The Statistical Mechanics Approach to the Study of Genetic Algorithm Dynamics

Jonathan Shapiro[1], Magnus Rattray[1], Adam Prügel-Bennett[2]
[1]*Department of Computer Science, Manchester University, Manchester M13 9PL. UK*
jls@cs.man.ac.uk.
[2]*NORDITA, Blegdamsvej 17, DK-2100 Copenhagen Ø, Denmark.*

1. Introduction

One of the difficulties in understanding genetic algorithms is that it is difficult to solve any specific examples. We try to understand the behavior of genetic algorithms through qualitative or general notions, such as the schema theorem or the building block hypothesis. Although these give a qualitative picture of how GAs might work which can be tested experimentally[1], and may inform the design of good problem representations[2], it is difficult to apply them to specific problems, or to specific realizations of the genetic algorithm. Thus, many questions about GAs, both practical and foundational, remain very difficult to answer. We feel that understanding of genetic algorithms will be improved if there are some examples which can be solved and studied. It would be particularly beneficial of these solvable examples possessed the properties which seem important to GA behavior: strong selection, relatively small population sizes, realistic recombination models, fitness functions which contain some complexity found in optimization problems, and so forth.

We have developed a formalism based on statistical mechanics which allows one to study the dynamics of genetic algorithms searching for solutions to a number of simple combinatorial problems[3,4,5,6,7,8]. This approach addresses some of the challenges raised above. The method can describe problems of realistic size and includes finite population effects, which prove to be fundamental in understanding how the GA searches. Results are possible for all values of selection pressure, although there is a simplification for weak selection. Both transient and equilibrium behavior can be modeled under the formalism and fluctuations about the expected trajectory can also be calculated. Comparisons with simulations show accurate agreement with high significance for the problems considered. The examples on which the GA has been analyzed using the approach contain some aspects of realistic problems, including multiple optima and interactions between sites. However, they are nonetheless toy problems. In particular, the problems considered so far lack the types of spatial structure which reveal substantial differences between the different forms of crossover.

In this paper, we shall give an introductory overview of the statistical mechanics method, survey some of the problems to which it has been applied, and discuss some future directions. We will assume no particular background in statistical mechanics. We will assume knowledge of GAs at the level of Goldberg[9]. The system we will be describing will be the simple GA as described by Goldberg: it acts on binary strings, uses generational, roulette-wheel selection, and contains the usual forms of mutation and crossover.

2. The Statistical Mechanics Approach

2.1 Macroscopic Versus Microscopic Descriptions of Genetic Algorithms

If developing a mathematical formalism for describing the dynamical behavior of a simple genetic algorithm, the first consideration might be: at what level of detail should the description be? Since the genetic operators are going to take a population at time t and produce a new population at time $t+1$, an obvious level of description would one which considers precisely which strings are in the population, and attempts to predicts how the constitution of the population changes in time. A description which takes into account precisely which individuals are in the population at each time step could be called *microscopic*, as it considers all of the small-scaled details of the population.

The appeal of such an approach is that it might be straightforward - no transformation of the system is required, and it could be rigorous, since all of the information required to predict the changes are available at every timestep. There are obvious drawbacks as well. There are an astronomical number of possible populations, many differing in insignificant ways, and a model at this level captures tiny changes as well is significant ones. Typically the types of questions one wishes to ask about GA behavior involves much higher-level properties - the evolution of the best, the diversity, and so forth - properties which are considerably removed from a full description of all strings. In addition, high-dimensional, non-linear systems are notoriously difficult to analyze, and the dimension of a genetic algorithm described microscopically is horrendously large.

One important example of a microscopic description of GA dynamics is the Markov Chain formulation of population dynamics (see, for example, Ewens[10], or Kimera and Crow[11] for application in population biology, and, for example, Vose and collaborators[12,13] for this approach in GA context). Here one derives deterministic equations for the probability of each possible string being present in the population at time t in terms of the probabilities at time $t-1$. High dimensionality certainly rears its ugly head in this description. Considering binary strings of length N, the dynamics can be described by the probability for each possible string - a vector of dimension 2^N . Mutation is a linear operator - it is described by a matrix of $2^N x 2^N$ components; crossover is bi-linear, so a tensor of $2^N x 2^N x 2^N$ describes it. Considering the degree of complexity here, what progress has been made represents considerable achievement. However, most results are for special situations - infinite population, neutral selection, or very small strings - none of which are particularly applicable to genetic algorithms (see reference 13 for an alternative opinion).

