Issues in Designing a Neutral Genotype-Phenotype Mapping

Rob Shipman

Intelligent Systems Laboratory, BTexact Technologies

Abstract: This paper discusses issues that arise when a neutral genotype-phenotype mapping is used in the context of a real-world evolutionary computation application. Previous studies have suggested that a neutral encoding may alleviate the problem of local optima by allowing drift at the same fitness level. We show that neutrality can indeed produce a search space free of local optima but that it must be carefully introduced with due consideration to details of the application being considered and its associated search space. Although the details of the neutral encoding are specific to the application, we seek to deduce some heuristics that are likely to prove useful for designing genetic encodings for other problems to facilitate search for fitter phenotypes.

1 Introduction

The impact of neutrality in the mapping from genotype to phenotype is attracting an increasing amount of interest in the artificial evolutionary community [1]-[5]. In these mappings a number of different genotypes map onto the same phenotype such that there are typically many examples of a given phenotype distributed throughout genotype space. Under certain conditions the set of genotypes mapping onto a given phenotype are connected by single actions of a genetic operator (such as mutation) and thus can form large neutral networks that can percolate throughout genotype space. A population can drift along these networks until it encounters areas in genotype space that allow further increases in fitness and may thus avoid becoming trapped in sub-optimal regions of genotype space. Through alleviating the problem of local optima neutrality in the genotype-phenotype mapping may thus be beneficial to the evolutionary process. However, if used incorrectly it may also be detrimental. How and when to incorporate neutrality in an artificial genotype-phenotype mapping is an area of ongoing research and an aim of this paper is to contribute to this research. In particular we address these issues in the context of a real-world engineering search problem, while previous studies have typically considered abstract search spaces.

Interest in neutral genotype-phenotype mappings was fuelled by important work in molecular evolution. The folding of RNA molecules was studied as an example of a genotype-phenotype mapping. It was shown that this mapping contains high degrees of neutrality that contributes significantly to the molecule's evolvability [6]. This work encouraged artificial evolution practitioners to explore the effect of neutrality in abstract fitness landscapes [2] and existing artificial genotype-phenotype mappings [3]-[5]. In some of this work it was suggested that the neutrality was beneficial [3] and in others to be of little or no use [5]. This raises an important issue; neutrality alone is not sufficient for improving the efficacy of the evolutionary process. The *right type* of neutrality must be introduced into the genotype-phenotype mapping for it to be of benefit. Exactly how to do this is likely to be specific to a particular application – the use of domain knowledge, for instance, will remain important. However, the development of more general heuristics to aid in the design of beneficial neutrality in genotype-phenotype mappings would be a significant step forward.

In this paper we explore some of the issues that arose during the construction of a neutral genotype-phenotype mapping for a planning tool to aid in the design and growth of telecommunication networks. Although specific to this application the lessons learned are likely to be of interest to others constructing such mappings for different applications and are a step towards the development of a useful set of heuristics.

2. The NetGrow planning tool

The NetGrow planning tool aims to aid in the initial design and subsequent growth of telecommunication networks. Network designers often use a set of planning rules during this process that take the form of a series of IF-THEN rules. It is the manual application of these rules in a given context that produces the network design. The NetGrow tool recognises this fact and complements the work of the designers by encoding and optimising a set of planning rules rather than a final network design. Thus the output of the tool is an optimised set of planning rules that can easily be understood by human designers, can be used in conjunction with other planning tools and may be of more general use than a single network design for a given situation. The tool is described in more detail in [7].

Here we consider an application of the tool to a simplified version of a real network design problem - the growth of the Internet data network in the UK. The initial network consists of 3 high-capacity core nodes that currently handle all data traffic. Demand originating at exchanges that do not have a co-located core node is routed via the standard phone network (or PSTN) to an entry point into the data network. In order to accommodate increasing demand and reduce the load on the PSTN, lower capacity access nodes are to be added at a subset of 20 potential sites in order to draw off data traffic at an earlier stage. These access nodes connect directly to the nearest core node. Thus there are several costs that are incurred by a given network design, hardware costs resulting from the addition of access nodes and their connection to the nearest core node, the cost of routing calls via the PSTN and an added cost that is incurred for any demand that is not satisfied. A good network design will therefore strike a balance between these costs to minimise the overall cost of the network. Full details of the cost function and other aspects can be found in [7].

