Facing The Facts: Necessary Requirements For The Artificial Evolution of *Complex* Behaviour

CSRP 422

Nick Jakobi
School of Cognitive and Computing Sciences
University of Sussex, Brighton BN1 9QH, England
`email: nickja@cogs.susx.ac.uk`
`tel: (UK) 01273 678061`
`fax: (UK) 01273 671320`

Abstract

This paper sets out a conceptual framework for the open-ended artificial evolution of complex behaviour in autonomous agents. If recurrent dynamical neural networks (or similar) are used as phenotypes, then a Genetic Algorithm that employs variable length genotypes, such as Inman Harvey's SAGA, is capable of evolving arbitrary levels of behavioural complexity. Furthermore, with simple restrictions on the encoding scheme that governs how genotypes develop into phenotypes, it may be guaranteed that *if* an increase in fitness requires an increase in behavioural complexity, then it *will* evolve. In order for this process to be practicable as a design alternative, however, the time periods involved must be acceptable. The final part of this paper looks at general ways in which the encoding scheme may be modified to speed up the process. Experiments are reported in which different categories of scheme were tested against each other, and conclusions are offered as to the most promising type of encoding scheme for a viable open-ended Evolutionary Robotics.

Introduction

Early work in Evolutionary Robotics has succeeded in producing simple behaviours for autonomous agents [2, 5, 9, 1]. It is becoming increasingly clear, however, that there is an upper limit to the behavioural complexity that Genetic Algorithm (GA) optimization techniques alone may achieve. If artificial evolution is ever to become a practicable alternative to human ingenuity in the design and creation of control architectures for autonomous agents, then this limit has to be overcome. This paper sets out what has to be done to ensure that the evolutionary process underlying a viable ER is limit*less* in terms of the behavioural complexity it is capable of producing.

1 How to ensure that the artificial evolution of complex behaviour is possible

This section introduces a GA that employs variable length genotypes to code for arbitrarily complex phenotypes. If recurrent dynamical neural networks [11] or something similarly suitable are used as phenotypes, then this makes it *possible*, at least as far as the evolutionary process is concerned, to produce arbitrarily complex behaviours. This has major consequences for the way in which the evolutionary search space is normally viewed and a different, more intuitive model is offered. The familiar concept of the fitness landscape is then redefined in terms of this new model.

1.1 A GA capable of evolving arbitrary levels of complexity

According to conventional theories of how and why GAs work (such as Holland's schema theorem[6] or the building block hypothesis[3]) it is during the transition from initially diverged to finally converged that the power of a GA is employed. Even though with suitable mutation a GA will never completely converge (and we can never be sure that it has reached a stage where it *will not* discover any better solutions if left long enough), it is generally recognised that there comes a point at which the rate of both fitness-improvement and convergence drops to zero, and that this is the point at which optimization stops. In contrast, the type of open-ended artificial evolution required by Evolutionary Robotics must *not* stop (see Section 2). A GA based on or requiring one-way change (such as population convergence) contains a built in stopping point where that one-way change goes to its limit. If we are after an open-ended evolutionary process that is truly limitless in terms of the behaviour complexity it is capable of producing, therefore, we cannot rely on traditional GA optimization techniques.

In [4], Inman Harvey offers a straight-forward way of literally extending traditional GA models for the purposes of open-ended artificial evolution. He proposes a Species Adaptive Genetic Algorithm (SAGA) which differs from a conventional GA in two crucial respects. Firstly, the length of the genotype is under evolutionary control, thus ensuring that there is no upper bound to the complexity that can, in principle, be achieved by the algorithm. And secondly, from the initial seeding of the population to the time at which the particular run is terminated, the population displays a high degree of phenotypic convergence. This is why the GA is called Species Adaptive. It ensures the sort of steady state process that can potentially, like evolution in the real world, continue indefinitely.

1.2 The search space is not a space

GAs search the set of all possible genotypes for individuals that correspond to phenotypes of high fitness. Conventional GA optimization refers to this set as 'genotype space', normally interpreted to mean the space of all possible permutations of the fixed number of base units or parameters that make up a genotype. Distances in such a space are measured by the Hamming distance. The less alike two genotypes are, the greater the minimum number of single point mutations that are required to transform one into the other, and the further apart they are in 'genotype space'.

There are crucial differences, however, between the set of all possible genotypes searched by a variable dimensionality GA such as SAGA and that searched by a conventional GA. The number of base units or parameters that may be permuted is *not* fixed, Hamming distance is meaningless between two genotypes of different lengths, and point mutations alone are insufficient to transform a genotype of one length into a genotype of another. Clearly, this set can not be represented by a fixed-dimensionality genotype space.