Rather than considering precisely which individuals are in the population, an alternative approach is to describe the GA in terms of statistical properties of the population. We describe this approach as using a *macroscopic* level of description, and the particular statistical properties which are used to characterize the population, we deem *macroscopics*. The advantages of a macroscopic formulation of genetic algorithm dynamics are evident: dynamics in a huge number of degrees of freedom is reduced to dynamics in a few quantities. A non-linear system of a few degrees of freedom can be readily solved or numerically iterated. The number of macroscopic quantities need not scale with the size of the problem or the population, so perhaps systems of realistic sizes could be tackled. As the macroscopics could chosen to be closely related to the properties which one is interested in studying concerning the GA, it is perhaps easier to learn about how GAs work at this level of description.

There are obvious drawbacks. Since a small number of global quantities contains far less information about the population than a complete description, no matter how accurate the approach may turn out to be, it will not be derived from first principles. It will be approximate,

and it can be difficult at times to see where the approximations are. Since a macroscopic description throws information away, it is essential that the appropriate macroscopics are chosen. Thus, it is not a mechanical, procedural method. Some insight about what is important and what is inessential is required.

It is obviously straight-forward to describe the population in terms of macroscopic quantities; the challenge is to derive the effect of genetic operators on these quantities. Several researchers have derived the effects of genetic operators on the mean and variance in the context of genetic algorithms[14,15] and it has been widely studied in the field of quantitative genetics[16]. Our approach is to use statistical mechanics to do the derivation. Statistical mechanics techniques are used in two ways. First, the genetic operators are probabilistic; selection involves sampling, and mutation and crossover involve probabilistic changes to the strings. Statistical mechanics can be used to compute the expected results of these operators in a robust way, and to compute fluctuations around the expected values. Second, where information other than that found in the macroscopics is required, that information is inferred from the macroscopics by a maximum entropy method. Statistical mechanics models are used to compute the maximum entropy result. This is described in the next section.

2.2 The Assumption of Maximum Entropy and the Statistical Mechanics Model

The fundamental difficulty in describing the GA dynamics via a macroscopic description is that these quantities may not provide all of the information required to determine the effects of the genetic operators. In order to get around this, we use an ansatz. We assume that any quantity which cannot be determined trivially from macroscopics which are known can be computed using a maximum entropy distribution with the known macroscopics as constraints. A statistical mechanics model is used to compute the maximum entropy prediction[4].

What this means can be illustrated by a simple example. Suppose the macroscopics are the mean and variance of the fitness in the population. One can easily derive the effects of selection on these two quantities. This is not so for crossover, however. The effectiveness of crossover will between on how similar the strings are to each other, a quantity called the *correlation*. If the strings are very similar crossover will make very little change to the population; if the strings are very dissimilar crossover can have a big effect. How can one infer the correlation from the fitness mean and variance, given the fact that it is possible for strings to have similar fitness without being correlated? It can be done using a maximum entropy assumption. Consider all possible populations of the same population size and with the same mean fitness and fitness variance as the GA population. For each possible population, one can compute the average correlation in the population. For each correlation value one can assign an entropy, that is the minus the log of the number of populations which have that value. The correlation value which maximizes the entropy is the one which is produced by the most populations. (It is typically the case that this value would be found in almost all populations; the entropy is a very sharply-peaked function[17]). The nature of our assumption would be to assume that correlation in the real population is the maximum entropy value. The maximum entropy value can be computed from a statistical mechanics model.

The maximum entropy assumption is precisely that, an assumption. The nature of the assumption is that there is a high degree of mixing of the population in the space determined by fixing the known macroscopics. Thus, it is important that appropriate macroscopics are chosen for the particular problem. In addition, maximum entropy prediction makes predictions concerning the expected results averaged over multiple runs of the GA. The advantages of deriving the results from statistical mechanics are: the assumptions, and thereby the source of approximations are made explicit, systematic improvement is possible by taking a larger

number of independent cumulants (if the model can be solved), and quantities can be expressed in such a way that the fluctuations about the expected values are small and calculatable[8]. Thus, the approximations can be understood and controlled, at least in principle.