In order to determine which of the sites will house access nodes a simple planning rule is encoded that takes into account the traffic demand at a site and its distance away from an entry point into the data network, the following values are calculated for each site i:

$$\theta\_\mathrm{demand}_i = a * (b - \mathrm{demand}_i) \qquad \theta\_\mathrm{distance}_i = c * (d - \mathrm{distance}_i)$$

Where b and d are threshold parameters, a and c are scaling parameters, demand_i is the demand at site i and $\mathrm{distance}_i$ is the distance that site i is away from the nearest existing node. If both these values are above zero then the site becomes a candidate to house an access node:

IF ($\theta\_\mathrm{demand}_i > 0$) **AND** ($\theta\_\mathrm{distance}_i > 0$) **THEN** add site *i* to candidate list

A “firing strength” is then calculated for each site in the candidate list: $\mathrm{FiringStrength}_i = \theta\_\mathrm{demand}_i + \theta\_\mathrm{distance}_{i.}$ The site with the highest firing strength is chosen to house an access node. The candidate list is then recalculated and the process repeated until the candidate list is empty i.e. no more sites satisfy the planning rule conditions. This iterative process is necessary as the action of a rule affects the calculated conditions i.e. adding a node affects a site’s distance to the nearest existing node and thus the order in which nodes are added is important. The scaling parameters a and c enable the relative importance of the two conditions to be controlled i.e. if a was much higher than c then sites with high demand would fire more strongly than those a long way from the existing network and vice versa.

3. Planning Rule Encoding

Each of the 4 parameters a, b, c and d was represented by a 5-bit value and was thus quantised into 32 levels between a defined minimum and maximum value. The resulting genotype was 20 bits long giving 2^{20}=1048576 distinct genotypes. This is a manageable number allowing exhaustive enumeration of the space and thus definitive statements to be made about its structure. As there are also 20 potential sites there are also 1048576 possible configurations of access nodes i.e. network designs. In order to determine the extent of the neutrality in this mapping, the phenotype resulting from application of each possible genotype was determined. Out of the 1048576 possible phenotypes only 52 were generated by all possible instances of the encoded planning rule and there was thus large-scale neutrality in the mapping. On average approximately 20165 genotypes mapped onto each phenotype.

4. Fragmented Neutral Networks

The above encoding resulted in 52 sets of genotypes that map to a specific phenotype i.e. there are 52 neutral sets. However, in order to allow a population to explore these neutral sets in search of areas of genotype space that will allow further increases in fitness (when no immediate increase in fitness is possible) these sets must be connected by application of the genetic operator i.e. probable changes to the genotype must allow movement through genotype space without affecting the fitness value. It is thus important to determine the connectedness of the elements in these neutral sets. Here we restrict our analysis to the simplest modification to the genotype - single point mutation. Thus, the first genotype in the neutral set is moved into a subset. For each of the remaining genotypes it is determined whether they are a single mutation away from any of the genotypes in the subset. If so, they are moved to the subset. This process is repeated until there are no new additions to the subset.

This analysis reveals that many of the neutral sets are not fully connected but are separated into a number of smaller sub-networks. In this case a total of 86 neutral networks were present. This may be detrimental as it reduces the ability of the population to explore genotype space through neutral drift. A population may be restricted to one of the sub-networks. In order to assess whether this was potentially problematic, the accessibility between the neutral networks was assessed. All one-point mutants of each genotype on a neutral network were generated and the resulting phenotypes recorded. Should any of these one-point mutants give access to another network then the two networks were defined as accessible from each other. During evolution the main concern is to have access to networks that allow an increase in fitness. If all the networks allow access to networks of higher fitness (apart from the global optimum) then there are no local optima in genotype space. Thus, the number of networks of higher fitness that were accessible from each of the 86 sub-networks was recorded.