Instead of a Euclidean space, the set of possible genotypes searched by a variable dimensionality GA is best represented by a graph or network i.e. a set of interconnected nodes. Each node corresponds to a possible genotype and each connection corresponds to a single application of a genetic operator. In order to make this more explicit, consider a stripped down version of a variable dimensionality GA involving two genetic operators: mutation and a 'change length' operator. Two genotypes may then be defined to be maximally similar, though not identical, if they differ by a single genetic base unit (bit, character, nucleic acid etc.). This is the case if *either* they are the same length but differ by a single genetic base unit (the mutation operator) *or* one genotype contains exactly one more genetic base unit than the other but in all other respects they are identical (the 'change length' operator). A connection is defined to exist between two nodes on the graph of all possible genotypes if and only if their corresponding genotypes are maximally similar. The more similar two genotypes are to each other, the shorter the direct path between their corresponding nodes on the graph.

A GA that allows the artificial evolution of arbitrary levels of behavioural complexity, then, searches *a graph* of possible genotypes, rather than *the space* of all possible genotypes, for individuals that correspond to phenotypes of high fitness. The connectivity of the graph, in terms of neighbourhood relationships, is no longer an intrinsic function of the search space, but a function of the genetic operators employed by the GA. In fact, whether the graph does indeed contain a node for every possible genotype is itself a function of the genetic operators, since some genotypes may be unreachable. The exact connectivity of the graph can have far-reaching effects on the efficiency of the evolutionary process, with special operators such as translocation and inversion providing 'short cuts' between nodes that would otherwise be many steps away from each other. In order to keep the discussion tractable, however, this paper is concerned only with the simple graph that results from the stripped down variable dimensionality GA outlined above.

1.3 Elements of the fitness landscape

The search performed by a conventional GA is guided by measures of phenotypic fitness, and a GA capable of evolving arbitrary levels of complexity is no different. Although it is necessary to abandon the idea of a Euclidean genotype space, we may still retain the idea of a fitness landscape (and its attendant explanatory power), by regarding the fitness value of the phenotype corresponding to each node on the graph of possible genotypes as an altitude metric. Moving along a connection from one node in the graph to another is deemed up-hill if the second node corresponds to a phenotype of greater fitness than that of the first, down-hill if the reverse is the case, and horizontal if both nodes correspond to phenotypes of the same fitness. Since the job of the the GA behind open-ended Artificial Evolution is to find areas of higher and higher fitness, it is the broad character of the landscape, whether gentle slopes or vertical cliffs, many local maxima or no maxima at all, that has the single most profound effect on the speed and efficiency of the search.

Every fitness value is a function of the *total* process that results in its assignment to a genotype, from the encoding scheme under which a phenotype is developed to the nature of a fitness trial. It is a mistake to regard the topography of the fitness landscape, as overlaid on the graph of possible genotypes, as a function of any one component of this process. Changing the encoding scheme will have just as drastic effects on the relative fitness of individual nodes as altering the fitness test. For any given selection criteria (such as a particularly difficult ER task, for instance) it may be possible to shape the fitness landscape into something that the evolutionary process finds easy by the careful selection of an appropriate encoding scheme along with other components of the fitness assignment process. This issue is of overwhelming importance to the viability and practicability of artificially evolving complex behaviour. It is further discussed in section 3.

2 How to ensure that the artificial evolution of complex behaviour is inevitable

The evolutionary process behind a viable ER must be such that if greater behavioural complexity is *required* in order for the fitness of the population to improve then it *will* evolve. If this condition is not fulfilled, then there is no guarantee that the evolutionary process will not become 'stuck', leaving us with simple behaviours that are poorly adapted. In fact, this condition is more general, since the aim of ER is to produce autonomous agents that are maximally adaptive to their environment, and not necessarily maximally complex. We therefore require that *if* it is possible for the population to be fitter, it *will* become fitter. If there is a global fitness maximum in the fitness landscape, then we require that the evolutionary process eventually finds it, and where the fitness landscape is continually changing, as might be the case in a coevolutionary scenario for instance [12], we require that the population continues to adapt ad infinitum.

This property of the evolutionary process, here referred to as open-endedness, is logically equivalent to the population never becoming 'stuck' on a *local* fitness maximum in the fitness landscape[1]. It may be guaranteed to pertain in one of two ways. Either one sets up the GA so that if the population comes to rest temporarily on a local fitness maximum, a new and fitter region of the fitness landscape will always be found, or one ensures that the fitness landscape is such that there *are no* local fitness maxima. Both of these approaches, here called 'multiple mutations' and 'neutral networks' respectively, are discussed below.

