Project Number 001907

DELIS

Dynamically Evolving, Large-scale Information Systems

Integrated Project

Member of the FET Proactive Initiative **Complex Systems**

Deliverable D5.6.1

Classification of Families of Information Networks based on their Topological and Functional Structures and their Relation with the Underlying Fitness Landscape

Start date of the project:	January 2004
Duration:	48 months
Project Coordinator:	Prof. Dr. math. Friedhelm Meyer auf der Heide Heinz Nixdorf Institute, University of Paderborn, Germany
Due date of deliverable:	December 2005
Actual submission date:	January 2006
Dissemination level:	PU – public
Work Package 5.6:	The structure of tinkered landscapes
Participants:	Universtitat Pompeu Fabra (UPF), Barcelona, Spain Universita di Bologna (UniBO), Italy
Authors of deliverable:	Ricard V. Solé (ricard.sole@upf.edu) Javier Macia (javier.macia@upf.edu) Sergi Valverde (svalverde@imim.es) David Hales (hales@cs.unibo.it) Ozalp Babaoglu (babaoglu@cs.unibo.it) Stefano Arteconi (arteconi@cs.unibo.it)

Abstract

This report comprises the complete D5.6.1 deliverable as specified for workpackage WP5.6 in Subproject SP5 of the DELIS (Dynamically Evolving Large-scale Information Systems) Integrated Project.

The essential goal of the DELIS project is to understand, predict, engineer and control large evolving information systems. It is desirable for such systems to display functional robustness under internal and external noise and perturbations. Given that many biological and technological networks demonstrate this, it would be of value to understand how this is achieved. The approach of this workpackage is to consider, categorise and analyse the fitness spaces of classes of information networks and relate them to functional properties.

Results are presented from experiments with feed-forward binary networks which demonstrate some surprising and potentially useful properties. We relate some of these insights to possible implications for robust peer-to-peer protocol design[1].

[1] Most papers produced within DELIS are available from the DELIS website as DELIS Technical Reports. Where this is the case references are appended with the DELIS Tech Report number in square brackets. This indicates the paper was produced within the DELIS project, not some other project.

Contents

1 Introduction 3

2 Feed-forward Boolean Networks 3

3 Results 5

4 Summary 9

4.1 General implications 9

4.2 Implications for robust peer-to-peer network design 10

1 Introduction

The tempo and mode of technological change, as it occurs with biological evolution, are strongly tied to the underlying fitness landscapes where evolutionary change takes place. It is the topology of these landscapes what largely canalizes the ways in which innovation occurs, how reliable designs are or how evolvable are they expected to be. Some indirect evidence indicates that such technology landscapes might be neutral [1]: many changes have little effect of systems performance, whereas some particular changes can help a rapid fitness gain. In such landscapes, neutral domains percolate the entire space and help exploration of optimal solutions in a very efficient way [2]. In molecular biology, the neutral structure of fitness landscapes is the key for efficient search in a huge combinatorial space.

In technology, as opposed to biology, the structure of the landscape itself might evolve by using appropriate choices while developing the designs of interest. On the other hand, understanding how complex biosystems achieve their structure with high degree of reliability might help designing new types of technological artifacts and define good strategies that keep the landscape evolvable [4,5]. In order to explore these questions, simple but well-defined models need to be constructed [6]. Such models would include both simulation of evolved computational networks, evolved hardware and software designs. They would establish the appropriate criteria to define the mapping between structure and function. Such mapping is well known in biology, and is known as the genotype-phenotype map.

Understanding the properties of such mapping has been enlightening in providing a rationale for biological evolution and the potential for evolvability [7] but not yet exploited and understood in the context of man- made designs. What is more important: the way in which evolved designs is constrained might actually shape the landscape itself, strongly enhancing the possible ways of finding optimal designs. Current studies indicate the existence of strong constrains in the development of technological networks as described in terms of technology graphs [8] that could be avoided by imposing well-defined limitations to the way the system is designed.

In order to understand how improvements in technological design can take place and how the evolutionary dynamics of such designs can improve their final performance, it is essential to understand the nature of their fitness landscapes. The final goal of our approach is to understand the links between fitness landscape architecture and the nature of the functionalities that we want to improve through evolutionary optimization. A first step, to be discussed below, was to explore such structure under a small number of assumptions. Specifically, we decided to analyse the structure of the landscapes of feed-forward networks with predefined Boolean as targets of the optimization process. The results of this first study are remarkable, since they indicate that technology landscapes might be largely neutral, sharing most of the statistical properties displayed by RNA landscapes. These findings support a rather universal organization of fitness landscapes and provide a rationale for exploring their structure under general conditions as well as how they can be properly tuned to avoid evolutionary traps.

2 Feed-forward Boolean Networks

Network structure and function. The model used is a very simple feed-forward structure. The network has I inputs, O outputs, and a $H \times M$ block of hidden units, as figure 1 shows. Units in the hidden block can connect only to the layers above them (thus avoiding cycles and cyclic behavior), including inputs, and the outputs can connect to the hidden units but not directly to the inputs. In addition, the number E of connections is fixed.