3. Applying the Approach

We now sketch some of the methods used to apply this approach to specific problems. We will describe this for a class of problems. Thus, the definition of the problem will remain somewhat abstract. In the subsequent section, the application to specific problems will be presented. The notation used to describe the genetic algorithm will be presented first, followed by a discussion of methods of treating the specific genetic operators: selection, mutation, and crossover.

3.1 The Genetic Algorithm

The GA will consist of a population of binary strings of length *N*, with the population size denoted *P*. Each component of the strings take two values ± 1. The strings are denoted x_i^α where the subscript denotes a components of a given string, and the superscript labels the different strings.

The genetic algorithm is going to be attempting to optimize an objective function which is a function of the x_i^α's. This will be written E^α and will be referred to as the *objective function.* The genetic algorithm will be attempting to *minimize* this. Note: in this paper, the terms "objective function" and "fitness" refer to different things. The objective function is the measure of how good a particular string is at solving a problem. The fitness describes how each string is selected. Thus, the objective function is a property of the string, and has meaning independently of the GA (e.g. for a traveling salesperson problem, the objective function could be the length of a tour). The fitness is a property of the particular selection strategy employed in the genetic algorithm; it will be defined below. We hope this will not cause confusion.

We assume generational, fitness-proportional, roulette-wheel selection. The probability of selecting a string α is determined by the fitness F^α by the relation

$$p^\alpha = \frac{F^\alpha}{\sum_{\alpha'} F^{\alpha'}}. \qquad \text{Equation 1}$$

The particular form of selection is determined by the functional relationship between the objective function and the fitness,

$$F^\alpha = F(E^\alpha). \qquad \text{Equation 2}$$

(To avoid overabundance of notation, *F* is used for functional relation and, if superscripted, as the fitness of the string.) Most of our studies consider Boltzmann selection, in which

$$F(E) = \exp(-\beta E). \qquad \text{Equation 3}$$

Here β is a parameter controlling the amount of selection pressure. This form of selection has several advantages: it always yields a positive number so that suitably normalized it can be interpreted as a probability, it is unaffected by redefinitions of the zero of the objective function, the selection strength can be controlled explicitly by a parameter, and in an infinite population, a normally distributed objective function remains normally distributed with only a shift in mean. It also has the appropriate multiplicative nature of a probability; independent additive contributions to the objective should increase the probability of survival independently (thus muliplicatively). Thus, it is commonly used in quantitative genetics. However, other

forms of selection can be treated in this formulation. For example, truncation or culling selection would require,

$$F(E) = \begin{cases} 1 \text{ ; if } E > E_0 \\ 0 \text{ ; otherwise} \end{cases} \qquad \text{Equation 4}$$

for the appropriate value of E_0, and ranking and tournament selection can also be treated. A survey of selection schemes in this approach will appear in Rattray[18]. We emphasis that this is fitness proportional selection, but not objective-function proportional selection. We use this notation because from a biological point of view, increase of frequency in a population *defines* fitness.

Mutation and crossover in the GA will take usual forms. Mutation changes the sign of the components of strings independently with probability μ. Crossover takes two parents and produces two offspring. Crossover probability is taken to be *1*. We consider uniform crossover here; single-point crossover has also been considered[7].

3.2 The Macroscopic Description of the GA

There are two types of measures of the strings which will be used the characterize the population. The statistics of these are the measures of the population which it is our goal to predict. First, define a quantity h^α, to be referred to as the *field*, as

$$h^\alpha = \sum_{i=1}^{N} w_i x_i^\alpha \ . \qquad \text{Equation 5}$$

Here w_i are weights which are fixed for the given problem. The weights determine the relative importance of each bit for that problem. We will consider problems for which the objective function is a function of the quantity h^α,

$$E^\alpha = E(h^\alpha) \, . \qquad \text{Equation 6}$$

Different problems and different instances of the problem will determine the values of the w_i's and the form of *E(h)*. (Not all problems that have been considered in the statistical mechanics approach have this form, e.g. the spin-glass chain[7].)

