The Use of Linkage Learning in Genetic Algorithms

David R. Newman
drn101@ecs.soton.ac.uk

Abstract— **This report investigates the use of Linkage Learning in Genetic Algorithms (GAs) to try and determine why it is worthwhile using this technique to improve the power of GAs.**

This report first introduces GAs and how they go about converging on optimal solutions with comparison to other search algorithms. Then it discusses what linkage learning is and several of the techniques used to perform linkage learning. The GA from 'Learning Linkage' [1] is then analysed. A re-implementation of the analysed GA is then tested, modified and experimented on to gain greater understanding of how the GA learns linkage and what are the motivating factors behind the improvement in linkage.

From experimentation on the re-implementation it is concluded that performing crossover using recipient genomes that are the complement to the optimal building block and the truncated selection of only optimal building block donors are two of the factors that motivate improvement in linkage. Leading to the conclusion that Harik's linkage learning GA (LLGA) may not be able to learn both linkage and fitness at the same time. Further experimentation, which tests for both improvement in linkage and fitness is suggested, to support the conjecture that Harik's LLGA cannot learn both linkage and fitness simultaneously. Finally the conclusions from the experimentation are combined with the research on other GAs to answer the following questions:

1) **How can functionally dependent genes be aligned adjacently in LLGAs?**
2) **What conditions are required for a LLGA to learn linkage?**
3) **Why is important for linkage to increase?**

Index Terms— **Linkage Learning, Genetic Algorithms, Linkage Skew, Building Blocks**

I. Introduction

GENETIC ALGORITHMS (GAs) were first developed by John Holland and his colleagues/students at the University of Michigan in 1975 [2] and first discussed in his paper 'Adaptation in natural and artificial systems' [3]. Genetic Algorithms are an attempt to model the biological world's process of natural selection as a method of searching. Natural selection was a phenomena first described by Charles Darwin in his book the 'On the Origin of Species'[4]. He remarked that natural selection is a process that causes many species to become extinct but consequently maintains those well-suited to their environment. This remark was clearly not Darwin's biggest contribution to genetics but it is a helpful conceptualisation of the purpose of GAs.

Prior to the conception of GAs there were three main types of search methods used to try and determine the optimal set of parameters in a particular search space; these three methods were calculus-based, enumerative and random searching [2]. One of the most critical aspects of a search method is its robustness. The robustness of a search method is basically the likelihood that it will find the optimal solution in the search space. All three of the pre-existing search methods are either not totally robust or are computationally expensive because of the 'Curse of Dimensionality.'

Genetic Algorithms work by taking an initial population of parameters sets, also known as genomes (sets of genes). The gene values for each genome can be specifically chosen but it is best if they are chosen randomly, as it will help the algorithm search across the greatest amount of the parameter space[1], making it more robust. From the initial population a selection process takes place to determine the new population. Selection is controlled by a fitness function applied to performance of each genome, the 'fitter', (The better the performance of the genome in a prescribed tournament), a genome is the more likely that genome will be involved in the next generation. There are four different ways that a genome can be involved in the next generation:

1) The genome can be directly copied to the next generation
2) The genome can be mutated and then copied to the next generation. Mutation is the process of randomly selecting a gene from the genome and randomly changing its value.
3) Crossover can be performed between two individual and the resultant offspring can be copied to the next generation. Crossover is where two genomes are spliced together to produce an offspring which shares some genes values with the first parent and some gene values with the second. The number of 'points' in Crossover, is the number of times the genetic material in an offspring changes from being inherited from the first parent to being inherited from the second or vice-versa.
4) Crossover can be performed between two individual and the resultant offspring can then be mutated before being copied to the next generation. It does not actually make a difference if mutation is performed before or after crossover.

Individuals generated from one of the last three of these 'Involvements' are required for the GA to search the parameter space.

GAs have some advantages over the other search methods, which provide them with greater robustness. GAs do not use derivatives like calculus-based methods do [2], which means they are capable of handling parameters which are made up of decision variable sets[2]. The way that GAs search parameter space makes them less susceptible to getting stuck at local optima like some calculus-based methods, such as a steepest gradient hill-climber [2].

[1] A multi-dimensional space which encloses every possible set of parameter/gene values.

[2] A decision variable set could be the set {a,g,c,t}, where the ordering of this set arbitrary.

II. Linkage Learning

Linkage learning in GAs is the process of grouping together functionally (epistatically) dependent subsets of genes. Genetic (also known as Physical) linkage is a measure of distance between the loci of two or more specific functionally dependent genes, this is the same as what the biological world commonly interprets linkage to be, (see [5]). Linkage does not necessarily mean genetic linkage, any representation of the functional dependency between two or more genes, can be considered a representation of linkage. Many GAs that learn linkage operate using a two phase method, firstly they determine the subsets of functionally dependent genes and then they use some mechanism to improve the genetic linkage between these genes.

Both the biological world and many GA papers[3] have cited that the generation of genomes with good Linkage is advantageous because it helps maintain good subsets of genes. As defined in section I one of the ways that genomes can populatate the next generation is through crossover. When crossover is performed the group(s) of genes that affect the fitness of the genome,(often described as the active genes or the building block(s)), may be split up. In one-point crossover the chance of splitting up a group of active genes is proptionate to the distance between the first and last active gene of that group. If we take a simple case where there is only one building block, if one of the parents has an optimal or near optimal building block (i.e. a highly fit genome), close genetic linkage is advantageous because it reduces the probability of the building block being split up.

It is all well and good to say that functionally dependent genes should be placed adjacent to each other and then this will make the search more robust; unfortunately the functional dependency between genes is generally not known. This means that aligning of functionally dependent genes adjacently must occur during the search for the optimal genome. Now that we have a better undertanding of what linkage is this report can address in greater detail several questions:

1) How can functionally dependent genes be aligned adjacently in GAs?
2) What conditions are required for a LLGA to learn linkage?
3) Why is important for linkage to increase?

