The use of neutral genotype-phenotype mappings for improved evolutionary search

R Shipman, M Shackleton and I Harvey

In natural systems, the organism or phenotype is the result of a complex developmental process that is played out as the genetic information is interpreted. This is in stark contrast to many artificial evolutionary systems in which the phenotype is represented directly in the genetic information and there is no such development. As well as overcoming the obvious practical impossibility of directly specifying an organism in the genotype, the developmental process may yield other desirable properties. One such property is neutrality in which many genotypes develop into the same phenotype. This paper examines the effect of neutral genotype-phenotype mappings on artificial evolutionary systems through examination of an abstract redundant mapping based on a random Boolean network (RBN). It then goes on to examine the genotype-phenotype mapping within a planning tool that evolves instructions for growing telecommunications networks. It is demonstrated how the right kind of redundancy has the potential of significantly aiding the evolvability of a system.

1. Introduction

In natural evolutionary systems, the inherited genetic material is expressed in an organism via a complex developmental process. The genotype encodes many molecules that are assembled as the organism develops. These molecules interact with each other in complex and intricate ways in order to form a phenotype that is able to survive in its environment and propagate the genetic material to the next generation.

This component of natural evolutionary systems is, however, missing from the majority of artificial evolutionary systems. In these systems, the phenotype is often directly encoded in the genotype and there is no such development. As well as overcoming the obvious practical impossibility of directly specifying an organism in the genotype, a developmental process is likely to yield many other desirable properties for an evolving population. This paper is concerned with one of these properties — neutrality.

Development can be thought of as a mapping between a genotype and a phenotype. Mappings can be constructed that introduce the possibility of a number of different genotypes producing the same phenotype, i.e. a many-to-one mapping. Thus it is possible to change the genotype, via mutation for example, without affecting the phenotype. Such mutations are called neutral mutations. It has been theorised [1] that in natural systems a considerable fraction of all mutations are neutral with only a minute fraction of non-neutral mutations being beneficial. This paper explores whether neutral mutations could be beneficial for evolutionary search by increasing the evolvability of a system.

Evolution is commonly viewed as a process of ‘hill-climbing’ on a fitness landscape. On such a landscape the peaks represent genotypes of high fitness and the troughs those of low fitness. Genetic operators such as mutation encourage diversity in an evolving population and selective pressure pulls the population to the peaks. A major problem with such a view is the presence of local peaks, i.e. locally optimal but globally sub-optimal regions of the landscape. Once a population has reached such regions, further adaptation is very difficult as the population must forgo its current fitness in order to achieve loftier heights. This is illustrated in Fig 1.

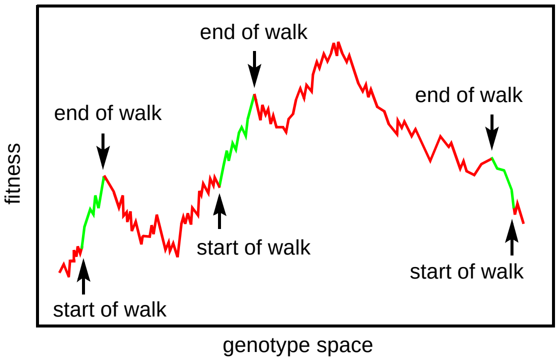

Fig 1 Adaptive walks on a fitness landscape. A population is able to climb the hills of the landscape. It can become ‘stuck’, however, at locally optimal, but globally sub-optimal, peaks. Diagram adapted from Schuster [2].

It is possible that a population could simultaneously occupy several peaks and that recombination between individuals on different peaks could allow further progression. However, this relies on maintaining sufficient diversity in an evolving population — an outcome against which selective pressure and, indeed, random genetic drift are continuously fighting (see Harvey and Thompson [3]).

The presence of neutrality in the genotype-phenotype mapping offers another possibility of overcoming these difficulties. In Fig 1 a change in the genotype always results in an associated change in fitness value.

However, neutral mutations allow movements in genotype space with no changes to the associated fitness value. Rather than moving up or down a hill, the population is able to drift along neutral ridges as shown in Fig 2. This neutral drift may significantly aid evolutionary search. Instead of becoming trapped in sub-optimal regions of the landscape a population is able to continue moving through genotype space in search of areas that allow further increases in fitness.

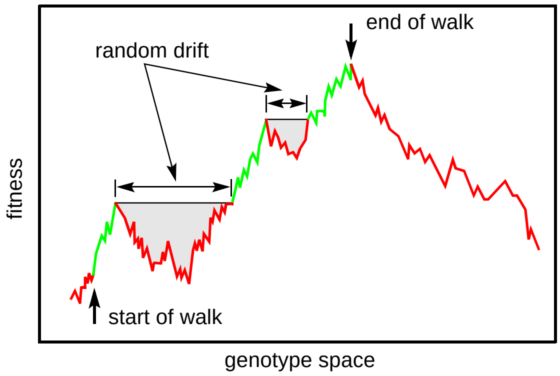

Fig 2 An adaptive walk on a fitness landscape with neutrality. Random neutral drift allows the discovery of areas of genotype space where further increases in fitness can be achieved. Diagram adapted from Schuster [2].

