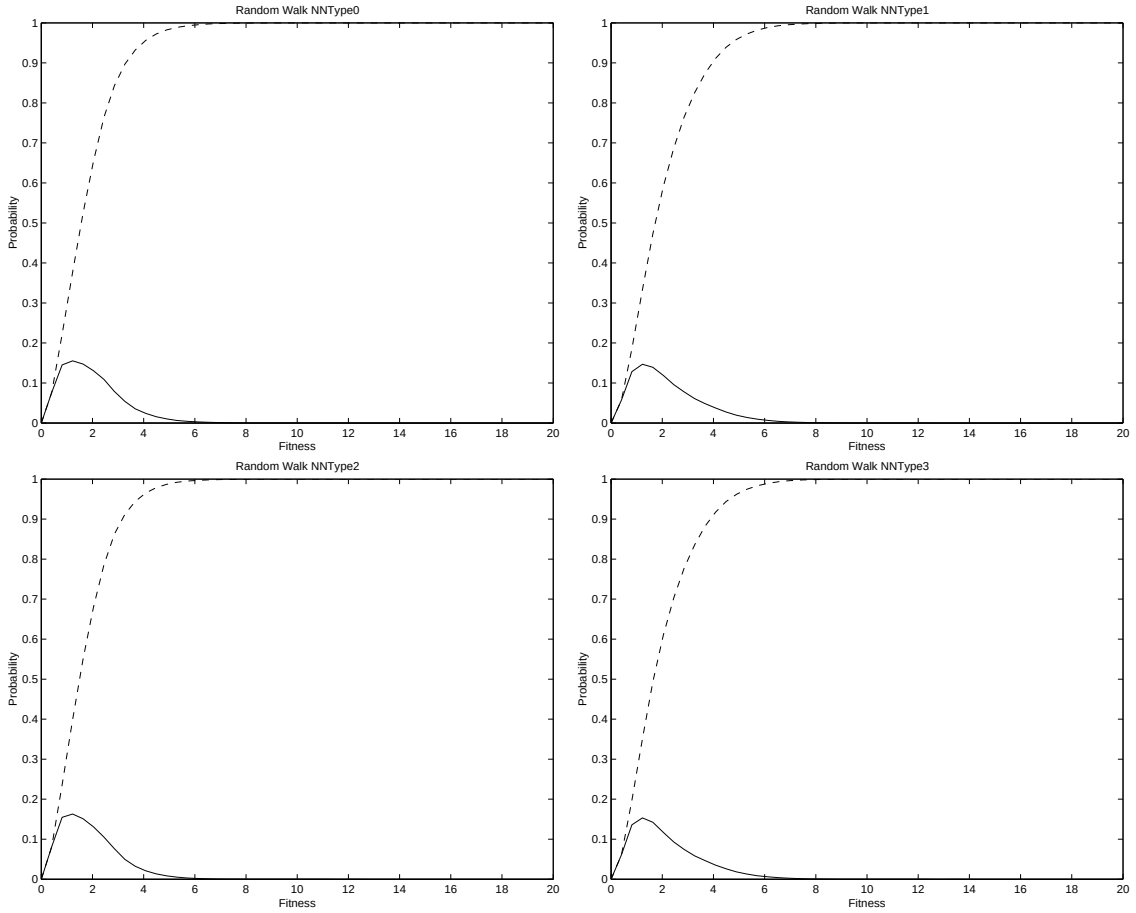

Figure 4.11: Density (solid) and cumulative (dashed) probability distribution of solutions obtained using random walk for ANN architecture 1. NNType0 (top left), 2. NNType1 (top right), 3. NNType2 (bottom left), 4. NNType3 (bottom right). X-axis: Fitness, Y-axis: Probability.

is illustrated in Figure 4.11 for all four ANN architectures. The shape of the curves was very similar to the ones obtained with random search. These graphs show that a random walk of the genotype space yields a very high percentage of low fitness solutions centered around a fitness of only 1. This supports the earlier observation from random search that the distribution of solutions in the objective space is highly non-uniform. As with random search, random walk had a slightly better probability of encountering fitter genotypes with NNType1 and NNType3 architectures compared to the NNType0 and NNType2 architectures (as evidenced by the slightly

larger areas under the curves in Figures 4.11.2 & 4.11.4 compared to the curves in Figures 4.11.1 & 4.11.3) as the probabilities approached 0.