There are a number of papers that discuss problems that both hill-climbing search algorithms and simple GAs struggle or find impossible to solve[6]. Simple GAs find some Walsh polynomials difficult to solve. Walsh polynomials are based on combined sets of Walsh functions[7] defined by a polynomial. Walsh polynomials are not necessarily difficult for to solve, it is dependent on the co-efficients defined for the polynomial. Some polynomial co-efficients produce low-order, short-defining length schema, that are easy for GAs to optimize. Other polynomial co-efficients produce high-order, long-defining length schema that are more difficult for GAs to optimize because high-order schema are more likely to be deceptive[4] and long-defining length schema are more likely to be split-up by crossover.

There are many other problems similar to these Walsh polynomials that simple GAs struggle to solve, due to the fact that crossover destroys many fit genomes because of large distance between the first and last active genes. In fact a ”random-mutation hill-climber”, a algorithm that follows local gradients but with genetic overtones, outperforms simple GAs on fitness landscapes such the ”Royal Road” function [8] [9]. Linkage learning has been taken up as one of the techniques that could be used to solve these problems because it can reduce the probability of groups of functionally dependent genes being broken up. This makes it easier for a GA to traverse the fitness landscape towards an optimal solution.

One of the first attempts to produce a GA that could learn linkage was Goldberg, Korb and Deb's Messy GA (mGA)[10]. The mGA relies on the fact that the problems that it needs to resolve are decomposable into a set of sub-problems that can be solved independently. Having a set of sub-problems means that an individual in the mGA does not have to specify a value for every gene. The advantage of being able to concentrate on separate sub-problems means that each functionally dependent set of genes can be developed independently into a tightly linked building block before gradually bringing back together all the building blocks to generate an optimal individual. The mGA is a moving-locus mechanism, that is to say the actual positions of the genes move about the genome to allow improvement in genetic linkage and produce more tightly-linked building blocks. The mGA was the first GA that was shown to converge on the global optimum for a provably difficult problem, i.e. a deceptive problem. It was compared against a simple GA which could only get 25% of the sub-functions correct in tests performed by Goldberg, Korb and Deb.

In [11], Kargupta acknowledged that the mGA was the first serious attempt at tackling the issue of linkage learning in quasi-decomposable[5] problems. Kargupta also observed that for linkage learning to be successful it needs to be able to represent complex subsets such as 0000, 1111, that he described as ‘similarity subsets.’ The mGA does not have the richer relationships required to easily capture such complex subsets. Kargupta states that GAs needs to be able to detect order-k delineable relations efficiently, (where k is a constant integer), rather than just first order relationships between specific genes. To simplify a framework to store these higher-order relationships is required, as opposed to just aligning functionally dependent genes adjacently. Kargupta makes clear that at press time this was purely a philosophical idea but since then several implementation have been developed that try and tackle the representation of complex subsets using higher order relationships (e.g. BOA, that is discussed later in this section).

The mGA showed that it can solve problems that simple GAs could not, however there are still problems that the mGA struggles to solve [12]. Watson and Pollack cited that the mGA is partial commitment algorithm because it can be broken down into individually solvable sub-problems. When solving the sub-problems only genes associated with that sub-problem

[3] mostly those written in the last 20 years

[4] I.e. converge on local optima rather than the global optimal

[5] Problems that can be broken down into a set of sub-problems.

need to be committed to and all other gene values can be ignored making it only 'partial' committment. Watson and Pollack then go on to say that partial committment GAs should be successful whether they are moving locus algorithm, where the ordering of genes can change or a fixed locus algorithm, which relies on recording links between functionally dependent genes instead. They also cited that the mGA uses a two phase operation:

1) Limited Commitment
2) Full Commitment

Watson and Pollack stated that by replacing these two phases with an integrated incremental approach, the algorithm becomes more powerful and able solve problems that the original mGA could not solve. They called this new GA the Incremental Commitment GA (ICGA). This ICGA is capable of solving hierarchical problems such as the Hierarchical If and only If (H-IFF) function that resembles a 'Royal Road' fitness landscape but unlike the 'Royal Road' fitness landscape this function cannot be solved by any type of hill-climber[13]. The ICGA was inspired by the mGA but it uses fixed loci and relies on building links between functionally dependent links rather than aligning functionally dependent genes adjacently.

Another approach to incorporate linkage into GAs was proposed by Pelikan, where he uses Bayesian networks in his Bayesian Optimization Algorithm (BOA)[14]. Bayesian networks rely on positive and negative instances, which can be generated by thresholding the genomes at a particular fitness function value and setting all genomes above that value to positive instances. Bayesian networks are a logical way to tackle this problem because the way they are constructed means that genes that have functional dependencies on each other are close nodes in the network. The joint distribution encoded by the Bayesian network can then be used to generate new genomes to replace some of the genomes in the current population to produce the next generation. Pelikan found that the linkage that the Bayesian network provides produces a GA that is more efficient than a simple GA [14].

Like the mGA, BOA struggles to solve hierarchical problems so Pelikan adapted BOA to produce a hierarchical BOA (hBOA)[15]. hBOA can solve hierarchical problems, such as hierarchical trap problems, by proper decomposition over a number of levels, chunking [6] and preservation of alternative candidate solutions.

BOA and hBOA are quite clever solutions, which can deal with continuous, discrete and decision variable gene values. BOA uses machine learning techniques to estimate the functional dependency between genes and then improves the genetic linkage of the new population. To gain a greater understanding of genetic linkage it is better to look at a more classical example of linkage learning which is unfettered by non-genetic methods of optimisation.

Harik and Goldberg in their paper entitled 'Learning Linkage' [1], suggest using a method of crossover that allows the re-arrangement of gene order in binary genomes, so that linkage learning could occur. One of the objectives of my paper is to re-implement the environment that Harik defined in [1] to gain a greater understanding of linkage in a purely genetic mechanism, (no machine learning, like in BOA) and to try and obtain detailed answers to the three questions stated previously.

