A Statistical Model of GA Dynamics for the OneMax Problem

Bulent Buyukbozkirli[1,3] and Erik D. Goodman[2,3]

[1] Department of Mathematics, Michigan State University
[2] Department of Electrical and Computer Engineering, Michigan State University
[3] Genetic Algorithms Research and Applications Group (GARAGe)
Michigan State University
East Lansing, MI 48824
buyukboz@msu.edu, goodman@egr.msu.edu

Abstract. A model of the dynamics of solving the counting-ones (OneMax) problem using a simple genetic algorithm (GA) is developed. It uses statistics of the early generations of GA runs to describe the dynamics of the problem for all time, using a variety of crossover and mutation rates. The model is very practical and can be generalized to cover other cases of the OneMax, such as weighted OneMax, as well as the deceptive function problem, for *high enough* crossover rates. Proportional selection with and without Boltzmann scaling have been modeled; however the Boltzmann extensions are not described here. In the development of the model, we introduce a new quantity that measures the effect of the crossover operation in the counting-ones problem and is independent of generation, for practical purposes.

1 Introduction

Theoretical models of Genetic Algorithms (GAs) fall into three main categories. The Markov chain model, as developed by Nix, Vose and Liepins [1][2], completely describes the probabilistic behavior of the GA. However, this model is too costly to implement computationally for problems with realistic population size and chromosome length. The statistical mechanics approach, developed by Prügel-Bennett, Shapiro and Rattray [3][4], gives fairly good results in modeling the OneMax problem with Boltzmann scaling, for a crossover rate of 100%, however it is not developed for lower crossover rates or to handle other benchmark problems of GA such as deceptive functions. The approach of modeling GAs by considering building blocks (Goldberg [7] and Goldberg, Deb, Thierens [6]), on the other hand, gives us a good idea about the appropriate population size or the convergence time of the OneMax and help us determine the failure boundaries in the "control maps". But the question of finding the most appropriate crossover or mutation rate is answered, so far, only by experimental results. We still lack a model that describes the behavior of the OneMax problem for different crossover and mutation rates together and allows us to choose the best parameters.

K. Deb et al. (Eds.): GECCO 2004, LNCS 3102, pp. 935–946, 2004.

Studying the OneMax problem is important not for solution of that problem, per se, but because many real-world problems solved via genetic algorithms consist of a set of separable sub-problems for which the optimum is to optimize each individually, which is reminiscent of OneMax.

In this paper, we develop a model that describes the mean allele dynamics of the OneMax problem for very high crossover rates. Then, we modify the model by using statistics of very early generations from GA runs, to describe the complete dynamics for different (lower) crossover rates. The model is developed to estimate the average GA dynamics, but it can be used for an individual run of the GA and has the potential to apply to other cases of GA-based solution of the OneMax problem, such as using Boltzmann scaling, and the weighted OneMax, or to benchmark problems involving deceptive functions. The authors hope to extend the approach to model solution of more representative real-world problems with various degrees of OneMax similarity and various amounts of deception.

2 Problem Description and Visual Representation of GA Dynamics

We consider the simple genetic algorithm in which two-point crossover, fitness-proportional selection and mutation are applied in the order given. We develop a model on the OneMax problem with a population consisting of P chromosomes of length L. Let $S(t)$ be the set of all chromosomes at time t, *chrom* an element of this set, and *chrom(i)* the allele at the i^{th} locus of this chromosome. The fitness of a chromosome, *chrom*, will be denoted as *f(chrom)*, which is equal to $\sum_i chrom(i)$ for the simple OneMax problem. The variables of the population that we are interested in are the mean fitness $\kappa_1(t)$, the variance of the fitness $\kappa_2(t)$, and the set of the mean of the alleles at each locus i $\{\alpha_i(t)\}_{i=1,\ldots,L}$, at time t. Define $A_h(t)$ to be the number of $\alpha_i(t)$'s whose value is less than or equal to h,

$$A_h(t) = \#\left\{ \alpha_i(t) \,\middle|\, \alpha_i(t) \le h, i = 1,\ldots,L \right\}. \tag{1}$$

By this definition, for example, $A_0(t)$ gives the number of loci where all of the chromosomes have value 0, while $A_{0.6}(t)$ gives the number of loci where at most 60% of the chromosomes have value 1.