The units s_i of the network have a Boolean nature (i.e. $s_i \in \{0, 1\}$), and perform a simple integer threshold function of the inputs, that is,

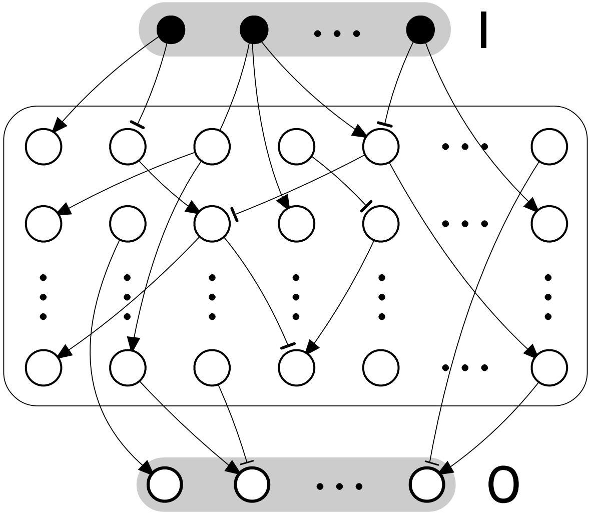

Figure 1: Topology of the model of FFN used. It consists of inputs (I units), hidden units ($H \times M$ units), and outputs (O units). Units can connect strictly to the layers above, thus avoiding cycles, except for the outputs, which cannot connect directly to the inputs.

$$s_i(t+1) = \Theta\left(\sum_{j=1}^{N} w_{ij} s_j(t)\right). \tag{1}$$

The Θ function is defined as: $\Theta(x) = 0$ for $x \leq 0$, and $\Theta(x) = 1$ for $x > 0$ (thus the XOR function is not possible with only one unit). The weights w_{ij} are drawn from the set $\{+1, -1, 0\}$ representing positive, negative, or absent regulation, respectively. When an input is presented, the output can be computed propagating the inputs in a non-dynamical way just as if all units changed at once.

By this definition, the input layer of the circuit models external states (or the result of sensing external states) being presented to the network, and the bottom layer models the output, representing needed response. The network, therefore, "computes" the appropriate set of responses for each external state. This feed-forward topology is widely used in artificial neural networks. This distinction is what enables us to more apropriately define a genotype and a phenotype. Additionally, the model presented has an asymmetrical treatment of activation since to produce activity in one unit, its preceding units have to be also active (an all-zero input always produces an all-zero output).

Wiring-function mapping. Given this structure, we can easily define a genotype and a phenotype. The *genotype*, W_i, is defined as the ordered string of all weights w_{ij}. To compute the phenotype we first calculate all the input-output pairs, with all possible different inputs from $I_1 = \{0, 0, 0, \ldots, 1\}$ to $I_{2^N-1} = \{1, 1, \ldots, 1\}$ (with the exception of $I_0 = \{0, 0, \ldots, 0\}$ which, by definition, yields an all-zero output). The entire list of outputs fully describes the Boolean function Φ_i, or *phenotype*.

Two sets, $\mathbf{W}$ and $\mathbf{\Phi}$, describe the universe of possible wirings and functions, i.e. the sets of all possible genotypes and phenotypes. The genotype-phenotype map between wiring and function is then defined as

$$\mathbf{\Omega} : \mathbf{W} \longrightarrow \mathbf{\Phi}. \tag{2}$$

For each genotype $W_i \in \mathbf{W}$, we have a phenotype $\Phi_i \equiv \mathbf{\Omega}(W_i) \in \mathbf{\Phi}$. Evolution and adaptation occurs through changes in wiring eventually leading to changes in function. How adaptation proceeds largely depends on the nature of the mapping $\mathbf{\Omega}$.

In order to characterize $\mathbf{\Omega}$, a metric or topological measure is needed (ref). Given the discrete

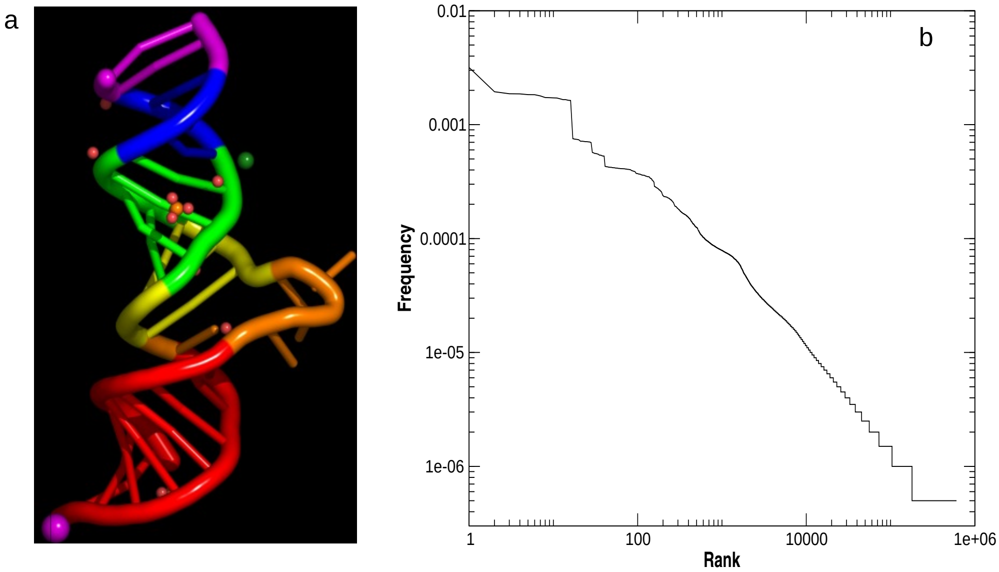

Figure 2: Neutrality in landscapes. Previous studies on neutral landscapes have concentrated on folding properties of RNA molecules (a). In particular, it was found that the frequency distribution of folds follows a rather universal power law distribution. Such distributions are very similar to those found in studying the landscapes of feedforward nets. In (b) we show the observed rank-frequency distribution of functions for our FFNs. A network with $I = 3, O = 4, H = 8, M = 3$ and average connectivity $\langle k \rangle = 3$ was chosen and a sample of 2×10^6 was taken. The distribution follows a general form of Zipf's law, i.e. $P(r) = a(b + r)^{-\alpha}$ with $\alpha \approx 0.75$.