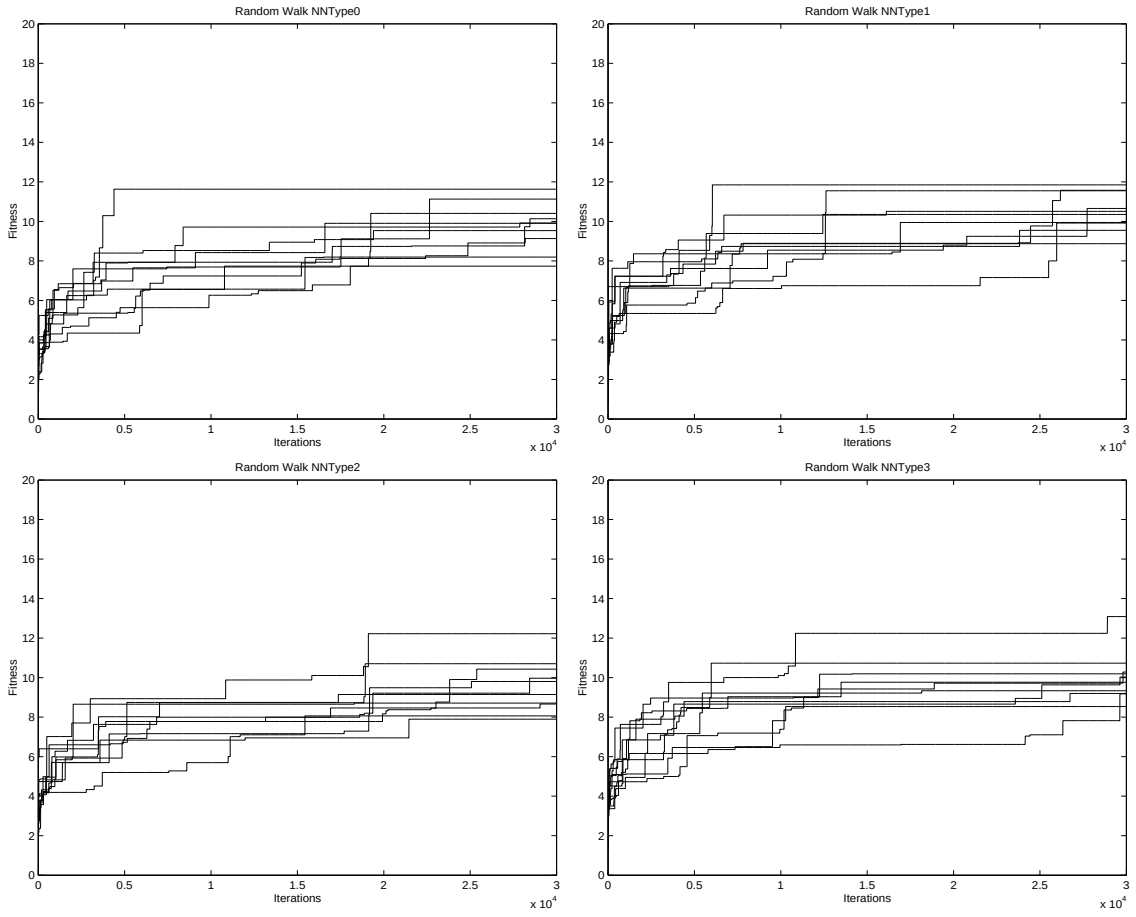

Figure 4.12: Best fitness for solutions obtained over time for 10 runs using random walk for ANN architecture 1. NNType0 (top left), 2. NNType1 (top right), 3. NNType2 (bottom left), 4. NNType3 (bottom right). X-axis: Iterations, Y-axis: Fitness.

The best solution obtained over the 30,000 iterations of random walk for the 10 independent runs is depicted in Figure 4.12. Compared to hill-climbing, the best solutions obtained for all four ANN architectures were less clustered within a specific range of fitness and had a fair spread of solutions between 7 and 12. Again this supports the earlier observations that the fitness landscape may be quite rugged and thus in a random walk, which does not have the constraint of having

to search within a neighborhood area of the best solution found so far, may have a better chance of finding a better solution by virtue of its random trajectory through different objective sub-spaces. Additionally, the progression of the best solution over time is more gradual in random walk compared to hill-climbing and the periods where the best fitness does not improve is also much shorter as evidenced by less occurrences of long plateau regions.