The values of the variables ($\kappa_1(t)$, $\kappa_2(t)$, $\{\alpha_i(t)\}_{i=1,\ldots,L}$) and $A_h(t)$ change from one GA experiment to another even if we have the same initial population. In terms of experimental results, we run a GA, with fixed parameters of selection, mutation and crossover, many times. For each run of the GA, we measure these quantities at each generation and take the average over all of the runs. The goal of our model is to estimate average values of these variables, hence the average behavior of the GA. In order to simplify the notation, we will use the same symbols ($\kappa_1(t)$, $\kappa_2(t)$, $\{\alpha_i(t)\}_{i=1,\ldots,L}$) and $A_h(t)$ for values of a specific run of a GA, or of an experimental average of these values, or of the estimated theoretical average in our model. Which one is denoted will be clear from the context. We will use the superscripts *c*, *cs* or *csm* in order to distinguish these variables after crossover, selection or mutation is applied,

respectively. So, $\alpha_i^{cs}(t)$ represents the mean of alleles at the i^{th} locus at the t^{th} generation after the crossover and selection are applied, and $\alpha_i^{csm}(t)$ equals $\alpha_i(t+1)$. Note that the crossover operation does not change the mean allele values. Thus, $\alpha_i(t)$ equals $\alpha_i^c(t)$.

The study of the time evolution of $A_h(t)$'s for several values of h gives a very practical insight into the behavior of the GA. Figure (1) shows the graphs of $A_h(t)$ for h = 0, 0.1, … 0.9, the crossover rate is 25% and the mutation rate is 0.1%. The population size is taken as 50 chromosomes and the chromosome length is 100 genes. Each time slice of such a graph can be seen as a "bar graph" of the mean allele distribution at the given instant. In other words, the vertical distance between two curves gives the number of gene locations at which the mean allele is between the corresponding values, averaged across runs. For example, at t=100, at about 18 gene locations, none of the chromosomes (i.e. h=0) have value 1; at about 5 locations from 1 to 5 chromosomes (i.e. 1% to 10% of the population, i.e. 0<h 0.1) have a 1 and the rest have a 0; and at about 50 locations from 45 to 50 chromosomes (i.e. 91% to 100% of the population, i.e., 0.9<h 1) have a 1 and the rest have a 0, etc. The closer the curves are to each other, the smaller the variation in the population. We observe that although the population converges to a more or less stable configuration after 100 generations, there is still some variation within the population due to the existence of mutation, which has the potential of creating new chromosomes.

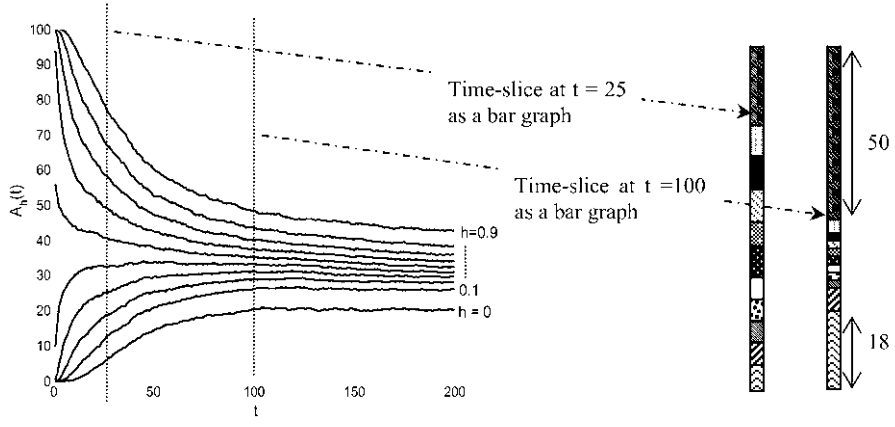

Fig. 1. The experimental average values of $A_h(t)$ as a function of time for h =0, 0.1, 0.2, … 0.9 , and the bar graph interpretation. The population size is 50 and the chromosome length is 100 genes. The GA parameters are p_c = 0.25, p_m = 0.001. The average is taken over 100 experiments

When h is quantized with a gap of 0.1 between two consecutive values as above, we get 10 regions formed between the graphs, including the region above the top graph. We will use the index h' to count these regions, h' = 1, 2, …, 10, given by

$$R_{h'} = \left\{ (y,t) \,\middle|\, A_{(h'-1)/10}(t) \le y \le A_{h'/10}(t) \right\}, \tag{2}$$

where $A_1(t)$ is defined as the constant function 1.

