Maximum Entropy Analysis of Genetic Algorithm Operators

Jonathan L. Shapiro[1] and Adam Prügel-Bennett[2]

[1] Department of Computer Science
University of Manchester Manchester, M13 9PL, UK
[2] NORDITA Blegdamsvej 17, DK-2100 Copenhagen Ø, Denmark

Abstract. A maximum entropy approach is used to derive a set of equations describing the evolution of a genetic algorithm involving crossover, mutation and selection. The problem is formulated in terms of cumulants of the fitness distribution. Applying this method to very simple problems, the dynamics of the genetic algorithm can be reduced to a set of nonlinear coupled difference equations which give good results when truncated to four variables. These equations correctly predict the best fitness after convergence as a function of the genetic search parameters. Suggestions concerning annealing rates for time-dependent search parameters are also discussed.

1 Introduction

Genetic Algorithms are increasingly used to search for solutions to hard problems. Genetic algorithms are used for a variety of reasons: they can be naturally applied to discrete problems where other techniques are more difficult to use, they parallelize well, and, most importantly, they have been shown to work in a number of applications.

Although a great deal of understanding about how genetic algorithms work has come from empirical observations, genetic algorithms are not well understood theoretically. This may be a controversial view, because many theoretical approaches have been proposed. These include static studies of the fitness landscape, such as hyperplane analysis [1] and deception analysis [2], Markov chain analysis [3, 4], and direct methods for small or special problems (e.g. [5]), and others. Although these techniques may be useful in providing insight as to how GAs work (or in some cases, it has been argued [6], erroneous insight), it is still difficult to get quantitative results concerning the effectiveness of GA search for different parameter settings, for different types of problems, or for different types of representations. This is an important goal of a theoretical description of genetic algorithms.

In previous papers [7, 8], we proposed a "statistical mechanics" approach to the study of genetic algorithm dynamics. This approach was fairly crude and approximate, although we found good agreement with simulations in the cases we tested. We have since developed a more rigorous formulation [9] and better understanding of the assumption which was implicit in that approach; this is basically a maximum entropy assumption. Although, we have only considered simple problems and concentrated our efforts in deriving the dynamical equations, the results suggest that this formulation can be used to study the effectiveness of GA search and suggest new strategies. Some results of that work are described here.

2 The Maximum Entropy Ansatz

Ideally, one would like to solve the dynamics of genetic algorithms exactly and rigorously. This could be done, in principle, either by studying the stochastic equation directly, or by using a Markov Chain formulation to analyse the deterministic equation for the probability of being in a given population. Both of these approaches have been extensively studied (see references above). However, it is very difficult to make progress viewing the problem in this way, because one must solve a high-dimensional, strongly interacting, nonlinear system which is extremely difficult to analyse. The high-dimensionality is because one must deal with a population of variables; selection and particularly crossover introduce the strong nonlinear interactions between these variables.

In both of these approaches, you must consider precisely which individuals are in the population, or you must index each possible population, and this contributes to the complexity of these methods. In our approach much less information about the population is assumed to be known. Only the behaviour of the distribution of fitnesses in the population is considered. For example, imagine that you have a histogram of fitnesses in the current population. With that information alone, we propose to predict the fitness histogram after mutation, crossover, and selection. (More than that, we predict the fitness distribution after any arbitrary number of generations).

It is clear that the histogram of fitnesses does not provide all of the information. It does not determine the individuals which constitute the population. For example, suppose you know that all of the strings in the population have a fitness 100, i.e. the fitness distribution is a delta-function. It is possible that all of the individuals in the population are identical; alternatively, they could all be quite different except that they are equally fit.

We infer the constituents of the population by ansatz – we assume that the distribution of strings within the population given the fitness distribution maximizes the entropy. That is, of all the populations which have the same distribution of fitnesses, the more numerous ones are most likely to describe the actual one (see, for example, [10] for a general description of the maximum entropy method). In earlier work, we have called this a statistical mechanics approach to studying GAs. This is in analogy with the way the statistical mechanics infers properties about microstates such as energy distributions from thermodynamics. We need to make clear that this is not simply a formalism applied to a problem; it is an assumption about the likelihood of each of the populations. The assumption is likely to be true if the genetic operators are sufficiently mixing to make all populations with given fitness properties equally likely, and if the fitness properties chosen are appropriate in describing the system. Some direct numerical tests appear in [9] and show that the assumption is very good (but not perfect).