NNType	Overall Best Fitness	Average Best Fitness ± Standard Deviation	t-statistic (against NNType0)	No. of Hidden Units
0	11.6325	9.7725 ± 1.2009	-	7.4 ± 2.1
1	11.8494	10.4776 ± 0.9616	1.66	7.4 ± 2.2
2	12.2220	9.5602 ± 1.3372	(0.43)	8.2 ± 2.0
3	13.0900	10.0333 ± 1.2535	0.42	7.0 ± 1.4

Table 4.3: Comparison of best solutions found using random walk over 10 independent runs.

Table 4.3 shows the overall best f_1 fitness obtained from the 10 independent runs of random walk along with the average best fitness and standard deviations for all four ANN architecture types. A slightly different picture is given by random walk compared to hill-climbing in terms of the ease of searching for good controllers across the four ANN architectures. The results obtained were less differentiating for the overall best fitness and especially with the average of the best fitness. Here, although the overall best fitness was highest for NNType3 followed by NNType2, then by NNType1 and finally NNType0, the averages had NNType1 with the highest best fitness followed by NNType3, then by NNType0 and the worst was NNType2. Taking into consideration the standard deviations, the means of these best solutions were not very different from each other. As such, there are no strong indications as to what effect allowing recurrency and direct input-output connections have on the ease of searching for good quality controllers. It is also interesting to note that a comparison of the average best fitness obtained with random walk was much higher than those obtained with hill-climbing. This lends further confirmation to the fact that hill-climbing is highly inefficient in searching this landscape and that even a random walk is better in chancing upon a fitter solution. A t-test at both

$\alpha = 0.05$ and $\alpha = 0.01$ showed no significant differences for the four different ANN architectures in terms of the best solutions obtained over 10 independent runs.

As with hill-climbing, there was little variation in terms of the size of the hidden layer among the best controllers found using random walk. However in this case, NNType3 required on average the least number of hidden units, NNType0 and NNType1 had similar requirements while NNType2 required the highest number of hidden units. Also, on average across all architecture types, random walk used approximately 1 hidden unit more than the best controllers found using hill-climbing and approximately 2 hidden units less than random search.

4.5.3.1 Information Content Analysis

NNType	ϵ	$H(\epsilon)$	$M(\epsilon)$	Exp. No. of Optima	$h(\epsilon)$
0	0	0.4067 ± 0.0009	0.6226 ± 0.0050	9339 ± 75	0.5727 ± 0.0030
	1	0.6733 ± 0.0269	0.2406 ± 0.0219	3608 ± 328	0.3993 ± 0.0293
	2	0.3442 ± 0.0522	0.0771 ± 0.0186	1156 ± 279	0.1588 ± 0.0310
	5	0.0233 ± 0.0135	0.0023 ± 0.0015	34 ± 22	0.0059 ± 0.0039
	9	0.0002 ± 0.0004	0.0000 ± 0.0000	0 ± 0	0.0000 ± 0.0001
	12	0.0000 ± 0.0000	0.0000 ± 0.0000	0 ± 0	0.0000 ± 0.0000
1	0	0.4066 ± 0.0005	0.6302 ± 0.0042	9452 ± 64	0.5681 ± 0.0026
	1	0.6951 ± 0.0358	0.2581 ± 0.0275	3870 ± 413	0.4266 ± 0.0309
	2	0.3837 ± 0.0539	0.0914 ± 0.0226	1371 ± 339	0.1856 ± 0.0354
	5	0.0338 ± 0.0170	0.0031 ± 0.0019	47 ± 29	0.0091 ± 0.0054
	9	0.0003 ± 0.0004	0.0000 ± 0.0000	0 ± 0	0.0000 ± 0.0001
	12	0.0000 ± 0.0000	0.0000 ± 0.0000	0 ± 0	0.0000 ± 0.0000
2	0	0.4069 ± 0.0007	0.6241 ± 0.0042	9361 ± 63	0.5718 ± 0.0025
	1	0.6703 ± 0.0220	0.2379 ± 0.0169	3568 ± 254	0.3924 ± 0.0236
	2	0.3327 ± 0.0418	0.0720 ± 0.0144	1079 ± 215	0.1514 ± 0.0248
	5	0.0186 ± 0.0079	0.0017 ± 0.0008	26 ± 12	0.0045 ± 0.0020
	9	0.0003 ± 0.0004	0.0000 ± 0.0000	0 ± 0	0.0000 ± 0.0001
	12	0.0000 ± 0.0000	0.0000 ± 0.0000	0 ± 0	0.0000 ± 0.0000
3	0	0.4067 ± 0.0005	0.6303 ± 0.0037	9454 ± 55	0.5680 ± 0.0023
	1	0.6976 ± 0.0365	0.2608 ± 0.0281	3912 ± 421	0.4267 ± 0.0306
	2	0.3817 ± 0.0548	0.0911 ± 0.0218	1366 ± 327	0.1845 ± 0.0366
	5	0.0330 ± 0.0189	0.0032 ± 0.0020	48 ± 30	0.0090 ± 0.0060
	9	0.0002 ± 0.0004	0.0000 ± 0.0000	0 ± 0	0.0000 ± 0.0001
	12	0.0000 ± 0.0000	0.0000 ± 0.0000	0 ± 0	0.0000 ± 0.0000

