

Artificial Evolution: A Continuing SAGA

Inman Harvey
Centre for the Study of Evolution
Centre for Computational Neuroscience and Robotics
School of Cognitive and Computing Sciences
University of Sussex
Brighton BN1 9QH, UK
inmanh@cogs.susx.ac.uk

Abstract

I start with a basic tutorial on Artificial Evolution, and then show the simplest possible way of implementing this with the Microbial Genetic Algorithm. I then discuss some shortcomings in many of the basic assumptions of the orthodox Genetic Algorithm (GA) community, and give a rather different perspective. The basic principles of SAGA (Species Adaptation GAs) will be outlined, and the concept of Neutral Networks, pathways of level fitness through a fitness landscape will be introduced. A practical example will demonstrate the relevance of this.

1 Artificial Evolution

Every day we come across sophisticated, highly-tuned machinery that is far far more complex than human designers could begin to imagine designing by standard techniques. I am referring to the animals (including other humans), plants and other organisms that we live amongst. These are self-regulating, mostly self-repairing, self-sustaining machines that can cope with changing situations in an incredibly flexible and adaptive fashion. Their designs are the product of billions of years of natural Darwinian evolution.

Proponents of Artificial Evolution aim to capture and exploit the core parts of this natural design methodology, and use it to design artificial complex systems to have comparable properties: adaptive and robust robot

control systems, self-repairing electronic circuits, telecommunications networks that grow and rearrange themselves around disruptions, pharmaceutical drug molecules that match up with a range of targets. We do not have the resources of billions of years of experimentation on one or more planets that Natural Evolution has had, so we must be as efficient as possible, and learn the crucial tricks that Nature can show us.

The context of evolution is a population (of organisms, objects, agents ...) that survive for a limited time (usually) and then die. Some produce offspring for succeeding generations, the 'fitter' ones tend to produce more than the less fit. Over many generations, the make-up of the population changes. Without the need for any individual to change, the 'species' changes, in some sense adapts to the prevailing conditions. There are three basic requirements for Darwinian evolution by Natural Selection:

1. **Heredity:** Offspring are (roughly) identical to their parents ...
2. **Variation:** ...except that they are not exactly the same
3. **Selection:** The 'fitter' ones are more likely to have more offspring than the 'unfit' ones

Variation is usually random and undirected, whereas Selection is usually non-random and in some sense directed. In the natural world, direction does not imply a conscious director. Rather, it reflects the fact that those organisms that are not as well designed for their particular ecological niche as their conspecifics will be less likely to survive and have offspring; the others thereby automatically qualify as 'fitter' for that particular niche — whatever that niche might be. If antelopes are often chased by lions, then it is reasonable to talk of Selection providing a selective pressure for a population of antelope to increase their speed over successive generations, other things being equal.

In Artificial Evolution, unlike Natural Evolution, the human experimenter decides what is going to count as 'fit', in what direction Selection should alter the population over generations. In this sense it resembles agricultural practice, where for thousands of years farmers have been selecting the cows that produce more milk, the crops that produce more grain, and breeding from them for future generations. Even without necessarily understanding the genetic details, the manipulation of DNA underlying the process, farmers have long implicitly understood the basic principles of Heredity, Variation and Selection sufficiently well to improve their crops over the centuries.

2 DNA

As we now know (but Darwin did not), a core mechanism underlying Heredity and Variation is the DNA that we (and other organisms) inherit from our parents and pass on to our offspring. DNA is often treated as though it is a ‘blueprint’, or a set of instructions setting out how an organism will develop from an initial single cell. Many biologists would say that this view of DNA is in important respects misleading; however, in Artificial Evolution, where we can pick and choose those biological ideas that suit us regardless of whether they give the whole biological picture, we typically do indeed take this simplistic view of Artificial DNA as a blueprint. The crucial aspects of DNA that we borrow for our own purposes are:

1. DNA can be treated as a meaningless string of symbols — Cs Gs As and Ts in humans, perhaps 0s and 1s in a Genetic Algorithm (GA) — that are just mindlessly copied to provide Heredity; perhaps with occasional copying errors to provide Variation.
2. The genotype, the full sum of DNA that an organism inherits, has a crucial role in determining the phenotype, the form and the physical and behavioural traits of an organism.

So to give a simple illustration of Artificial Evolution applied to finding a good design for a paper glider, one could invent a set of symbols that specified how a piece of paper, initially square, is folded. For example, A could mean ‘fold the paper towards you about a vertical line through the middle’; B could mean ‘fold the paper away from you about a diagonal line from NE to SW’. An appropriate set of such symbols could cover all the possible standard folding moves, and any particular list of such symbols, e.g. GABKJNPD, can be used in each of the two ways listed above: firstly, as a string of symbols that can be mindlessly copied and passed on, secondly as a blueprint detailing the successive folds that turn a plain sheet of paper into some folded object.