This analysis revealed that 3 of the networks did not allow access to higher fitness networks via single-point mutations. One of these networks corresponded to the global optimum; however, the other two networks correspond to local optima. Both of these networks contained 3069 genotypes and were one sub-network of 2 distinct sub-network sets. That is, both represented phenotypes that were represented by two neutral sub-networks that were inaccessible from each other. In both cases the other sub-network did allow access to higher fitness networks. Thus, fragmentation of the neutral networks had generated local optima. In order to assess the cause of this fragmentation, the differences between the two sub-networks was assessed in each case. Sections of the genotype that were crucial to the development of the phenotype would be constant for all genotypes in a specific sub-network whereas sections in which variance was permissible (for the given sub-network) would by definition vary from genotype to genotype. Differences between genotype sections, which were constant with respect to a single neutral sub-network, but varied with respect to its corresponding "partner" neutral sub-network are problematic. These lead to fragmentation of the overall neutral set into the respective sub-networks. Thus, such genotype sections were ascertained and in both cases it was found that the value encoding the demand threshold corresponded to just such a varying section. In one case the value for one sub-network was 7 and for the other the value was 8. Although these values resulted in very similar demand thresholds and generated the same phenotype, in genotype space they were separated by a hamming distance of 4 i.e. 4 single-point mutations away. Thus, the choice of a binary encoding had resulted in fragmentation of the neutral networks and hence the generation of local optima.

In order to address these difficulties it was necessary to ensure that contiguous values of the parameters i.e. those that are a single quantised level from each other, were also close to each other in genotype space. This was achieved by adopting the familiar Gray encoding scheme. Here, contiguous values are always a hamming distance of 1 away from each other and hence the problem highlighted above would not arise. The above analysis was repeated for the new encoding scheme and revealed that in this case there was no fragmentation of the neutral networks. Each of the 52 phenotypes was represented by a single fully connected neutral network. This removed the local optima that were found for the binary encoding.

5. Designing Phenotype Space

The mapping explored above was highly neutral. A total of 2^{20} genotypes mapped onto only 52 phenotypes. However, there were 2^{20} possible network designs and the 52 that were generated by the planning rule may not have been of the highest quality - good solutions may have been thrown away in the design of the planning rule. In order to assess whether this was the case, the fitness of each of the possible network designs was generated. This revealed that there were indeed network designs that were of a higher fitness than those allowed for by the initial planning rule. Of the 1048524 network designs that were not possible, only 2 were of slightly higher fitness than the best of the 52 possible network designs. Although in many cases this may be an acceptable price to pay for the prospect of removing local optima, it would be desirable if the best solution achievable was allowed for by the planning rule. The reason this is not so in this example can be understood by examining the relationship between the planning rule and the fitness calculation. The planning rule takes into account the demand at a site and its distance away from an existing node. It is the latter that is problematic. If a site does not contain a node the demand is routed via the PSTN to the nearest node. However, the cost for this is the same independently of the distance of that node. In effect, this part of the planning rule is irrelevant and serves only to slow the evolutionary process down. This is evidenced by examining the best of the encoded planning rules, which set the distance threshold at a low level so that the condition is always true. Conversely, demand is a very important indicator for where to place access nodes - nodes need to be placed where there is high demand as this reduces the necessity for routing calls via the PSTN and reduces the chance of demand not being satisfied. This part of the planning rule relates well to the fitness calculation. In fact, the best of the 52 solutions could easily have been achieved by simplifying the planning rule and encoding only the demand threshold. However, demand alone is not sufficient for generating the best possible network design. For this, further conditions are required that relate well to the fitness calculation.