[6]representing and manipulating solution pieces to low-level sub-problems as if they were single variables

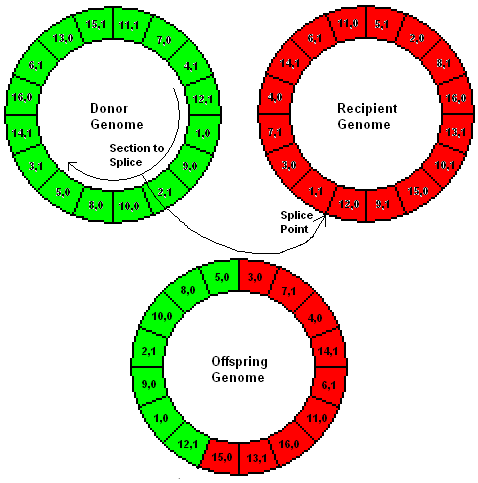

Fig. 1. Harik's Crossover Mechanism

III. Harik's Linkage Learning

The crossover method that Harik and Goldberg use in [1] is similar to that which occurs in bacterial cells, where the conservation of gene order is much less likely than for the crossover that occurs meiosis. Bacterial cells are quick at adapting to produce versions of themselves that are highly fit, such as those resistant to antibiotics [16]. Therefore a crossover method that resemble that of bacterial cells should produce a GA that can quickly find the optimal solution.

The main question about Harik's LLGA is, "Does it cause linkage learning to occur?" To determine this, firstly the method of crossover must be analysed. In Harik's paper, he describes the crossover mechanism in detail. Figure 1 is a pictorial representation of Harik's crossover mechanism.

Firstly he defines that his genomes are circular, to negate any differentiation in gene position relative to a gene's closeness to the end of the genome. As a consequence circular genomes must have two-point crossover or crossover with an even number of points. Secondly two genomes, a donor and a recipient, contribute to produce only ONE offspring. A randomly long splice is taken from the donor genome and is spliced into the recipient genome at a random point. From the splice point in the recipient genome, there is an iterative process through each gene, (firstly through the donor splice and then through the whole recipient genome) removing all duplicate genes. By removing duplicate genes, the gene order of the offspring genome is significantly different from the order of either the donor or recipient genome, which may make it have better (tighter) or poorer (weaker) linkage than its parents. As tight linkage preserves fitter genomes better, the selection process should tend to favour genomes with tighter linkage.

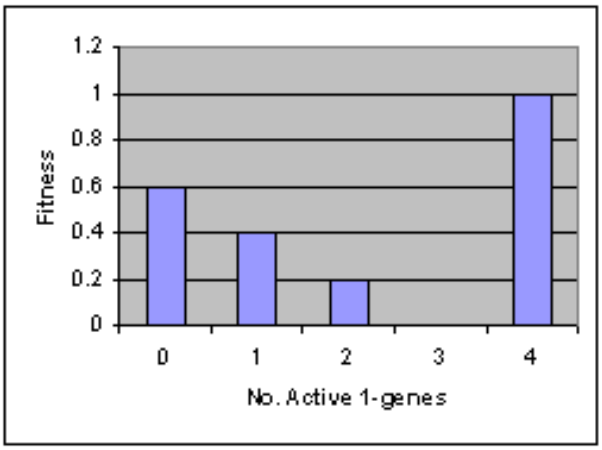

Fig. 2. Fitness Function Distribution for 4-Bit Deceptive Trap Function

Harik defined an implementation with a functionally dependent four gene building block, as a part of a 150 gene genome. He used the fitness function distribution in Fig. 2 for this building block.

This distribution is known as a deceptive trap function because if a calculus-based search algorithm, such as a steepest gradient hill-climber, was to be used, it would almost certainly converge on genomes with the four zeros building block, called the deceptive building block, instead of genomes with the optimal four ones building block. Harik chose this type of fitness function distribution because it demonstrates a case where a GA that promotes tight linkage can find the global optimum when a calculus-based hill-climber or a simple genetic search algorithm is unlikely to do so.

Harik performed two control experiments to test whether predictions on how linkage should improve were correct. Harik defines linkage as being the largest gap between two active genes divided by the total number of genes in a genome, therefore the best linkage that can be achieved is $(150-4)/150 = 0.973$. He measured linkage after the selection stage but before performing crossover. Both these experiments are somewhat contrived, in the sense that they always cross a selected optimal block building genome with a randomly ordered (generated on the fly) deceptive building block genome, that is also the optimal building blocks complement. The first experiment 'Linkage Skew' where the selected optimal genome was the donor, showed identical results to those predicted by Harik. The second experiment 'Linkage Shift', is very similar to the first experiment, apart from the optimal selected genome is the recipient rather than the donor. The results for this experiment were slightly different to Harik's prediction even after he re-adjusted his prediction but they did show an improvement in linkage.

After performing the two control experiments Harik ran his main experiment, which was not contrived like the control experiments. This main experiment used a similar deceptive trap function for measuring the fitness of all selected genomes. The deceptive trap function used was slightly different to the control experiments because the building block was only of size 3 in a genome of size 100.

The mechanism that Harik used for selecting genomes is quite interesting and can be changed by varying a parameter called the selection rate. This selection rate defines the tournament size to use from which the fittest individual is selected. Take for example a selection rate of 1.5, this selection rate means that half of the selected genome are selected from a tournament size of two and the other half is selected from a tournament size of 1 (random selection).

Harik used selection rates from 1.2 up to 2.0 and tested to see both whether the linkage increases and what value the building block converges on. For all the selection rates apart from 1.2 the population converges on the optimal building block but with no discernible increase in linkage for optimal building blocks. When a selection rate of 1.2 was used there was a discernible increase in linkage for the optimal building blocks but the algorithm converged on the deceptive building block. So the optimal building block population's linkage reaches a peak and then quickly drops down to zero, as the number of optimal building blocks becomes very small and then zero.

These results show that it seems very difficult to strike a balance between converging on the optimal building block and learning linkage using Harik's mechanism. The control experiments that show linkage are contrived and have no consideration of the average fitness of the population and the main experiment, that is concerned about fitness only shows linkage when the GA converges on the deceptive building block.

The reason that Harik gives for the difficultly in learning significant linkage in an uncontrived experiment is the 'Homogeneity Effect'. This is the effect that selection process reduces the diversity in the population, (making the population more homogeneous) but the crossover mechanism fails to increase the diversity so that over time the diversity of the population decreases as the population converges on the optimum. Having identical genomes means there is no way of producing non-optimal genomes in the future that provide the pressure for increased linkage.