The person who wants to design a paper glider using artificial evolution would then start with perhaps 30 sheets of paper, and write on each piece a random sequence of the symbols. Then she would take each piece of paper in turn, interpret the symbol string, the artificial DNA, as instructions to fold the paper, and see what shape results. The next step is to open a window high up in a building, and throw all 30 folded shapes out of the window.

She would then go outside and see how the different shapes have fallen to the ground below the window. Some may have fallen straight down,

some may have accidentally been caught by some wind, some shapes may have possibly glided a metre or two. This is where Selection comes in, and the ones that have not travelled far are discarded while the one that went furthest are chosen to form the parents for the next generation. A new set of 30 sheets of paper is prepared, and strings of symbols, of artificial DNA, are copied onto them based on the surviving parents from the previous generation. This can be done in a variety of ways, any of which are likely to work.

The simplest option might be the asexual one, in which perhaps the best 50% of the previous generation each have 2 offspring, who inherit their single parent's DNA with some small probability of a mutation altering, deleting or adding a symbol. Alternatively, a form of sexual reproduction can be used, wherein the parents are brought together in pairs, and their offspring inherit some DNA from each parent, again with the possibility of further mutations. As long as the method chosen maintains the population of the next generation at the same size as the same as the initial generation, and obeys the rules for Heredity and Variation, then the stage is set for a further round of Selection on the new generation. Continuing this over many successive generations should result in increasingly successful paper gliders that fly further and further out of the window.

You can change the problem to that of designing real aircraft wings, or control systems for robots; and you can change the set of symbols to a new set specifying the curvatures and thicknesses of parts of a wing, or the type and connectivity of artificial neurons in an artificial neural net. Then the underlying methodology of Artificial Evolution will basically remain the same, except that the Selective process, the evaluation of the fitnesses of different members of the population, is likely to be far more expensive than throwing paper gliders out of the window.

When you change to a different problem, you have to create a new and appropriate method for interpreting strings of symbols, the artificial DNA, as potential solutions to the problem. For some problems it may be appropriate to use real-valued numbers as symbols in the DNA, in which case there is a potentially infinite range of values at such a locus on the genotype. In the work discussed from here on, however, it is explicitly assumed that, as in natural DNA, there is only a limited range of symbols, quite possibly limited to the binary range of 0 and 1.

3 The Microbial Genetic Algorithm

There are many varieties of Evolutionary Algorithms, many different ways to implement, for problem solving, the three main requirements of Heredity, Variation and Selection. I shall now describe one little known but effective method, that is so simple to implement that the core of the program can be reduced to a single line of code. I call it the Microbial Genetic Algorithm because it is loosely based on the way microbes can exchange genetic material, DNA, ‘horizontally’ between different living members of the population as an alternative to ‘vertically’ from one generation to the following one.

There are three particular tricks used here that are subtly different from the basic algorithm described above in the paper gliders example. The first is the use of a ‘Steady State’ method rather than a ‘Generational’ method. Instead of accumulating a complete new generation of offspring, and then discarding the older generation and replacing it wholesale by the new, it is very reasonable to just generate a single new offspring at a time; then (in order to maintain the population size constant) choose one member of the population to die and be replaced by the new one. The Selection criterion will be satisfied by either biasing the choice of parent(s) for the new offspring towards the fitter members, or biasing the choice of which is to die towards the less fit. There are at least two advantages of the Steady State method over the generational method: it is usually much easier to implement, and it allows for efficient parallel implementations where it is not actually necessary to keep the evaluations of all members of the population in step with each other. Despite the fact that the generational method is usually the first to be discussed in the textbooks, these advantages mean that many serious users of evolutionary algorithms favour the Steady State method.

The second trick is to use a rank-based method of selection, and in particular tournament selection. The textbooks generally present ‘fitness-proportionate’ selection (where for instance if one member has twice the fitness of another member of the population it can expect twice as many offspring) as the main method used in GAs. This is probably for historical reasons, and because the formal analysis of GAs is mathematically easier when using this method. However, professional practitioners are far more likely to use a rank-based selection method, where the expected number of offspring of any member is based on (in the simplest case, linearly proportionate to) its ranking in the population. To give a simple example with a population of size 5, they can be ranked in order on the basis of their fitness and then allocated an expected number of offspring in this ratio: $4/2$ $3/2$ $2/2$ $1/2$ $0/2$. In this fashion the top-ranking member will have twice the