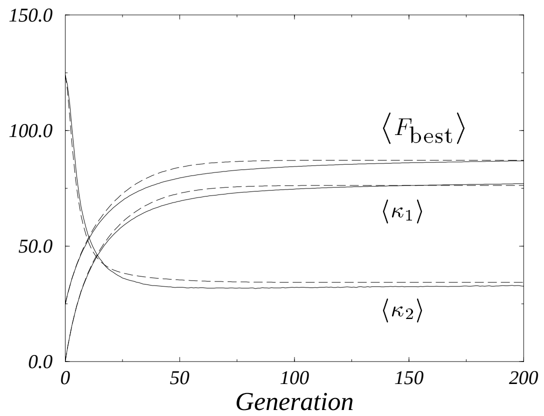

Fig. 1. The solid curves show the evolution of the first two cumulants and the fitness of the best member of the population. These curves were created by averaging over 10 000 samples for $N = 127$, $P = 50$, $\beta_o = 0.1$ and $p_m = 1/2N$. The ground state is at $F_{\text{best}} \approx 101$. The dashed curve shows the theoretical prediction.

The long term goal of this work is first to use the maximum entropy ansatz to derive equations predicting the fitness distribution at time $t+1$, in terms of that at time t, and then to solve or study these equations to learn about effectiveness of GAs for given parameters or representations for given problems. The first step has involved considerable effort, and the bulk of our work thus far has been in carrying out these derivations.

There are several advantages to the maximum entropy approach. It reduces the dynamics of the GA to a few variables. We have considered a representation which reduces to four coupled nonlinear difference equations. This is a considerable improvement over the NP variables in the initial problem of P strings of size N. Figure 1 shows the result of solving the four variable equation as the theoretical curve as compared to simulation, where P is 50 and N is 127. There are no restrictions on the length of the strings or the size of population studied, although the approach is most appropriate in the large string limit. It is intrinsically dynamical – the effect of the genetic operators are determined by current state of the system predicted by the dynamics and inferred from the ansatz. Thus, it does not make the errors which static deception analysis makes, that the probability of finding a particular state depends on properties of the fitness and not where the actual the population is.

There are disadvantages. Significant analysis is required simply to derive the equations describing the dynamics of the GA, even prior to attempting to find solutions to these equations. Thus, it may be impossible to apply to realistic problems. Although the method works in the cases we have looked at, there is no argument that the ansatz must be satisfied. It may not be for classes of problems; we do not know.

In addition to the maximum entropy ansatz, there are two sources of approximation in our approach. First, to characterize the fitness distribution, we use a cumulant expansion. The first two cumulants are the mean and the variance of the distribution respectively, the third and fourth are related to the skewness and kurtosis respectively; they measure the difference of the third and fourth moments from Gaussian moments. The cumulant expansion is exact when an infinite number of cumulants is used. However, we truncate this expansion, typically to four cumulants. This approximation will be good if the fitness distribution is similar to a Gaussian, as it is in our problems (see figure 2). However, if the fitness distribution should become two-peaked or strongly non-Gaussian in another way, this approximation will break down. The cumulants are denoted κ_n for the nth cumulant.

A second source of error is sample to sample fluctuations. We calculate average behaviour. The dynamical equations for the cumulants are nonlinear, and nonlinear functions of cumulants do not necessarily self-average. In figure 1, is is apparent that the theory is qualitatively the same as the simulation, but they disagree quantitatively for intermediate times (These results are typical of the parameters and the problems we have looked at). At least some of this is due to sample-to-sample fluctuations, because we can derive equations for the nonlinear functions of the cumulants which remove this discrepancy. As the asymptote and the convergence rate is predicted correctly, we do consider the results to be useful without attempting to fix this.

3 Test Problems

The method has been applied to four problems. The simplest task is one with zero epistasis, *counting random numbers.* In this problem, each bit of the string contributes individually an arbitrary amount. The fitness is,

$$F[\boldsymbol{\tau}] = \sum_{i=1}^{N} J_i \tau_i .$$

Here $\boldsymbol{\tau}$ represents the string of length N, and J_i is a random number (usually drawn from zero-mean, unit-variance Gaussian distribution).

