Genetic Algorithms, Selection Schemes, and the Varying Effects of Noise

Brad L. Miller
Department of Computer Science
University of Illinois
at Urbana/Champaign
Urbana, IL 61801
bmiller@illigal.ge.uiuc.edu

David E. Goldberg
Department of General Engineering
University of Illinois
at Urbana/Champaign
Urbana, IL 61801
deg@uiuc.edu

Abstract

This paper analyzes the effect of noise on different selection mechanisms for genetic algorithms (GAs). Models for several selection schemes are developed that successfully predict the convergence characteristics of GAs within noisy environments. The selection schemes modeled in this paper include proportionate selection, tournament selection, (μ, λ) selection, and linear ranking selection. An allele-wise model for convergence in the presence of noise is developed for the OneMax domain, and then extended to more complex domains where the building blocks are uniformly scaled. These models are shown to accurately predict the convergence rate of GAs for a wide range of noise levels.

Keywords

Genetic algorithms, selection, noise, selection intensity, tournament selection, linear ranking, stochastic universal selection, proportionate selection, uniformly scaled building blocks.

1. Introduction

Selection schemes primarily determine the convergence characteristics of genetic algorithms (GAs). Good progress has been made in developing models for several different selection schemes that successfully predict the convergence characteristics of a GA within a deterministic (noiseless) environment. However, these models are not designed for noisy environments, where fitness functions only approximately measure the true fitness of individuals. This paper seeks to model the convergence characteristics of several selection schemes for noisy environments for the OneMax domain. It then applies this allele-wise analysis for more complex domains where the building blocks are uniformly scaled. This paper focuses on the effect of noise on the selection pressure of a GA; it does not address the equally interesting issue of how noise affects the performance characteristics of the recombination and mutation operators.

Convergence models of different selection schemes were first broached in Goldberg (1989a) and later expanded in Goldberg and Deb (1991). Mühlenbein and Schlierkamp-Voosen (1993) introduced the use of selection intensity for convergence analysis of GA selection schemes. The convergence characteristics of several different selection schemes have recently been successfully modeled for deterministic environments. In Thierens and Goldberg (1994), convergence models for deterministic environments were developed for several selection schemes, including proportionate selection, binary tournament selection, and truncation selection. Both Bäck (1995) and Miller and Goldberg (1996) applied order statistics to extend the tournament selection model of Thierens and Goldberg (1994) to handle tournament sizes larger than 2. Bäck (1995) also used order statistics to develop a model for (μ, λ) selection. A convergence model for linear ranking was presented by Blickle and Thiele (1995).

The purpose of this paper is to further our understanding of how selection pressure works, with special attention on how noise alters the effects of selection pressure. Convergence models at the allele level are then developed that utilize our new understanding of how noise affects selection pressure. Convergence models for noisy environments are developed for several selection schemes, including tournament selection, (μ, λ) selection, linear ranking, and stochastic universal selection. These models accurately predict the convergence of GAs and are verified for a wide range of noise levels using the OneMax domain. The allele-level model is then extended to work at the building block level and verified using several different domains, including the Royal Road level-0 and MaxTrap domains.

Section 2 provides the basic background needed to understand this paper and includes a discussion of selection scheme types, selection intensity, noise definitions, and the OneMax domain. Section 3 expands the selection intensity equation to handle noise and adapts the noisy selection intensity model for the OneMax domain. This forms the basis of the convergence models for both proportionate-based and ordinal-based selection. The experimental methodology and results are then presented for the OneMax domain, comparing actual convergence performance of several selection schemes with predicted performance. Section 4 extends the allele-level model developed in Section 3 to the building block level and presents experiments demonstrating the accuracy of this approach for the Royal Road level-0 domain, the MaxTrap domain, and a domain denoted as the OneMax building block domain. Finally, Section 5 presents some general conclusions from this research.

2. Background

In this section, a brief overview of the basic background information needed to understand this paper is given. The role of selection schemes in GAs is discussed, and a general selection intensity model is presented. Next, noise and noisy fitness functions are explained. The last subsection deals with the OneMax domain, as well as using the general selection intensity model to predict performance in the deterministic OneMax domain.