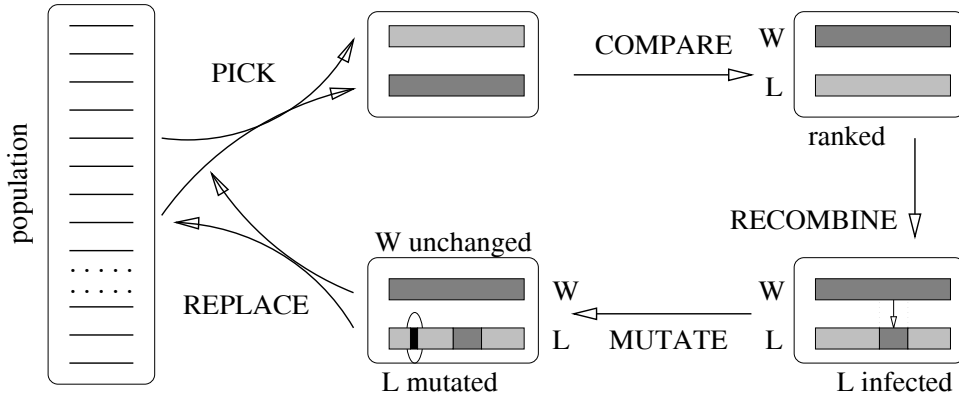


Figure 1: *A single tournament in the Microbial Genetic Algorithm.*

expected number of offspring of the middle-ranking member, irrespective of whether it is 100 times fitter or only 1% fitter.

A cheap and cheerful method of implementing this type of rank-based selection, particularly appropriate for the Steady State case, is to pick out 2 members of the population at random and compare their fitnesses in a ‘Tournament’. Then picking the winner to be a parent (or alternatively, picking the loser to be the individual that dies to make way for a new offspring) gives exactly the same expected selection bias as described in the previous paragraph. There are at least three advantages of this Tournament Selection method over the orthodox fitness proportionate selection method: it is usually much easier to implement, it avoids many scaling problems of the standard method, and it implements a form of elitism for free. Elitism in this context means that the currently fittest member of the population will always remain preserved unchanged.

Now we build on these two tricks by moving on to the third trick of the Microbial GA. It is perfectly acceptable to operate a GA by picking two members at random to be parents and generate a new offspring; and then pick a further two members at random, and using Tournament Selection choose the loser to die and be replaced by the new one. It may seem initially strange to have no bias towards choosing fitter members as parents, but the bias in choosing who is to die is what satisfies the criterion of Selection. The trick here is to combine all this into one operation.

So the Microbial method is to pick just two members of the population at random, who will be parents of the new offspring; and the least fit of the two parents is chosen as the one to die and be replaced. I have used so far the conventional language of ‘parent’, ‘offspring’ and ‘die’, but in fact this is equivalent to horizontal transmission of genetic material from the

‘Winner’ of the tournament to the ‘Loser’. The Winner remains unchanged in the population, and the Loser receives copies of some genetic material (not necessarily restricted to 50%) from the Winner, with the opportunity for further mutations also.

The Microbial Genetic Algorithm is illustrated in the diagram, where the population of genotypes of ‘artificial DNA’ is represented by the set of lines on the left. Initially these will each be a random string of symbols, for instance a random binary string, and there will be some method for translating any such string into a trial solution for the design problem being tackled. This is where the human designer has to be creative in matching the genotype-to-phenotype translation to the requirements of the task. But then, provided that there is a suitable method for testing and scoring any such potential solution, giving it a ‘fitness’, all the rest of the work can be left to the algorithm. Two strings are picked out at random, and evaluated to see which is the Winner and which the Loser (W and L on the diagram). Then with some probability each locus (genotype position) of the Winner may be copied over the corresponding locus of the Loser, followed by a separate mutation process of changing at random some small proportion of the Loser loci. The two strings are re-inserted into the population — in fact the Winner is unchanged.

This Microbial GA obeys the 3 rules of Heredity, Variation and Selection, is effective, yet is so simple that it can be reduced to a single line of code. If we assume that, in C , the genotypes are in a binary array $gene[POP][LEN]$, and that the function $eval(i)$ returns the fitness of the i^{th} member of the population, the one-liner goes something like this:-

```
for (t=0;t<END;t++)
  for (W=(eval(a=POP*rand())>eval(b=POP*rand()))?a:b),
    L=(W==a?b:a),i=0;i<LEN;i++)
      if ((r=rand())<REC+MUT)
        gene[L][i]=(r<REC ? gene[W][i] : gene[L][i]^1);
```

4 Searching through Fitness Landscapes

Evolutionary algorithms, including the Microbial GA, can be thought of as search methods in a high-dimensional search space. Turning back to the paper glider folding example, if there are 8 possible folding instructions, and a succession of 25 folds, then there are 8^{25} possible versions of folding a glider. Only a tiny proportion of these will have any sort of flying ability, and an even smaller proportion will fly properly. If one considers all the 8^{25}

designs as spread out over a landscape, with similar designs (differing by say just one fold) nearby to each other, then one can imagine the search process as searching across this landscape. Now treat the ‘fitness’ of each possible design as the ‘height’ of the corresponding position in this landscape, and we have a hilly fitness landscape where the peaks represent our goal. Typically the majority of this landscape will be foothills of negligible height, but it is reasonable to expect that the higher mountains form connected ranges that are the areas on which the search should be focused.