3 The Model

The model is developed first for the case with a "high enough" crossover rate. Then it is modified to model the cases with lower crossover rates. The first case involves three main steps. First, mean alleles after the crossover and the selection are estimated assuming that the crossover rate is "high enough". Then, the effect of mutation on the mean allele is determined. The last step involves the estimation of fitness variance given the mean allele values.

The second case, in which the crossover rate takes more realistic values, is modeled by observing some statistical properties of the GA at early generations.

At any GA stage, the mean fitness, κ_1, is always the sum of the mean alleles across loci at that moment, i.e.

$$\kappa_1(t) = \sum_i \alpha_i(t) \ . \tag{3}$$

The following sections describe each of these steps in detail.

3.1 Mean Allele After Selection and Crossover with "High Enough" Crossover Rate

First, consider the case in which the crossover rate is so high that the alleles at any locus are distributed essentially randomly among the chromosomes. We will call this crossover rate a "high enough" crossover rate. In fitness-proportional selection, each chromosome has a selection probability proportional to its relative fitness within the population. If we denote as q_j the probability of selecting the j^{th} chromosome, then

$$q_j = \frac{f_j}{\sum_{k=1}^{P} f_k} \ , \tag{4}$$

where f_k is the fitness of the k^{th} chromosome.

Let p_i be the probability that a chromosome that is selected randomly with the above probability scheme after the application of crossover, has 1 at its i^{th} locus. As with the other symbols, we will use the notation $p_i(t)$ for values of a specific run of a GA at time t, or of an experimental average of these values at time t, or of the estimated theoretical average in our model, depending on the context. It is easy to estimate $p_i(t)$ theoretically in terms of $\kappa_1(t)$ and $\alpha_i(t)$ when the crossover rate is "high enough". For this purpose, define the subsets $S_0^i(t)$ and $S_1^i(t)$ of $S(t)$ as

$$S_0^i(t) = \left\{ chrom \in S(t) \,\middle|\, chrom(i) = 0 \right\} \quad \text{and} \tag{5}$$

$$S_1^i(t) = \left\{ chrom \in S(t) \,\middle|\, chrom(i) = 1 \right\} .$$

Then, we have

$$\overline{\sum_{chrom \in S_0^i(t)} f(chrom)} = P\left(1-\alpha_i(t)\right)\left(\kappa_1(t)-\alpha_i(t)\right) \quad \text{and} \tag{6}$$

$$\overline{\sum_{chrom \in S_1^i(t)} f(chrom)} = P\alpha_i(t)\left(1+\kappa_1(t)-\alpha_i(t)\right) ,$$

where the bar over the summation means the average over all possible configurations of gene distributions, in which we assume that the genes are distributed randomly satisfying the given mean allele values, since the crossover rate is "high enough". Thus, the estimated average value of $p_i(t)$ is

$$p_i(t) = \frac{P\alpha_i(t)\left(1+\kappa_1(t)-\alpha_i(t)\right)}{P\alpha_i(t)\left(1+\kappa_1(t)-\alpha_i(t)\right)+P\left(1-\alpha_i(t)\right)\left(\kappa_1(t)-\alpha_i(t)\right)} , \tag{7}$$

which simplifies to

$$p_i(t) = \alpha_i(t) + \frac{\left(1-\alpha_i(t)\right)\alpha_i(t)}{\kappa_1(t)} . \tag{8}$$

In the process of fitness proportional selection, we apply selection of chromosomes P times with replacement. Each time, the probability that the selected chromosome has 1 as its i^{th} allele, is $p_i(t)$. So, the expected number of 1's at the i^{th} locus, after the selection is over, can be obtained by using a binomial distribution. Let $B(n,P,p_i)$ denote the probability of having n successes after P trials, when the success probability is p_i for each trial. Then, the expected theoretical value of $\alpha_i^{cs}(t)$ is

$$\alpha_i^{cs}(t) = \frac{1}{P}\sum_{n=1}^{P} n \cdot B\left(n, P, p_i(t)\right) , \tag{9}$$

when the crossover rate is "high enough".