As only the control experiments that Harik ran showed improvement in the linkage of optimal building blocks these experiments should be focused on to see if the reasons for increased linkage can be determined. As Harik's 'Linkage Skew' experiment most closely resembled his predicted results it makes sense to focus on this mechanism as the results that it produced are best understood.

Harik's 'Linkage Skew' experiment used an initial population of 6400 genomes, all with the optimal building block but in random orders. These genomes where then used as donors and crossed with randomly ordered genomes with deceptive building blocks to produce a new population of 6400 genomes. I.e. an individual with four active 1-genes is crosed with an individual with four active 0-genes in random locations. The objective the 'Linkage Skew' experiment was to show that the four active 1-genes get closer to each other over evolutionary time.

As the optimal building block genome is the only type of donor that can be selected the fitness function is not a trap function (see Figure 2) but a truncated, at the optimal building

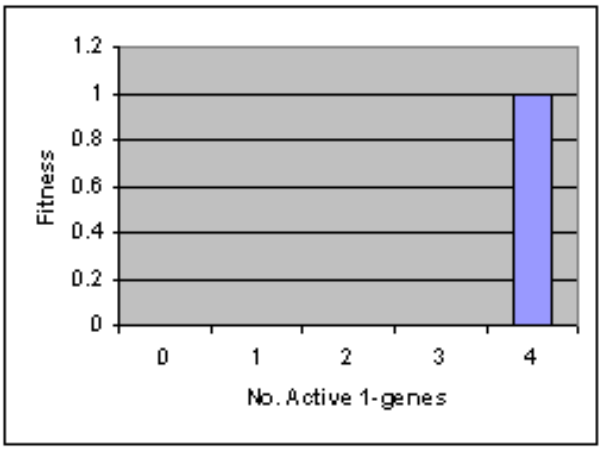

Fig. 3. Truncated Fitness Function Distribution for 4-Bit Building Block

block, fitness function (see Figure 3). This means that although Harik states that the recipient is the deceptive building block it is really only the optimal building block's complement, as the four zeros building block is no longer a local optima.

The crossover process to produce a new population was repeated over 50 generations and it was found, as already stated, that linkage did indeed increase. This result was significant for two reasons; firstly it showed that there was some force encouraging tighter linkage; secondly it showed that the increase in linkage between generations was dependent on the current linkage, which Harik had hypothesised before the experiment[7].

By having a basis from which linkage learning can be seen to occur, i.e. Harik's 'Linkage Skew' experiment it should be possible to develop this experiment to work out which conditions provide pressure for increased linkage and which do not. Harik's 'Linkage Skew' experiment has several conditions, listed below, that make it contrived and therefore make the search algorithm much less robust.

1) The initial donor population is already optimal.
2) The recipient genomes used are always the complement of the optimal building block.
3) The selection mechanism for the donor only allows optimal genomes to be selected.

By gradually relaxing the conditions listed above it should be possible to make the search algorithm more robust and help us gain a better idea of what condition(s) are the biggest contributers to the experimental results that show increased linkage.

IV. Re-Implementation

As stated in section III, there are several conditions in Harik's 'Linkage Skew' experiment that can be relaxed. This section takes these conditions and implements experiments that gradually relax these conditions to determine those that have the greatest affect on the improvement of linkage.

[7]Harik hypothesised that the increase in linkage should equal the linkage variance in the current population divided by the current linkage.

Matlab was used to implement all the experiments in this re-implementation, as it is the best tool for prototyping this type of algorithm because it allows fast development and easy management of multi-dimensional numerical arrays.

For each experiment a certain number of parameters need to be defined, as these parameters can directly affect the results produced. For all the experiments in this paper these parameters have been kept as close to the Harik's 'Linkage Skew' experiment as possible. To save explanation for each experiment, there follows a listing of all the parameters, their descriptions and values:

- N: The number of genes in the genome = 150
- K: The number of active genes in the genome = 4
- P: The population size at each generation = 500
- G: The number of generations of genomes produced = 50

The parameter values shown are almost identical to those used by Harik in his 'Linkage Skew' experiment. The only main difference is the population size is 500 instead of 6400. Several tests were run to see how population size affects results and it was found that a population size of 500 gives almost identical results to 6400.[8]. Harik himself observed in [17] that the population size is very important, too small a population will have a lack of diversity and will not thoroughly traverse the search space, too large a population and the algorithm becomes too computationally expensive. This was considered when the population size for these experiments was chosen. Using a population size the same as Harik, for all of the experiments would have been too computationally expensive but by proving the similarity in the test results for population sizes of 500 and 6400, shows that the diversity could not have been to drastically affected. The size of the population becomes more important as the size of the building block increase, for each extra gene in the building block, the number of building blocks permutations doubles, therefore the population size itself needs to be doubled. Fortunately the building block size for these experiments is quite small.

Harik's 'Linkage Skew' experiment measured its linkage on the donor genomes selected at each generation to maintain consistency across all these experiments, this is how linkage is measured.

A. Experiment I

Before testing to see whether Harik's 'Linkage Skew' experiment can be repeated, it would be good to see how quickly Harik's mechanism for crossover can increase linkage, if the fitness function for selecting both the donor and recipient genomes explicitly selects based on the linkage of the genomes. By knowing the maximum rate of linkage learning for a GA that is explicitly selective, (unlike Harik's 'Linkage Skew' experiment), gives a basis for comparison for experiments that are implicit. Figure 4 shows the results for an experiment where both the donor and recipient are

[8]Comparing the results for Harik's 'Linkage Skew' experiment and this paper's re-implementation in Experiment II there is almost no difference

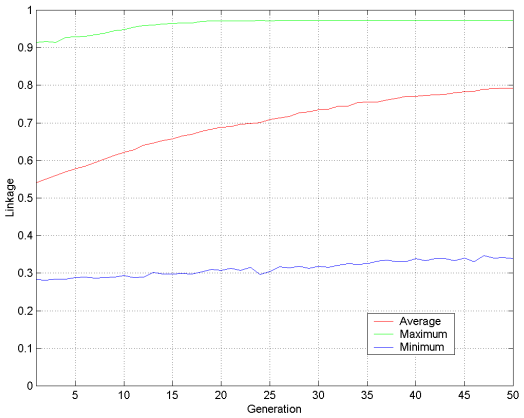

Fig. 4. Experiment I(a): Linkage Learning averaged over 30 runs where Donor and Recipient are selected using a linear Linkage-based fitness function