Weak interactions between bits can be introduced by considering the task of maximizing the energy of a *spin-glass chain.* Here the fitness is

$$F[\boldsymbol{\tau}] = \sum_{i=1}^{N} J_i \tau_i \tau_{i+1} .$$

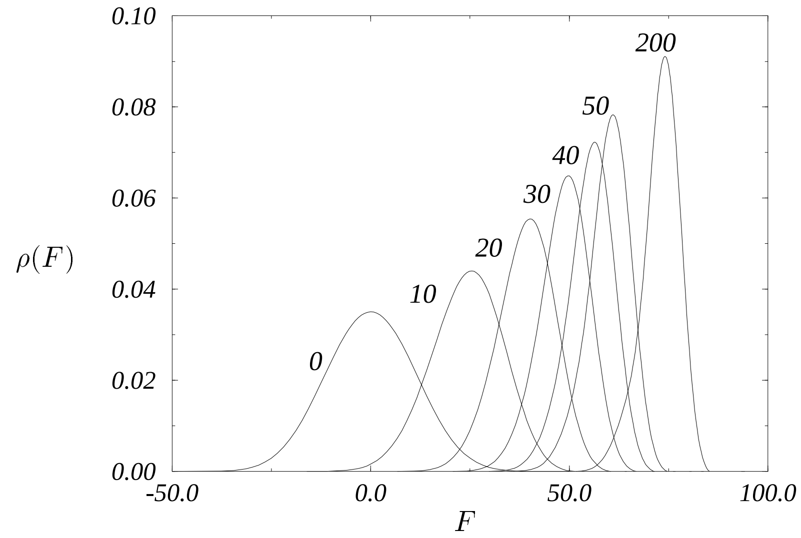

Fig. 2. The distribution of fitness is shown after 0, 10, 20, 30, 40, 50 and 200 steps of the GA applied to the spin-glass chain for the same set of simulations as those used to produce figure 1

This is an easy task in that, although there are enormous numbers of local minima, the barriers between them are small. One can make this systematically more difficult by making increasing the number of τ's in the interaction from two, or by increasing the complexity of the chain from the chain to a planar graph to a three dimensional graph and so forth. The latter makes the problem NP-complete, and we have not yet looked at it. The two tasks are analysed in great detail in [9].

A third task to which this approach has been applied is the task of training in a binary perceptron[11]. This is a truly hard problem for which there is no known efficient algorithm; simulated annealing and genetic algorithms give the best results. There has been progress, but not complete success, toward deriving the dynamical equations for this problem. The techniques have also been applied to the subset sum problem[12], an NP-complete problem which can be solved by pseudo-polynomial algorithms. These results will not be discussed in this paper.

4 Genetic Search Operators

We now discuss some of the results which have been found concerning the genetic search operators. We consider the simplest form of a genetic algorithm consisting of a form of roulette wheel selection, bitwise mutation and single-point crossover. There is a mutation rate and a selection rate, but for simplicity all strings participate in crossover and each pair of parents produces a pair of offspring. Changing the rate of crossover is easily done in a fuller treatment.

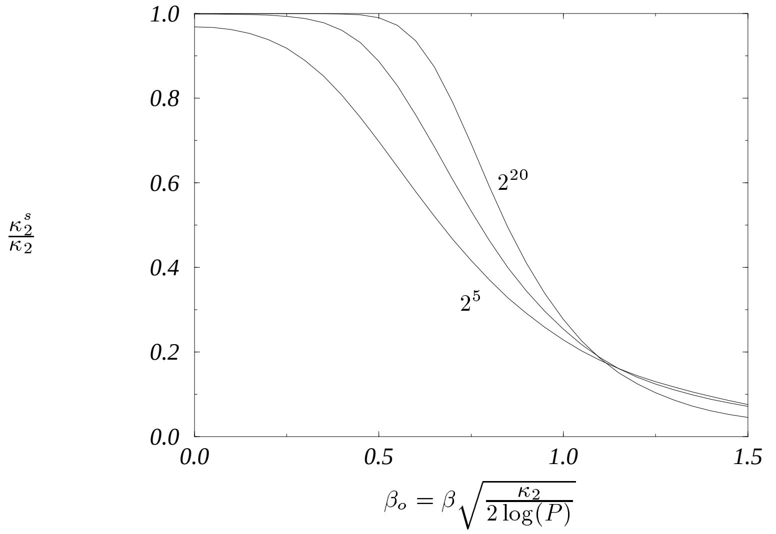

Fig. 3. The change in width of fitness distribution after selection starting from a Gaussian distribution for $P = 2^5$, 2^{10} and 2^{20}. The solid lines are calculated by numerical integration.