nature of both spaces, phenotypic and genotypic distances can be defined, respectively, as

$$d_P(\Phi_a, \Phi_b \in \mathbf{\Phi}) = \sum_k \left| \Phi_a^k - \Phi_b^k \right|, \tag{3}$$

$$d_G(W_a, W_b \in \mathbf{W}) = \frac{1}{2} \sum_k \left| W_a^k - W_b^k \right|. \tag{4}$$

Phenotype distance is, therefore, equivalent to the Hamming distance of a bit string, and genotype distance is similar, measuring the number of different connections (that is, either displaced or with reversed sign, which contribute 2 to the sum, hence the 1/2 factor).

Throughout the work, we have used small networks, usually with $I \in \{3, 4\}$, $O \in \{4, 5\}$, $H = \{7, \ldots, 11\}$, and $M \in \{3, 4\}$, with an average connectivity of $\langle k \rangle \approx 3.0$ which allowed us to more exhaustively explore genotype and phenotype spaces.

Network wiring changes. Mutation is implemented as the simplest random procedure that alters the wiring of the network: an existing edge is chosen at random and it is removed, and a new edge is chosen also at random and it is added, with a negative weight with probability 1/3 and a positive weight with probability 2/3. The bias in the weights tries to compensate for the fact that a balanced network is less active overall.

3 Results

Frequencies of shapes. The frequencies of different functions were obtained using a sample of 2×10^6 random wirings and computing the input-output table by the rules given. The rank-plot of this data is shown in figure 2 (left), evidencing a general form of power law. Thus, there are some

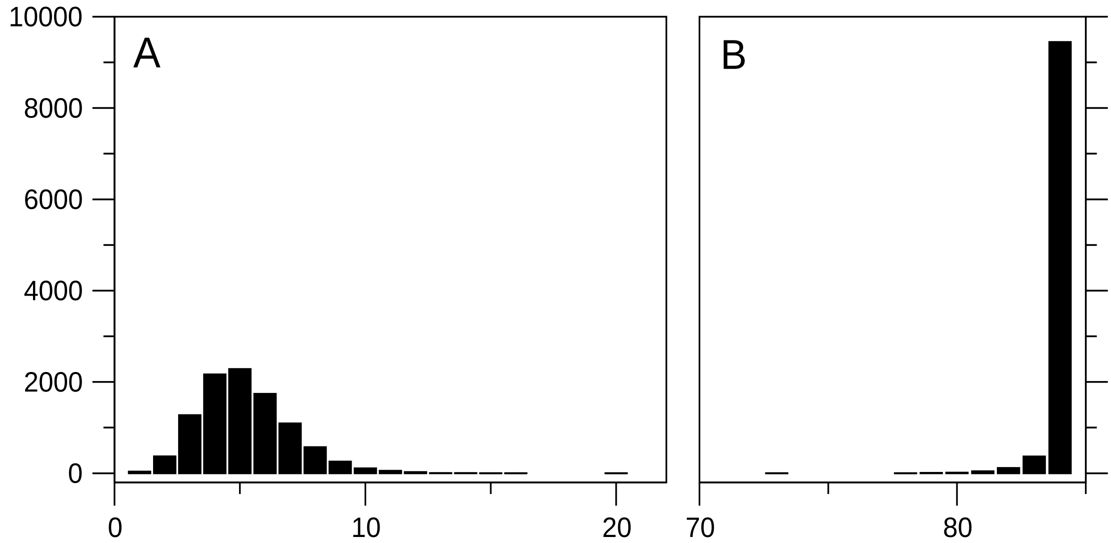

Figure 3: Probability densities of genotype distance d_G in two different experiments. (**A**) A target wiring is chosen at random, and an approach in genotype space to this target is tried by 10^4 different wirings. For each trial wiring, a neutral path is performed in which neighbours have to maintain function and at the same time be closer to the target (average final distance to the target: 5.18 ± 1.82). (**B**) A similar neutral path is tried. In this case, a random genotype is chosen and a trail copy is mutated, step by step pushing it away in genotype space, while again conserving function (94.4% of genotypes were completely rewired while still maintaining phenotype).

frequent functions and many rare ones. The most frequent is the all-zero outputs function: there is a certain probability that no activation path exists between inputs and outputs, although overall, the probability is rather low (3.1×10^{-3}). The small plateau following this first value at the low rank zone corresponds to those functions with an equal output for all different inputs. Also, within groups of 10 or more genotypes with the same phenotype, we calculated the average genotypic distance. In all cases we obtained the same distance (within 1%) as that of a random sample, confirming that phenotypes have genotypes distributed uniformly over genotype space.

