Neutral search spaces for artificial evolution: a lesson from life

Rob Shipman[1], Mark Shackleton[1], Marc Ebner[2], Richard Watson[3]

[1] BT Labs at Adastral Park, Admin 2-5, Martlesham Heath, Ipswich, IP5 3RE, UK. rob.shipman@bt.com, mark.shackleton@bt.com

[2] Universität Würzburg, Lehrstuhl für Informatik II, Programmiersprachen und Programmiermethodik, Am Hubland, 97074 Würzburg, Germany. ebner@informatik.uni-wuerzburg.de

[3] Brandeis University, Volen Center for Complex Systems, Mail Stop 18, Waltham MA 02454-9110, USA. richardw@cs.brandeis.edu

Abstract

Natural evolutionary systems exhibit a complex mapping from genotype to phenotype. One property of these mappings is neutrality, where many mutations do not have an appreciable effect on the phenotype. In this case the mapping from genotype to phenotype contains redundancy such that a phenotype is represented by many genotypes. Studies of RNA and protein molecules, the fundamental building blocks of life, reveal that this can result in neutral networks - sets of genotypes connected by single point mutations that map into the same phenotype. This allows genetic changes to be made while maintaining the current phenotype and thus may reduce the chance of becoming trapped in sub-optimal regions of genotype space. In this paper we present several redundant mappings and explore their properties by performing random walks on the neutral networks in their genotype spaces. We investigate whether the properties found in nature's search space can be engineered into our artificial evolutionary systems. A mapping based on a random boolean network was found to give particularly promising results.

1. Introduction

Natural evolution differs in many respects from the evolutionary algorithms typically employed today. One such difference is highlighted by the neutral theory of evolution. According to this theory a considerable fraction of all mutations are neutral and only a minute fraction of the remainder are actually beneficial (Kimura 1994). This results in a redundant genotype-phenotype mapping with typically many genotypes representing any given phenotype. The redundancy manifests itself in a number of different ways and at a number of different levels - from the genetic code, consisting of 64 codons mapping into only 20 amino acids, to the complex interplay of molecules forming an organism. A particularly important and thorough study of the effects of such redundancy was performed in the context of the folding of RNA, and to a lesser extent, protein molecules (Huynen 1996, Huynen, Stadler, and Fontana 1996). These studies revealed a number of interesting properties in the nature of the secondary structures that the primary structures folded into. There were a number of common secondary structures each represented by a very large set of primary sequences i.e. there was large-scale redundancy in the genotype-phenotype mapping. The density was such that these sets were often connected by single-point mutations forming so-called *neutral networks.* Thus, it was possible to traverse the set of genotypes through the simplest of mutations without changing the represented phenotype.

This brought about the possibility of neutral drift allowing larger areas of genotype space to be explored in search of more adaptive secondary structures. However, during such a process there is no pressure influencing movement to areas of genotype space in which more adaptive phenotypes can be found and there is thus a danger of prolonged periods of random drift. It is important, therefore, that the neutral networks representing each of the phenotypes are intertwined with many access points between them. This encourages beneficial transitions from one network to another and minimizes the amount of time spent drifting randomly. Studies of RNA folding suggested that this was the case. As the neutral networks were traversed a relatively high, and roughly constant, number of new structures were discovered at each step (Huynen 1996).

These properties may have a significant impact on the evolvability of a system. Instead of becoming trapped in sub-optimal regions of genotype space, adaptation is able to continue through genetic changes that do not alter the phenotype but enable movement in genotype space to areas that are closer to genotypes representing potentially more adaptive phenotypes. Following on from previous studies (Ebner 1999, Shipman 1999) this work explores a number of redundant genotype-phenotype mappings with a view to ascertaining whether these fundamental properties of living systems can be encouraged in our artificial systems. For related work that aims to exploit the developmental process in engineered systems see (Hoile and Tateson 2000).

The structure of this paper is as follows - section 2 details the four mappings that were studied, section 3 details the methods that were used to ascertain the properties of these mappings, section 4 presents our findings, section 5 gives a discussion of these results and section 6 concludes.