There are many possible strategies for searching such fitness landscapes, including Simulated Annealing, Hill-Climbing, Tabu Search. Since the search spaces are too big to search exhaustively, then all search methods involve sampling in turn successive points, checking their fitnesses and using this knowledge to guide the continuation of the search from what has been explored so far. The distinctive feature of evolutionary approaches such as genetic algorithms is the use of a *population* of search points, searching in parallel although not independently.

5 Conventional Genetic Algorithm Assumptions

If you read the Genetic Algorithm textbooks, you will find (explicitly or implicitly) a number of assumptions as to what makes the GA work effectively. One of the major worries is that of getting stuck on a local optimum in a fitness landscape; indeed this is the reason that most people are sceptical about simple Hill-Climbing methods.

It is generally assumed that GAs make a good effort to avoid getting trapped on such local optima through two properties. Firstly, by starting with an initial randomly spread population, there is more chance that the foothills to many different ranges will be encountered. The parallel population search will be eventually won by those climbing the mountain range that turns out to be the highest of those seen, and there is less chance of being trapped in one of the lower ranges.

Secondly, when there is recombination between different members of the population, this means that the searches are not truly independent. Even if two different members are in effect trapped on separate foothills (or local optima), then their offspring will, through recombination, occupy a new point on the fitness landscape somewhere that is in effect in-between these foothills. Hence such an offspring could escape from the local traps that each of its parents might be in.

From these ideas flow some further assumptions, widespread in the GA

literature, that I shall argue are completely misleading. One major, and mistaken, worry is about ‘premature convergence’. If you follow the above intuitions about how a single member of the population may get trapped in a local optimum, then you *need* a widely varied population to avoid this problem. Once all the variation in the population has disappeared over time, so that it is in effect multiple copies of the same individual, then it can get stuck on a local optimum however big the population of clones is. Unless new variation is injected into the population, then this genetic convergence will happen; and if it happens before the global optimum has been found, then this is the disaster of so-called ‘premature convergence’. I shall give a different picture below.

The GA textbooks usually present a theorem derived by John Holland, the architect of GAs, called the *Schema Theorem*. This proves that under specific limited circumstances the ‘useful parts’ of genotypes in the population will grow exponentially as the GA produces the next generation from the current one. Although formally correct, it is usually misinterpreted as if this exponential growth continues unchecked over successive generations, whereas in fact it is only valid for a single generation; the calculations of fitnesses within the population have to be done afresh each time. So although the Schema Theorem is formally correct, pragmatically it is useless and irrelevant.

The Schema Theorem is associated with a commonly held dogma in the GA community, that *recombination*, the mixing and matching of various parts of the genotype from different parents to produce the offspring, is the driving force of evolutionary search. This feeds back to the worries about premature convergence spelled out above. There is an alternative viewpoint, however. Those who advocate the evolutionary methods of Evolutionary Programming (or EP) tend to emphasise the role of *mutation* rather than recombination, and for slightly different reasons so do I here.

6 SAGA: Species Adaptation Genetic Algorithms

In the natural world, of course, evolving populations are genetically highly converged. If this was not so, then the task of the Human Genome Project, assembling the genotype of a typical human being, would be pointless. The genetic differences between two human beings (or two members of any other species) are of course significant. They lie behind the subtle differences of human form and behaviour, of eye colour and temperament; they allow the possibility of DNA identification. But these differences are tiny compared

to the similarities, to what makes an identifiable and coherent *species*.

Each and every species on this planet probably shares a common origin some 4 billion years ago. From this origin of life, variations have branched out with many such branches terminating, as species become extinct. But if we imagine following the historical trace of a currently-existing species, such as humans, we will find that for some 4 billion years there has been the phylogenetic pathway of a population changing from a single cell to the complex creatures we are today. At every point in this history, this population would have been genetically very converged, the genetic differences between individuals would have been minimal compared to their similarities.

In thought experiment at least, we could imagine this historical trace represented by a single individual from each generation, displaying our phylogenetic history. This history would be one of long-term change almost entirely through mutation; the interesting possible exceptions being when transfer of genetic material between species may occasionally bring together branches of the Tree of Life after they have previously bifurcated. So apart from such exceptions, all the accumulated design of a present day organism has come through the occasional lucky mutations that have been incrementally incorporated. Whatever the role of recombination might be in natural evolution — and the jury is still out on this question — it is mutation that is the driving force.

With this in mind, some ten years ago I started to develop a framework using similar ideas for long-term artificial evolution, for the incremental design methodology needed for such tasks as Evolutionary Robotics. This I call SAGA, or Species Adaptation Genetic Algorithms, as a genetically converged population, in effect a species, is involved. The early stages of SAGA (Harvey, 1992; Harvey, 1993) were based on the realisation that if long-term evolution meant that genotype lengths were initially relatively small (for encoding e.g. relatively simple robot control systems) and then slowly increased in length over generations to accommodate more complex designs as evolution progressed, then it was inevitable that the population would be genetically converged throughout. But later it came to be recognised that actually even with genotype lengths remaining constant, in practice an evolving population is genetically converged in any case. So it has turned out that SAGA ideas are far more widely applicable than their original domain.