Although it is actually a trivial matter to demonstrate how the evolutionary process may always find its way to a fitter part of the space (if there is one) given enough time... it is a far from trivial matter to ensure that it does it quickly enough to safeguard artificial evolution as a viable design alternative. Any method of ensuring the open-endedness of the process must, at least in principle, have the potential for massive future performance improvements. If, using a certain method, it is a priori impossible to speed up the evolutionary process, then this method must be rejected.

2.1 Multiple mutations

The first approach to guaranteeing open-endedness considered in this paper places the emphasis squarely on the underlying GA. To appreciate how it works, we consider a population that is temporarily 'stuck' on a local fitness maximum, and look at ways in which the GA could be guaranteed to produce a fitter individual.

In it's most basic form, the GA could guarantee the *eventual* production of an individual of greater fitness by introducing a single random genotype per generation until one is found. This does indeed guarantee open-endedness, but it constitutes nothing more than the random search of the graph of possible genotypes. It certainly does not provide the power necessary to render open- ended artificial evolution a practicable proposition. The question on which the viability of this approach rests, then, is whether it is possible to alter the GA in such a way that it escapes from local fitness maxima significantly quicker than random search.

The answer is that it is not possible without some foreknowledge of the fitness landscape. There are many ways to bias the search that still guarantee the open-endedness of the evolutionary process. For example, we could introduce multiple single point mutations at each offspring event according to a Poisson distribution, thus safeguarding the minute possibility of a vastly mutated offspring while biasing the search towards the population average. But in order for this method to out-perform totally random search, the fitness landscape must be known to be of a certain shape e.g. local maxima must always be near to areas of higher fitness. We *may* constrain the shape of the fitness landscape by altering elements of the fitness value assignment process (as outlined in section 1.3), and this might provide us with the appropriate foreknowledge. However, since it is an easy matter to constrain the fitness landscape to such an extent that there *are no* fitness maxima (see below), this sort of hybrid ap-

---

[1] On the graph of possible genotypes, a local fitness maximum is constituted either by a single node or by a subnetwork of inter-connected nodes that are all of the same corresponding fitness. It may be recognised by the fact that all the nodes directly connected to it are of lower corresponding fitness.

proach should not constitute the primary focus of enquiry.

2.2 *Neutral networks*

The other way to guarantee that artificial evolution is open-ended places the emphasis on the fitness assignment process. It ensures that there are no local fitness maxima in the fitness landscape by placing restrictions on the encoding scheme; an evolutionary process employing nothing much more complicated than hill-climbing is thus guaranteed to be open-ended. The restrictions may take many forms, but for a simple example let us look at an encoding scheme in which it is always possible to add extra genetic material (in the form of extra bits, characters etc) to the genotype without effecting the phenotype, and it is always possible to switch segments of the genotype 'on' or 'off' by way of single point mutations. These restrictions may seem strange but they are in fact true of the encoding scheme behind natural development. In order to show that there are no local fitness maxima in the resultant fitness landscape, consider a worst case scenario - the genotype coding for a particular phenotype cannot undergo a normal single point mutation anywhere along its length without suffering a loss in fitness. Extra genetic material that is 'off' can always be added to the genotype without effecting the phenotype, however, and this will eventually lead, after a monkeys-typing-Shakespeare length of time, to the evolution of a stretch of 'junk dna' that codes for a fitter phenotype (if there is one) than that expressed by the current 'on' stretch of genotype. Since a single-point mutation can always switch an 'on' stretch of genotype to 'off' and an 'off' stretch of genotype to 'on', it is therefore *possible* that the 'junk dna' is expressed while the rest of the genome is switched 'off', thus producing a fitter phenotype.

Under this encoding scheme, we can guarantee that no node or set of nodes on the graph of all possible genotypes constitutes a local fitness maximum. All nodes will connect to at least a few other nodes of the same corresponding fitness thus forming large neutral networks (the term here is adapted from its use in [10]). In every neutral network there will be one or more nodes that also has an up-hill connection to a node in a neutral network of higher fitness. It is possible, therefore, to find a path from *any* node through the graph of possible genotypes that monotonically increases[2], with respect to corresponding fitness, ad infinitum.

If this approach is taken seriously then the major part of the evolutionary process behind an open-ended ER becomes a matter of searching neutral networks for connections that lead up-hill on the fitness landscape, and not hill-climbing itself. There *are* hills in the typical ER fitness landscape, but consider what these represent. To travel up a fitness hill on the graph of possible genotypes amounts to a gradual altering of the parameters of the corresponding phenotypes until a fitness plateau (which is actually a neutral network) is reached. But this gradual shifting of the parameters is unlikely to result in a change to the *functionality* of expressed phenotypes. In other words, although changing the parameters in this fashion may make phenotypes *better* at amassing fitness points, it is unlikely to change *how* they amass fitness points.