Neutral paths. Two experiments were performed to check for the existence of neutral networks (i.e., regions in $\mathbf{W}$ consisting of neighboring genotypes with the same phenotype), both involving neutral paths with monotonously decreasing (and increasing) distance from a reference sequence. In the first (fig. 3), a target wiring W is chosen at random, and a second random wiring is chosen as trial genotype, T. Next, if a random neighbour T' of T conserves the phenotype and has a smaller $d_G(W,T')$, it is accepted as the new T. The process is repeated 10^4 times. The final $d_G(W,T)$ is an upper bound of the minimum distance of the two phenotypes Φ^W and Φ^T. It is remarkable that this distance is on average 5 (out of 84).

In the second experiment, a random wiring W is chosen, and a copy of it is taken as trial, T. At each step, if a random neighbour T' of T has the same phenotype as W and $d_G(W,T')$ is larger, it is accepted as the new T. The process is repeated 10^4 times. The final $d_G(W,T)$ correlates with the size in genotype space of the neutral networks. In this experiment, 94.4% of the genotypes could be completely rewired (maximum genotype distance of 84) while keeping the phenotype (the smaller distance being 73). Neutral networks, therefore, percolate through genotype space.

Map Structure. To understand how the map $\mathbf{\Omega}$ is seen from the viewpoint of an average genotype W, we evaluated the probability that another genotype W' at distance $d_G(W,W')$ had a phenotype at a certain distance d_P (the Structure Density Surface, or SDS. This probability was evaluated by

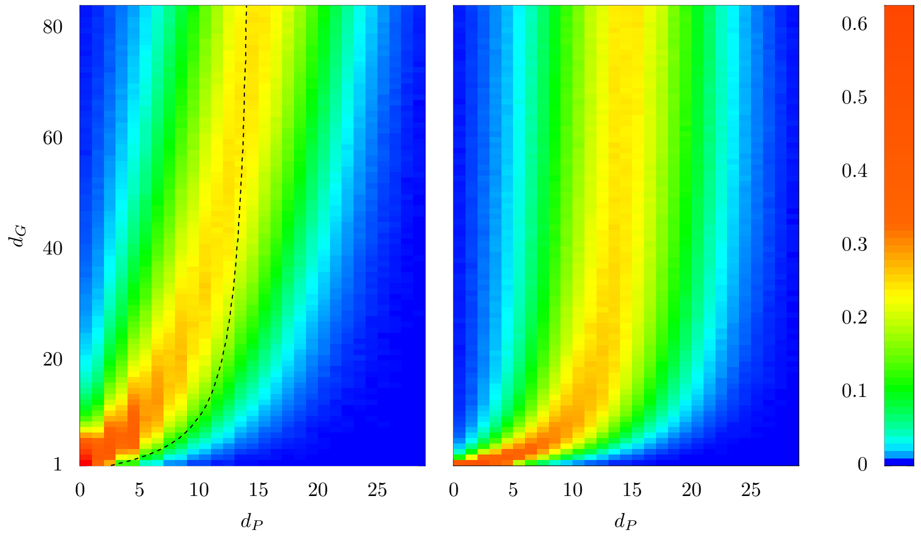

Figure 4: The genotype-map structure as seen from a group of genotypes in two different numerical experiments $((I, O, H, M, \langle k \rangle) = (3, 4, 8, 3, 3.0))$. **Left**. From 10^3 random wirings, the histogram of phenotypic distances of 10 other wirings at progressively higher distances starting at them. **Right**. The same histogram starting at the 80 selected genotypes with increased mutational diversity. The average phenotypic distance of the right part is plotted as a dashed black line in the left part for comparison.

producing progressively distant mutants from a given starting wiring, and evaluating the distance d_P of their respective functions. We took 10^3 reference wirings, and for each one, we chose 10 different wirings at all values of d_G and computed d_P. Figure 4 (left) shows the resulting two-dimensional histogram. As d_G increases, we have a picture of how the average phenotypic distance behaves. For values of d_G between 1 and 20, there is correlation with the phenotype (a few changed wires usually produce a few changes in function), but after that, the distance to phenotypes is progressively similar to the random case, i.e. we can expect to find an almost random phenotype if we change 20 or more of a given network's links (a 25% of the total).

Together with the covering of genotype space by the average phenotype, these results suggest the presence of a neighbourhood (a high-dimensional ball) around a particular FFN whose wirings include all common functions, in consonance with the RNA case [5]. However, there are some differences. Firstly, a few changes in an RNA sequence mostly result in a changed shape: even in the case of only one mutation, an RNA molecule can have a drastically different structure (up to a 66% change in phenotype distance). This is in contrast to FFNs, which in general are more robust for a small number of mutations (fig. 4 left). Secondly, the radius of the high-dimensional ball around which a genotype can find all common functions is somewhat smaller in the RNA case (around 15%). These differences lead us to think whether a special group of FFNs could be more sensitive to mutations, and therefore deeply alter the perception of genotype space in the same experiment with them as starting points.

Mutational Sensitivity. To test this hypothesis, we searched FFNs with a higher average sensitivity. Starting at a random genotype W, we measured its mutant diversity with two parameters: μ (satisfying $0 < \mu < 1$), indicating the fraction of mutants with a different phenotype (i.e., non-neutral), and δ (satisfying $0 < \delta < 1$), measuring the fraction of unique phenotypes within the non-neutral group (or diversity). A pair (μ, δ) with values $(0.85, 0.1)$ describes a robust FFN with an 85% of neutral mutants, in which non-neutral phenotypes (the remaining 15%) are repeated 10 times on average.

