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 beha(GA)

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 viaproach should not constitute the primary focus of enquiry.

2.2 Neutral network

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.

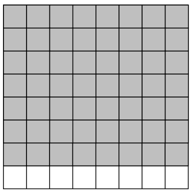
[2]monotonically increasing means never going down, not always going up.

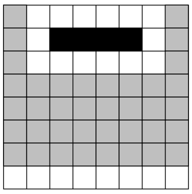
2.3 How a imple GA would work in the context of neutral network

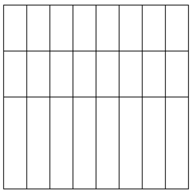
So far, no explanation has been given of *how* an open-ended GA would work. This is because, in order to give a satisfactory account, some knowledge of the nature of the search space is required. There is no point, for instance, in spending time explaining and ensuring how an open-ended GA will never 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.

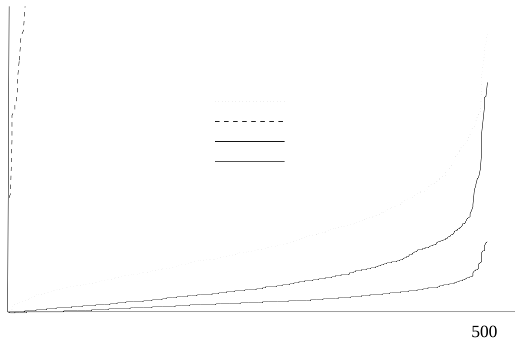
If the fitness landscape consists of neutral networks connected together by slopes, we certainly require the GA behind open-ended artificial evolution to perform hill-climbing type search. This is only half the story, however. It is important to realize that as well as being connected to nodes of equal 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








500

encoded and searched for,