3.2 Mean Allele After Mutation

In this section, we want to estimate $\alpha_i^{csm}(t)$ given the values of $\alpha_i^{cs}(t)$. Each gene of a chromosome has the probability p_m of changing its value from 1 to 0 or from 0 to 1 by mutation. When we consider the possible changes at the i^{th} locus only, the expected number, N, of total allele changes due to mutation can be found by using a binomial distribution as

$$N = \sum_{n=1}^{P} n \cdot B(n, P, p_m) \ . \tag{10}$$

Since the percentage of 1's at the i^{th} locus is $\alpha_i^{cs}(t)$, $\alpha_i^{cs}(t)N$ of these changes are going to be from 1 to 0, and $(1-\alpha_i^{cs}(t))N$ of the changes are from 0 to 1, on the average. This means that the number of 1's at the i^{th} locus, which is $P\alpha_i^{cs}(t)$, will become $P\alpha_i^{cs}(t) - \alpha_i^{cs}(t)N + (1-\alpha_i^{cs}(t))N$ after the mutation. Simplifying this quantity and dividing by P gives the mean allele for the next generation as

$$\alpha_i(t+1) = \alpha_i^{csm}(t) = \alpha_i^{cs}(t) + \frac{1-2\alpha_i^{cs}(t)}{P} N \ . \tag{11}$$

3.3 Estimation of Fitness Variance for "High Enough" Crossover Rates

The fitness variance by definition is

$$\kappa_2(t) = \frac{1}{P}\left(\sum_{k=1}^{P} f(chrom_k)^2 \right) - \kappa_1(t)^2 \ . \tag{12}$$

If we write the fitness of $chrom_k$ as the sum of its gene values a_k^i and change the order of summation after expanding the square sign above, we obtain

$$\kappa_2(t) = \kappa_1(t) + \frac{1}{P} \sum_{i \neq j}^{L} \sum_{k=1}^{P} a_k^i a_k^j - \kappa_1(t)^2 \ . \tag{13}$$

The term $\sum_{k=1}^{P} a_k^i a_k^j$ in Equation (13), counts the number of chromosomes in which loci i and j both contain 1's. In the case of "high enough" crossover rates, this count is estimated by using α_i^{cs} and α_j^{cs} as follows. The probability, $p(i,j,n)$, that locations i and j have n common 1's is found by $\binom{P\alpha_i^{cs}}{n} \times \binom{P - P\alpha_i^{cs}}{P\alpha_j^{cs} - n} \div \binom{P}{P\alpha_j^{cs}}$, where n could take any value between $max(0, P\alpha_i^{cs} + P\alpha_j^{cs} - P)$ and $min(P\alpha_i^{cs}, P\alpha_j^{cs})$ and the product of P with α's is rounded to the nearest integer in order to calculate the combinations. Thus, the estimation of the fitness variance in the case of "high enough" crossover rates is found by using

$$\kappa_2^{cs}(t) = \kappa_1^{cs}(t) + \frac{1}{P} \sum_{i \neq j}^{L} \sum_{n} n \cdot p(i,j,n) - \kappa_1^{cs}(t)^2 \ . \tag{14}$$

The estimation of fitness variance after mutation is done by Prügel-Bennett and Shapiro, [5]. Their formula gives us

$$\kappa_2^{csm} = (1-2p_m)^2 \kappa_2^{cs} + \left(1-\frac{1}{P}\right) p_m (1-p_m) \sum_{i=1}^{L} w_i^2 \quad , \tag{15}$$

where w_i is the weight of the i^{th} locus. In other words, the fitness of a chromosome $(a_1, a_2, \ldots, a_L)$ is calculated by the weighted summation $\sum w_i a_i$. In our special case, the values of w_i's are all 1. So, we use the formula

$$\kappa_2^{csm}(t) = (1-2p_m)^2 \kappa_2^{cs}(t) + \left(1-\frac{1}{P}\right)(p_m - p_m^2) \sum_{i=1}^{L} 1 \quad , \tag{16}$$

$$= (1-2p_m)^2 \kappa_2^{cs} + L\left(1-\frac{1}{P}\right)(p_m - p_m^2) = \kappa_2(t+1) \quad ,$$

to estimate the fitness variance after mutation.