The slopes of ER fitness landscapes span the phenotypic space between a bad version of a control strategy and a perfected version of that same control strategy. As an example, imagine a population of control architectures that evolves from the situation in which a simple robot reacts slightly to one of its whiskers, to the situation in which the robot travels round a room efficiently using its whiskers as obstacle-detectors. The neutral networks or fitness plateaus of ER fitness landscapes span the phenotypic space between different or augmented control strategies. Such a transition might involve the previous population of control architectures, that has evolved to make the robot avoid obstacles using its whiskers alone, evolving further to make use of the robot's infra-red sensors instead or as well.

The point is that fitness slopes are relatively easy to climb since directed search is possible (although there may be problems caused by noise). Evolutionary search from a perfected version of one control strategy to a bad version of another (that might well subsume the first), on the other hand, *is blind*. There are no hills to climb. In our example, the fittest members of the population continue to make the robot avoid obstacles using its whiskers in exactly the same way until, eventually, a mutant arises which also reacts *slightly* to infra-red as well. Until this happens, the population is wandering randomly on a neutral network.

The question on which the viability of this particular approach rests, then, is whether it is possible to significantly reduce the time spent by the evolutionary process searching neutral networks from the monkeys-typing-Shakespeare level. The answer to this question is, unlike the previous approach, that it may well be possible and Section 3 explores how further restrictions on the encoding scheme can improve performance significantly. For although it is true that search on neutral networks is blind, it might be feasible to reduce the expected time of arrival at an up-hill connection to an area of greater fitness by many orders of magnitude.

---

[2] monotonically increasing means never going down, not always going up.

2.3 How a simple GA would work in the context of neutral networks

So far, no explanation has been given of *how* an open-ended GA would work. This is because, in order to give a satisfactory account, some knowledge of the nature of the search space is required. There is no point, for instance, in spending time explaining and ensuring how an open-ended GA will never settle on a local fitness maximum if there are no local fitness maxima in the fitness landscape. Having shown that the most promising way to think about fitness landscapes for open-ended artificial evolution is in terms of inter-connected neutral networks, we are now in a position to give an exposition of what we require from a simple GA in order for it to operate on such a landscape, and how we expect one to meet these requirements. This will give a general picture of the evolutionary process underlying an open-ended ER which we may use to point the way forwards for performance improvements.

If the fitness landscape consists of neutral networks connected together by slopes, we certainly require the GA behind open-ended artificial evolution to perform hill-climbing type search. This is only half the story, however. It is important to realize that as well as being connected to nodes of equal corresponding fitness, every node on a neutral network is connected to many that are of lower corresponding fitness. The application of genetic operators to a particular individual will result in a genotype of lower corresponding fitness just as easily (usually much more easily) as one of equal corresponding fitness. This means that the GA must also prevent the population from 'falling off' whichever neutral network it happens to be on while continuing the search.

If the rate at which genetic operators (such as mutation or a 'change length' operator) are applied is kept low, the selection pressure in conjunction with the constant renewal of the population will ensure a high degree of genetic convergence. The population will thus cluster together on the graph of possible genotypes exploring a compact region thoroughly. Since individuals of higher fitness produce a greater number of offspring than individuals of lower fitness, the area around their corresponding nodes on the graph of possible genotypes will be more densely populated, and thus better searched. If this search results in the discovery of nodes with even higher fitness, then these in turn will quickly come to be the locus of rapid local population expansion, and the average fitness of the population will increase. The GA is thus performing simple hill-climbing type search.

If the fittest member of the population is never replaced, then hill-climbing search is sufficient, also, to stop the population from 'falling off' a neutral network. We may either explicitly implement an elitist policy in which the fittest member of the population automatically goes through to the next generation (or is never replaced in the case of a steady-state GA), or we may set the rate at which genetic operators are applied to such a low rate that it is extremely *unlikely* that a fit member of the population will *not* go through to the next generation unchanged. Unlikely, however, does not mean impossible, and over evolutionary time we may expect the population to sometimes 'fall of' the neutral network[3]. The best thing to do may be to adopt a simple steady state GA policy in which elitism comes for free. Such a policy is at the basis of the experiments performed in the next section.