The importance of neutrality in evolutionary search has been studied in the context of evolvable hardware experiments [3], robot control systems [4] and more abstract genotype-phenotype mappings [5, 6]. Extensive studies of molecular folding have also provided useful insights [7, 8].

This paper demonstrates the potential benefits of neutrality in the context of a mapping based on an abstraction of a genetic regulatory network — a random Boolean network (RBN) [9]. The usefulness of neutrality is then explored in a genotype-phenotype mapping that forms an integral part of a tool aiming to aid in planning the growth of telecommunications networks [10].

2. An investigation of a random Boolean network mapping

2.1 *The random Boolean network*

In natural evolutionary systems, genes and their molecular products interact with each other in huge regulatory networks. The transcription of a gene can be influenced by other genes — in effect they can turn each other on or off. These regulatory systems have been modelled using a random Boolean network. The binary string of the network represents the genes (and their products) and the 'wiring' of the network the regulatory connections between them. A rule table is maintained for each gene in the network, which defines whether that gene is on or off depending on its regulatory inputs. This is illustrated in Fig 3.

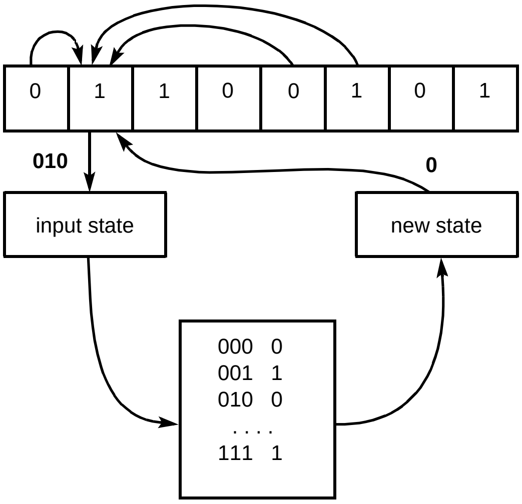

Fig 3 Illustration of the random Boolean network. Each gene is randomly assigned a number of other genes that regulate it. The state of these genes forms an index into a randomly initialised rule table that gives the new state. Each gene has its own rule table.

In order to produce a system in which generic properties of regulatory networks can be studied, the wiring of the network and the rule tables are usually randomly initialised. The genes are continuously and synchronously updated according to their rules and inputs. Eventually, the system converges on to an attractor in which a subset of states is persistently cycled through. This final state of the system can be thought of as a phenotype.

It should be noted that attractors in synchronous RBNs differ significantly from those in asynchronous RBNs [11]; in the present paper exclusively synchronous update is studied.

2.2 *The RBN mapping*

In order to produce a genotype-phenotype mapping based on the RBN it would be useful if a given genotype (initial state) always mapped on to a single phenotype (final state). This is not the case in the standard RBN as the system often cycles through a number of final states. For this reason, the 'physics' were modified for the RBN mapping. Rather than waiting for the system to settle into a final state independently of the number of updates, it was updated for a fixed number of iterations and the resulting state then interpreted as the phenotype.

A further modification was also made. An RBN with given wiring and rule tables produces a system with a number of attractors — changes to the genotype allow selection between these attractors. However, the genotype in the RBN mapping was extended to encode not only the initial state but also the wiring and rule tables. Thus, changes to the genotype could both select between existing attractors and change the attractors themselves.

In the results presented in the following sections the network length was fixed at 16 bits and the number of regulatory inputs at 3. Thus, the encoding of each bit required the following:

- initial state (1 bit),
- addresses for each regulatory gene (3 × 4 = 12 bits),
- the rule table defining the gene's next state (2^3 = 8 bits).

This resulted in a genotype of length 16 × 21 = 336 bits. A total of 2^{336} genotypes map on to 2^{16} possible phenotypes and there is therefore a very large degree of neutrality in this mapping.

2.3 *Evaluation of the mapping on a random neutral walk*

Consider an evolutionary search in which single point mutation is the only available genetic operator. For a binary genotype with no neutrality, the maximum possible number of new phenotypes accessible from any point in genotype space is simply its length, 16 in this case. If none of these phenotypes were better adapted than the current one, the search may halt at a local optimum. It has been hypothesised that neutral drift (as made possible when changing in this example to the 336-bit encoding as opposed to the 16-bit encoding) can decrease the likelihood of such an occurrence by allowing the discovery of more new, and potentially better adapted, phenotypes. In order to test whether this was the case for the RBN mapping, random neutral walks were performed. Starting from a given genotype, a single mutation neighbour was randomly selected that mapped into the same phenotype, i.e. a neutral mutation was made. Each single mutation neighbour of the new genotype was then assessed to determine whether any new phenotypes were discovered. This process was repeated for a fixed number of steps and the cumulative number of different phenotypes recorded. This is illustrated in Fig 4.