We will be interested in measures of the distributions of fields in the population. The first four cumulants will be used, where κ_n denotes the *n*th cumulant. The first cumulant is the population mean

$$\kappa_1 = \frac{1}{P} \sum_{\alpha=1}^{P} h^\alpha \ , \qquad \text{Equation 7}$$

the second cumulant is the population variance,

$$\kappa_2 = \frac{1}{P} \sum_{\alpha=1}^{P} \left(h^\alpha - \kappa_1 \right)^2 \ , \qquad \text{Equation 8}$$

the third cumulant is related to the skewness, and the fourth is related to kurtosis (see appendix A for an introduction to cumulants). Note that these are averages over the actually population, not over all possible strings. Thus, they characterize the population and will change with time.

A measure of the similarity between the strings in the population will also be required. The correlation $q^{\alpha\beta}$ measures the similarity between the two strings α and β, weighted by the relative importance of each site

$$q^{\alpha\beta} = \frac{1}{N}\sum_i x_i^\alpha x_i^\beta . \qquad \text{Equation 9}$$

and the average correlation, q, is given by

$$q = \frac{1}{P(P-1)}\sum_\alpha \sum_{\beta\neq\alpha} q^{\alpha\beta} . \qquad \text{Equation 10}$$

In summary, the goal is to determine how the cumulants of h and how q change with time. We now describe how the effects of the genetic operators on these quantities is computed.

3.3 Sketches of Calculational Details for Genetic Operators

3.3.1 Selection

If $\rho(h)$ is the distribution of fields in the population before selection, one might expect that the distribution after to be proportional to $\rho(h)F[E(h)]$, that is the probability of a value of the field being in the population times the probability of selecting it. That is not correct in a finite-size population, because it ignores sampling fluctuations. To compute the effect of selection on statistics of the fields properly, first consider the generating function for moments,

$$Z_s(\gamma) = \sum_{\alpha=1}^{P} F\left[E\left(h^\alpha\right)\right]\exp(\gamma h^\alpha) \qquad \text{Equation 11}$$

Here γ is meaningless parameter; differentiation of $Z_s(\gamma)$ in terms of γ generate moments of h. See appendix A for an overview of generating functions. The logarithm of this quantity is the generating function for the distribution of the fields after selection, (distinguished by the superscript s)

$$\kappa_n{}^s = \frac{\partial^n}{\partial\gamma^n}\log Z_s\bigg|_{\gamma=0} . \qquad \text{Equation 12}$$

Selecting involves drawing P values of h from its distribution independently and with replacement. Thus, the generation function after selection in terms of the field distribution before selection is

$$\left\langle \log Z\right\rangle_{\rho(h)} = \left(\prod_{\alpha=1}^{P}\int_{-\infty}^{\infty}\rho(h^\alpha)dh^\alpha\right)\log Z_s \qquad \text{Equation 13}$$

Here, the angled bracket means average over the distribution $\rho(h)$. Equation 14, with Z_s as defined in equation 15, is the fundamental result for calculating the effect of selection on the distribution of any quantity which is functionally related to the objective function. It contains the distribution before selection in $\rho(h)$ and predicts the cumulants of this distribution after selection by taking derivatives in terms of the dummy variable γ.

Equation 16 looks quite hopeless to actually use, as it involves a P-fold integration. However, using a trick employed by Derrida[19] for the Random Energy model, this can be transformed into a two-dimensional integral. This can be calculated very accurately

numerically, or asymptotic analysis can be performed on it. For example, for large population size and selection not too strong (which can be defined precisely),

$$\left\langle \log Z_s \right\rangle_{\rho(h)} = \log P\left\langle F(h)\exp(\gamma h)\right\rangle_{\rho(h)} - \frac{1}{2P}\left(\frac{\left\langle F(h)^2\exp(2\gamma h)\right\rangle_{\rho(h)}}{\left\langle F(h)\exp(\gamma h)\right\rangle^2} - 1\right)$$

$$+O(P^{-2}) \qquad \text{Equation 17}$$

Here *F(h)* means *F[E(h)]*. The first term is precisely the generating function one gets in the infinite population limit, which is the generating function for the distribution $\rho(h)F(h)$, as suggested above. The next term is the leading finite-population correction. For example, this gives the well-know reduction in genetic variance of *1-1/P* due to genetic drift in the weak selection limit. Taylor expanding the above in terms of γ gives all of the cumulants of $\rho(h)$ after selection in terms of averages of *F(h)* over $\rho(h)$ before selection. These averages can be expressed in terms of the cumulants before selection either by averaging over a $\rho(h)$ which has the correct known cumulants[7] or, if *F(h)* is Taylor expandable, by expressing the averages in terms of the cumulants before selection directly.