2. Redundant mappings

A number of different mappings were constructed for this work, this paper reports on four of them. In order to allow the calculation of statistics, the phenotype space was fixed at 8 bits, giving $2^8 = 256$ possible phenotypes. Both the length of the genotype and the number of alleles varied between the mappings.

2.1 Voting mapping

The first mapping, shown in figure 1, is based on a voting approach where each bit of the phenotype is influenced by several bits from the genotype. Each phenotype bit is determined by looking at all the bits of the genotype to which it is linked. A bit of the phenotype is set to one if the majority of connected bits in the genotype ”vote” in favor of this. Thus, depending on the values of the other relevant bits, a point mutation may or may not have an effect on the phenotype. It is important to note that the set of genotype bits linked to a particular phenotype bit will typically *overlap* with the sets corresponding to other phenotype bits. It is this aspect that permits multiple phenotype bits to potentially be changed simultaneously by a single point mutation. Together with the redundant “majority voting” aspect of the mapping, this permits the scene to be set for future transitions to another phenotype, without actually changing the current phenotype encoded in the genotype. The links between the genotype bits and the phenotype bits are determined in the following way. For each bit of the phenotype we select a number of bits of the genotype which will vote for that phenotype bit, which is typically a constant odd number. For each of the voting bits, we randomly choose whether a set bit will vote in favor of the corresponding phenotype bit being set, or against it being set. Thus each gives either a positive or a negative vote. For instance, in the results reported later, a genotype of 24 bits was used with sets of 17 genotype bits being chosen for each of the 8-phenotype bits. There is thus significant overlap among the sets of voting bits.

In order to confirm that the above mapping was introducing “useful” redundancy, we compared it with a trivial voting mapping in which each phenotype bit was linked to 3 genotype bits *without overlap*. This mapping exhibits redundancy, but in other respects behaves exactly like a direct encoding where exactly one genotype represents each phenotype. The redundancy in this case was not useful.

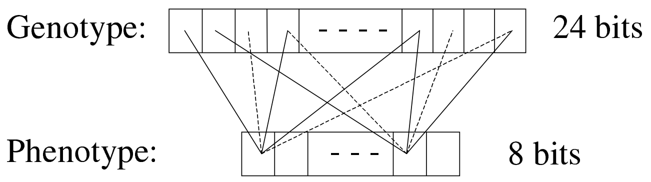


Figure 1. Illustration of the voting map; each of the phenotype bits receives input from an odd number of genotype bits. These bits vote either positively (solid lines) or negatively (dashed lines) for the corresponding genotype bit to be turned on. A positive sum results in a 1 for the genotype bit and a negative sum a 0.

2.2 Cursor based mapping

The second mapping uses a developmental approach where the phenotype is interpreted as a linear sequence of commands similar to the genetic programming paradigm (Koza, 1992). The commands control a write-head, which moves over a write area as shown in figure 2.

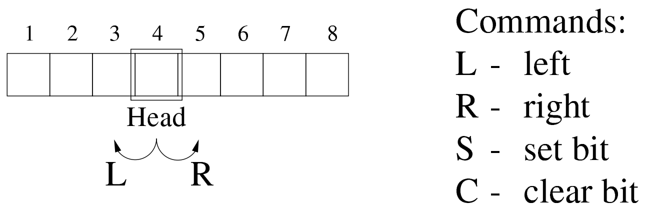


Figure 2. Illustration of the cursor based mapping, the write head can move over the write area and set or clear bits. Four commands are used: L - moves the write head one bit to the left, R – moves the write head one bit to the right, S – sets the bit under the write head, C – clears the bit under the write head. If the write head moves out of the write area it reappears on the other side.

Four different commands are used, two to control the movement of the head and two to set and clear bits on the 8-bit write area. The genotype bits thus consist of 4-alleles to represent each of these commands. All bits on the write area are initially cleared and then all commands represented in the genotype are executed sequentially. The resulting bit string is interpreted as one of the 256 phenotypes. This process is illustrated in figure 3 for a genotype length of 11. This work used a genotype of 32 commands.