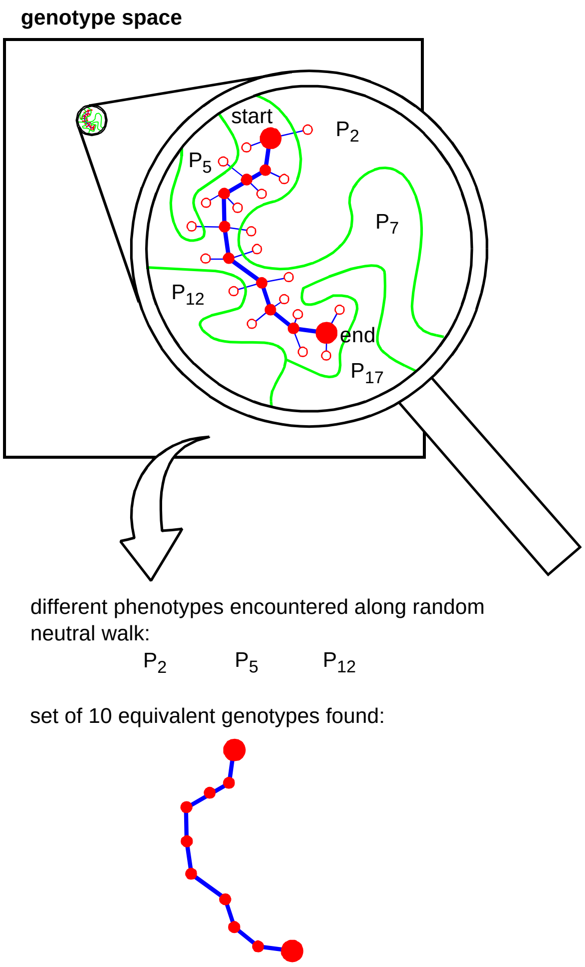
genotype space
start
P_2
P_5
P_7
P_{12}
end
P_{17}
different phenotypes encountered along random neutral walk:
P_2 P_5 P_{12}
set of 10 equivalent genotypes found:

Fig 4 A random neutral walk in genotype space. The walk begins with a given genotype and then proceeds by randomly selecting a single mutation neighbour that maps into the same phenotype. The number of different phenotypes falling within a single mutation neighbourhood of genotypes on the walk is recorded.

A total of 1024 random neutral walks were performed for the RBN mapping, each starting at a genotype which was chosen randomly from the set of all possible genotypes. The length of each walk was 500 steps. Figure 5 shows the average number of phenotypes discovered on these walks. The results reveal that neutral drift allowed the discovery of many more phenotypes than would be the case for a direct encoding with no redundancy. The total number of phenotypes discovered by the end of the walk was over 4600 on average. This greatly reduces the possibility of all reachable phenotypes being of a lesser fitness than the current one and hence the possibility of entrapment at local optima. The gradient of the slope shown in Fig 5 remains high at the end of the walk indicating that further discovery of new phenotypes is very likely.

2.4 *Evaluation of the mapping on a fitness-adaptive walk*

The previous section demonstrated how neutrality in the RBN mapping allowed for the discovery of many more phenotypes than would be the case for a non-neutral

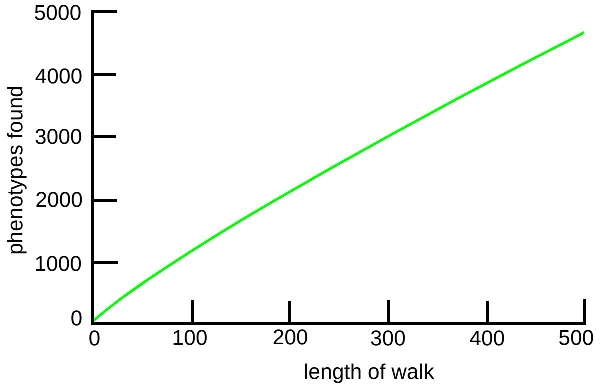

Fig 5 The number of phenotypes reachable averaged across 1024 random neutral walks of 500 steps in the genotype space of the RBN mapping. The graph shows that each step on the walk, i.e. a neutral mutation, allows the discovery of many more phenotypes. This is the property of 'constant innovation' [7], in this case there is an upper bound of 2^{16} = 65 536.

mapping. In this section the effect this increased accessibility has on the efficacy of a fitness-adaptive walk is determined. Similar walks to those presented in the last section are used; however, each phenotype is now assigned a fitness value at random (recall that there are 2^{16} different phenotypes (and fitnesses) corresponding to 2^{336} genotypes). As for the neutral walk, a random starting genotype is chosen. The walk then proceeds by adaptive moves to the single mutation neighbour with the highest fitness, provided this fitness is greater than or equal to the fitness of the current phenotype. Thus, in addition to neutral-drift, steps are taken up hill on the fitness landscape whenever possible. If neither an adaptive nor a neutral step is possible, the walk ends.

Fitnesses are assigned to phenotypes randomly in the range [0,1] with higher values denoting higher fitness. However, the fitness assignment is not uniform. The random number, r, is instead remapped in order to produce many more low fitness values than high fitness values. The fitness is given by:

$$f = e^{100(r-1)}$$

The fitnesses are distributed in this way to generate a search space in which there are few good solutions among many poor solutions, which is a property of many evolutionary search spaces. A total of 1024 walks of length 500 were performed and the average attained fitness is shown in Fig 6. The diagram also plots the fitness attained with no mapping between genotype and phenotype for comparison.