3 How to *approach* ensuring that the artificial evolution of complex behaviour is viable

The last two sections have put forward several conditions that must be fulfilled if ER is to be capable of producing arbitrary levels of behavioural complexity. In order for ER to be *viable* as an alternative to other design methodologies, however, the monkeys-typing-Shakespeare levels of efficiency mentioned in section 2 are unacceptable. ER must be able to produce satisfactory results within a reasonable time period. This is by no means the only matter on which the viability of an open-ended ER rests (see [7] for issues involving real robots) but it is certainly a necessary and pressing requirement.

If the conceptual framework outlined previously in this paper is accepted, then it is clear that by far the largest portion of the evolutionary effort is spent on the process of searching neutral networks for areas of higher fitness. We have already seen in section 2.2 that it is restrictions on the encoding scheme that ensures every node on the graph of possible genotypes lies on a neutral network in the first place. This section goes further to investigate how the encoding scheme may be used, in conjunction with domain-specific heuristics, to alter the shape and nature of neutral networks in ways that vastly improve the performance of the entire evolutionary process.

3.1 How the encoding scheme may be used to favorably alter the fitness landscape

If a phenotypic property p exists such that phenotypes that are p are, on the whole, fitter than phenotypes that are $p/$ (i.e. are not p), then we may use the encoding scheme to apply this heuristic in one of two ways. Either we *bias* the encoding scheme so that there are lots of ways in which genotypes may code for p phenotypes but only a

---

[3] The biological analogue of this phenomena is known as Muller's Ratchet (see [8]). This might be one area in which artificial evolution may effortlessly improve on Natural Evolution.

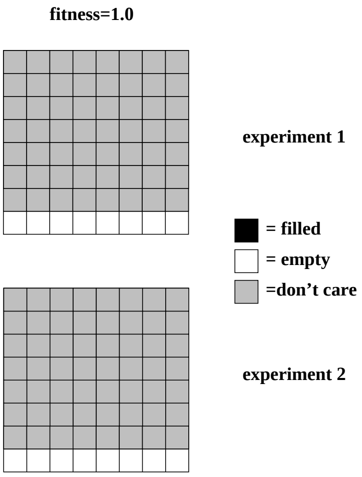


Figure 1: This figure shows phenotypic templates that define neutral networks on the graph of possible genotypes. In both experiment 1 (top pair) and experiment 2 (bottom pair), any phenotype matching the left hand template was assigned a fitness of 1.0. Any phenotype also matching the relevant right hand template was assigned a fitness of 2.0.

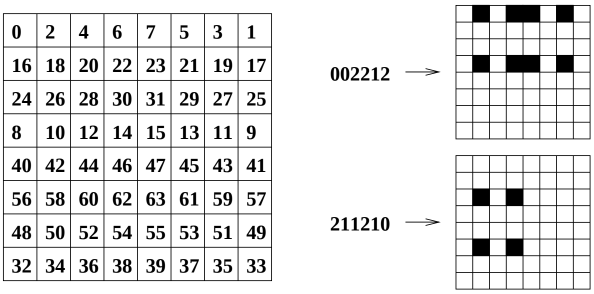
few ways in which a genotype may code for p' phenotypes, or we *restrict* the encoding scheme so that genotypes are incapable of coding for some or all of the phenotypes that are p'. The first method uniformly increases the proportion of nodes on the graph of possible genotypes that code for p phenotypes, and the second method uniformly decreases the number of nodes on the graph of possible genotypes that code for p' phenotypes. Each method has the effect of both increasing the relative frequency of nodes that code for p and increasing the likelihood of any particular node being connected to a node that codes for p. If the evolutionary process is searching a neutral network for connections to a fitter area in the fitness landscape, therefore, and the nodes that constitute that area code for phenotypes whose characteristics include p, then we expect both methods to improve the average time taken for the evolutionary process to find that area.

Despite the fact that there are almost as many encoding schemes in use as there are ER researchers, these are in fact the only two ways in which the encoding scheme may use heuristic knowledge to favorably alter the fitness landscape. As an example, let us take symmetry as the p property of phenotypes. This is highly relevant to ER since being able to act and react equally to the left and right seems an important constituent of adaptive behaviour. Under a so-called *direct* encoding scheme, the proportion of nodes on the graph of possible genotypes that code for symmetrical phenotypes will be the same as the proportion of all possible phenotypes that are symmetrical i.e. practically zero. One way in which we may remedy this situation is by increasing the length or cardinality of the genotype so that a single application of a genetic operator (such as mutation) may not only cause the right *or* the left of the phenotype to change, but may also cause the right *and* the left of the phenotype to change equally. This has the effect of increasing the number of ways that a genotype may code for a symmetrical phenotype and the benefits of biased encoding schemes, as outlined above, will apply. Another way in which the proportion of genotypes that code for symmetrical phenotypes may be increased is simply to make it impossible to encode asymmetrical phenotypes. If this is done, then the evolutionary search is wholly restricted to the space of symmetrical phenotypes.