Selection acts on the objective function, so its effect on this quantity or a quantity which is functionally related to it, like *h*, can be calculated in problem-independent form without the need for assumptions, as shown above. The correlation measure *q*, however, is not directly related to the objective function. A method for calculating the effect of selection on the correlation was recently developed[5]. There are two contribution to this. First, there is a change in the correlation due to that fact that selection duplicates strings. Strings are completely correlated with themselves, so duplication increases the population correlation. Additionally, as the mean and other cumulates of the objective function change, the expected correlation will change. For example, if the population has a very high fitness, the population may have to be very correlated due to the fact that there are few configurations with high fitness, so most of the strings must be very similar. This is determined from maximum entropy. This can be seen by considering selection as producing a multinomial distribution. Let n^α be the number of times that the string labeled α is selected. The correlation after selection will be

$$q^s = \frac{1}{P(P-1)}\sum_{\alpha=1}^{P} n^\alpha\left(n^\alpha - 1\right) + \frac{1}{P(P-1)}\sum_{\alpha=1}^{P}\sum_{\beta\neq\alpha}^{P} n^\alpha n^\beta q^{\alpha\beta} \; . \qquad \text{Equation 18}$$

The first term is the duplication component, it is caused by the correlations between duplicated strings (which have a correlation of *1*). Averaging this yields,

$$q^s = \sum_{\alpha=1}^{P} p^{\alpha\,2} + \sum_{\alpha=1}^{P}\sum_{\beta\neq\alpha}^{P} p^\alpha p^\beta q^{\alpha\beta} \; . \qquad \text{Equation 19}$$

Here p^α is the probability of selection, as defined in Equation 1. The expected value of q^s is tricky to compute; the details can be found in reference 5. The essential thing to note is that the second term is a maximum entropy value; the first takes into account the effect of correlation through duplication. The duplication term is a finite-population effect, but it is very important, as it makes a major contribution to the correlation of the population, and this, the decreasing effectiveness of crossover. It is particularly important to include this term when there are multiple optima, as the maximum entropy prediction would yield an expectation of equally occupied optima (assuming identical properties at each peak).

3.3.2 Mutation

Mutation takes a bit of a string x_i^α and replaces it with $-x_i^\alpha$, with probability μ. It is straightforward to average over such changes. The effect of mutation on the cumulants of can be shown to be

$$\kappa_1{}^m = \kappa_1 + 2\mu\left(\kappa_1^{\text{random}} - \kappa_1\right), \qquad \text{Equation 20}$$

$$\kappa_2{}^m = \kappa_2 + 4\mu(1-\mu)\left(\kappa_2^{\text{random}} - \kappa_2\right), \qquad \text{Equation 21}$$

$$q^m = (1-2\mu)^2 q, \qquad \text{Equation 22}$$

and so on. The superscript "random" refers to the cumulants of a random population, which is the population found by repeated applications of mutation. The expressions for these cumulants are easily calculated.

3.3.3 Crossover

The effect on the cumulants of the field distribution can be computed in an analogous fashion. For problems for which the objective function is a function of the field, uniform crossover is the natural form to use; results are quoted here for that form. Assuming that both children are retained at each crossing, the distribution of bits at each site is preserved after crossover. Thus crossover has no effect on the mean of h, κ_1, or on the correlation q. However, crossover does effect the higher cumulants,

$$\kappa_2{}^c = \kappa_2 - \frac{P}{2(P-1)}\left[\kappa_2 - \frac{N(1-1/P)}{4}(1-q)\right]. \qquad \text{Equation 23}$$

The effectiveness of crossover to increase diversity is related how similar the strings are, as measured by the correlation q. Equations for higher cumulants can also be derived; the higher cumulants are returned by crossover to "natural" values which are those of a maximum entropy population.