3.4 Lower Crossover Rates

Equation (8) gives the probability p_i when the crossover rate is very high. In such a case, as in Section (3.1), we are able to treat the 1's at a fixed locus of different chromosomes as identical to each other in terms of their roles in selection because of the high mixing rate of the crossover operator, which makes chromosomes look similar to each other, on the average. However, for lower and more realistic crossover rates, there will be some correlation between alleles within a chromosome and Equation (8) will no longer hold. Let's keep the usage of notation $p_i(t)$ for the probability of selecting a 1 at the i^{th} locus in the case of the "high enough" crossover rate and denote the corresponding probability in the case of a lower crossover rate by $\tilde{p}_i(t)$. To remedy this situation and estimate $\tilde{p}_i(t)$ correctly, we consider imaginary weights, $c_i(t)$, for each locus in order to reflect the average change in the role of 1's played in the selection process due to correlation between alleles. The correction weights, $c_i(t)$, are defined implicitly by

$$\tilde{p}_i(t) = \alpha_i(t) + \frac{(1-\alpha_i(t))\alpha_i(t)c_i(t)}{\kappa_1(t)} \quad . \tag{17}$$

The reason why we defined the correction weights as in Equation (17) is because if we write Equation (8) for a fitness function of the form $f(chrom) = \sum_i w_i \cdot chrom(i)$, with weights w_i, we would get an equation exactly like Equation (17) with c_i replaced by w_i. Our correction weights play a similar role at each locus as w_i's would, except that c_i's change over time.

The next step will be to estimate the c_i's statistically by means of some data gathered from experiments. In order to do this, the GA is run with fixed rates of p_m and p_c up to a pre-selected generation, say t_0 . Let us call this generation G_0. The crossover operation with the current rate, p_c, is applied to G_0 many times. Each time,

$\tilde{p}_i(t)$ values are calculated from the experimental data for each locus i, and the corresponding c_i values are found using Equation (17). This process is repeated for many runs of the GA to obtain statistical measures. It is observed that the value of c_i strongly depends on the values of α_i, as expected. Because of this dependence, it makes more sense to group the c_i according to their corresponding α_i values before finding the statistics of the data gathered from experiments. So, define $C_k^h(t_0)$ as the set $\left\{ c_i \text{ values of the } k^{th} \text{ experiment of GA such that } h \le \alpha_i(t_0) < h + 0.1 \right\}$, for h = 0, 0.1, …0.9 . The mean of the correction weights is obtained by finding $\mu(h',t_0) = \underset{k}{mean}\left(mean\left(C_k^h(t_0)\right)\right)$, h' = 1, 2, …,10, where the relationship between the index h and h' is given by $h = (h'-1)/10$, to be consistent with definition (2). In order to measure how much the correction weights vary from one experiment to another, we also calculate the standard deviation $\sigma(h',t_0) = \underset{k}{std}\left(mean\left(C_k^h(t_0)\right)\right)$.

The experimental results show that, when p_c is not too low (below about 4%), μ and σ remain more or less at the same value regardless of the time, t_0. Moreover, μ shows a linear-like behavior while σ shows a quadratic-like behavior as a function of h'. This behavior of the crossover operator allows us to use the linear approximation of $\mu(h',5)$ to predict $\tilde{p}_i(t)$ for the following generations. Figure (2) shows the graphs of μ for two different rates of crossover with t_0 at generations 5, 15 and 30. We have observed that the inclusion of μ in our model is good enough for describing the effects of c_i distributions and the information coming from σ does not play a significant role in the counting-ones problem. However, for other problems, such as OneMax with Boltzmann scaling, σ might be needed in the model. The deviations from the linear behavior, in Figure (2), at h'=1 or 10 are due to the statistical averaging in which there were not enough data points available for these border values.