These results support those from the random neutral walk given in the last section. The direct encoding is unable to continue improving the fitness value and becomes trapped after a very small number of steps. The RBN mapping in contrast continues improving fitness before approaching an asymptote at a fitness value of 0.99. Note that the fitness assignment did not guarantee that a phenotype of maximum fitness existed in the space and it may not have been possible to achieve higher fitness values. The results demonstrate the potential of the mapping to improve the efficacy of evolutionary search.

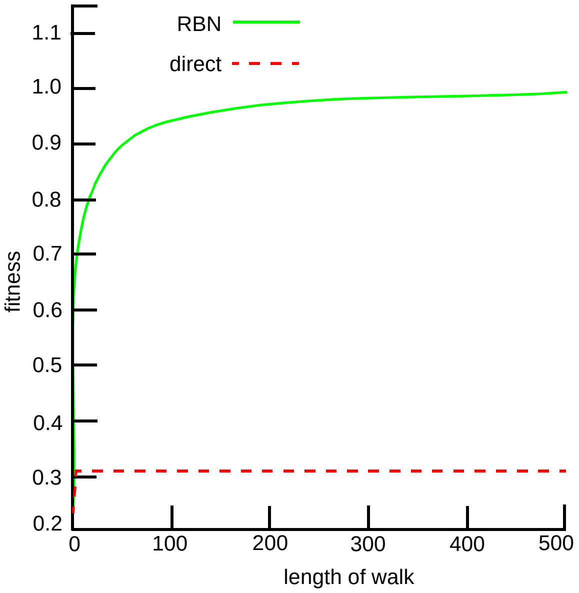

Fig 6 Results for the fitness-adaptive walks. The graph shows the average fitness attained on 1024 walks for the RBN mapping and a direct encoding, i.e. no genotype-phenotype mapping. The latter quickly becomes trapped at local optima and is unable to reach high fitness values. The RBN mapping, however, is able to continue improving the fitness value, reaching an average fitness of 0.99 after 500 steps. Note that the fitnesses were assigned in the range [0, 1], but there was no guarantee that any of the phenotypes had the maximum fitness value.

3. An investigation of a mapping within the NetGrow planning tool

3.1 The NetGrow planning tool

In order to meet the increasing demand on BT's telecommunications networks, they must be grown. The ever-increasing complexity of this task necessitates tools to aid the network designers and planners — NetGrow is one such tool and is described in detail in a companion paper [10]. In practice, real networks are often built according to various 'planning rules' which embody the experience of the relevant experts. It is the optimisation of these rules that is the target of NetGrow. Rather than evolve a network design directly, planning rules are evolved which are in turn used to produce the network design. This is a more general solution as the rules can be used in a range of situations and in conjunction with other tools. The tool consists of the following three components.

- Network simulator

  This allows the simulation and evaluation of potential new solutions.

- Planning rules

  These rules are used to make decisions as to how to grow the network in the context of the simulated environment.

- Artificial evolution

  This technique is used to optimise the planning rules with the overall effect of producing good-quality, low-cost networks.

The use of evolution within NetGrow is more analogous to the way in which evolution operates in nature. In biological evolution, the genotype encodes a set of molecules. As already noted, these molecules form complex networks of interaction to produce the organism. In NetGrow the equivalent of the encoded molecules are encoded rules, which interact with each other in the context of the simulated network environment and demand for traffic. It is the final result of the actions of these rules that forms the overall network design. This analogy is illustrated in Fig 7.

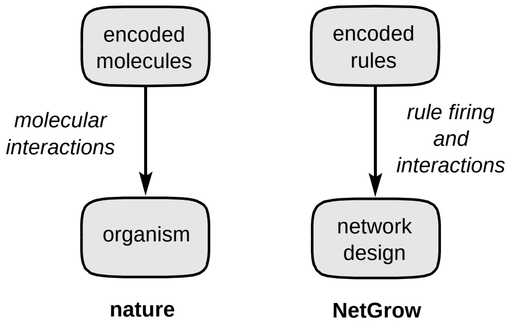

Fig 7 The developmental metaphor used in NetGrow. The combined action of the encoded rules produces the final network design in analogy to the combined actions of encoded molecules producing an organism.

The rules take the form of condition action statements such as the one shown below:

If (X < bandwidth < Y) **Then** modems = Z

Thus, conditions in the simulated network are assessed and any rules whose conditions are satisfied 'fire', which results in the action being carried out. In the example described in the following section the parameters of the rules (e.g. X, Y and Z) are encoded in the genome. Evolution can thus control the conditions in which rules fire in order to modify the final network design.

The mapping from genotype (encoded rules) to phenotype (network design) will typically be many-to-one, i.e. many different configurations of the rule parameters will produce the same network design. In the example above, a bandwidth of 40 will equally well fit within the parameters X=10, Y=50 as X=30, Y=70, and many others besides. There is thus a considerable potential for neutral mutations. The potential usefulness of these mutations is discussed in the context of the example introduced in the next section.