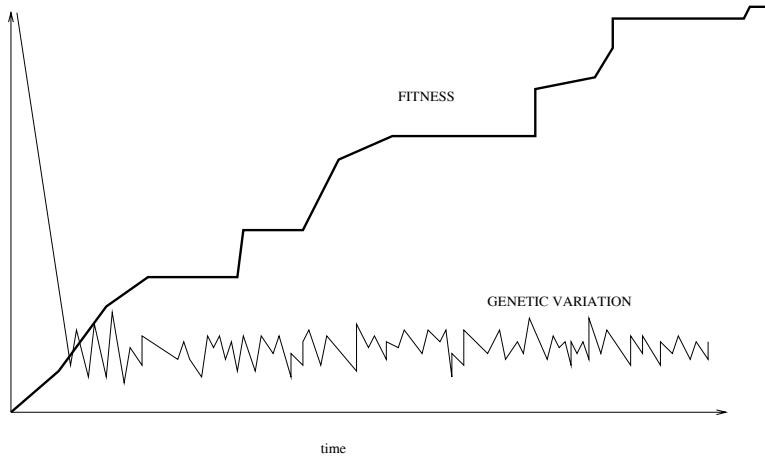


Figure 2: *Fitness typically continues to increase after the genetic variation has settled down.*

7 Evolving a Genetically Converged Population

In artificial evolution, typically one starts with a randomly dispersed initial population, sampling widely across the search space. One useful measure of the genetic variation in a population is the average Hamming distance between two members; the average number of places on the genotype where there genetic material is different. For a random population of binary genotypes, any two will differ in about half the positions, but as successive rounds of selection discard the less fit and concentrate the population around the fitter ones, this variation will dramatically decrease. If one plots this genetic convergence in a practical example of artificial evolution, then one sees just how surprisingly fast this occurs; even with a large population, most of the variation will have disappeared within 10 generations or so.

Yet if one also plots the fitness of the population over time, one can see that the fitness continues to increase after the genetic variation has been reduced to its (noisy) minimum range of values.

In the absence of any mutation, selection will concentrate the population at the current best. The smallest amount of mutation will hill-climb this current best to a local optimum. As mutation rates increase, the population will spread out around this local optimum, searching the neighbourhood; but if mutation rates become too high then the population will disperse completely, losing the hill-top, and the search will become random. If a balance is achieved between selective forces and those of mutation (as modified by recombination), then some elements of the population can crawl down

the hill far enough to reach a ridge of high selective values. As discussed in (Eigen McCaskill and Schuster, 1988), this results under selection in a significant proportion of the population working their way along this ridge, and making possible the reaching of outliers further in Hamming-distance in that particular direction from the current fittest. The term ‘ridge’ is used here to fit in with intuitive notions of fitness landscapes; in fact in high-dimensional search spaces such ridges may form complex neutral networks, percolating long distances through genotype space.

If any such outliers reach a second hill that climbs away from the ridge, then parts of the population can climb this hill. Depending on the difference in fitness and the spread of the population, it will either move *en masse* to the new hill as a better local optimum, or share itself across both of them.

So in a SAGA setup of evolution of a converged species, we want to encourage through the genetic operators such hill-crawling down towards ridges to new hills, subject to the constraint that we do not want to lose track of the current hill. Eigen and co-workers use the concept of a quasi-species to refer to a similar genetically converged population in the study of early RNA evolution. To quote from (Eigen McCaskill and Schuster, 1988):

In conventional natural selection theory, advantageous mutations drove the evolutionary process. The neutral theory introduced selectively neutral mutants, in addition to the advantageous ones, which contribute to evolution through random drift. The concept of quasi-species shows that much weight is attributed to those slightly deleterious mutants that are situated along high ridges in the value landscape. They guide populations toward the peaks of high selective values.

8 SAGA and Mutation Rates

Although progress of a species through a fitness landscape is not discussed in the standard GA literature, in theoretical biology there is relevant work in the related field of molecular quasi-species (Eigen, 1987; Eigen McCaskill and Schuster, 1988). In particular, analysis of ‘the error catastrophe’ shows that, subject to certain conditions, there is a maximum rate of mutation that allows a quasi-species of molecules to stay localised around its current optimum. This critical maximum rate balances selective forces tending to increase numbers of the fittest members of the population against the forces of mutation that tend, more often than not, to drag offspring down in fitness away from any local optimum. But a zero mutation rate allows for no further local search beyond the current species, and other things being

equal increased mutation rates will increase the rate of evolution. Hence if mutation rates can be adjusted, it would be a good idea to use a rate close to but less than any critical rate that causes the species to fall apart. A further possibility, in the spirit of simulated annealing, is to temporarily allow the rate to go *slightly* above the critical rate — to allow exploration — and then cut it back again to consolidate any gains thus made.