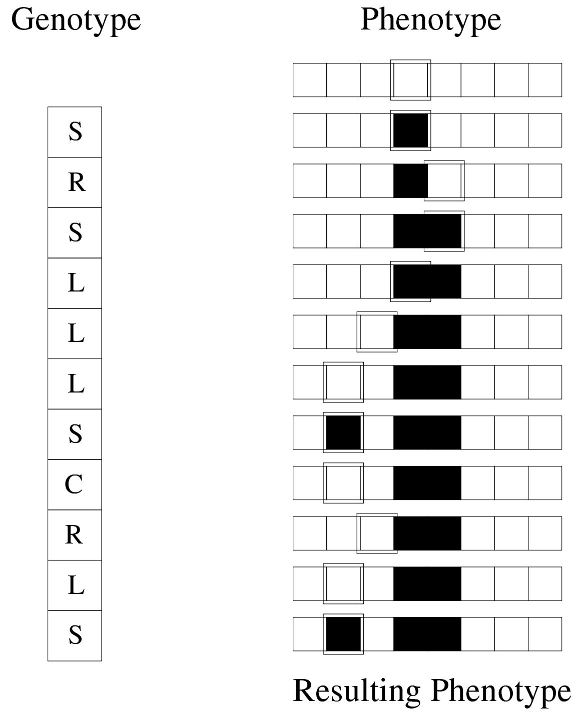


Figure 3. Development of the phenotype using a cursor based mapping. The genotype is interpreted as a sequence of commands, which either move the write head or set or clear bits on the write area. Initially all bits of the write area are cleared, the binary string following execution of all commands represents one of the 256 phenotypes.

2.3 Cellular automaton mapping

The third mapping also uses a developmental approach in which the genotype specifies a cellular automaton (Wolfram 1984) that is used to form the phenotype. In this work a non-uniform one-dimensional cellular automaton was used (Sipper 1997), which consists of a linear array of cells and a rule table for each of the cells specifying how the state changes over time. The state of the cell and its two immediate neighbors are used to form an index into the rule table for that bit, which specifies whether the next state should be 1 or 0. Thus, 8 entries per phenotype bit are required to fully specify the behavior of the automaton. The initial state is also encoded in the genotype resulting in a total of 72 bits for the 8-bit phenotype used in this work.

To perform the mapping from genotype to phenotype the state of the cellular automaton was initialized with that specified in the genotype. The automaton was then run for a fixed number of time steps, 20 in this work, and the resultant binary string interpreted as one of the 256 phenotypes. This is illustrated in figure 4 for an example cellular automaton and 16 updates.

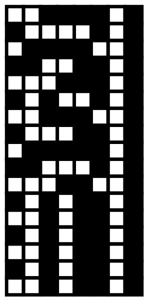


Figure 4. Development of a cellular automaton with the following rule table: 11011100, 00011001, 11100100, 11101010, 01110110, 01001110, 10101001, 10110100. The starting state (encoded in the genotype) is 00111101. The development is shown for 16 time steps after which the cellular automaton has settled into a periodic cycle. A snapshot of the automaton at a fixed time step is interpreted as one of the 256 phenotypes.

2.4 Random boolean network mapping

The fourth mapping is a generalization of the cellular automaton mapping. In this case the wiring of the automaton is also specified in the genotype together with the rule tables and initial state. Thus, the neighborhood of each cell is no longer fixed as the cell itself and its two immediate neighbors but can be any of the phenotype bits as specified by the genotype. This is illustrated in figure 5. More information on random boolean networks can be found in (Kauffman 1993).

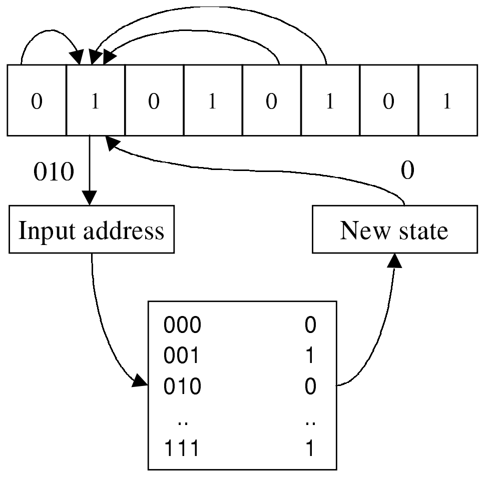


Figure 5. Illustration of the RBN mapping; the genotype specifies the rule table for each cell, the initial state of each cell and the inputs to each cell. For the example bit, the inputs specify that the third rule is used to give its next state.