3.2 *The dial-IP access network example*

With the explosive growth of Internet traffic an important current problem is planning the growth of the Internet protocol (IP) network in order to meet projected demand. This section considers a simplified version of one component of this problem — the access network for dial-up Internet calls. The problem involves the placement of IP access nodes at sites around the country in order to satisfy cost effectively the projected peak demand for calls. The problem considers the ability of the network to establish calls, i.e. provide enough modems, rather than the performance of the network following the establishment of the calls.

The initial network is taken to consist of three core nodes at London, Manchester and Birmingham. In addition, 20 sites are distributed around the country, which are potential homes for access nodes. Each of the core nodes can simultaneously handle 2000 calls and each access node an additional 1000. Each of the potential sites has an associated peak demand for data calls, which ranges from 0 to 1500. The following simple rule is used to make decisions regarding which of the sites should house an access node:

If (demand > X) **And** (distance to nearest node > Y)
Then (add an access node)

Note that the action of the rule affects the calculated conditions, i.e. the addition of a new node will affect other sites' distances to the nearest node. Thus, 'one-shot' application of the rules is not possible. An iterative process is therefore used in which the rule that is firing the most strongly (as calculated by the amount the conditions are above the thresholds) is activated. The conditions are then re-assessed and the process repeated until no further rules fire. As in the previous example, it can be seen that this rule produces a many-to-one mapping from genotype to phenotype as many different parameter configurations will produce the same network design.

In addition to the parameters X and Y, two more parameters were defined that allowed evolution to control the relative importance of the two conditions. These parameters multiplied the amount the calculated conditions were above the thresholds and could therefore adjust the rules that were firing the most strongly. Consider the following example. Site A has a demand of 50 and is a distance of 30 away from the nearest node. Site B has a demand of 100 and is a distance of 15 away from the nearest node. If the parameters X and Y had values of 30 and 10 respectively, site A's rule would fire with a value of

(50 – 30) + (30 – 10) = 40 and site B’s rule with a value of (100 – 30) + (15 – 10) = 75. The latter would therefore be chosen to house an access node, which could result in site A no longer being the required distance away from the nearest node and so becoming ineligible to house an access node in further iterations. Now consider two new parameters, *J* and *K*, which scale the amount the demand and distance are above their thresholds. Values of *J* = 1 and *K* = 4 for example would cause site A to fire with a strength of (50 – 30) × 1 + (30 – 10) × 4 = 100 and site B with a strength (100 – 30) × 1 + (15 – 10) × 4 = 90. Site A would now be chosen to house an access node, which could have the effect of precluding site B from doing so in future iterations. These parameters, therefore, have the potential to modify the final network design. They will be important in later discussions of useful versus useless neutrality.

The quality of networks produced by the application of the planning rule was calculated using a combination of the following four costs.

- Node cost

  This is the cost of each new access node.

- Link cost

  This is the cost of linking a new access node to the nearest core node.

- Dropped calls

  A cost is incurred for every call that could not be established by the network.

- PSTN calls

  If the demand for data calls at a site could not be satisfied by a collocated IP node, it was routed to the nearest available IP node via the normal telephone network (PSTN). Each call incurred a cost.

A good rule would thus find a suitable balance between the increases in hardware costs and decreases in quality of service costs resulting from the addition of access nodes. The overall calculated cost was used as the fitness value for the evolutionary algorithm. Full details are available in a companion paper [10].

3.3 Evolving the planning rule

The four parameters *X*, *Y*, *J* and *K* were each encoded as a 5-bit number, giving a genotype of length 20. Each of the parameters was given a minimum and maximum value and the 5-bit representation quantised this range into 32 levels. The range of the demand threshold was 0 to 1500 (the maximum possible demand), distance threshold 0 to 50 (the maximum possible distance) and both scaling parameters 0 to 5. A genetic algorithm [12] was used with roulette wheel selection, a per-bit mutation rate of 0.01, a crossover rate of 0.7 and population size of 20. The network costs were as follows — each access node incurred a cost of 800, each link a cost of 25 per unit distance, each dropped call a cost of 10 and each PSTN call a cost of 5. Note that these values have no units, it is their relative rather than absolute values that are important.

Figure 8 shows a sample plot of the evolutionary process over 1180 generations. All encoded rules initially produced high-cost networks, as would be expected for a randomly initialised population. However, better quality networks were quickly discovered and a relatively low-cost network was produced within the first 75 generations. This network was not improved upon for around a thousand generations until two better quality networks were discovered in relatively short succession. The final network design was the endpoint of a series of five networks that correspond to the plateaus in Fig 8. The rule parameters that produced the first network on each of these plateaus are shown in Table 1.