4.1 Selection

Selection has been described in detail elsewhere [7, 8], so will be considered briefly here. We have studied Boltzmann selection in which each member of the population is selected with probability

$$p_\alpha = \frac{\mathrm{e}^{\beta F_\alpha}}{Z}, \qquad Z = \sum_{\alpha=1}^{P} \mathrm{e}^{\beta F_\alpha}. \tag{1}$$

where β is determines the strength of selection and F_α is the fitness of string α.

The effect of selection truly depends only on the fitness distribution in the population, so the results involve no assumptions and hold independent of the problem. The analysis revealed how β should scale as the evolution occurs and the population converges, namely,

$$\beta = \beta_o \sqrt{\frac{2\log P}{\kappa_2}} \tag{2}$$

where β_o is constant. The equation for the first two cumulants due to selection, for example, is

$$\kappa_1^s = \kappa_1 + f_1(\tilde{\beta}_s)\sqrt{\kappa_2} \tag{3}$$

$$\kappa_2^s = f_2(\tilde{\beta}_s)\kappa_2 \tag{4}$$

where $\tilde{\beta}_s = \sqrt{2\log(P)}\beta_o$. This shows that selection increases the mean fitness in proportion to the standard deviation and decreases the variance by failing to select certain individuals. All higher cumulants or indeed the distribution function itself can also be calculated.

The functions $f_1(\tilde{\beta}_s)$ and $f_2(\tilde{\beta}_s)$ are functions of the selection parameter. Their form suggests how to chose an optimal selection rate. Figure 3 shows the function $f_2(\tilde{\beta}_s)$; $f_1(\tilde{\beta}_s)$ is the integral of $f_2(\tilde{\beta}_s)$. Since $f_2(\tilde{\beta}_s)$ is initially flat while $f_1(\tilde{\beta}_s)$ grows, the increase in average fitness can be made with almost no loss of diversity. However, when the selection parameter reaches the point where $f_2(\tilde{\beta}_s)$ starts to fall off, diversity starts to be lost. Thus, under the assumption that the best selection gives the biggest improvement with the least loss of diversity, an optimal selection rate is predicted.

4.2 Mutation

We have calculated the effect of mutation on the first four cumulants. The first two cumulants can be computed rigorously. The effect of mutation on the higher cumulants depends upon the configurations of the populations, so the maximum entropy ansatz is required for these.

To show how this reveals properties of genetic search, consider the first two cumulants only and only selection and mutation together (no crossover). By considering only two cumulants, we are making a Gaussian approximation. This must break down as the optimal state is approached (since there can be no high fitness tail there), and will be cumulatively worse for later generations. Nonetheless, it is a useful first approximation.

The effect of mutation on the first two cumulants is to change them by an amount,

$$\begin{aligned} \kappa_1^m &= \Gamma\kappa_1 \\ \kappa_2^m &= \kappa_2 + (1-\Gamma^2)(N-\kappa_2) \end{aligned} \tag{5}$$

where $\Gamma = 1-2p_m$ for *random-number counting* and $\Gamma = (1-2p_m)^2$ for the *spin-glass chain*, and p_m is the mutation probability. This shows that mutation makes the average fitness worse in proportion to the average fitness. This is because the better the solution the more likely mutation will mess it up. Mutation also drives the variance towards that of a random population.

Applying the selection and mutation operators together produces a change in the cumulants of

$$\kappa_1(t) = \Gamma\kappa_1(t-1) + \Gamma f_1(\tilde{\beta}_s)\sqrt{\kappa_2(t-1)} \tag{6}$$

$$\kappa_2(t) = \Gamma^2 f_2(\tilde{\beta}_s)\kappa_2(t-1) + (1-\Gamma^2)N \tag{7}$$

The population converges to a dynamic equilibrium in which the improvement in the mean fitness due to selection balances the decrease in fitness due to mutation.

It is straight-forward to solve these equations. The equilibrium variance is

$$\kappa_2(\infty) = \frac{(1-\Gamma^2)N}{1-\Gamma^2 f_2(\tilde{\beta}_s)} \tag{8}$$

the equilibrium average fitness is

$$\kappa_1(\infty) = \frac{f_1(\tilde{\beta}_s)\Gamma\sqrt{\kappa_2(\infty)}}{1-\Gamma} \tag{9}$$

The final fitness depends upon selection rate and the mutation rate. The selection optimum suggested in the previous section occurs by initially increasing κ_1 through $f_1(\tilde{\beta}_s)$. However, when the selection rate reaches an optimum, $f_2(\tilde{\beta}_s)$ starts to decreases, making κ_2 and thereby κ_1 smaller.