In this work the number of inputs per cell was fixed at 3 and thus 9 bits were required to specify the inputs for each cell. This resulted in a total of 72 bits to specify the wiring of the 8-bit boolean network, which together with the 72 bits to specify the rule tables and initial state gave a genotype of length 144. To perform the mapping from genotype to phenotype the network was constructed and initialized using the information encoded in the genotype. As for the cellular automaton, it was then run for 20 time steps and the resulting binary string interpreted as one of the 256 phenotypes.

3. Evaluation of the mappings

The main tool used to evaluate the properties of the mappings described above was a random neutral walk (Huynen 1996). This procedure allowed a measure to be made as to the benefit of neutral drift through assessing the number of new phenotypes encountered. The walk began by choosing a genotype mapping into a given phenotype at random. All one-point mutants of this genotype were assessed and the number of new phenotypes encountered was logged. These are termed innovations. In addition a list of neutral neighbors, i.e. one-point mutants mapping into the same phenotype, was formed. One of these neutral neighbors was chosen at random and the procedure was repeated for a given number of steps, 100 steps were used in this work. If no neutral neighbors are found, the walk remains in the same position and no further innovation is possible. This process is illustrated in figure 6. A number of statistics were calculated using this procedure (see Ebner et al. 2000), the following two are reported on in this paper:

3.1 Total number of innovations

For each of the 256 phenotypes, a genotype mapping into that phenotype was chosen at random. A random neutral walk consisting of 100 steps was performed for each of these genotypes and the number of new phenotypes encountered at each step was logged. This procedure was repeated for four independent walks for each of the phenotypes. This resulted in a total of 1024 walks and the averaged cumulative number of innovations was plotted at each step of the walk for each mapping.

3.2 Phenotypic accessibility

For the same set of 1024 walks described in the previous section an accessibility plot was formed that showed which phenotypes were encountered on the neutral walks for all 256 phenotypes. This resulted in a 256 by 256 plot with one axis showing the phenotype that neutral walks were being performed for and the other axis the phenotypes encountered on those walks. This plot gave some impression of the connectedness of phenotype space via the neutral pathways in genotype space.

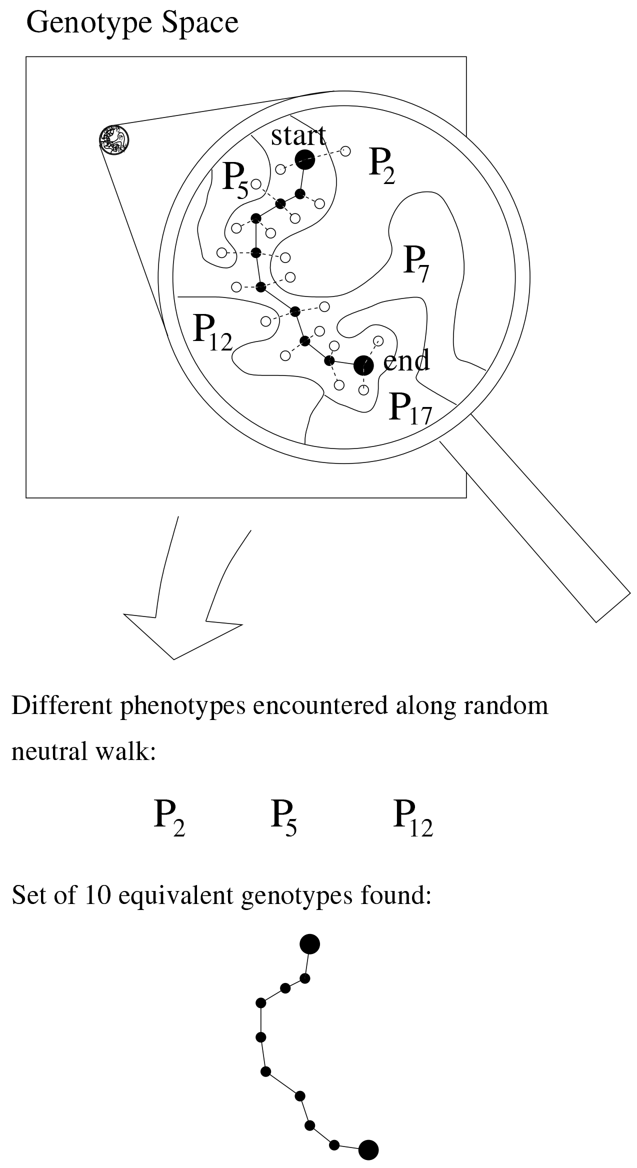
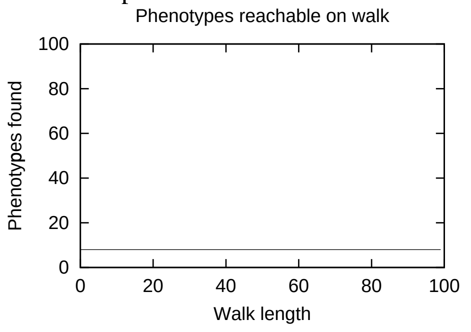