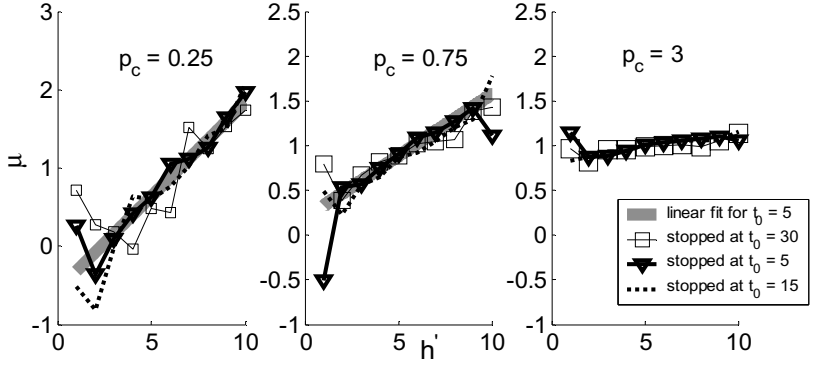

Fig. 2. The mean value of the correction weights as a function of mean allele levels, h', for crossover rates p_c = 0.25, 0.75 and 3. The statistical average is found over 100 experiments of GA. Population size is 50 and the chromosome length is 100 genes

4 Simulation of the Model and Comparison of the Results with the Experiments

The simulation of the model for high enough crossover rates starts with selecting a set of $\alpha_i(0)$ values chosen by considering a binomial distribution for each locus in which we have P selections with a 50% chance of selecting a 1 each time. Equations (9) and (11) are applied to estimate the mean alleles after the crossover, selection and mutation operations. This process is iterated for each generation to obtain a dynamic simulation of the mean allele. At any moment, the mean fitness is estimated by Equation (3), and the fitness variance in the case of "high enough" crossover rates is estimated using Equations (14) or (16), depending on whether we are considering the variance right after the selection process or after the mutation, respectively. The following simulation results are obtained by taking an average over 10 runs of this model.

In the case of normal crossover rates Equation (9) is replaced by Equation (17), in which the c_i-values are pre-determined by the linear approximation of the data gathered at the 5th generation of a set of GA runs as described in Section (3.4), Figure (2).

In Figures (3)-(6), we see the comparison of the experimental results with the simulation of the model for population size 50 and the chromosome length 100 genes. In all these graphs, the black lines represent the experimental results, which are obtained by averaging 100 runs of the GA, and the thick gray lines represent the results obtained by the simulation of our model. The "high enough" crossover rate in our case is p_c = 300%, which means that 100% crossover is applied 3 times in a row before the selection. This rate of crossover is verified experimentally as "high enough" by observing that there is no significant change in the graphs of $A_h(t)$, $\kappa_1(t)$ and $\kappa_2(t)$ if a higher value of p_c is used. It can also be verified from Figure (2), since the correction weights, when p_c = 300%, are all very close to 1. In Figure (3), the time variation of A_h is shown for crossover rates of p_c = 4%, 25%, 75% and 300%. We observed in our experiments with the model that when the crossover rate is to low, such as p_c = 4%, the statistics of correction weights taken only from the 5th generation is not enough and we needed adjustment by using the statistics at the 15th generation. Figure (3.a) shows the graph with this adjustment. For p_c = 25% and 75%, the statistics from only the 5th generation are used. The simulation for p_c = 300% is obtained by taking all c_i's as 1. In all four cases, no mutation is applied — i.e., p_m = 0. The graphs look quite similar to each other, except that there is a slight variation in the value to which the lines converge as time goes to 200 generations. We see that the limit value decreases from around 40 to 30 as p_c increases from 4% to 300%. This slight decrease observed in the experimental graphs is well captured by the model simulations.

In Figure (4), the time evolution of A_h is shown for two different mutation rates, in both of which p_c is kept constant at 50%. In the first case the mutation rate is very low at p_m = 0.1%, while in the second case it is p_m = 2%. In both cases, the model predicts the mean allele behavior very well.