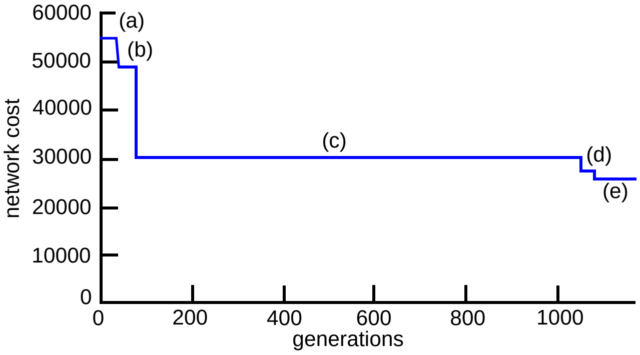

Fig 8 An example evolutionary run is shown for the IP access application. The cost of the network produced by the current best rule is shown.

Table 1 The rule parameters that generated the first network on each of the plateaus in Fig 8.

	Distance threshold	Distance scaling	Demand threshold	Demand scaling
(a)	18.75	3.59	421.88	4.69
(b)	6.25	4.84	375	3.24
(c)	0	2.03	375	1.88
(d)	0	2.66	187.5	3.12
(e)	1.56	2.66	234.38	2.5

3.4 Useful and useless neutral mutation

The encoding of the rule used to generate the above networks introduced significant possibility for neutral mutations. For any given value of distance and demand, the only threshold mutations that had the potential of changing the network design were those that resulted in a value becoming greater than a threshold it was previously below and vice versa. Many mutations did not have this effect and were thus neutral. In addition, many changes to the scaling parameters did not affect the final network design, as they did not modify the relative strength of rule firing to a great enough degree. However, care needs to be taken when introducing the possibility for neutral mutation. Although such a possibility would always allow for neutral drift, it would not always allow for beneficial neutral drift. In order to decrease the likelihood of entrapment at local optima, neutral drift must allow access to phenotypes that would otherwise have been inaccessible.

A situation where this is not the case can be illustrated by considering the thresholds. In this example the range of the thresholds are set to be equal to the minimum and maximum of the values themselves. For example, the demand threshold varies between 0 and 1500, which is the range of demand that is possible at a site. It would be possible, however, to increase the maximum of this range to 3000 for example. Thus, it would be guaranteed that all encoded values of 1500 and above would not change the network design as demands greater than this value are not possible. Mutations that maintained the threshold in the range 1500—3000 would therefore be neutral. However, these neutral mutations would not be of any benefit in reducing the likelihood of local optima, as they could never contribute to the fitness of the phenotype. For any given network design it would make no difference whether the threshold was 1500, 3000 or any value in between. A neutral mutation would only be useful if it produced a value that, given changes elsewhere on the genotype, played a role in the development of the phenotype, i.e. a value that had the potential of losing its neutrality. The neutrality created through increasing the maximum value of the threshold would not improve evolutionary search but simply encourage non-beneficial random drift, which may slow the search down.

3.5 Neutrality in the scaling parameters

The neutrality described in the previous section is at one end of a scale — guaranteed to never directly contribute to the fitness of the phenotype. However, neutrality could also be introduced that was potentially useful but very unlikely to be so. For example, the entire genotype could be duplicated and an extra bit employed that determined whether the original or its duplicate would be interpreted. Thus, at any one time an entire (original) genotype would be free to be mutated at will without affecting the current fitness. It is possible that neutral mutations could produce a high-quality phenotype that could then be interpreted as the current phenotype through mutation of the ‘switch’. However, this is highly unlikely — local optima do not exist in this search space but the transition points between phenotypes (i.e. areas of genotype space in which further increases in fitness could be made) are very sparse. This is akin to randomly drifting along an enormous neutral ridge looking for a single point that allows access to a different phenotype. It is very unlikely that the needle in this haystack would be discovered. To improve evolutionary search, the presence of many needles needs to be encouraged.

An example of where this was not achieved is in the choice of scaling parameters in this work. These two parameters were assigned the ranges 0 to 5 and, as already discussed, scaled the amount that the conditions were above their thresholds. However, these amounts were likely to be very different for the two conditions. The maximum distance between two points in the simulated geography was 50 and the maximum demand was 1500. Thus, the amount the two values could exceed their thresholds was 0—50 for the distance and 0—1500 for the demand. However, the range of their scaling was the same. There was thus an implicit bias in the encoding towards placing access nodes at sites with high demand before those that were a large distance away from the nearest node. This bias was exacerbated by the fact the placement of sites was such that the actual maximum distance between sites was much less than the maximum possible distance of 50. It was still possible for these biases to be overcome (with a high-distance scaling parameter and a low-demand scaling parameter), but the bias made it less likely that mutations to the scaling parameters would have an effect on the overall network design.

The encoding of the scaling parameters introduced a significant probability of neutral mutation. However, the bias shifted the balance between neutral mutations and non-neutral mutations too much in favour of the former. Neutral ridges had been formed but the accessibility between them had been reduced — too much hay, not enough needles. The effect of the scaling parameters would have been increased if their ranges were dependent on the amount that the associated value was above its threshold. For example, the maximum distance threshold could have been increased to thirty times that of the demand threshold. This would have balanced the effect of the scaling parameters and increased the likelihood of mutations having an effect on the final network.