For an infinite asexual population, it can be shown (e.g. in (Eigen, 1987)) that these forces just balance for a per-genotype mutation rate m equal to the logarithm of σ ; where σ is the *superiority* parameter of the fittest member of the population — the factor by which selection of this sequence exceeds the average selection of the rest of the population. Recombination makes some degree of difference (Ochoa, Harvey and Buxton, 1998), but the end result stays in the same general area. The rule of thumb is that if the selection pressure used is that associated, for instance, with the Microbial GA using tournaments of size 2, the optimal mutation rate is in the region of one mutation per genotype, after taking account of any junk or neutral mutations (see below). In other words, the rate should be set so as to expect around one fitness-altering mutation in the whole genotype; if, for example, 50% of the genotype is redundant or neutral, such that mutations in those regions make no difference, then a rate of 2 mutations per complete genotype gives an expected one mutation in the non-redundant part.

When applying such mutation rates in a GA, it is essential that the probability of mutation is applied independently at each locus on the genotype. This gives a binomial distribution (approximating a Poisson distribution for long genotypes) for the number of mutations per string, so that genotypes with an expected m mutations have this as the average value with a wide variance (including the possibility of zero mutations).

9 Neutral Networks and Drift

Mutations in a genotype encoding a fit phenotype are often deleterious, and occasionally advantageous. There is a third possibility, that a mutation is neutral and leaves the fitness unchanged.

Neutral mutations can in turn be subdivided into two kinds, with a rather grey area between them. They can be in parts of ‘junk DNA’, such that the decoding of the genotype ignores the values in that part. In this case it is only the functional part of the genotype, the part that is capable of causing some difference in the fitness, that counts towards effective genotype length when deciding upon mutation rates. For example, if a genotype of length 1000 is 90% junk, then a mutation rate set at the rate of one per

effective genotype length should be implemented at the rate of 1/100 per locus, rather than 1/1000. It is often difficult to estimate what proportion of a genotype is junk, however, as this shades into the second class of neutral mutation.

This second type of mutation may leave the phenotype unchanged, yet open the possibility of a further mutation making some difference. At its simplest level, a binary genotype with two loci, whose fitness is given by the logical **AND** of the alleles at each locus, retains a fitness of 0 during mutation from **00** to **01**; yet this opens up the possibility of a further single point mutation reaching **11**, with a fitness of 1 which was not achievable from the starting point. Such neutral mutations can in a high-dimensional space allow extended neutral paths that can percolate through vast areas of sequence space. Neutral drift of a population through such pathways means that it is much more difficult than one might think to get stuck on a local optimum. In addition, the percolation of such paths through sequence space tends to mean that it does not matter too much where in sequence space a converged population starts; under many circumstances it is possible to reach all possible fit regions from most starting points.

The SAGA selection and mutation rates encourage just such exploration through neutral drift in sequence space.

10 Recombination

With a genetically converged population, sections of genotype that are swapped in recombination are likely to be fairly similar. With species evolution recombination does not have the prime significance it has in standard GAs — asexual evolution is indeed feasible — but nevertheless it is a useful genetic operator.

There are two roles recombination has which are opposite sides of the same coin. On the one hand, it allows two fortunate mutations that happen to have occurred independently in two different lineages within the population to be combined into one which has both; something not possible with asexual reproduction. On the other hand, it allows parents with a detrimental mutation to produce an offspring which does not have it; also impossible asexually, in the absence of highly improbable back-mutations. This latter effect in general allows higher mutation rates to be used with recombination than were suggested above for asexual populations, thus promoting exploration without risking loss of a currently achieved local optimum.

Recombination is particularly powerful when combined with a distributed GA. Here each member of the population is allocated a different position

in some notional geographical space, often a two-dimensional toroidal grid. Recombination between individuals is only allowed for pairs within a certain distance of each other on this grid, which thus comprises a number of overlapping neighbourhoods. This combines the virtues of small and large populations; small interrelated local populations allows through random drift more extended search through genotype space, but the overlapping nature of such localities means that any improvement found percolates through the whole population.

11 But does it work?

The SAGA approach to artificial evolution assumes that evolution can incrementally improve the fitness of a population over the long term, despite the population being genetically converged. This requires the existence of ridges or neutral networks, to avoid getting trapped on local optima. Now clearly many fitness landscapes do not have these useful properties of neutral networks, of escape routes; indeed, almost all the benchmark problems and test suites in the GA literature do not have such useful properties. So why should one expect a difficult practical problem to have such neutral networks?

Here we should start by appealing to some mathematical intuitions, before progressing to look at an actual practical example. Suppose we have a problem encoded with binary genotypes of length 1000, so the genotype search space has 2^{1000} points and is impractically large to search exhaustively. If the problem is to find appropriate robot behaviours, and a genotype specifies a robot control system, then typically there will be *very* many different genotypes that will produce the same phenotype (or robot behaviour). For the sake of an example, let us suppose that the genotype is 50% redundant, implying that there are 2^{500} different phenotypes; and each of these can be specified, on average, by 2^{500} different genotypes. One can think of this as a 500-dimensional phenotype space mapped into a 1000-dimensional genotype space.