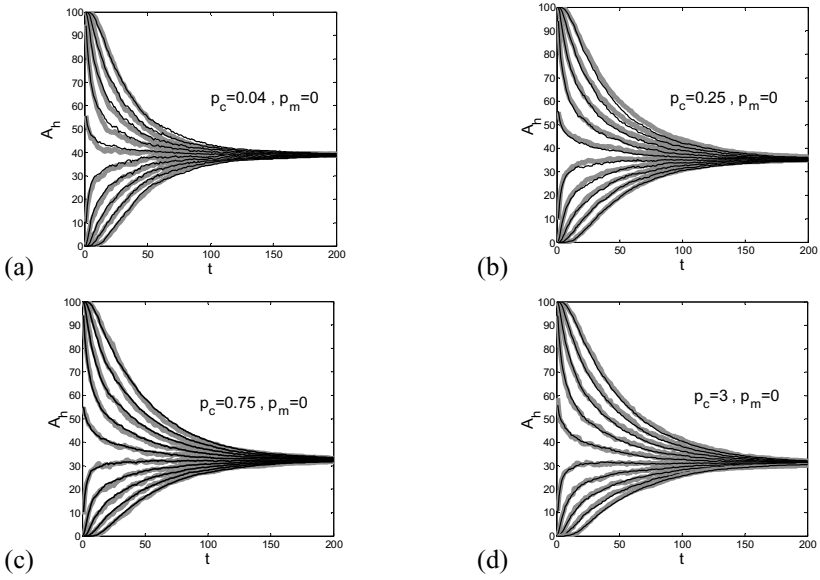

Fig. 3. A_h as a function of time for four different cases where the crossover rate is 4, 25, 75 and 300 percent, respectively. There is no mutation in all four cases. Black lines are the experimental averages over 100 GA runs and thick gray lines are the results obtained by model simulations. Population size is 50 and the chromosome length is 100 genes

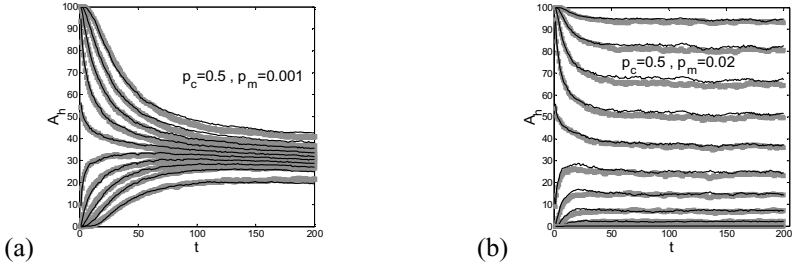

Fig. 4. A_h as a function of time for two different cases where the mutation rate is 0.1 and 2 percent, respectively. The crossover rate is 50% in both cases. Black lines are the experimental averages over 100 GA runs and thick gray lines are the results obtained by model simulations

The experimental results and the model estimations of mean fitness for several values of p_m with crossover rates of p_c = 25%, 75% and 300% are shown in Figure (5). The impact on the mean fitness of changing p_c from 300% to 25% is more visible when p_m is low, around 0 or 0.1%. The effect of higher mutation rates dominates the dynamics of mean fitness evolution, and decreases the amount by which different crossover rates affect the mean fitness. The estimation of the fitness variance when p_c = 300% is shown in Figure (6) for various mutation rates, together with experimental averages. In all these graphs, we see that these dynamics of mean fitness and fitness variance are well captured by the simulation of the model.

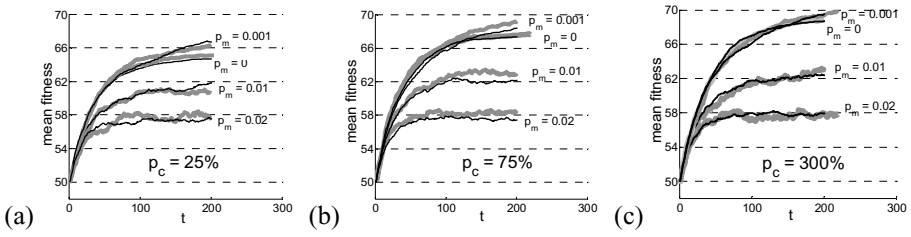

Fig. 5. The mean fitness for three different crossover cases, namely p_c = 25%, 75% and 300%. Each figure shows the graphs for four different mutation rates, p_m = 0%, 0.1%, 1% and 2%. Black lines are the experimental averages obtained by averaging over 100 GA runs and thick gray lines are the results obtained by model simulations. Population size is 50 and the chromosome length is 100 genes