We chose a group of 80 random FFNs and with each one, we performed a hill climbing process,

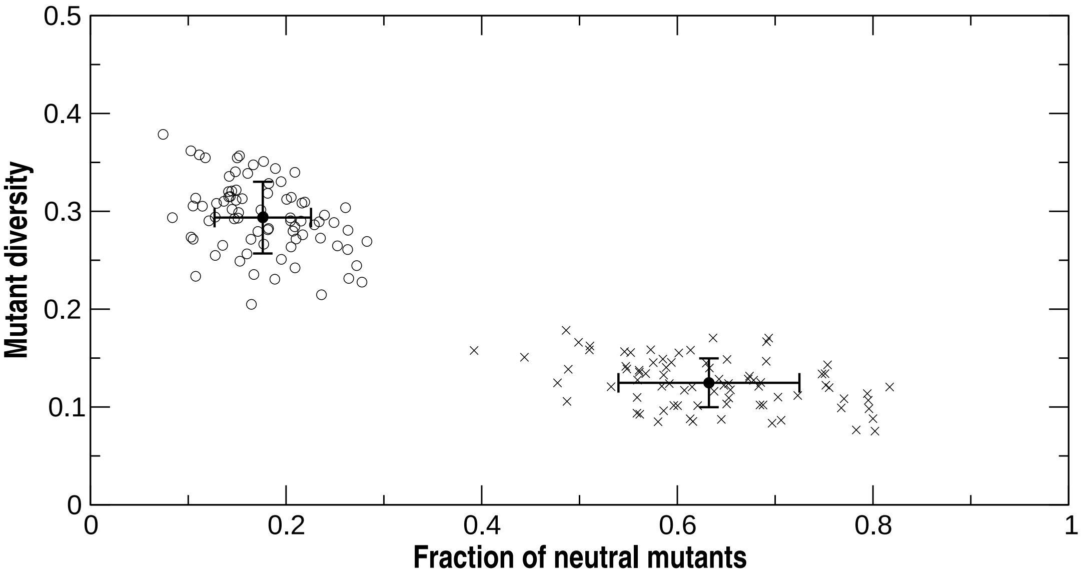

Figure 5: A set of 80 random genotypes (crosses) was optimized for higher mutant diversity, conserving their phenotype (circles). A hill-climbing process was performed with each one so that either the fraction of neutral mutants (μ) decreased, or the diversity of the different mutants (δ) increased. The initial average values of $(0.63, 0.12)$, indicating very robust networks, were changed into $(0.17, 0.29)$, suggesting the big differences in the sensitivity to mutations of genotypes within the same neutral network.

successively choosing mutants with either lower μ or higher δ, but conserving phenotype. The size of the mutant samples was 2×10^3. The starting and ending sets are shown in figure 5. It is immediately clear that the differences in mutation sensitivity are enormous. This differences suggest that within a neutral network, special FFNs could serve as gateways, giving populations access to a very high number of different phenotypes from the same spot.

This is confirmed by the structure of the genotype-phenotype map as viewed from the sensitive group of FFNs (fig. 4). The average phenotype distance is plotted as a dashed line on the left, for comparison. The distance separating this group from a random genotype is halved, indicating a much smaller search space for this special FFNs.

Dynamical transitions. The structure of the mapping is further made clear by studying evolutionary dynamics. Following previous approaches we did some optimization experiments in which a population of FFNs evolves towards a target function. We choose a very unfrequent target phenotype Φ_T (the phenotype with the highest average output-pair entropy in a sample of 10^5), and a group of 10^3 FFNs chosen at random serves as initial population. At each iteration, a new population results from fitness-proportionate reproduction, with the fitness of a genotype W_i being $F_i = e^{-d_P(\Phi_i, \Phi_T)}$. Every reproduced FFN has a probability $p = 0.3$ of being mutated.

An example of the dynamics displayed by this kind of process is shown in figure 6. The average distance to the target decreases with time, showing punctuated events in which fitter genotypes spread rapidly within the population. Between these transitions, a stable regime characterized by an increase in genetic diversity takes place. A sample of the average d_G of FFNs in the population shows increasing values, which drop abruptly whenever a fitter genotype takes over. This is the typical result that should be expected from the random drift of a population within a genotype space in the presence of neutrality.