The reduced effect of the scaling parameter is highlighted by the results presented in Table 1. All the transitions to networks of higher fitness were the result of mutations to the thresholds and not to the scaling parameters. The changes in the scaling parameters during periods of neutral drift between individual transitions were coincidental rather than necessary. However, that is not to say that neutral drift in the scaling parameters was unimportant. Consider the transitions from network (a) to (c) for example. Changes in the scaling parameter were not necessary to allow the transition from network (a) to (b) or the transition from (b) to (c). However, changes to the scaling parameter were necessary to allow the transition from network (a) to (c), i.e. the scaling parameters used in the rule that generated network (a) could not have been used in the rule that produced network (c). Thus, although neutral drift in the scaling parameters was not important for individual transitions, it was important when considering the series of transitions.

3.6 Neutrality in the threshold parameters

Now consider the mutations to the threshold parameters that were the driving force behind the transitions. The transitions from network (b) to (c) and from network (d) to (e) were the result of single mutations. In the former case the mutation reduced the distance threshold from 6.25 to 0. In the latter case the demand threshold was increased from 187.5 to 234.38. The plot in Fig 8 reveals that these mutations were quickly discovered. However, the transitions from network (a) to (b) and network (c) to (d) required two mutations. In the former case a single mutation was required to both the demand and distance thresholds. In the latter case a dual mutation to the demand threshold was required. However, Fig 8 shows that they took very different lengths of time to discover. The time taken for the transition from network (a) to (b) was of the same order as that for a single mutation; the transition from (c) to (d) took a great deal longer.

Figure 9 shows the effect of all possible mutations on the genotype producing network (a) at the start of the plateau. Also shown in Fig 9 is that all possible single mutations are either deleterious or neutral. It is not possible to increase the fitness of the network by single-point mutations from this point in genotype space. Also shown is that the two required mutations are individually neutral — either can be made without affecting the current network. Thus, the scene can be set for the second required mutation, which does increase the fitness of the resulting network. This is a good example of useful neutrality. The individual mutation has no immediate effect on the phenotype but, given changes elsewhere on the genotype, comes to play an important role in its construction.

Figure 10 shows the effect of all possible mutations on the genotype producing network (c) at the start of the plateau. Again all mutations are either deleterious or neutral. The figure reveals that in this case both the mutations that are required to produce the transition from network (c) to (d) are individually deleterious. It is thus not possible for these mutations to be made without forgoing the current fitness value. Individuals with these mutations would quickly be removed from the population through selective pressure and the stage cannot be set for the second mutation. In this case it is likely that both mutations must be made simultaneously. The probability of this occurrence is very much less than that of a single mutation. This is highlighted by the number of generations that were required for the dual-mutation to occur. In effect, a local optimum has been reached at network (c) and reliance is being made on improbable multiple mutation events that can jump the search over a valley. Note that the jump could potentially also be made via a beneficial crossover event. However, the functional parts of the genotypes in the population were typically highly converged during periods of neutral drift and therefore crossover had little effect.

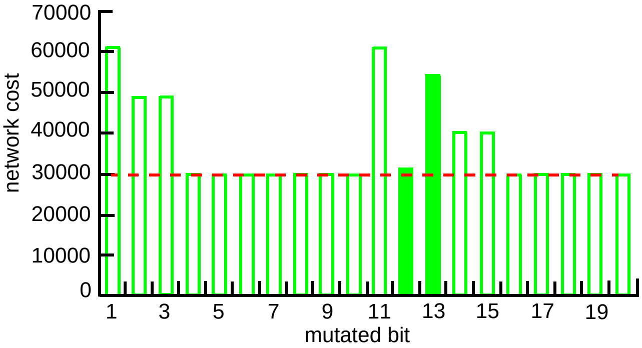

Fig 9 The effect on network cost of all possible mutations of the genotype producing network (a). No mutations are immediately beneficial; however, the two mutations required to cause a transition to network (b) (shaded columns) are both individually neutral. They can thus be made without affecting the current network in order to 'set the scene' for the second mutation. The dashed line shows the cost of network (a).

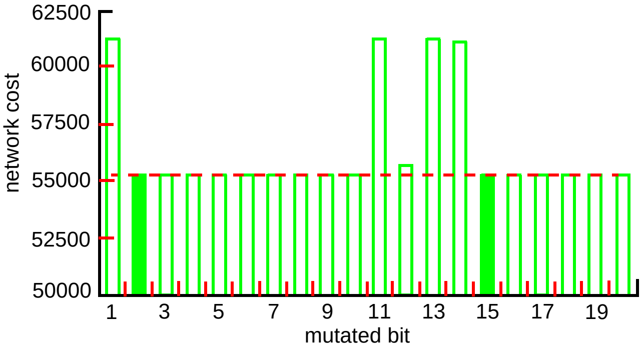

Fig 10 The effect on network cost of all possible mutations of the genotype producing network (c). Again, no mutations are immediately beneficial. In this case both the mutations required to transition to network (d) (shaded columns) are individually deleterious and cannot be made without forgoing the current fitness value. Stage-setting neutral mutations are thus not possible and the mutations must be made simultaneously. The dashed line shows the cost of network (c).