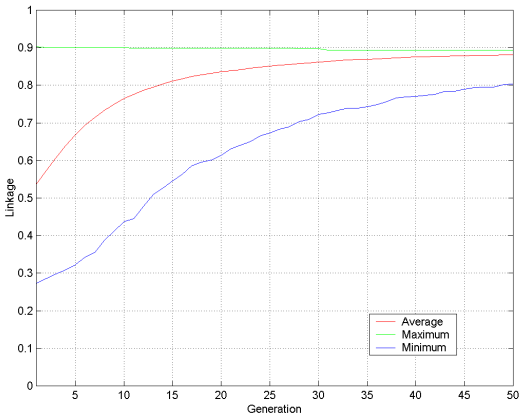

Fig. 5. Experiment I(b): Linkage Learning averaged over 30 runs where the offspring is selected directly from the current population with NO crossover

selected based on a linear linkage-based fitness function[9] From Figure 4 there is a clear improvement in Linkage from an initial Linkage of 0.55 to 0.79 after 50 generations. There are two mechanisms at work here, one is Harik's method of crossover that reorders the genes and has the potential to produce offspring with improved linkage; the other is the fitness function which should select genomes with better than average Linkage. Figure 5 show results where no crossover was performed but the same fitness function was used. From Figure 5 there is a discernible greater and faster increase in linkage when no crossover is performed. This suggests that Harik's crossover mechanism is possibly destructive to linkage and to a certain extent this is true. It may therefore seem counter-intuitive to use Harik's method of crossover but all mechanisms that change the order of the genes will on average tend to decrease the linkage of the offspring in comparison to its parents, if those parents have linkage better than that of

[9]I.e. a genome with linkage = 0.8 is twice as likely to be selected as a genome with linkage = 0.4. Linkage is measured by dividing the largest gap between two active genes and dividing by the total number of genes, the same measure as used by Harik.

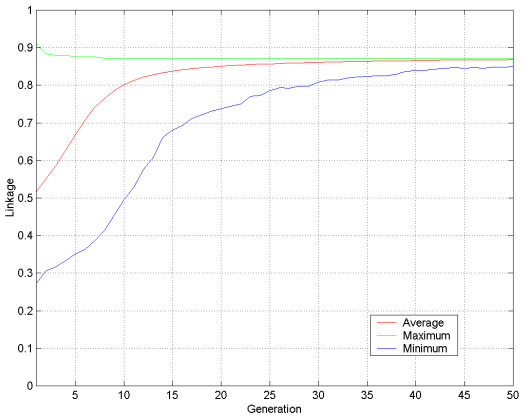

Fig. 6. Experiment II: Re-implementation of Harik's 'Linkage Skew' experiment averaged over 30 runs

a random population. Basically as the linkage of the parents becomes greater and greater than that of a random population, the probability that the offspring will have weaker linkage than its parents increases. This is because the crossover mechanism cannot work against the linkage distribution gradient that drives offspring closer to that of a random population's average linkage.

The motivation for using Harik's crossover method can be seen by comparing the trends of the maximum linkage genomes for 4 and 5, when crossover is used this maximum value increases, whereas without crossover the maximum value stays the same or decreases. This demonstrates that the crossover method helps to maintain the diversity of linkage in the population allowing the creation of genomes with greater linkage than any of the genomes in the previous population.

B. Experiment II

Now that the means by which Harik's crossover mechanism affects the linkage of genomes is better understood, a re-implementation of Harik's 'Linkage Skew' experiment can be analysed with a greater insight to what is happening between one generation and the next. Figure 6 shows the results to this re-implementation. By comparison between the results from Figure 6 and those for Harik's paper, the only main difference is the linkage of the initial population, which is approximately 0.12 higher. The reason for the difference in the initial population Linkage is probably due to Harik's method of randomizing the order of genomes[10]. The curves for both graphs are very similar and the average linkage value after 50 generations is 0.87 in both graphs.

C. Experiment III

Now that we have an implementation, which is as similar to Harik's implementation as possible, alterations to the conditions can be made to determine what conditions in Harik's implementation provide a drive for increased linkage.

[10]The method that randomises the initial population's gene order is not defined in Harik's paper

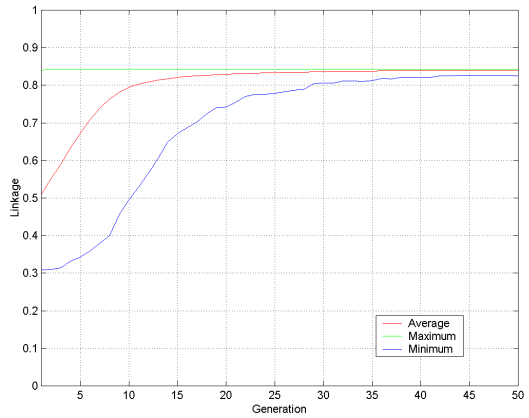

Fig. 7. Experiment III: Linkage Learning averaged over 30 runs where the initial population is random not optimal

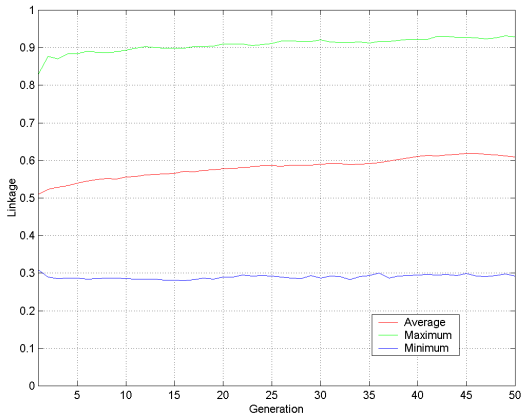

Fig. 8. Experiment IV: Linkage Learning averaged over 30 runs where the recipient genome's building block is random

As already discussed in section III Harik's 'Linkage Skew' experiment is rather contrived in its attempts to obtain results that show improved linkage. By relaxing the 'Linkage Skew' mechanism gradually, one component at a time, it should be possible to determine which components drive an increase in linkage.