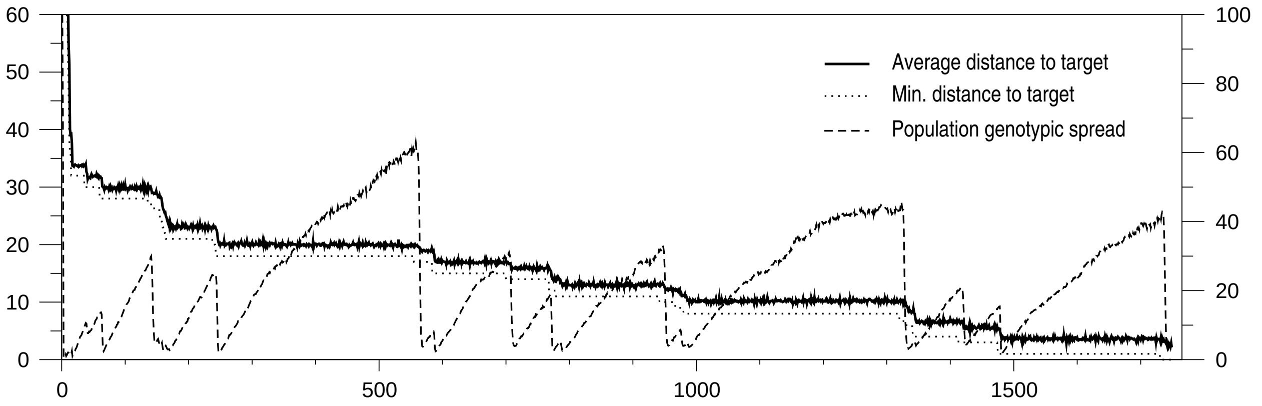

Figure 6: An example of the evolutionary optimization towards a specially chosen phenotype (see text). The parameters of the FFNs are $I = 6$, $O = 5$, $H = 10$, $M = 5$, $L = 3.0$. The initial population consists of 10^3 random FFNs. The population finds the target in 1749 generations, and the dynamics shows punctuated events in which population diversity falls abruptly. The vertical axis on the left shows distance to the target, or d_P. On the right, it shows genotypic distance, or d_G. Genotypic spread is the average d_G between 10^3 random pairs of genotypes within the population.

4 Summary

4.1 General implications

In FFNs, neutrality is a consequence of the numerous connections in a specific network that can be added or removed without directly affecting its functionality [6]. Therefore, it is a robust result that is insensitive to the parameters I, O, M or H, but depends on the existence of a threshold at each unit, as the consistent results we have observed suggest. The exception is the rank-frequency distribution, which already for the values of $I = 6$ and $O = 6$ turns out to be too large to sample with enough significance, and therefore, is different if sampled with the same density.

The parameters affect other aspects of the FFNs, such as their ability to compute a randomly chosen phenotype. As in the case of neural networks, the number of hidden layers (H here) and their size (M here), affects the complexity of the computations available to the system, or its capacity, and, in the case of FFNs, to the attainable phenotypes (which also depends on the number of links E). In this sense, many dynamics experiments failed when performed with a target phenotype chosen at random (and never with a phenotype computed from a random *genotype*). In general, we do not know what is the precise dependence between an increase in M or H and the diversity of phenotypes, but we expect to find an increasing coverage of phenotype space as H and M increase.

Despite the limitations implicit in our approximation, and as already discussed in [5], the presence of neutrality in the genotype-phenotype map has immediate consequences for an evolutionary process, and our results extend those of RNA and combinatorial molecules to a new domain and provide a good example of a more general applicability of these ideas. In the case of biology, living systems have evolved mechanisms of computation able to optimize their chances of survival. As a consequence, convergence towards networked structures able to integrate and process external inputs into reliable outputs has been widespread.

On the other hand, many attempts have been made at the evolutionary design of digital (and analog) circuits, using genetic representations. In this context, some authors have already pointed out the importance of neutrality in artificial circuit evolution and genetic programming. However, most of this work has been focused on the production of small electronic circuits and their application to other domains is not straightforward. The results presented here could also contribute to this field.

Although there is a consensus about the need for a so called complex systems engineering [4], success is still modest.

A very informative result in these lines is the remarkable difference in sensitivity among the genotypes in a neutral network, suggesting that these networks have finer structure inside. This opens up interesting new questions, and in particular it cleanly visualizes the already mentioned, and somewhat controversial issue of evolvability, about which precise models are scarce. If special FFNs have a broader spectrum of mutants, they could more readily access new functions, and hence adapt more rapidly. If this is translated in some way to the domain of circuit evolution, adaptation would be greately speeded up, maybe allowing us to design larger circuits. This sensitivity is in contrast with the natural tendency of populations to drift towards the more connected parts of a neutral network, which are occupied by the more robust phenotypes. It remains to be seen if such sensitivity is found in biological networks, how it could be maintained in the course of an artificial adaptation experiment, and, perhaps more importantly, what is the underlying structure that supports it.

Available theoretical developments and previous agreement between molecular evolution on neutral landscapes and predictions strongly indicate that evolution of complex structures and designs is likely to happen on neutral landscapes. The surprising similarities between the statistical properties exhibited by evolved computational graphs, such as feed-forward networks [6] and evolved RNA structures strongly supports the presence of a rather universal (and unexpected) structure of a disparate class of landscapes. Such universality is at the root of a theoretical basis for evolved structures.

The straightforward functional interpretations of given designed structures, from computational capacities to resistance to bug propagation are easily mapped into well- defined topological features that are suitable for theoretical analysis. If true, a general theory of evolution of designs and networks would be available to exploration, with all the desirable consequences for information networks. Future work in this context might include different types of fitness functions imposing restrictions on the network topology based on information measures [8].

4.2 Implications for robust peer-to-peer network design

Peer-to-Peer (P2P) networks consist of interconnected computer systems (nodes) working together to solve, collectively, computational tasks. The majority of such deployed networks are based on the concept of an “overlay topology”, in which each node maintains some logical links (such as an IP address on the internet) to other nodes, which may implement unstructured, structured or semi-structured topologies.

In general such networks are highly dynamic with nodes and links under constant change (or churn). This is particularly true in so-called open P2P systems where no central authority or control is exercised over the nodes at run-time.