3.2 *Testing different categories of encoding scheme against each other: experimental setup*

In order to test the relative neutral network searching abilities of different categories of encoding scheme, simple experiments were devised and performed. In all, four different schemes were tested against each other: a direct encoding scheme, a biased encoding scheme, a restricted encoding scheme and a hybrid encoding scheme which was both biased and restricted. Two different sets of experiments were performed, the first involving all four schemes, and the second involving only the biased and the direct encoding scheme. In both experiments, schemes were judged on the average time taken by a GA to search a neutral network of a particular fitness for nodes in a second fitter network. Precise details of the neutral networks and encoding schemes are given below.

Phenotypes consisted of the pattern of filled squares on an eight by eight grid. Their fitnesses were assigned according to the templates shown in figure 1. As can be seen from the diagrams, in both experiments the templates of fitness 2.0 are special cases of the templates of fitness 1.0. This is analogous to the type of incremental ER process in which



0	2	4	6	7	5	3	1
16	18	20	22	23	21	19	17
24	26	28	30	31	29	27	25
8	10	12	14	15	13	11	9
40	42	44	46	47	45	43	41
56	58	60	62	63	61	59	57
48	50	52	54	55	53	51	49
32	34	36	38	39	37	35	33

Figure 2: The numbering system used to fill in squares on the phenotypic grid. On the right are two example genotype fields from the biased encoding scheme, and their corresponding phenotypic effect

encoding scheme	*experiment 1*	*experiment 2*
direct	175220	133700
biased	1054.1	1650600
restricted	2312.4	
hybrid	404.5	

Table 1: This shows the average number of evaluations for each encoding scheme, taken over a series of 500 runs, that the GA took to find the neutral network of fitness 2.

a particular control strategy, such as obstacle avoidance using whiskers, is subsumed by another more complex control strategy, such as obstacle avoidance using infra-red sensors as well (see section 2.2). Indeed, the way in which phenotypes blindly evolve so that patterns that fit the left hand template suddenly come to fit the right hand template is analogous to how we would like the redundant connectivity of an artificial neural network, for instance, to blindly grow and evolve until it suddenly begins to display some useful functionality.

The genotypes in the direct encoding scheme consisted of a precise number n of 6-bit binary fields joined together to make a string, where n was under genetic control. In order to decode a phenotype from a genotype, a decimal value is calculated from the binary value of each field, and the relevant square filled in on the grid according to the special numbering shown in figure 2. Thus each phenotype has as many squares filled in as there are fields on the genotype.

The genotypes in the biased encoding scheme consisted of a precise number n of 6-bit *ternary* fields, where n was under genetic control. Of the three possible bit values, 0 represents 0, 1 represents 1, but 2 is a wild card character. Therefore each 6-bit ternary field maps to 2^k 6-bit binary numbers, where k is the number of 2s in the ternary field. When decoding a ternary field, a square is filled in on the grid (according to the special numbering shown in figure 2) for every binary number that matches. Two examples of decoded fields are given in figure 2. It is important to note that although the biased encoding scheme is capable of encoding any phenotype, genotypes that code for a high degree of symmetry and order will be much more common than those that don't.

The genotypes in the restricted encoding scheme consisted of a precise number n of 5-bit binary fields, where n was under genetic control. During decoding, each field is treated as if it had an extra bit of value 2 on the end, and decoded in the same way as for the biased encoding scheme. This means that for each field on the genotype a symmetrical pair of squares is filled in on the phenotype. All phenotypes are vertically symmetrical.

The genotypes in the hybrid encoding scheme consisted of a precise number n of 5-bit ternary fields, where n was under genetic control. During decoding, each field is treated as if it had an extra bit of value 2 on the end and decoded in exactly the same way as if it were a six-bit ternary field of the biased encoding scheme. This means that many more squares may be filled in on the phenotype than there are fields on the genotype and that all phenotypes are vertically symmetrical.

3.3 Testing different categories of encoding scheme against each other: experimental results

In both experiments, the GA[4] was run 500 times for each encoding scheme. On every run the population was started from the same point on the first neutral network, namely the node that corresponds to a genotype of length 0 (which represents a blank grid under every encoding scheme). On each run, the number of fitness evaluations performed before the GA found the second neutral network was counted. A table of the average scores per encoding scheme per five hundred runs per experiment is given in Table 1. Because the distributions underlying the numbers of evaluations taken approximate a Poisson distribution, the variances of the scores are of the same order of magnitude as the means, and conventional graphical representations are largely uninformative. To produce Figures 3 and 4, the scores from each set of five hundred runs were sorted in order of magnitude and plots were made of rank against value. This give a fairly good picture of the abilities of each encoding scheme to perform neutral network search. The shallower the gradient, the better.