4. Example Systems

In general, to apply these techniques, first one must decided what macroscopics will be used to characterize the system. The initial population is taken as random, thus those macroscopics must be calculated for a random population. Finally, the equations which express the effects of each operator on the macroscopics are used to evolve the system. If any quantity which is not trivially related to the chosen macroscopics appears in the results, those are calculated from a maximum entropy assumption by using the statistical mechanics model. Several problems have been studied using the approach; typically predictions of the theory agree with computer simulations to within a few percent, when averaged over many runs.

Taking as the macroscopics those described above, interesting toy problems can be studied by changing the objective function. In general what happens is as follows. Selection can increase the mean of h if it is not at its optimal value, and it reduces the variance. Selection can also cause an increase of the higher cumulants which is a consequence of the fact that the extremes cannot increase more than the current best. Selection also increases q, correlating the population. Crossover does not affect the mean of h or the mean objective function, as that is a function of h. Crossover has its effect through higher cumulants. It increases the variance,

mediated by q and decreases the magnitude of the higher cumulants. Mutation, of course, moves all cumulants towards their random values.

4.1 Non-interacting Alleles - The Random Paramagnet

The simplest problem for a GA is one in which there is no interaction between the sites of the strings. In this case we take

$$E^{\alpha} = -h^{\alpha} = -\sum_i w_i x_i^{\alpha} \ . \qquad \text{Equation 24}$$

This is like the well-studied one's counting or one-max problem, except each site of the string contributes a different amount to the objective function. If we take the weight w_i to be positive random numbers, there is a single optimum - the configuration with all x_i's equal to one.

Because there is a single optimum, it is easier to infer the correlation from the field distribution statistics. In the original studies of this problem, four cumulants were evolved[7]. Selection is directional, it increases the mean at each generation until convergence. Selection also increases skewness, as the tail of the distribution for good values of the objective function cannot be increased past the value of the best, but there is a long tail on the suboptimal side of the mean. Crossover affects search by reducing higher cumulants including the skewness, thereby populating the tail on the optimal side of the mean. It was also shown in that work that a two cumulant description does not include finite-population effects; higher cumulants are required to get these effects.

4.2 Directional and Stabilizing Selection - The Subset Sum problem

Problems with no interactions between sites and only a single local optimum are not very good examples of optimization problems, as they are trivial to solve. A more realistic example was considered in reference 5 - the subset sum problem. This problem is defined by a set of weights w_i, which are positive numbers. The task is to find a subset of the weights whose sum is as close as possible to a goal value G. Thus, the objective function can be expressed

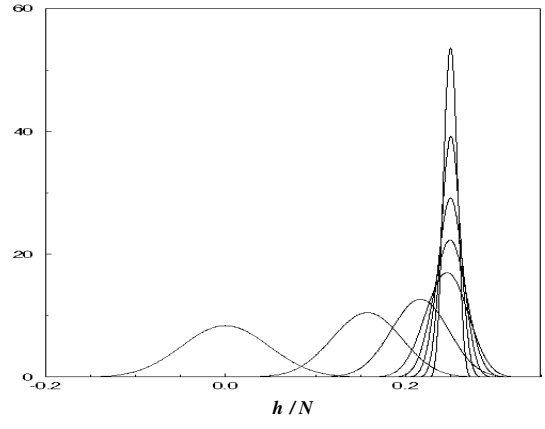

Figure 1- Evolution of the GA for the subset sum problem averaged over 500 runs. The distribution of fields in the population is shown at generation 0, 10, 20, 30, 50, 80, 110, and 140. The goal value, G/N, is at 0.25. Population size P is 80, string length N is 150. The two stages of search - directional and stabilizing - can be seen.

$$E^{\alpha} = \left(h^{\alpha} - G\right)^2 . \qquad \text{Equation 25}$$

This problem is NP-complete, although only weakly so, as it can be solved for most cases by traditional methods in polynomial time[20]. Thus, this is still a toy problem, but it is related to more complex problem like bin-packing, to which GAs have been successfully applied.

An interesting feature of this problem is that it exhibits both directional selection - during which the mean of the field moves towards the goal value *G*, and stabilizing selection - during which the mean of the field stays constant near its goal value but the variance decreases as the members of the population close in on the goal value. This is shown in Figure 1.