In this class of P2P system a major task for the designer is to create a protocol (a set of rules) that govern how the nodes in the network interact, that has the property of robustness (see deliverable D5.1.1 for a discussion of this in the context of life-like properties). Proposed protocols for such systems are generally evaluated under various regimes of “churn” with those displaying high insensitivity, at the level of function, being considered superior to those which breakdown [2].

This kind of robustness to churn is generally achieved in two main ways: 1) by maintaining topologies and protocols with some degree of redundancy (for example by having multiple possible routes between nodes for message passing); 2) by implementing within the protocols themselves the ability to self-repair and rebuild desirable topologies from undesirable topologies.

The insights gained from the experiments discussed in the previous sections support a third (though inter-linked) approach: that networks can be constructed that are robust by the nature of their structural / functional relationship (i.e. their genotype / phenotype mapping in evolutionary language). This could be termed “structurally realised functional robustness” (SRFR).

Essentially, for a given collective computational function F (and in P2P this generally involves supplying services to each node within the network) there are generally many possible topologies possible that can implement F.

A P2P designer needs to construct a protocol in order to realise and maintain at least one of those topologies. In some sense the protocol constitutes a kind of genotype (G) which when executed on the nodes maps to a phenotype (P) which hopefully achieves F. A significant sub-property of G, in a P2P network, is the topology (T)[2].

For a given function F it will be advantageous to selected a T that achieves F with SRFR. This means, a T under change (mutation or churn) continues to perform F. What appears to be the case is that certain topologies have this property for certain functions over large contiguous regions of the T-space under certain kinds of mutation assumptions.

In the previous sections the mutation assumption was one of randomly rewiring by removing one link and adding another. In P2P work the change assumptions are generally the removal of links and nodes and the insertion of new nodes and links - because this is what occurs in running systems as nodes join and leave the network. This can be equated to a different kind of mutation operator applied to T.

Perhaps it could be possible to explore the T-space (in a similar way to that shown in figure 5) to find those topologies that have a high SRFR for a given F for some kinds of P2P domains or conversely to find those locations where SRFR is low (not shown in figure 5). Given this information, a P2P designer could choose a candidate T based on the principle of structural robustness in achieving F.

There are of course some problems with this, both practical and theoretical. Practically, P2P networks tend to be large - this is what makes them powerful - and traditional searching techniques would not be applicable. However, it may be possible in some instances to use various heuristics to cut-down a specific search space. Theoretically, even if we could locate particular topologies that have high SRFR this does not mean that an efficient protocol can be constructed that can implement those topologies in a P2P.

Coming back to the G-space (which codes the entire protocol) it is not theoretically impossible to consider exploring the the entire G-space for robust networks but this involves evolving the entire protocol. Even for small networks this is not practical (even with the current best Genetic Programming techniques [5]) due to the huge size of the G-space, its brittle nature and the large computational effort required to evaluate the fitness of a network (how close it comes to F).

Another interesting implication of the work is the idea of functional "portals" in the T-space. At such portal locations in the T-space, function can change dramatically via only a small change in T. A network in such a topology is in a highly unstable state - only requiring small changes to produce large functional transitions. In the context of SRFR these are to be avoided. However, such portals could be very important, essential even, in order to transition a large open network from performing one computational function F to another function F' without the need to reprogram or control many nodes. Although how this could predictably be done and what F transitions are possible for given protocols is an open question, we do have evidence of such kinds of transition points (see figure 7) related to sudden network wide functional changes for protocols supporting cooperation between nodes [2, 3]. However, the relationship between topology and function in these examples is complex and dynamic.

It could be argued that P2P designers already do a kind of "eyeball and hacking" hill-climbing

[2] In some sense it might be more realistic to consider the topology as a sub-property of the phenotype (P). Then the mapping between G and T involves the P2P programmer designing code that when executed produces T (often utilising some form of self-organisation in the process). Here we do not consider that mapping other than to note that many possible G would give identical T but that often (since G is handcrafted computer code) G is highly brittle, meaning that a small change in G leads to radically different P and failure to achieve F. One attempt to obviate this is to consider genetic programming (GP) trees [5] rather than traditional software languages. We briefly discuss this later.

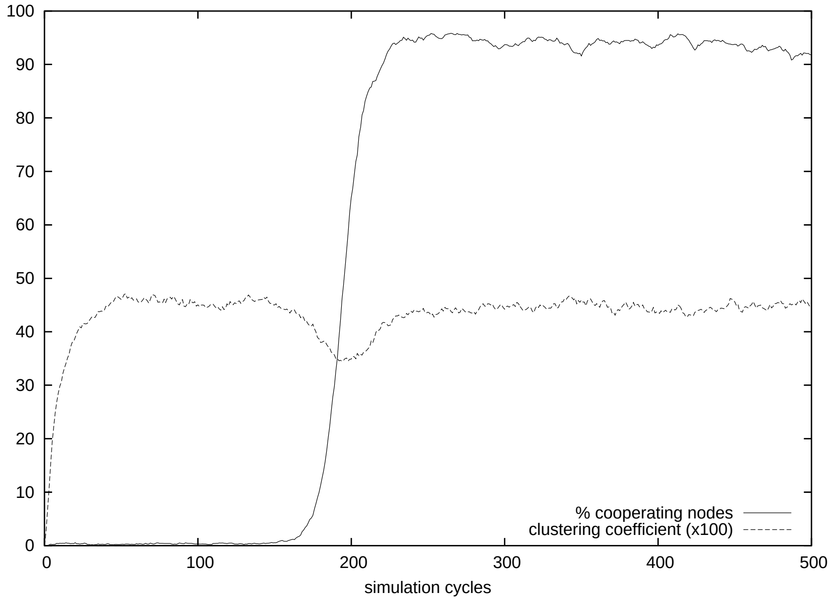

Figure 7: Typical single simulation run of the SLACER P2P protocol. SLACER supports high levels of cooperation between nodes [2, 3]. Notice the clustering coefficient dips during the same period that high cooperation breaks out. This indicates a different kind of topology that emerges during rapid transition to high cooperation. Here the network is initialised as a random topology - hence very low clustering coefficient initially.