Table 4.4: Information content analysis using random walk for the 4 ANN architecture types.

The information characteristics associated with the search spaces of the four different types of controller architectures are presented in Table 4.4. This analysis does not indicate any discernable differences between the four different fitness landscapes in terms of information content. All architectures have a similarly high information stability ($H(\epsilon) = 0$) of approximately 12 indicating that the differences in fitness between neighboring solutions is very high. $H(0)$ is also quite large and therefore this indicates that the diversity of shapes on the landscape is also relatively high. $M(0)$ is also large indicating that a high degree of modality was encountered during the walk, which is also evident from the large number of expected optima on the landscape. Surprisingly, $h(0)$ is significantly large, indicating that there are diverse flat and smooth landscape sections. As such, the information content analysis points to the fact that there is a mixture of both rugged as well as smooth areas in the fitness landscapes for all four architectures, the characteristics of which were both encountered a significant proportion of the time during the random walk. This is an indication that depending on which sub-spaces were being explored, the characteristics of the landscape may differ very substantially from highly rugged to very smooth. As such, the ability for search algorithms to find increasingly better solutions may be highly dependent on the initialization and trajectory of the search on the fitness landscape.

4.6 Limitations and Future Work

The idea of neutral plateaus within search spaces, where large numbers of genotypes have similar phenotype fitness values, has recently been of particular interest (Huynen 1996; Barnett 1998; Smith, Philippides, Husbands, and O'Shea 2002). It was suggested that problems with high degrees of neutrality, whether an inherent feature of the original problem or artificially introduced through genotype redundancy, tend to produce landscapes that are easier for EAs to escape local optima and find better solutions (Shackleton, Shipman, and Ebner 2000; Vassilev and Miller 2000). However, both autocorrelation and information content measures

are unable to provide any information regarding the presence of such neutral areas within search spaces (Barnett 1998; Smith, Philippides, Husbands, and O'Shea 2002). A new methodology for elucidating neutrality was proposed by Smith, Husbands, Layzell, and O'Shea (2002). However, these methods measure for evolvability rather than providing direct characterizations of the actual fitness landscapes. Furthermore, there is also evidence that neutrality does not necessarily improve the evolutionary search process (Smith, Husbands, and O'Shea 2001a) and hence, the general significance of neutrality within search spaces remains somewhat inconclusive.

In order for both autocorrelation and information content measures to work, an important assumption needs to be made — that the fitness landscapes being analyzed are statistically isotropic (Weinberger 1990; Vassilev, Fogarty, and Miller 2000). Importantly, it was highlighted by Smith, Husbands, and O'Shea (2001b) that the search space for an evolutionary robotics task environment displayed strong indications of anisotropy. Therefore, the results obtained from landscape analysis methods that make the explicit assumption of isotropy needs to be treated with some caution. Another important observation made by Smith, Husbands, and O'Shea (2001b) is that results obtained from using sparse sampling methods such as random sampling and random walk may provide a highly inaccurate picture of the actual fitness landscape when the distribution of solutions in the search space is non-normal and highly skewed, as was the case in the experiments carried out in this study. This problem of heterogeneity is present in many hard problems and may lead to inaccurate characterizations of landscapes when using techniques that assume homogenous distribution of solutions. Another landscape feature that may also affect the efficacy of the evolutionary search process is the degree of deceptiveness of the problem and again is not a characteristic that can be ascertained with current landscape measures (Smith, Husbands, and O'Shea 2001b).