The first thing to say about these results is that they prove quite conclusively that the choice of an appropriate encoding scheme which exploits domain knowledge in a heuristic way, does indeed significantly speed up the neutral network search space. Of the four encoding schemes tested in experiment 1 only the direct encoding scheme does *not* exploit the fact that the template for the second neutral network is symmetrical. This makes it two orders of magnitude worse than its nearest rival. The hybrid encoding scheme, which exploits the symmetry of the template in a dual way, seemed to work the best.

In experiment 2, an asymmetrical template defines the neutral network of fitness 2. Only the biased and the direct encoding schemes are capable of representing a phenotype of this type, and the experiment was performed in order to test the

---

[4] Population 100. Steady state. Tournament selection. Genetic operators included a mutation operator and a 'change length' operator capable of adding or subtracting a random genotype field. These operators were applied at a rate of *one* (of the two operators) applications per offspring event in the of ratio 4:1, respectively. Crossover was employed at all offspring events.

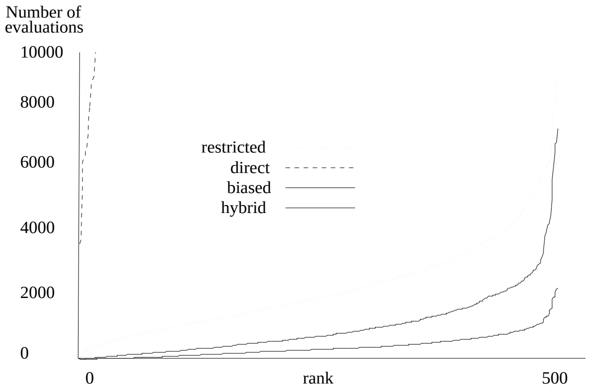


Figure 3: Results of experiment 1: the relative ability of the four encoding schemes to perform neutral network search on a symmetrical task. For each encoding scheme, the results of all five hundred runs are displayed (see text).

biased scheme's ability to search for asymmetrical phenotypes when it is biased in favour of symmetrical ones. As can be seen, the biased encoding scheme performs an order of magnitude *worse* than the direct encoding scheme on this task. This is not surprising if one considers that increasing the proportion of nodes code for symmetry on the graph of possible genotypes decreases the proportion of nodes that code for asymmetry.

3.4 The choice is restricted

No definite conclusions can be drawn from these results as to which type of encoding scheme is the most promising for a viable open-ended ER. Filling in squares on a grid is not the same as evolving complex behaviours for autonomous agents. The results are, however, extremely suggestive. Even though the performance difference between the hybrid encoding scheme and the biased encoding scheme seem slight, it is nevertheless some sort of (biased) restricted encoding scheme rather than a pure biased encoding scheme that is advocated here.

To appreciate why, consider how each type of scheme would encode something that involved lots of complex repeated structure such as a millipede, for example. A hybrid restricted scheme would encode a single leg once on a genotype and reuse this code 1000 times. Perhaps it would also code for a small number of position dependent operators, such that the front legs differed from the back legs according to some spatial transformation (see [13] for examples of this sort of thing occurring in nature). A true biased encoding scheme, on the other hand, must safeguard the possibility that any leg may evolve independently to any other, since every phenotype is potentially representable. Thus, although it could code for the millipede in a similar way to the hybrid restricted scheme (by repeating the code for 1 leg 1000 times), it must also be able, at least in principle, to code for each of the 1000 legs individually. However it is done, this will involve a longer or higher cardinality genotype resulting in larger neutral networks with a lower frequency of nodes connected to areas of greater fitness.

Many researchers will feel uneasy about the idea of adopting encoding schemes that are unable to encode all possible phenotypes. They may feel that, even if this does lead to performance increases, it amounts to prejudging the problem and prevents the evolution of unexpected solutions - precisely the reason evolution is being used in the first place. There is one further thing to be said, however, that may quell these fears. An encoding scheme that is biased for the evolution of certain traits, is biased *against* the evolution of other traits. This is clear from experiment 2. Therefore in order for a biased scheme to even come close to competing with a hybrid restricted scheme at evolving complex phenotypes (such as in the millipede example above), the level of bias would be such that the evolution of a particular phenotype which the scheme is biased *against* would take a totally impractical length of time. In fact, as can be seen from experiment 2, it would actually take many, many orders of magnitude longer to evolve than if a simple direct encoding scheme was used. Thus, by removing the possibility of representing such phenotypes, we are actually losing very little.