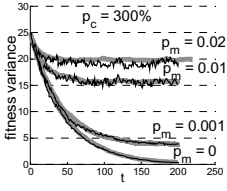

Fig. 6. The fitness variance for four different rates of mutation, p_m = 0%, 0.1%, 1% and 2%, with crossover rate at 300%. Black lines are the experimental averages obtained by averaging over 100 GA runs and thick gray lines are the results obtained by model simulations

5 Conclusions and Future Work

In this paper, we have developed a new and very practical model for the GA dynamics of the OneMax problem, which, for modeling the case of typical crossover rates, uses some statistics of early generations of the GA in order to predict the rest of the evolution. The simulation results in Section (4) show that the model describes the GA dynamics for the OneMax problem very well for different crossover and mutation rates with fitness proportional selection. The correction weights introduced by Equation (17) are a new way of analyzing the crossover operator, and they work very well for two-point crossover, in our GA problem.

Note that our model for "high enough" crossover rates is covering a different case than the statistical mechanics model of Prügel-Bennett and Shapiro [3]. The maximum entropy assumption of Prügel-Bennett and Shapiro essentially models a situation in which the crossover operator is assumed to be effective enough to allow a relocation of the alleles which is probabilistically most likely to occur, under the constraints of the given mean fitness and fitness variance, when the alleles move freely. On the other hand, our model of "high enough" crossover rates does not assume any constraint in relation to how much the allele can be mixed. The lower crossover rates are modeled relative to this extreme case using correction measures.

The authors have applied the model to the weighted OneMax[1] problem as well as to a class of deceptive functions; for the case of "high enough" crossover rates, it predicts the dynamics very well. They are now modifying the model to cover the effects of more typical, lower crossover rates for these benchmark problems. The future work to improve the model would also include the estimation of the fitness variance for normal crossover rates, and an investigation of the predictive power of the model in the presence of external noise.

The model can be applied to these benchmark problems even when the crossover, mutation and the selection rates (in the case of Boltzmann scaling) are changed at predetermined generations during a GA run. Because of this capability of the model, it is unique, to the best of the author's knowledge, among the current models of the GA.

The method of building blocks for modeling parallel genetic algorithms is applied by Cantú-Paz [8] in the case where the migration occurs only when all the populations are converged. Since our model estimates the mean value at each locus at any generation, it can be used to determine a suitable migration time as well as the migration rate for parallel genetic algorithms (in the island model case) when migrations are allowed at any generation.

Acknowledgment. The authors would like to thank Prof. Charles R. MacCluer for helpful and inspiring discussions.

References

1. Nix, A.E., Vose, M.D.: Modeling Genetic Algorithms with Markov Chains. Ann. Math. Art. Intell., (1991), 5:79-88
2. Vose, M.D., Liepins, G.E.: Punctuated Equilibria in Genetic Search. Complex Systems, (1991), 5:31-44
3. Prügel-Bennett, A., Shapiro, J.L.: An Analysis of Genetic Algorithms Using Statistical Mechanics, Phys. Rev. Lett., (1994), 72(9):1305-1309
4. Rattray, L.M.: Modeling the Dynamics of Genetic Algorithms Using Statistical Mechanics. PhD thesis, University of Manchester, Manchester, U.K., (1996)
5. Prügel-Bennett, A., Shapiro, J.L.: The Dynamics of a Genetic Algorithm for Simple Random Ising Systems. Physica D, (1997), 104:75-114
6. Goldberg, D.E.: The Design of Innovation. Kluwer Academic Publishers, Boston, Dordrecht, London, (2002)
7. Goldberg, D.E., Deb, K., Thierens, D.: Toward a Better Understanding of Mixing in Genetic Algorithms. Journal of the Society of Instrument and Control Engineers, (1993), 32(1), 10-16
8. Cantú-Paz, E.: Efficient and Accurate Parallel Genetic Algorithms. Kluwer Academic Publishers, (2001)

[1] The fitness function is given as a weighted sum of alleles.