Genotype Space
start
end
P_2
P_5
P_7
P_{12}
P_{17}
Different phenotypes encountered along random neutral walk:
P_2 P_5 P_{12}
Set of 10 equivalent genotypes found:

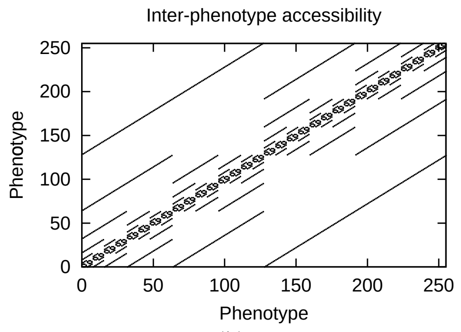
Figure 6. A random neutral walk in genotype space. Starting from a randomly chosen genotype, all innovations i.e. previously unseen phenotypes, are logged. A list of neutral neighbors, i.e. neighboring genotypes that map into the same phenotype, is also formed. One of these neutral neighbors is then chosen at random and the procedure repeated for a number of steps.

4. Results

This section shows the statistics that were calculated for each of the redundant mappings. In addition, the statistics for a direct encoding with no redundancy are included in order to allow comparison.



(a)



(b)

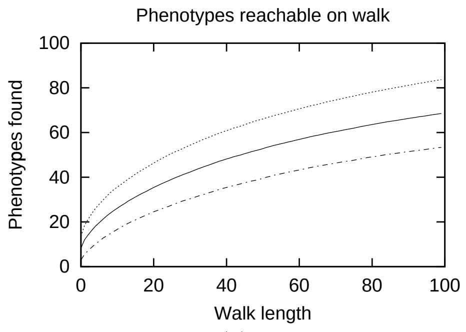
Figure 7. Results for a direct binary encoding. (a) The number of phenotypes found remains constant at 8, reflecting all possible one-point mutations. There are no neutral neighbors and thus there can be no innovation through neutral drift. (b) The accessibility is very sparse reflecting the fact that only 8 new phenotypes are accessible from each phenotype.

4.1 Direct binary encoding

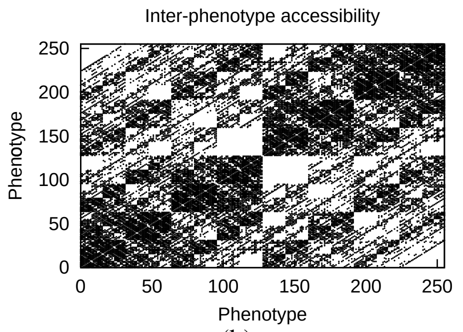
Figure 7 shows results for a direct encoding in which an 8-bit genotype directly specified the phenotype. In effect there was no genotype-phenotype mapping as is commonly the case in artificial evolution.