4 Conclusions

The first part of this paper shows how the evolutionary process underlying ER may be constrained to ensure not only that the evolution of complex behaviour is possible, but that if it is necessary for the increased fitness of the population, it *will* evolve. Section 1 showed how the correct choice of GA leads to a view of genotype space as a graph upon which arbitrary levels of complexity may be

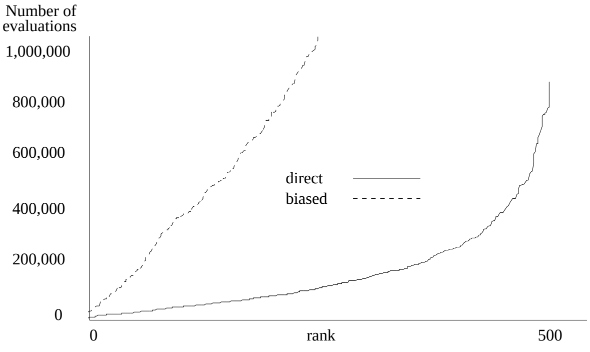


Figure 4: Results of experiment 2: the relative ability of the biased and direct encoding schemes to perform neutral network search on an asymmetrical task. For each encoding scheme, the results of all five hundred runs are displayed (see text).

encoded and searched for, and Section 2 showed how suitable restrictions to the encoding scheme result in the formation of neutral networks on this graph, thus guaranteeing the open-endedness of the evolutionary process.

Neither Section 1 or Section 2, took into account the time constraints that underly the viability of ER as a design alternative, however, and in Section 3 methods of using the encoding scheme to speed up the evolutionary process were explored. All encoding schemes were shown to belong to one of a set of distinct categories, and experiments were performed to test these categories against each other. As a result, hybrid restricted encoding schemes were advocated as the most powerful type of encoding scheme for evolving complex behaviour.

Although many ER researchers share the intuition that the encoding scheme is, somehow, at the root of the problem, there is very little consensus or theory at the basis of this intuition. It is hoped that with the aid of the conceptual framework laid out in Sections 1 and 2 the conclusions resulting from the experiments of Section 3 may be extended to provide the basis for a viable open-ended ER.

Acknowledgements

I would like to thank Phil Husbands, Inman Harvey and others at the School of Cognitive and Computer Sciences for various crucial discussions. Thanks also to the school of COGS itself for the bursary that allows me to undertake this work.

References

[1] R.D. Beer and J.C. Gallagher. Evolving dynamic neural networks for adaptive behavior. *Adaptive Behavior*, 1:91–122, 1992.

[2] D. Floreano and F. Mondada. Automatic creation of an autonomous agent: Genetic evolution of a neural -network driven robot. In D. Cliff, P. Husbands, J.A. Meyer, and S. Wilson, editors, *From Animals to Animats*, volume 3, 1994.

[3] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading, Massachusetts, 1989.

[4] I. Harvey. Species adaptation genetic algorithms: the basis for a continuing saga. In F. J. Varela and P. Bourgine, editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 346–354, Cambridge, Massachusetts, 1992. M.I.T. Press / Bradford Books.

[5] I. Harvey, P. Husbands, and D. Cliff. Seeing the light: Artificial evolution, real vision. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, 1994.

[6] John Holland. Genetic algorithms and classifier systems: Foundations and future directions. In J. J. Grefenstette, editor, *Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pages 82–89, Hillsdale, N.J., 1987. Lawrence Erlbaum Associates.

[7] M.J. Mataric and D. Cliff. Challenges in evolving controllers for physical robots. *Robot and Autonomous Systems*, forthcoming.

[8] Martin Nowak and Peter Schuster. Error thresholds of replication in finite populations, mutation frequencies and the onset of Muller's ratchet. *Journal of Theoretical Biology*, 137:375–395, 1989.

[9] C.W. Reynolds. Evolution of corridor following behaviour in a noisy world. In D. Cliff, P. Husbands, J.A. Meyer, and S. Wilson, editors, *From Animals to Animats.* MIT Press/Bradford Books, 1994.

[10] P. Schuster. Artificial life and molecular evolutionary biology. In *Proceedings of the European Conference on Artificial Life*, pages 3–19. Springer-Verlag, 1995.

[11] H.T. Siegelmann and E.D. Sontag. On the computational power of neural nets. *Journal of Computer and System Sciences*, 1995.

[12] N. C. Stenseth and J. Maynard Smith. Coevolution in ecosystems: Red queen evolution or stasis? *Evolution*, 38:870–880, 1984.

[13] D.W. Thompson. *On Growth and Form.* Cambride University Press, 1942.