From our experiments, we have also noted five additional limitations associated with these existing landscape analysis methods. Where relevant, we provide some pointers on open research questions and possible future work that will further

extend the usability and generalization power of these techniques.

1. The problems being analyzed are generally problems that exist in high-dimensional space whereas the landscape analysis methods such as the autocorrelation and information content measures only provide some statistical characterization of the actual search space. As such, these methods only capture a limited amount of information along a particular dimension. A methodology that is able to capture more information from the high-dimensional search spaces would thus be desirable in order to give a more comprehensive and accurate appraisal of the actual fitness landscape.

2. Obtaining the landscape points through a random walk results in evaluating the search space in one particular direction. This implies a bias in the way in which the landscape is being characterized. In order to reduce this bias, multiple walks need to be carried out. However, generating multiple random walks is extremely time-consuming and the time spent on characterizing the fitness landscape may actually take longer than that needed to simply proceed with the actual optimization process of finding a solution. More research effort is required towards designing a computationally more efficient technique for obtaining fitness values in fitness landscape analyses.

3. The operators involved in generating landscape points function only in the genotype space. As such, these landscape measures do not provide any information whatsoever concerning the genotype-to-phenotype mapping. Again, it would be highly desirable to have a technique that is able to give some insights into how different genotype-phenotype mappings affect the fitness landscapes.

4. Current landscape analysis methods only work with a single fitness function. If the problem is multi-objective, then none of the existing measures are able to generalize to higher dimensional objective spaces. For example, the parameter ϵ from the information content analysis can only be used to perform analysis on one objective function at a time. Furthermore, it does not show the degree of correlation between the objective functions. With the resurgent interest

in multi-objective optimization approaches for solving real-world problems, an analysis technique able to characterize such multi-functioned landscapes would be of great value to both researchers and industry practitioners alike.

5. The last and perhaps most serious drawback to current landscape analysis techniques is their inability to capture the true landscape of evolving populations in EAs. Current methods rely on generating a single genotype and tracing its single path through the fitness landscape. It must be remembered that in EAs, an entire population is moving through the fitness landscape, not just a single individual. Considerations need to be given to the coverage of the search space achieved by the evolving population as the optimization progresses. Additionally, concurrent evolutionary paths somehow need to be tracked in order to provide a more accurate picture of how the actual formations present on the EA landscapes affect the transition of populations from one generation to the next.

4.7 Chapter Summary

An analysis on the fitness landscape for four different types of ANN architecture yielded the following results:

- The advantages or disadvantages of having recurrent connections and/or direct input-output connections in the ANN for controlling the artificial creature remain unclear. In terms of average best solutions found, random search performed better using NNType1 and NNType3, hill-climbing performed better using only NNType1 whereas random walk showed no performance differences whatsoever between the four types of architectures. Furthermore, hill-climbing performed worse than both random search and random walk in three out of the four architectures and worse than random walk in the remaining case. As such, whether the search space difficulty is lowered by adding recurrent and/or input-output connections to a standard feed-forward ANN architecture cannot be concluded with certainty.

- The fitness landscape of all four ANN architectures is highly similar. The landscape analysis conducted using informational measures did not show any discernable differences between the four search spaces.

- The fitness landscape of these evolutionary search spaces has both rugged and smooth sections depending on the sub-spaces being explored. Additionally, the variety of rugged shapes on the landscape is high indicating that epistatic interactions between genes in the genotype are high. A correspondingly high degree of modality in the fitness landscape was also noted.

- The solution space is highly heterogeneous — a uniform sampling of the genotype space yielded a highly skewed distribution of solutions in the objective space.

- There are serious deficiencies associated with current landscape analysis methodologies, especially for analyzing non-homogenous and anisotropic search spaces, such as in artificial creature evolution. Additional limitations were also noted for characterizing evolutionary search spaces using such techniques.

It remains unclear as to whether the NNType0, NNType1, NNType2 or NNType3 architecture provides a more amenable search space. As such, experimentation on all four architectures will be required in searching for fit artificial creature controllers. In the next chapter, we will present our evolutionary optimization algorithm using a Pareto multi-objective methodology for evolving controllers based on these four ANN architectures.