If there is some degree of correlation, some tendency for similar phenotypes to map into similar genotypes, then one can expect the genotypes corresponding to some specific phenotype P to be partially correlated, and indeed largely connected in genotype space; in fact to form a neutral network. And it is entirely reasonable to expect some form of such correlation in any practical problem, because otherwise such problems would be completely intractable.

These are the intuitions, but in at least one example this has been tested

in practice. Adrian Thompson has pioneered intrinsic Hardware Evolution at Sussex (Thompson, 1998), using artificial evolution to design electronic circuits for pattern recognition tasks on reconfigurable silicon chips. The evolutionary method used was based on SAGA principles outlined above. In one experiment he deliberately set out to test the hypothesis that one can reasonably expect there to be pathways through genotype space to a (near-) perfect solution, that do not get trapped in local optima. With this in mind, he used a population size one in a form of Hill Climbing.

A genotype specified the current population, and we can simplify the description somewhat to say that this was effectively a binary genotype of length 1900, in other words a genotype search space of 2^{1900} . At each step in the search process, a minimal mutation was applied to the genotype, and the new fitness compared to the previous one. If the mutation resulted in a decreased fitness, that step was abandoned; but if fitness either increased *or* remained the same, the step to the new mutant was taken. This means that the eventual pathway seen through genotype space consisted of only upward or horizontal steps in the fitness landscape; in practice far more horizontal ones than upward. In fact the fitness graph took the typical form of punctuated equilibria.

The mutational steps were not actually single mutations. On SAGA principles, the mutation rate should be set so as to generate an expected one fitness-altering mutation per genotype. It was already known, from previous experiments, that around 2/3 of the genotype was redundant, so that a mutation rate of around 3 mutations per genotype could be expected to produce on average one fitness-altering one. A further complexity that we need not pursue further here was that fitness measurements took place on the real silicon chip, with the inevitable noise associated with real physical processes; this meant that care was needed in taking re-evaluations of the fitness when it was felt that a previous one may have been, through noise, misleading.

As can be seen from the fitness graph, there was indeed a connected pathway from the low-fitness starting point to the final good solution. Much of the time was spent on the level sections where fitness basically remained constant (within the limits of noise and measurement). Now possibly this time was wasted, merely mutating redundant parts of the genotype until a lucky break was hit upon. Thompson set out to test whether this was the case by looking at what happened as the search travelled along a specific plateau.

This plateau started at generation 13000, and continued for over 3000 successive 'accepted mutations' until between generation 16144 and 16145 a new fitness-increasing mutation was found. Along the plateau there was

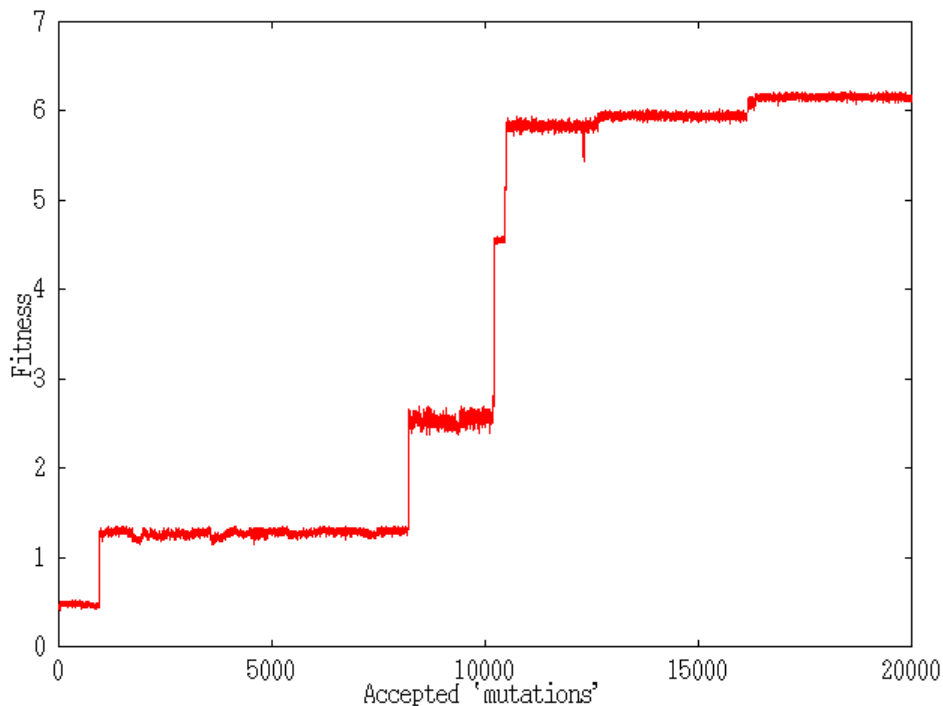


Figure 3: *The graph of fitness rising over successive mutational steps, with plateaus or ‘equilibria’ punctuated by rises in fitness.*

clearly a lot of genetic change, and it was possible to investigate the associated phenotypes, or useful functional parts of the genetically specified electronic circuits. In the diagrams shown, the irrelevant parts of the circuits are not shown, but one can see significant changes in the functional part over these 3000 or so generations. Despite the change in circuits, their behaviour, as tested directly on the silicon chip faced with a signal recognition task in real time, was unaltered.