The ability of neutral mutations to improve evolutionary search is evident from this example. In the cases where immediate improvement was not possible, the existence of beneficial neutral mutations allowed the search to quickly progress. Their absence caused it to drastically slow and even halt but for unlikely multiple mutation events.

4. Conclusions

This paper has explored the usefulness of neutral mutation for evolutionary search. A genotype-phenotype mapping, based on an abstraction of a genetic regulatory network, was first investigated. This mapping introduced a large amount of neutrality that was found to allow access to a far greater number of (potentially higher fitness) phenotypes than would be the case for a non-neutral mapping. This increased accessibility was found to significantly aid evolutionary search on a fitness-adaptive walk. Secondly, the NetGrow planning tool [10] was investigated. It was found that the genotype (encoded planning rule) to phenotype (network design) mapping contained large amounts of neutrality. The usefulness of this neutrality was discussed and it was demonstrated how the presence of beneficial neutrality significantly aided the search process.

The presence of local optima has always been a problem for search algorithms. This paper adds to the work that is beginning to demonstrate that consideration of neutrality in the encoding of the problem offers real potential for overcoming problems associated with local optima and increasing the efficacy of the search.

Acknowledgements

Part of this paper draws on work performed with our colleague Dr Marc Ebner, Universität Würzburg, who produced Fig 4. Rob Shipman also acknowledges the support of the EPSRC.

References

1 Kimura M: ‘Population genetics, molecular evolution, and the neutral theory: selected papers’, The University of Chicago Press, Chicago (1994).

2 Schuster P: ‘How does complexity arise in evolution?’, University of Vienna, Theoretical Biochemistry Group, Working Paper 96-05-026 (1996).

3 Harvey I and Thompson A: ‘Through the labyrinth evolution finds a way: a silicon ridge’, Proc of the First International Conference on Evolvable Systems: ‘From Biology to Hardware (ICES’96)’, Berlin, Springer Verlag (1996).

4 Shipman R: ‘Genetic redundancy: desirable or problematic for evolutionary adaptation?’, Proc of the Fourth International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA’99), New York, Springer Verlag (1999).

5 Shackleton M, Shipman R and Ebner M: ‘An investigation of redundant genotype-phenotype mappings and their role in evolutionary search’, to appear in Proceedings of the 2000 Congress on Evolutionary Computation (2000).

6 Shipman R, Shackleton M, Ebner M and Watson R: ‘Neutral search spaces for artificial life: a lesson from life’, to appear in Proceedings of the Seventh International Conference on Artificial Life (alife VII) (2000).

7 Huynen M: ‘Exploring phenotype space through neutral evolution’, Santa Fe Institute Working Paper, 95-10-100 (1995).

8 Huynen M, Stadler P and Fontana W: ‘Smoothness within ruggedness: the role of neutrality in adaptation’, Proc of the National Academy of Science, 93, pp 394—401 (1995).

9 Kauffman S A: ‘Origins of order: self-organisation and selection in evolution’, Oxford University Press (1993).

10 Shipman R, Botham P and Coker P: ‘Coupling developmental rules and evolution to aid in planning network growth’, BT Technol J, 18, No 4, pp 95—102 (October 2000).

11 Harvey I and Bossomaier T: ‘Time out of joint: attractors in asynchronous random Boolean networks’, in Husbands P and Harvey I (Eds): ‘Proc of the Fourth European Conference on Artificial Life’, MIT Press (1997).

12 Holland J: ‘Adaptation in Natural and Artificial Systems’, MIT Press (1994).

Rob Shipman has worked in the area of nature-inspired systems for several years at Cyberlife Technology Limited and, since April 1998, BT’s Future Technologies group at Adastral Park.

Following early-life sporting preoccupations, he received a first class BEng in Electronic Systems and Control Engineering from Sheffield Hallam University, specialising in artificial intelligence, and was also awarded for exceptional performance by the Armourers and Brasiers company.

In addition, he holds an MSc in Evolutionary and Adaptive Systems from the University of Sussex which was awarded with distinction. From 1994 to 1995 he studied in the Speech, Vision and Robotics group at the University of Cambridge.

He has also held several positions as a professional software engineer.

Mark Shackleton graduated from Sheffield University in 1986 with a BSc in Computer Science. He first worked for Singer Link-Miles, manufacturers of commercial flight simulators, developing real-time 3-D computer graphics algorithms and systems. He joined the Image Processing and Computer Vision group at BT in 1989. In this group he designed and implemented a number of systems in areas such as automatic face recognition, and content retrieval from images and video sequences. During this time he spent two periods of six months seconded to the MIT Media Laboratory working closely alongside researchers there. In 1996 he moved across to the Future Technologies group at Adastral Park, whose remit is to develop novel solutions to BT’s problems using a nature-inspired approach. He now leads this group, working on research and applications within the domain of evolutionary computation and adaptive systems.

Inman Harvey has degrees in Mathematics, Philosophy and Social Anthropology from Cambridge, and more recently an MSc and a Doctorate from Sussex where he is Senior Lecturer.

There he is a founder member of the Evolutionary and Adaptive Systems Group, and specialises in Artificial Evolution.