Harik's 'Linkage Skew' experiment has an initial population with only optimal building blocks in it, (although the linkage of population is random). A typical GA would not start with an optimal population, as what would be the point in trying to construct new optimal genomes when every genome is already optimal. Instead of starting with an optimal population, what would happen if you started with a random population? Figure 7 shows the results obtained after this slight modification was made.

Figure 7 produces results very similar to Experiment II. This is hardly surprising as the selection mechanism for the donor only selects optimal genomes. In Experiment II, each time a donor genome was selected there was an equal chance it could be any genome from the population of 500. In Experiment III, each time a donor genome is selected, the population to choose from is on average only $500/16 = 31.25$ because the probability of an optimal building block being generated randomly is 1/16th, therefore the initial selectable population size is a lot smaller. The most noticeable difference is that the final linkage value is about 0.03 less than Experiment II, this is due to a number of genomes with good linkage in the initial population being eliminated because they are not also optimal. Due to the very small change in results it is safe to conclude that this component of Harik's 'Linkage Skew' experiment is not a significant driver for improved linkage.

D. Experiment IV

A further slight modification that can be made to Harik's 'Linkage Skew' experiment to make it slightly less contrived, is to change the way that recipient genomes are generated for crossover. Currently all the recipient genomes are randomly generated genomes with random linkage but a deceptive building block, (i.e. four zeros on the active genes). The reason that Harik chose the deceptive building block he did was because it is the complement of the optimal building block. When you cross an optimal donor with a recipient that is its complement, the only way to produce an optimal genome, which can be used in the next generation, is for all the active genes to come from the donor. Due to the way that Harik's crossover mechanism works, it is impossible for linkage of the offspring to be better than that of the donor in his 'Linkage Skew' experiment. This makes it comparable to Experiment I(b) where no crossover takes place at all. If you compare the maximum linkage genome over time in Experments I(b), II and III, all three of them have either a downward or level trend.

By changing the recipient genome generation mechanism so that the building block is random not the complement of the optimal building block makes the experiment less contrived. The building block distribution for randomly generated building blocks is defined below:

- Zero active 1-genes = 1/16
- One active 1-genes = 4/16
- Two active 1-genes = 6/16
- Three active 1-genes = 4/16
- Four active 1-genes = 1/16

By using this distribution for recipient building blocks provides the potential for an increase in linkage between the donor and the offspring. Figure 8 shows the results using the new recipient building block distribution.

Figure 8 shows a much smaller improvement in Linkage in comparison to Experiments II and III, roughly an increase of 0.1 rather than 0.3-0.35 of the earlier experiments. As this experiment was run 30 times an increase of 0.1 can still be considered significant trend. Another significant result is the upward trend of the maximum linkage genome, which is a clear contrast to the level or downward trends in Experiments II and III. This shows that Harik's crossover mechanism can generate offspring with better linkage than its donor parent, even if the fitness function for the donor does not explicitly select on linkage.

Due to the sharp contrast in these results it can be concluded

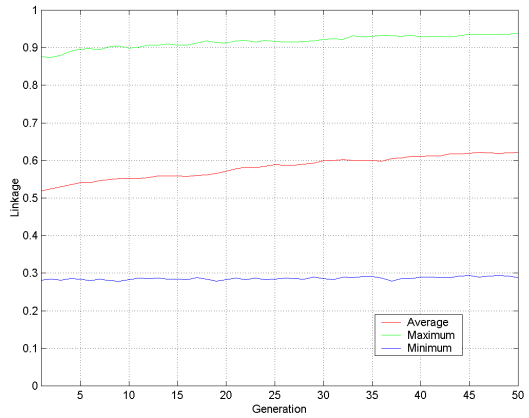

Fig. 9. Experiment V(a): Linkage Learning averaged over 30 runs where the donor genome is selected using a super-linear ramping fitness function

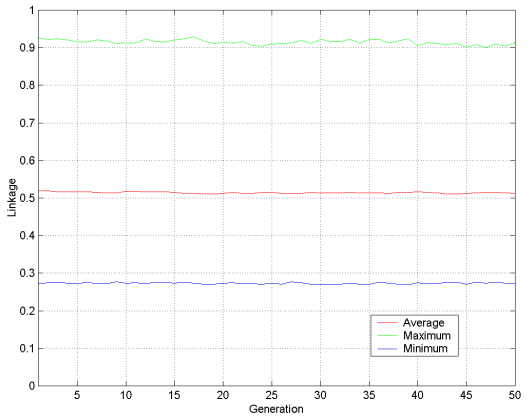

Fig. 10. Experiment V(b): Linkage Learning averaged over 30 runs where the donor genome is selected using a linear ramping fitness function

that crossing with a complementary building block, is one of the driving factors for good linkage. Unfortunately as already discussed, the reason the previous two experiments had good improvement in linkage was because by crossing with complementary building blocks, the effect of Harik's crossover mechanism is nullified. When crossover is nullified the effect that it has of pulling offspring's average linkage closer to that of a random population is also eliminated, making the divergence from a random population's average linkage faster.

E. Experiment V

So far every alteration made to the Harik's mechanism has reduced the rate of Linkage improvement but there is still a discernible increase over 50 generations. One of the main components that makes Harik's GA contrived, that has yet to be investigated is the fitness function for the donor genome. Presently it is 'truncated' with only the capacity to select genomes with optimal building blocks. By relaxing this fitness function so that it can select non-optimal building blocks would make it less contrived.

Although Harik defined the fitness function as being a trap function (see Figure 2 because it was truncated so that only optimal building blocks could be selected, when relaxing the selection mechanism, it does not really matter what the fitness function distribution is as long as the optimal building block has the highest fitness value. Figure 9 uses a super-linear ramping fitness function. Below is a listing of the fitness function value for each type of building block.

- Zero active 1-genes = 0^{10} = 0
- One active 1-gene = 1^{10} = 1
- Two active 1-genes = 2^{10} = 1024
- Three active 1-genes = 3^{10} = 59049
- Four active 1-genes = 4^{10} = 1048576

By using this super-linear fitness function the probability in the first generation of the donor selection mechanism picking an optimal genome is only 0.76, assuming the random building block distribution, as defined in section IV-D, which should gradually increase as the linkage becomes tighter and the population distribution tends towards more optimal building blocks.