2.1 Selection Schemes

Genetic algorithms use a selection scheme to select individuals from the population to insert into a mating pool. Individuals from the mating pool are used by a recombination operator to generate new offspring, with the resulting offspring forming the basis of the next generation. Since the individuals in the mating pool pass their genes on to the next generation, it is desirable that the mating pool comprise "good" individuals. A selection scheme in GAs is simply a process that favors the selection of better individuals in the population for the mating pool. The *selection pressure* is the degree to which the better individuals are favored: the higher the selection pressure, the more the better individuals are favored. This selection pressure drives the GA to improve the population fitness over succeeding generations.

The convergence rate of a GA is largely determined by the magnitude of the selection pressure, with higher selection pressures resulting in higher convergence rates. Genetic algorithms are able to identify optimal or near-optimal solutions under a wide range of selection pressure values (Goldberg, Deb, & Thierens, 1993). However, if the selection pressure is too low, the convergence rate will be slow, and the GA will unnecessarily take longer to find the optimal solution. If the selection pressure is too high, there is an increased chance of the GA prematurely converging to a suboptimal solution. In addition to providing selection pressure, selection schemes should also preserve population diversity because this helps to avoid premature convergence.

There are two basic types of selection schemes commonly used today: proportionate-based selection and ordinal-based selection. Proportionate-based selection selects individuals on the basis of their fitness values relative to the fitness of the other individuals in the population. Some common proportionate-based selection schemes are proportionate selection (Holland, 1975), stochastic remainder selection (Booker, 1982; Brindle, 1981), and stochastic universal selection (Baker, 1987; Grefenstette & Baker, 1989). Ordinal-based selection schemes select individuals not according to their fitness, but on the basis of their rank within the population. The individuals are ranked according to their fitness. This entails that the selection pressure is independent of the fitness distribution of the population and is solely based on the relative ordering (ranking) of the population. Some common ordinal-based selection schemes are tournament selection (Brindle, 1981; Goldberg, 1989b), (μ, λ) selection (Schwefel, 1981), truncation selection (Mühlenbein & Schlierkamp-Voosen, 1993), and linear ranking (Baker, 1985, 1987; Grefenstette & Baker, 1989). Ordinal-based selection schemes are normally preferred over proportionate-based selection schemes for a variety of reasons, the most prominent of which is the scaling problem (Forrest, 1985; Goldberg, 1989a; Whitley, 1989).

2.2 Selection Intensity

The *selection intensity* I measures the magnitude of the selection pressure provided by a selection scheme. The selection intensity of GAs, as defined by Mühlenbein and Schlierkamp-Voosen (1993), is the expected average fitness of a population after selection is performed on a population whose fitness is distributed according to the unit normal distribution $N(0, 1)$. If the selection intensity I of a selection scheme is known, and the population fitness at generation t is distributed $N(\mu_t, \sigma_t^2)$, the expected mean fitness of a population after selection can be determined:

$$\mu_{t+1} = \mu_t + I\sigma_t \qquad (1)$$

An important assumption of this model is that population fitness is normally distributed before selection. In practice, this is true or approximately true for many domains because recombination and mutation operators have a normalizing effect on the population fitness distribution.

Table 1 gives the selection intensity for several common selection schemes. Bäck (1995) and Miller and Goldberg (1996) independently applied order statistics to derive the selection intensity for tournament selection. The order statistics are for the unit normal distribution $N(0, 1)$; thus, $\mu_{i:j}$ represents the expected value of the ith biggest sample from a sample of size j drawn from the unit normal distribution. The maximal order statistic $\mu_{s:s}$ determines the selection pressure of a tournament of size s. The study by Bäck (1995) also derives the selection intensity for (μ, λ) selection. In (μ, λ) selection, the best μ individuals are selected from a random sample of λ individuals. The selection pressure is simply the mean of the top μth-order statistics of sample size λ. The selection intensity of linear ranking is given by Blickle and Thiele (1995), where n^+ denotes the number of desired copies of the best individual. Linear ranking selects each individual in the population with a probability linearly proportional to the rank of the individual. Implicit in the selection intensity value for linear ranking is that $1 \leq n^+ \leq 2$, and $n^+ + n^- = 2$, where n^- is the number of desired

Table 1. Selection intensity of common selection schemes.