As already mentioned, when a site does not contain a node its demand is routed to the nearest node. Thus, if a node is added to a site it may find that additional demand is generated from nearby nodes. It may be important, therefore, for the planning rules to take into account not only demand at a site but also demand in nearby sites. These requirements are captured by the following rule, which is applied to each site:

IF (demand$_i$ > *a*) **OR** (demand_in _radius$_i$(*b*) > *c*) **THEN** Add access node

Where *a* and *c* are threshold parameters, demand$_i$ is the demand at site i and demand_in_radius$_i$(*b*) is the demand within a radius *b* from site i. Note that in this case the action of the rules does not affect the calculated condition and thus the iterative process is not required. The 3 parameters *a*, *b* and *c* were represented using 5-bit Gray encoded values resulting in a 15-bit genome. The neutral sets resulting from these planning rules were generated and their connectedness assessed as previously. The new rule set produced a total of 137 phenotypes and thus, with a genotype space of 2^{15}=32768, approximately 239 genotypes mapped onto each phenotype on average. Analysis was also carried out to determine whether any of the neutral networks corresponded to local optima. In this case no such local optima existed. The new rule set had thus created a set of phenotypes that included the best possible network design and a genotype space free of local optima. This was achieved by more carefully embedding domain knowledge into the design of the genotype-phenotype mapping.

6. Discussion

The above analysis shows that it is possible to create a search space that is free of local optima through appropriate use of neutrality in the genotype-phenotype mapping, which is an important property as evolution is used for more difficult real-world problems and the associated search spaces become more rugged. The problem investigated in this work was relatively trivial and could have been solved using exhaustive enumeration or a search technique using a direct encoding (although a typical direct encoding consisting of 20-bits where each bit dictated whether a site housed an access node resulted in several local optima). However, the purpose of this paper is to investigate the issues associated with a neutral mapping on a manageable problem with a view to developing heuristics that would be applicable to more difficult problems when simpler techniques may not be as successful. Some of these heuristics will inevitably be application specific and require the use of domain knowledge. However, it may also be possible to develop more general heuristics that are applicable to a wider range of real-world problems. One example of this is the encoding that is used. Many of the problems highlighted in this work resulted from fragmentation of the neutral networks, which ultimately was an artefact of the binary encoding. The problems were alleviated by the use of Gray encoding, however, the problems may not have arisen at all if a different encoding scheme would have been used. For example, the parameters could have been represented by real values and the mutation operator either increase or decrease the value by a given amount. In effect, in this case the quantisation level is set by the amount the mutation operator modifies the value by (which may also be under evolutionary control as in an evolutionary strategy) and small changes in the parameter value would not require disproportionate and impossible changes to the encoded values. The causes of the fragmentation for the binary encoding would thus be removed. However, it is not obvious that this would produce better results during search. Percolating neutral networks are a means to an end, they allow more of genotype space to be probed for better-adapted phenotypes. The real value encoding may reduce this probing by reducing the number of mutations available for a given individual. This and other encodings will have their own set of issues that need to be assessed. The lessons learnt in this work, however, suggest that in situations where similar values of a parameter are likely to result in similar fitness', an encoding that preserves this relationship in genotype space is necessary to reduce fragmentation of the neutral networks and thus facilitate search for better adapted phenotypes.

7. References

[1] R. Shipman, M. Shackleton, M. Ebner and R. Watson, "Neutral Search Spaces for Artificial Evolution: a Lesson From Life," Artificial Life VII Conference, Portland, Oregon, 2000.

[2] L. Barnett, "Ruggedness and neutrality: The NKp family of fitness landscapes," Artificial Life VI Conference, Los Angeles, California, 1998.

[3] I. Harvey, A. Thompson, "Through the Labyrinth Evolution Finds a Way: A Silicon Ridge," First International Conference on Evolvable Systems: From Biology to Hardware. Berlin: Springer Verlag, 1996.

[4] R. Shipman, "Genetic Redundancy: Desirable or Problematic for Evolutionary Adaptation?," 4th International Conference on Artificial Neural Networks and Genetic Algorithms, New York Springer-Verlag, 1999.

[5] T. Smith, P. Husbands, M. O'Shea, "Neutral Networks and Evolvability with Complex Genotype-Phenotype Mapping," European Conference on Artificial Life: ECAL2001. pp272-281. Prague, Czech Republic, 2001.

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

[7] R. Shipman, P. Botham, P. Coker, "Coupling Developmental Rules and Evolution to aid in Planning Network Growth," BT Technology Journal, Vol 18:4, pp 95-102 , 2000.