The fitness of the optimal string is of order N. However, it is apparent from the above equations that the equilibrium fitness will be of order $\sqrt{N}$ unless the mutation rate is very small, of order $1/N$. (In other words, only one or two bits can be changed per string, rather then changing a finite fraction of bits per string). It is interesting to note that if the mutation rate is large the system will reach a point where there is high diversity in the population (variance of order N), but no improvement. When the mutation rate is small (order $1/N$) the system can get much closer to the optimum, at which point there is very little diversity (variance of order 1). Note, however, that the convergence rate for a fixed mutation rate is also related to the mutation rate; it is given by $-\log\Gamma^2 f_2(\tilde{\beta}_s)$. If the mutation rate is small enough to find a good solution, the convergence will be very slow. (It has been shown in the ones-counting problem that the optimal mutation rate is also $1/N$[4].)

Thus, there is a trade-off. Small mutation rate gives slow convergence to a good solution; larger mutation rate gives rapid convergence to a poor solution. Since the equilibrium reached depends only on the mutation rate at the time it is reached, a possible way around this is to use large mutation rates for early times and then anneal the mutation rate.

The equations are easily solved for time-dependent mutation rate. In this case, the convergence rate is

$$\frac{1}{t}\sum_{i=1}^{t} \ln\Gamma(i)^2 f_2(\tilde{\beta}_s).$$

The convergence fitness is determined by the mutation rate at convergence. It is not clear how to find the best annealing schedule to determine how $\Gamma(t)$ should change with time.

A heuristic can be used to set the annealing schedule. At each stage, take the mutation rate which maximizes the best member of the population. In the Gaussian approximation, the maximum fitness in terms of the cumulants is (ignoring logarithmic corrections),

$$F_{\text{best}} = \kappa_1 + \sqrt{2 \ln P \kappa_2}.$$

This shows another form of the trade-off. Mutation decreases the first term but increases the second. At each generation, the mutation rate can be found which maximizes this. For early times this gives a large mutation rate (initially 1/4); for latter times this goes to 0.

This can all be done to four cumulants, although the equations are quite complicated and F_{best} must be worked out numerically. These results suggest optimal values for mutation rate, and suggest an annealing schedule which needs to be tested on realistic problems.

4.3 Crossover

For the zero epistasis task, crossover has no effect on the first two cumulants. For the spin-glass problem, crossover decreases the fitness for the same reason mutation did, at the crossover point it is likely to make the fitness worse in proportion to the average fitness. However, these effects are small. Thus, in the Gaussian approximation, crossover does not appear important, because it only makes the mean slightly worse. For single-point crossover, the reduction in the mean is given by,

$$\Delta\kappa_1 = -\kappa_1 \kappa_2 / N^2.$$

The benefits of crossover come in its effect on the third cumulant. Crossover decreases the third cumulant. This is beneficial, because the third cumulant measures the skewness. The distribution is skewed in that the low fitness tail is longer and the high fitness tail is short. By reducing the skewness, crossover replaces low fitness individuals for high fitness ones.

The effect is seen by considering the equations for three cumulants. We will take the case of the spin-glass chain with single-point crossover, mutation, and selection. The derivations of these equations are quite complicated and can be found elsewhere. For simplicity, we keep only the leading nonlinearity. We assume the crossover rate is unity.

The equations for selection are,

$$\kappa_1^s = \kappa_1 + f_1(\tilde{\beta}_s)\sqrt{\kappa_2} + g_1(\tilde{\beta}_s)\frac{\tilde{\kappa}_3}{\sqrt{\kappa_2}} \tag{10}$$

$$\kappa_2^s = f_2(\tilde{\beta}_s)\kappa_2 + g_2(\tilde{\beta}_s)\tilde{\kappa}_3 \tag{11}$$

$$\tilde{\kappa}_3^s = -f_3(\tilde{\beta}_s)\kappa_2 + g_3(\tilde{\beta}_s)\tilde{\kappa}_3 \tag{12}$$

where $\tilde{\kappa}_3 = \kappa_3/\sqrt{\kappa_2}$ and the f's and g's are selection rate dependent, which can be considered constants for fixed selection rate and population size. The equations for the effect of mutation on the first two cumulants are exact; the equation for the third cumulant is approximately