The macroscopics used to characterize this problem were the first two cumulants of *h* and the mean correlation *q*. The argument for the importance of the correlation is that a system with many peaks or local optima, it is particularly difficult to infer the correlation from the fitness values. By evolving the correlation, and particularly keeping track of the effect of duplications, this three parameter dynamics accurately captures the genetic algorithm. The accuracy is shown in figure 2. Typical results show agreement between theory and experiment with a few percent.

4.3 Noisy Fitness Evaluation - Training a Neural Network

Genetic algorithms are often used for training neural networks or other supervised learning models. We have recently applied the statistical mechanics theory to an extremely simple neural network learning problem - training a simple perceptron with binary weights[6]. Another

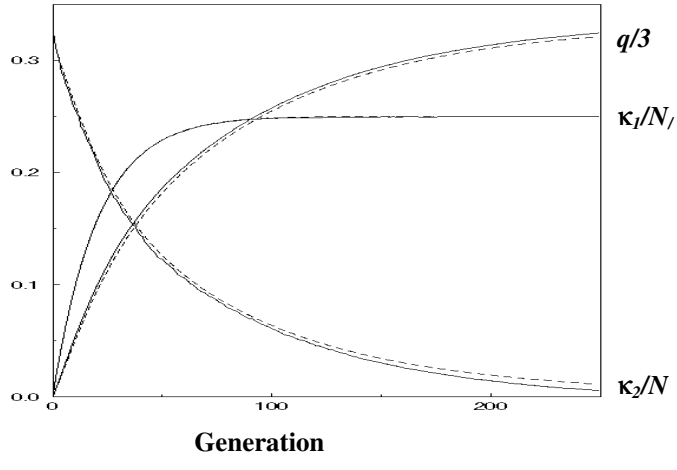

Figure 2 - A comparison between the predictions of the theory (dashed line) and computer simulations (solid line) for the subset sum problem. Population size is 80, string length is 150; G/N = 0.25. Results are averaged over 500 runs.

very interesting theoretic approach to this problem was developed by Baum[21]. This problem has an interesting feature. It is not possible to determine the performance of a potential solution to a problem exactly. One can only estimate the true error of a neural network from a testing set; as long as the testing set is only a sample of all possible patterns, there will be error

in this estimate. Thus, if we take as the objective function the generalization error of the network (i.e. the number of errors it makes on patterns different from those in the trainging set), it will not be a deterministic function of the string. Thus, the objective function is determined by a probability, $P(E^{\alpha}|h^{\alpha})$. This is a probability of getting a generalization error E^{α} given the field value h^{α}. For a simple perceptron, the performance is determined by the correlation between the weights of the network and a network which performs perfectly. This correlation is the meaning of the field.

It is possible to show that for any population size in the noiseless case, there is a population size in the noisy case on which selection has the same result. In other words, it is possible to remove the effect of this noise by suitable increasing the population size. The size of the noise depends upon the properties of the population, so this population rescaling occurs dynamically. For this problem, this can be done by changing the size of the training set for each generation.

For this problem it is also possible to optimize the computational cost. The bigger the training set, the smaller the noise so the smaller the population size need be, but the more computations required to evaluate the objective function. The effectiveness of the prediction is shown in Figure 3.

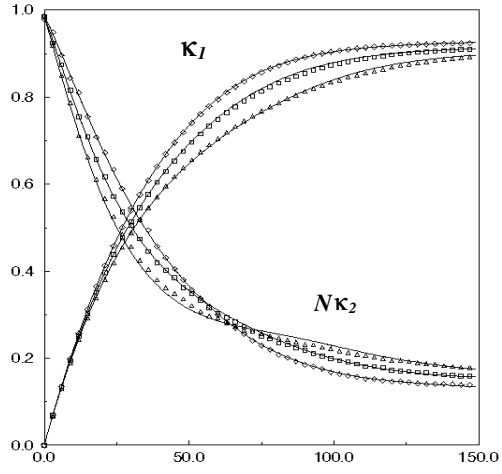

Figure 3 - Comparisons between theory and experiments for learning in a binary perceptron, averaged over 1000 runs. The solid lines are the theory for different size training sets. Here N is 155, population size is 80. This curve is from reference 6.