The plateau ended with a single lucky mutation that increased fitness, so the interesting question was posed: would that specific mutation have been as lucky if it had occurred earlier on in that plateau? The mutation was applied to generation 13000, and it was observed that fitness fell significantly as a result, rather than increased. Further experiments showed that there was indeed no possible lucky single mutation from generation 13000, and from this we can deduce that indeed the drift along the neutral network was indeed beneficial. By analogy, if we wish to reach the top of a multi-storey building, then walking along a level corridor may not immediately gain us height, but will still be beneficial if it leads us to a distant upwards stairway.

This experiment gives support to the conjecture that in some real prob-

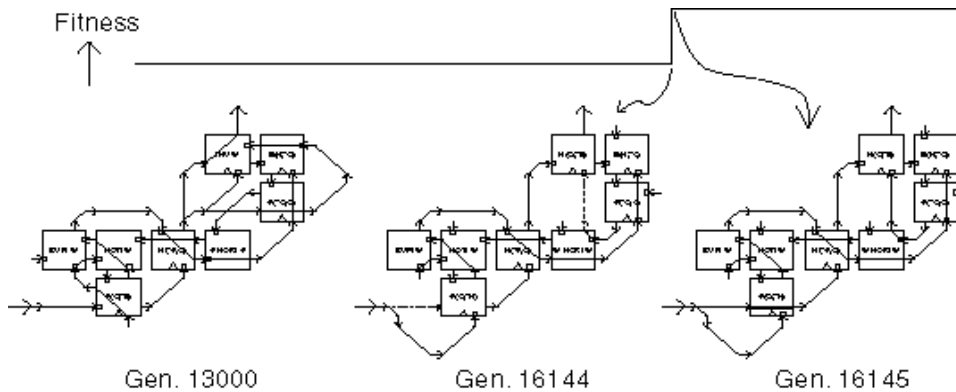


Figure 4: *Thompson's experiment to seek evidence for a neutral network. The phenotypes, or functional electronic circuits, at the beginning and end of a plateau; and immediately after the mutation that caused a rise in fitness.*

lems with very high-dimensional search spaces, using a genetic code with a discrete alphabet of symbols, there may indeed be useful neutral networks that assist in avoiding getting trapped in local optima.

12 Conclusions

I have given a very basic sketch of the underlying principles of artificial evolution, and illustrated how simple this can be by demonstrating the very minimalist Microbial Genetic Algorithm. I have argued for a rather different perspective on GAs than that conventionally presented. One should expect evolution to proceed even when a population is to a great extent genetically converged. SAGA principles suggest that mutation rates should be tuned to give an optimum amount of genetic convergence, arising from a balance between the inward forces of Selection, and the outward forces of mutation that add Variation. These principles have a wide applicability.

There are conventional worries about genetic convergence, often phrased in terms of 'premature convergence'. There are good reasons for believing that these worries are ill-founded in some very high-dimensional search spaces. Neutral Networks offer potential escape routes, and an example shows how this can work in practice.

Acknowledgments

I thank colleagues in COGS at the University of Sussex for fruitful discussions and collaborations on issues related to this paper.

References

- Eigen, M. (1987). New concepts for dealing with the evolution of nucleic acids In *Cold Spring Harbor Symposium on Quantitative Biology*, vol. LII, 1987.
- Eigen, M., McCaskill J., and Schuster, P. (1988). Molecular quasi-species. *Journal of Physical Chemistry*, 92:6881-6891, 1988.
- Harvey, I. (1992). Species Adaptation Genetic Algorithms: A basis for a continuing SAGA In *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, F.J. Varela and P. Bourguine (eds.). MIT Press/Bradford Books, Cambridge, MA, 1992, pp. 346-354.
- Harvey, I. Evolutionary robotics and SAGA: the case for hill crawling and tournament selection. In Langton, C. (ed.), *Artificial Life III, Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XVI*, pp. 299–326. Addison Wesley.
- Ochoa, G., I. Harvey, I., and H. Buxton, H. Error Thresholds and their Relation to Optimal Mutation Rates. In *Proceedings of the Fifth European Conference on Artificial Life (ECAL99)*, D. Floreano, J-D. Nicoud, F. Mondada (eds). Springer-Verlag, 1999.
- Thompson, A. Hardware Evolution: Automatic design of electronic circuits in reconfigurable hardware by artificial evolution. Distinguished dissertation series, Springer-Verlag, 1998.