4.2 Voting mapping

The results in figure 8 show that neutral walks for a voting mapping allow access to a greater number of phenotypes than for a direct encoding. The number of innovations continues to rise after 100 steps indicating an increased probability of finding more adaptive phenotypes through neutral drift. The accessibility plot is also less sparse indicating the discovery of more phenotypes on the neutral walks. Statistics were also gathered for a trivial voting map as discussed in section 2.1. As expected, these results were identical to those for a direct encoding.



(a)



(b)

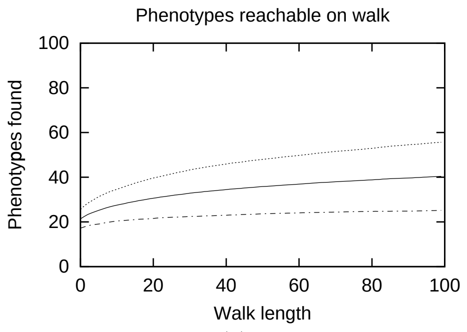
Figure 8. Results for the voting mapping. (a) The number of phenotypes found continues to increase throughout the walk indicating continual innovation. The number eventually reaches a total of 68. The dashed lines indicate +/- one standard deviation. (b) The accessibility plot is denser than for the direct encoding although some transitions were not possible.

4.3 Cursor based mapping

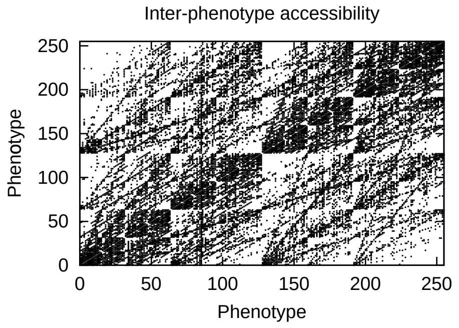
Figure 9 again shows that neutral walks for a cursor based mapping allow the discovery of more new phenotypes than a direct encoding. However, the number of phenotypes encountered at the end of the walk was smaller than that for the voting mapping. The curve is beginning to level off after 100 steps and thus the number of phenotypes found is not likely to increase much further even with longer walk lengths. The accessibility plot is again denser than that for a direct encoding however a number of phenotype transitions were not found on the neutral walks.

4.4 Cellular automata mapping

Figure 10 shows that the number of phenotypes found is greater for the CA than for the previous mappings. The curve is still relatively steep after 100 steps indicating remaining potential for discovering further phenotypes with increased walk length. The accessibility plot is also much denser indicating that many more transitions between phenotypes are possible for the CA mapping than the previous mappings.

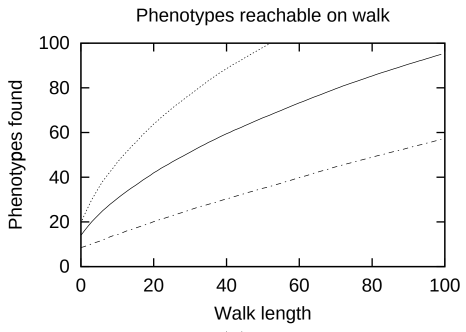


(a)

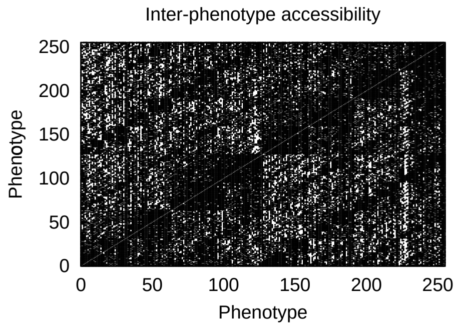


(b)

Figure 9. Results for the cursor based mapping. (a) The number of phenotypes found is an improvement over the direct encoding but not as high as for the voting mapping. The final number reached is 40. The dashed lines indicate +/- one standard deviation. (b) The accessibility plot is denser than a direct encoding but some transitions were not found on the neutral walks.