$$\tilde{\kappa}_3^m \approx \Gamma^3 \tilde{\kappa}_3. \tag{13}$$

Finally, crossover affects the third cumulant by reducing it by a factor of $\frac{1}{2}$. Putting these together, we find that the equilibrium average fitness is

$$\kappa_1(\infty) = \frac{\sqrt{\kappa_2(\infty)}}{1 - \Gamma + \kappa_2(\infty)/(2N^2)} \left(1 + \frac{f_1(\tilde{\beta}_s)}{g_1(\tilde{\beta}_s)} \frac{\tilde{\kappa}_3(\infty)}{\kappa_2(\infty)}\right), \tag{14}$$

where κ_2 and $\tilde{\kappa}_3$ evolve to equilibrium values

$$\kappa_2(\infty) = \frac{(2 - \Gamma^3 f_3(\tilde{\beta}_s))(1 - \Gamma^2)N}{(1 - \Gamma^2 f_2(\tilde{\beta}_s))(2 - \Gamma^3 g_3(\tilde{\beta}_s)) + \Gamma^5 g_2(\tilde{\beta}_s) f_3(\tilde{\beta}_s)} \tag{15}$$

$$\tilde{\kappa}_3(\infty) = \frac{-2\Gamma^3 f_3(\tilde{\beta}_s)(1 - \Gamma^2)N}{(1 - \Gamma^2 f_2(\tilde{\beta}_s))(2 - \Gamma^3 g_3(\tilde{\beta}_s)) + \Gamma^5 g_2(\tilde{\beta}_s) f_3(\tilde{\beta}_s)}. \tag{16}$$

Crossover increases the final fitness found by increasing the size of κ_2 and κ_3, but the effect is small for this problem.

5 Future Work

We have used the maximum entropy method to derive a set of equations describing the dynamics of genetic algorithms for some very simple problems. These equations use the evolution of a few cumulants to predict the average behaviour of a population undergoing genetic search. Some simple forms of these equations have been discussed here.

There are two directions in which work must be done, if this is to contribute to understanding of genetic algorithms. First, having derived the equations for the simple problems and tested them, they must be extensively studied. We have used these equations to suggest parameter choice and annealing schedules qualitatively. These must be demonstrated quantitatively. Second, it would be very useful if more realistic problems could be studied via this approach. This work is ongoing.

References

1. John H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press (Ann Arbor), 1975.
2. David E. Goldberg. Genetic algorithms and walsh functions: Part II, deception and its analysis. *Complex Systems*, 3:153–171, 1990.
3. A Nix and Michael D. Vose. Modeling genetic algorithms with markov chains. *Annals of Mathematics and Artificial Intelligence*, 5:79–88, 1991.

4. H. Mühlenbein. Evolution in space and time – the parallel genetic algorithm. In G. Rawlins., editor, *Foundations of Genetic Algorithms*. Morgan-Kaufman, 1991.
5. Darrell Whitley. An executable model of a simple genetic algorithm. In L. Darrel Whitley, editor, *Foundations of Genetic Algorithms 2*. Morgan Kaufmann (San Mateo), 1993.
6. John J. Grefenstette. Deception considered harmful. In L. Darrel Whitley, editor, *Foundations of Genetic Algorithms 2*. Morgan Kaufmann (San Mateo), 1993.
7. A. Prügel-Bennett and J. L. Shapiro. An analysis of genetic algorithms using statistical mechanics. *Phys. Rev. Letts.*, 72(9):1305–1309, 1994.
8. J. L. Shapiro, A. Prügel-Bennett, and M. Rattray. A statistical mechanical formulation of the dynamics of genetic algorithms. In Terence C. Fogarty, editor, *Evolutionary Computing: AISB Workshop, Leeds, U.K., April 1994, Selected Papers*, volume 864 of *Lecture Notes in Computer Science*, pages 17–27. Springer-Verlag, 1994.
9. A. Prügel-Bennett and J. L. Shapiro. Dynamics of genetic algorithms for the ising spin-glass chain.
10. Thomas Cover and Joy Thomas. *Elements of Information Theory*. Wiley, 1991.
11. Magnus Rattray. An analysis of a genetic algorithm training the binary perceptron. Master's thesis, University of Manchester, 1995.
12. Magnus Rattray. An analysis of a genetic algorithm solving the subset-sum problem. To appear in *Complex Systems*, 1996.

This article was processed using the LaTeX macro package with LLNCS style.