The super-linear ramping function is basically a linear ramping function (where the number of active 1-genes is the fitness function value) raised to the power of 10. A super-linear function is necessary to produce results that show an increase in linkage. This is because using only a linear function increases the number non-optimal building block donors selected and therefore also increases the probability that an optimal offspring will be generated from two non-optimal building blocks, where the linkage of the donor genome does not matter. By having optimal building blocks in the new generation that have not survived because of their tight linkage means that selecting an optimal building block does not necessarily mean you are selecting a genome with better than average linkage. Figure 10 is the same experiment as Figure 9 but using only a linear ramping fitness function, where the probability of selecting an optimal building block in the first generation is only 0.25. In [17] it states that for an an optimal building block to survive over time, the probability of one being selected must be greater than 0.5. This suggests that the number of optimal building blocks in the population are likely to decrease yet further and make it more and more difficult to learn linkage as the donor population becomes more dissimilar to that of the optimal population used in Harik's 'Linkage Skew' experiment, as shown in Figure 10.

By testing using varying powers to raise the super-linear fitness function it was found that a power of 5-6 is required to show a discernible improvement in Linkage, this is equal to roughly a 0.5 probability of selecting an optimal building block in the first generation. To confirm definitively the super-linear power required to demonstrate improved Linkage, would require further investigation with significance testing such as Student T tests[18].

The results from these experiments show that as long as the initial probability of selecting an optimal genome is a significant majority, then there is very little difference in the the improvement in linkage. The question of why reducing the probability of selecting an optimal genome eliminates any

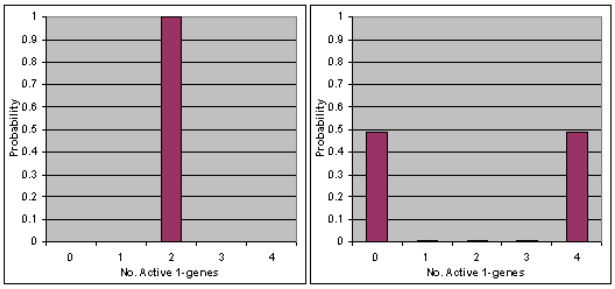

Fig. 11. Probability Distributions for building blocks after crossover using Experiment II Mechanism. (Left: Poor Linkage Donors, Right: Good Linkage Donors)

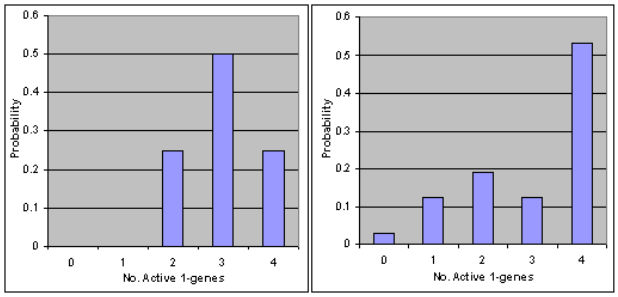

Fig. 12. Probability Distributions for building blocks after crossover using Experiment IV Mechanism. (Left: Poor Linkage Donors, Right: Good Linkage Donors)

sign of linkage has been addressed, however it would be useful to apply some mathematical logic to why there is no increase in linkage, as well as why the increase in linkage in the experiments was less than for Harik's 'Linkage Skew' experiment.

F. Experimentation Conclusion

From Experiment II to Experiment V, the rules of Harik's 'Linkage Skew' have been relaxed to try and created an experiment which is less contrived and could therefore be applied in more general circumstances. Although we have managed to show that with less contrived experiments linkage can still be learnt, the degree of linkage learnt continued to decrease throughout. The question is, '"Is it possible to fully understand why this phenomena occurs?"'

Lets first take Harik's 'Linkage Skew' experiment and work out the probability distribution of building blocks after crossover when they have both good and poor linkage. Figure 11 shows both probability distributions. These distributions are very different to each other but it should be noted that there is a distinct increase in the probability of the optimal building blocks surviving crossover when there is good linkage, which after all is the whole point of linkage learning. A second thing that should be noted about these distributions is that the average building blocks are the same, (2 active 1-genes per genome). This means that although Harik's mechanism will learn linkage it cannot converge on an optimal value, which further highlights how contrived Harik's 'Linkage Skew' experiment was.

Now lets take Experiment IV mechanism, it produces the probability distributions as shown in Figure 12, when linkage is poor and good. As can be seen from these probability distributions, the increase in the probability of the optimal building block is much less at 0.28 instead of 0.49. This difference explains why there is a much lower increase in linkage for Experiment IV than Experiment II. As already stated for Experiment II the average number of active one-genes after crossover is the same whether there is good or poor linkage. However with Experiment IV the average number of active one-genes is 3 instead of 2, which is closer to the optimal building block. This shows how the Experiment IV is less contrived than Harik's Linkage Skew experiment, where the average number of active 1-genes is the same as that for a set of random genomes.

Probability of selecting an Optimal Donor	Probability of Optimal Offspring after Crossover with Good Linkage	Probability of Optimal Offspring after Crossover with Poor Linkage	Difference between Good and Poor Linkage
1	0.53	0.25	0.28
0.5	0.26	0.12	0.14

Fig. 13. Probability of Optimal Building Blocks after Crossover as probability of Donor Building Blocks decreases

Experiment V has a lower probability of selecting an optimal donor as the raising power for the super-linear function is decreased. As the probability of selecting an optimal building block donor decreases, the probabilities for optimal building blocks after crossover also decreases, in both the poor and good linkage cases. Purely for illustrative purposes an example of this phenomena is given in Figure 13, looking at the values you can see how reducing the probability of selecting an optimal donor reduces the drive for increased linkage because the probability of producing an optimal offspring does not vary as much dependent on the linkage of the donor. In fact the probability for producing an optimal genome after crossover when linkage is poor is likely to be higher than 0.12 because optimal genomes can be produced from two non-optimal parents. This makes the difference in probability between good and poor linkage donors even less than 0.14, making the drive for increased linkage even smaller. That is why a super-linear function such as Experiment V(a) show increased linkage whereas Experiment V(b) that has a linear function does not.