(a)



(b)

Figure 10. Results for the CA mapping. (a) 95 phenotypes were found after 100 steps on the neutral walk. The curve is still relatively steep indicating continuing innovation. The dashed lines indicate +/- one standard deviation. (b) The accessibility plot is much denser than for the previous mappings indicating greater accessibility between the neutral networks associated with each of the phenotypes.

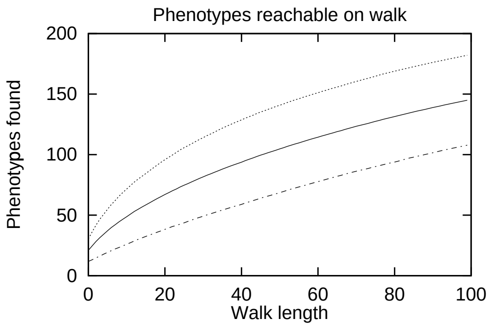
4.5 Random boolean network mapping

Figure 11 shows that the ability to modify the wiring of an automaton increases the number of innovations on a neutral walk. The number of phenotypes found after 100 steps is approximately 50% greater than for the cellular automaton mapping and the steep curve indicates the potential for further innovations. The density of the accessibility plot is also greater indicating even greater accessibility between the phenotypes.

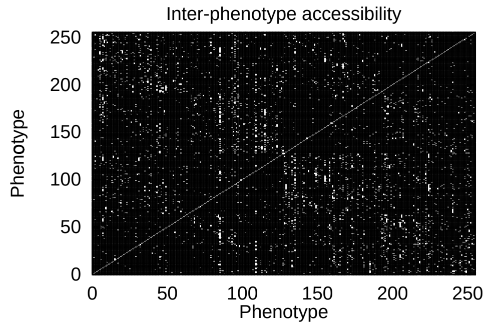
5. Discussion

It is common practice in artificial evolution to use a direct one-to-one mapping between genotype and phenotype. An example of such a mapping is the direct binary encoding, the results for which were presented in section 4.1. In such a scenario the number of differing phenotypes accessible from any given genotype is restricted to the length of the genotype i.e. all one-point mutants. In many situations it may be common for none of these phenotypes to be better adapted than the current one and thus adaptation will effectively halt at a local optimum.

This is a very different scenario to the seemingly open-ended innovation found in natural systems. The mappings explored in this work show that continuing innovation can be achieved with a suitable mapping between genotype and phenotype. Introducing the same kind of redundancy found in natural evolutionary systems into our artificial systems may increase the efficacy of artificial evolution through reducing the possibility of entrapment at local optima. However, the type of redundancy is crucial. This is highlighted by the voting mapping; the trivial voting mapping can be made to contain very high degrees of redundancy by increasing the number of genotype bits that are used to vote for a given phenotype bit. However, this redundancy will be of no benefit, regardless of how much is introduced. The redundancy is only beneficial if it increases the accessibility between phenotypes, i.e. if it allows more new phenotypes to be discovered than would be the case for a direct encoding. This is not the case for the trivial voting mapping as all mutations effect only one genotype bit and thus neutral mutations cannot enable moves closer to new phenotypes while maintaining the current phenotype. In order to allow for such a scenario a genotype bit must be able to effect more than one phenotype bit, which is the case when the sets of genotype bits voting for each of the phenotype bits are allowed to overlap. The results presented in section 4.2 show that this enables the discovery of many more phenotypes than is the case for the direct encoding.



(a)



(b)

Figure 11. Results for the RBN mapping. (a) The number of phenotypes found is higher than all the other mappings. A total of 145 different phenotypes were found after 100 steps along the neutral walk. The curve continues to rise indicating continuing innovation. (b) The accessibility plot is the densest of those presented in this paper with most transitions between phenotypes found.

In order to introduce beneficial redundancy, therefore, it is important not only to create a mapping in which there are large connected sets of genotypes, representing each phenotype of interest but also to create a mapping in which the boundaries between the sets are intertwined. In the case of trivial redundancy the sets only come within a point mutation of each other in one position, as is the case for a direct encoding. However, the addition of overlap both increases the number of boundary points and introduces new boundaries with different phenotypes. The cursor-based mapping has a similar effect but on the evidence presented here, is less effective at increasing and expanding the boundary points of the sets. This is highlighted by the smaller number of phenotypes encountered during the neutral walk.