in the G-space through repeated cycles of simulation and re-coding. By subjecting their candidate protocols to churn experiments and modifying them to get better performance they are partially searching the T-space for robust alternatives (see figure 8 for an example). Additionally, where certain topologies are theoretically understood to deliver robustness they are used - for example, random networks give robust communication infrastructures [4]. It is certainly the case that for many proposed and deployed P2P systems a small change in the links or node population would have little or no effect on the the functional performance.

However the potential for some kind of generic understanding of topology robustness for given functions and mutation operators potentially allows for a more principled design approach for the construction of P2P protocols.

References

[1] Fernandez, P. and V. Solé, R. (2005) From wiring to function and back: a case study in feed-forward networks. *Santa Fe Inst. Working Paper.*

[2] Hales, D.; Arteconi, S.; Babaoglu, O. (2005) SLACER: randomness to cooperation in peer-to-peer networks. In *Proceedings of the 1st International Conference on Collaborative Computing: Networking, Applications and Worksharing*, Workshop on Stochasticity in Distributed Systems (STODIS'05), IEEE Computer Society Press. [DELIS-TR-0119]

[3] Hales, D. and Arteconi, S. (2005) Friends for Free: Self-Organizing Artificial Social Networks for Trust and Cooperation. Submitted to IEEE Intelligent Systems Special Issue on Self-management through self-organization in information systems. Available: http://arxiv.org/abs/cs.MA/0509037. [DELIS-TR-0196]

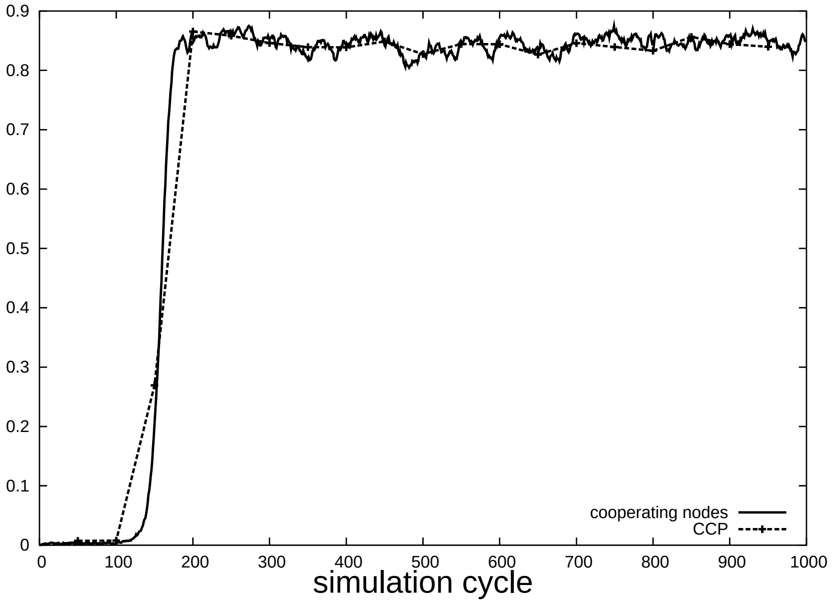

Figure 8: Typical single simulation run of the SLACER P2P protocol under churn. Each simulation cycle 20 nodes and their links are removed and new new nodes randomly added. The performance. in terms of cooperation and a measure of how many pairs of nodes are connected by cooperative paths (CCP), is not greatly reduced. For more details see [3].

[4] Jelasity, M., Montresor, A. and Babaoglu, O. (2005) Gossip-based aggregation in large dynamic networks. *ACM Trans. Comput. Syst., 23(1):219-252*

[5] Langdon, William B. and Poli, Riccardo. 2002. *Foundations of Genetic Programming.* Springer-Verlag.

[6] Lobo, J., Miller, J. H. and Fontana, W. (2004) Neutrality in technological landscapes. *Santa Fe Inst. Working Paper*

[7] Ottino, J. M. (2004) Engineering complex systems. *Nature 427, 399*

[8] Schuster, P. (1996) How does complexity arise in evolution? *Complexity 2, pp. 22-30.*

[9] Schuster, P. (1997) Landscapes and molecular evolution. *Physica D, 107, 351-365*

[10] Schuster, P., Fontana, W., Stadler, P. and Hofacker, I. (1994) From sequences to shapes and back: A case study in RNA secondary structures. *Proc. Roy. Soc. B255, 279-284.*

[11] Solé, R., Valverde, S., Montoya, J. and Ferrer-Cancho, R. (to appear) *Evolving Webs.* Princeton U. Press.

[12] Solé, R. and Valverde, V. (2004) Information Theory of Complex Networks: On Evolution and Architectural Constraints. In *Networks: Structure, Dynamics and Function*, Lecture Notes in Physics, Springer-Verlag,