The experiments show that as you make the mechanism less contrived and more like a realistic GA, the amount of linkage learnt decreases but as demonstrated mathematical the average number of active 1-genes gets closer to the optimum after crossover (whether the Linkage is good or poor). This clearly demonstrates that with Harik's crossover mechanism, a balance between learning linkage and finding optimal genomes needs to be struck, unfortunately from the experimental results and Harik's main experiment it appears that there may be no balance where you can achieve both of these at the same time. Further testing would be required to improve the certainty of this conclusion. All the experimentation in this paper has concentrated on increased linkage and not worried

too much about increasing the average fitness of genomes. More experimental evidence which shows fitness to increase but linkage to remain unchanged would lend further weight to the earlier conclusion.

G. Conclusion

To conclude lets consider the three questions set out in section II. Firstly, "How through the processes of linkage learning GAs do functionally dependent genes become adjacently aligned?" In Harik's 'Learning Linkage' paper genes can be adjancently aligned through the crossover mechanism. Harik demonstrated in his paper that two mechanisms are at work, 'Linkage Skew' and 'Linkage Shift'. Harik's implementation is one of only a few implementations discussed that uses a purely genetic technique to re-order genes and attempt to learn linkage, in contrast to BOA, that uses techniques that more resemble machine learning. Through the experimentation in section IV it was shown that even after relaxation of Harik's 'Linkage Skew' experiment there is still the potential for linkage learning. This demonstrates that functionally dependent genes can be aligned adjacently using non-genetic approaches such as machine learning Bayesian networks but also using a purely genetic approach as demonstrated by Harik's LLGA.

The second question asked, "What conditions are required for a LLGA to learn linkage?" It is clear from the experimentation that for linkage to increase in a GA that uses purely genetic mechanisms, very strict conditions need to be imposed, which unfortunately makes the GA very difficult to apply to generalised circumstances. Other GAs that learn linkage require less strict conditions because many of them can identify functionally dependent relationships and linkage can be changed explicitly to take into account the functional dependencies.

Finally, "Why is it important for linkage to increase?" The experimentation in this report probably gives the clearest demonstration of why it is important for linkage to increase. Figures 11 and 12 clearly show better linkage preserves optimal genomes better. Other GAs show that increased linkage is important because it allows the search algorithm to traverse the parameter space more efficiently and decreases the chance of it getting stuck at a local optima, which reduces the algorithms robustness.

drn
May 13, 2005

REFERENCES

[1] G. Harik and D. Goldberg, "Learning linkage," in *Foundations of Genetic Algorithms 4*, R. Belew and M. Vose, Eds. San Francisco, CA: Morgan Kaufmann, 1997, pp. 247–262. [Online]. Available: citeseer.ist.psu.edu/article/harik97learning.html

[2] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[3] J. Holland, *Adaptation in natural and artificial systems.* The University of Michigan Press., 1975.

[4] C. Darwin, *On The Origin of the Species*. John Murray, 1859.

[5] J. W. Kimball, "Genetic linkage and genetic maps," December 2004. [Online]. Available: http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/L/Linkage.html #An_example_of_linkage

[6] S. Forrest and M. Mitchell, "What makes a problem hard for a genetic algorithm? some anomalous results and their explanation," *Machine Learning*, vol. 13, pp. 285–319, 1993. [Online]. Available: citeseer.ist.psu.edu/forrest93what.html

[7] E. Weisstein., "Walsh function," MathWorld–A Wolfram Web Resource, 1999. [Online]. Available: http://mathworld.wolfram.com/WalshFunction.html

[8] S. Forrest and M. Mitchell, "Relative building-block fitness and the building-block hypothesis," in *Foundations of Genetic Algorithms 2*, L. D. Whitley, Ed. San Mateo, CA: Morgan Kaufmann, 1993, pp. 109–126. [Online]. Available: citeseer.ist.psu.edu/forrest93relative.html

[9] M. Mitchell and S. Forrest, "Fitness landscapes: Royal road functions." [Online]. Available: http://www.cs.pdx.edu/ mm/handbook-of-ec-rr.pdf

[10] D. Goldberg, B. Korb, and K. Deb, "Messy genetic algorithms: Motivation, analysis, and first results," *Complex Systems*, vol. 3, pp. 493–530, 1989.

[11] H. Kargupta and B. Stafford, "From dna to protein: Transformations and their possible role in linkage learning." [Online]. Available: citeseer.ist.psu.edu/26102.html

[12] R. A. Watson and J. B. Pollack, "Incremental commitment in genetic algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, Eds., vol. 1. Orlando, Florida, USA: Morgan Kaufmann, 13-17 1999, pp. 710–717. [Online]. Available: citeseer.ist.psu.edu/ra99incremental.html

[13] R. Watson, "The hierarchical-if-and-only-if (h-iff) ga test function." [Online]. Available: http://www.cs.brandeis.edu/ richardw/hiff.html

[14] M. Pelikan, D. Goldberg, and E. Cantú-Paz, "BOA: The Bayesian optimization algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, Eds., vol. I. Orlando, FL: Morgan Kaufmann Publishers, San Fransisco, CA, 13-17 1999, pp. 525–532. [Online]. Available: citeseer.ist.psu.edu/article/pelikan99boa.html

[15] M. Pelikan and D. Goldberg, "A hierarchy machine: Learning to optimize from nature and humans," *Complexity*, vol. 8, no. 5, 2003.

[16] D. Tenenbaum, "Microbes: What doesn't kill them makes them stronger," 1997. [Online]. Available: http://whyfiles.org/038badbugs/

[17] Harik, Cantu-Paz, Goldberg, and Miller, "The gambler's ruin problem, genetic algorithms, and the sizing of populations," in *IEEECEP: Proceedings of The IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, 1997. [Online]. Available: citeseer.ist.psu.edu/article/harik97gamblers.html

[18] D. Caprette, "'student's' t test (for independent samples)," January 2004. [Online]. Available: http://www.ruf.rice.edu/ bioslabs/tools/stats/ttest.html