The cellular automaton mapping and, in particular, the random boolean network mapping are especially effective at introducing the right kind of redundancy. The accessibility plots are quite dense indicating that the sets of genotypes representing each of the phenotypes have become very intertwined with a high number of boundary points between them. The number of innovations indicates that boundary points with other sets continue to be found throughout the walk. These properties could considerably aid an artificial evolutionary system.

The success of a neutral mapping is dependent on the balance between structure and randomness. In order to increase the possibility of discovering a given phenotype, it would be desirable for many genotypes mapping into that phenotype to be randomly scattered throughout genotype space. Thus, from any point in that space it is likely that a required genotype will be in relatively close proximity. However this scattering cannot be entirely random, as it is important to maintain a relatively high number of neutral neighbors in order to encourage the formation of connected neutral networks and allow substantial neutral drift. The good results of the RBN mapping reflect a good balance between these two aspects. A large number of different phenotypes were discovered on neutral walks whilst an average of approximately 50% neutral neighbors was maintained.

6. Conclusion

This work has explored the properties of four redundant genotype-phenotype mappings that were constructed in an attempt to mimic the desirable properties found in nature's own redundant search space, evidenced by the work on RNA folding (Huynen, Stadler, and Fontana 1996) for example. In all four cases the redundancy was found to be beneficial, in that movement on the resulting neutral networks allowed for the discovery of a larger number of phenotypes than would be the case for a direct encoding. Thus, the probability of entrapment at local optima when using these mappings would be reduced. One mapping, based on a random boolean network, was found to have particularly good properties and may be of real benefit in an artificial evolutionary system.

The results presented in this work used only a small number of phenotypes and a relatively small sample of the genotype space. With sizeable genotypes exhaustive enumeration of the space is impossible and some form of sampling is required. An intention of future work is to explore other statistics that can help to further reveal the properties of the spaces created by these and other mappings. The performance of the mappings on larger phenotype spaces and in the context of an adaptive fitness walk is also being explored (Shackleton et al. 2000).

Acknowledgments

Part of this work was supported by the Santa Fe Institute, Santa Fe, NM. Thanks go to Susan Ptak from Stanford University, Department of Biological Sciences for helpful discussions and comments. The authors would also like to thank their respective institutions for their continuing support.

References

Ebner, M. 1999. On the search space of genetic programming and its relation to nature's search space. In Proceedings of 1999 Congress on Evolutionary Computation, Volume 2, 1357-1361. IEEE Press.

Ebner, M., Shackleton, M., Shipman, R., and Watson, R. 2000. How redundant mappings influence the ability to adapt to a changing environment. Under submission.

Hoile, C., and Tateson, R. 2000. Design by Morphogenesis. To appear in Proceedings of the Seventh International Conference on Artificial Life. MIT Press.

Huynen, M. A. 1996. Exploring phenotype space through neutral evolution. *Journal of Molecular Evolution,* 43:165-169.

Huynen, M. A., Stadler, P. F., and Fontana, W. 1996. Smoothness within ruggedness: The role of neutrality in adaptation. *Proc. Natl. Acad. Sci. USA,* 93:397-401.

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

Kimura, M. 1994. Population Genetics, Molecular Evolution, and the Neutral Theory: Selected Papers. The University of Chicago Press, Chicago.

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

Shackleton, M., Shipman, R., Ebner, M. 2000. An investigation of redundant genotype-phenotype mappings and their role in evolutionary search. To appear in Proceedings of the 2000 Congress on Evolutionary Computation.

Shipman, R. 1999. Genetic Redundancy: Desirable or Problematic for Evolutionary Adaptation? In Proceedings of the 4th International Conference on Artificial Neural Networks and Genetic Algorithms, 337-344, New York. Springer-Verlag.

Sipper, M. 1997. *Evolution of Parallel Cellular Machines: The Cellular Programming Approach.* Springer-Verlag, Berlin.

Wolfram, S. 1984. Cellular Automata as models of complexity. *Nature,* 311(4):419-424.