5. Directions for Future Research

There are two direction in which the work needs to go. First, the problem which the search problems described above is that there is no interesting spatial structure to the interactions between sites. Although sites do interact if *E(h)* is a non-linear function, the interactions are

homogeneous. Each sites interactions with each other (statistically) equally. Real problems have much richer structure, for example the structures of three dimensional graphs. Nearest site interactions have been considered in reference 7, but this is the most trivial case. To put in spatial structure will involve solving a statistical mechanics model for the maximum entropy calculation with spatial correlations as constraints. That will be a most challenging computation.

It might also be interesting to try to develop ways to apply this approach to problems which cannot be solved. Methods for choosing search parameters at each generation have already been proposed, but have not been tested. It would be most interesting to develop methods for using these ideas to try to model and improve search at each timestep, without a full solution to the problem, by taking some input from the actual GA run.

Appendix A. - Generating Functions for Cumulant Expansion

Let $\rho(x)$ be a distribution. One way to calculate moments of this distribution is to evaluate a generating function for the moments.

$$Z(\gamma) = \int \rho(x) \exp(\gamma x) dx$$

This generates the moments of the distribution because derivatives of Z with respect to γ determine the moments

$$m_n = (-1)^n \frac{\partial^n}{\partial \gamma^n} Z \Big|_{\gamma=1} .$$

Cumulants are related to moments; however, those higher than the second are measured relative to Gaussians for which all cumulants higher than the second are zero. The generating function for cumulants is the log of the generating function for moments.

---

[1] Mitchell, M. Forrest, S. and Holland, J. 1991. "The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance", in *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life* (MIT Press: Cambridge, Ma).

[2] Radcliffe, N. J. 1991. *Complex Systems* **5** 183.

[3] Prügel-Bennett, A. and Shapiro, J. L. 1994. *Physical Review Letters* **72** 1305.

[4] Shapiro, J. L. and Prügel-Bennett, A. 1995. *Lecture Notes in Computer Science* **993** 14.

[5] Rattray, M. L 1996. "The Dynamics of a Genetic Algorithm under Stabilizing Selection" *Complex Systems* (to appear).

[6] Rattray, M L and Shapiro, J L 1996. "The Dynamics of Genetic Algorithms for a Simple Learning Problem" (submitted to *Journal of Physics A*).

[7] Prügel-Bennett, A. and Shapiro, J. L 1996. "The Dynamics of Genetic Algorithms for Simple Random Ising Systems" (submitted to *Physica D*).

[8] Prügel-Bennett, A. 1996 "Modelling Evolving Populations" (submitted to *J. of Mathematical Biology*).

[9] Goldberg, D E 1989 "Genetic Algorithms in Search, Optimization, and Machine Learning" (Addison-Wesley, Reading, Ma).

[10] Ewens, W. J. 1979, "Mathematical Population Genetics" (Springer).

[11] Crow, J.F. and Kimura, M. 1970, "An Introduction to Population Genetics Theory", (Harper and Row:New York).

[12] Vose, M. D. and Liepins, G. E. 1991. *Complex Systems* **5** 31.

[13] Nix, A. and Vose, M. D. 1991, *Annals of Mathematics and Artificial Intelligence* 5:79 88.

[14] Goldberg, D E, Deb, K and Clark J H 1992 ***Complex Systems* 6** 333.

[15] Mülenbein, H and Schlierkamp-Voosen D 1995 *Lecture Notes in Computer Science* **899** 142.

[16] Bulmer, M.G. 1980, "Mathematical Theory of Quantitative Genetics", Claredon Press: Cambridge

[17] Cover, T. and Thomas, J. 1991. "Elements of Information Theory" (Wiley).

[18] Rattray, M. L. 1996. University of Manchester Ph.D. Thesis. In preparation.

[19] Derrida, B. 1984. Physical Review B **24** 2613.

[20] Garey, M R and Johnson, D S 1979. "Computers and Intractability - A Guide to the Theory of NP-Completeness" (W. H. Freeman and Co: San Francisco).

[21] Baum, E. B., Boneh, D. and Garret, C. 1995. in *Proceedings of the 8th Annual Conference on Computational Learning Theory* p 230.