Selection Scheme	Parameters	Selection Intensity I
Tournament selection	s	$\mu_{s:s}$
(μ, λ) selection	μ, λ	$\frac{1}{\mu}\sum_{i=\lambda-\mu+1}^{\lambda}\mu_{i:\lambda}$
Linear ranking	n^+	$(n^+ - 1)\frac{1}{\sqrt{\pi}}$
Proportionate selection	σ_t, μ_t	σ_t/μ_t

copies of the worst individual. Mühlenbein and Schlierkamp-Voosen (1993) derived the selection intensity for proportionate selection, which directly depends on the current mean μ_t and standard deviation σ_t of the population in generation t. Proportionate selection selects individuals for the mating pool with a probability directly proportional to the individuals' fitness. The selection intensity equation for proportionate selection is used in this paper to predict the performance of stochastic universal selection, one of a handful of different proportionate selection schemes. The selection intensity of proportionate selection is unique in that it is the only one that is sensitive to the current population distribution.

2.3 Noise

Although there are many different definitions of noise, in this paper we are concerned with the factors that prevent the accurate evaluation of the fitness of individuals. This noise results in the fitness functions being inaccurate, so the fitness functions in turn are referred to as *noisy fitness functions.*

There are many factors that may necessitate the use of noisy fitness functions. In some domains, there may be no known fitness function that can accurately assess an individual's fitness, so an approximate (noisy) fitness function must be used. In domains where computational speed is paramount, fast but noisy fitness functions may be preferred over slow but accurate fitness functions (Fitzpatrick & Grefenstette, 1988). Noisy information can also negatively affect the accuracy of a fitness evaluation. Information noise can come from a variety of sources, including noisy data, knowledge uncertainty, sampling error, sensor input, and human error.

Because the selection process is based on fitness values, noisy fitness functions cause the selection process itself to be noisy. The noisy fitness value of an individual can be viewed as the sum of the real fitness of the individual plus a random noise component. In this paper, we assume that the noise component is randomly drawn from an unbiased (mean of zero) normal distribution and added to the true fitness value of an individual to obtain a noisy fitness value. The assumption of an unbiased, normally distributed noise source is true or approximately true in many noisy domains and allows the effects of noise to be more easily modeled.

A noisy fitness distribution with a known mean noise bias of x can be converted to an unbiased fitness distribution by simply subtracting x from every fitness evaluation. This results in an unbiased noisy fitness function; however, the distribution of the "fixed" fitness function will still have the same distribution as the original noisy fitness distribution, just shifted by x. This approach does not result in an unbiased normal fitness distribution unless the original biased fitness distribution is normal.

2.4 OneMax Domain

This paper uses the OneMax domain, also known as the counting 1's or bit-counting problem, to verify the accuracy of the selection scheme models. The ability to accurately predict the performance of a selection scheme operating within the OneMax domain demonstrates a basic understanding of the underlying selection mechanism. This section reviews the OneMax domain and derives the convergence model for the OneMax domain.

The OneMax domain is often used for analyzing GA performance because it has several strengths:

1. The alleles are independent of each other.
2. The alleles are uniformly scaled.
3. The initial proportion of optimal alleles p can be determined.
4. The mean and variance of the population fitness can be represented as functions of p.
5. There is no creation or deletion of building blocks by the recombination operators.

Item 1 states that the alleles independently contribute to the chromosome fitness, and item 2 indicates that the alleles contribute equally to the chromosome fitness. The alleles each have an equal weight of 1, so it is easy to see that item 2 is true. In item 3 the proportion of optimal alleles p refers to the proportion of alleles in the population that match the allele value in the optimal chromosome. For the OneMax domain, an optimal allele is simply an allele with a value of 1. Item 4 enables prediction of the convergence performance of the OneMax domain as a function of p. Item 5 is the result of the building block length for the OneMax domain being of length 1. Because the length is 1, standard recombination operators cannot disrupt existing building blocks, thus no new building blocks are created or destroyed by standard recombination operators. This results in the number of optimal building blocks in the population before crossover being equivalent to the number of optimal building blocks after crossover. This is a key point because it allows the research to focus on selection effects without having to worry about the effect of building block disruption, which is highly domain dependent.

Another major strength is that the OneMax domain can be a good jumping off place for exploring building block issues in GAs. Because the OneMax domain has a building block of length 1, it is a good starting point for the analysis of larger building blocks in more complex domains. Later in this paper we extend the results of the OneMax allele-wise convergence analysis to larger building block sizes.

Within the OneMax domain, the real fitness of an individual is simply the number of one bits in the chromosome: $\sum_{i=1}^{m} w_i bb_i$, where m is the number of building blocks in the chromosome; bb_i is the fitness of building block i; and w_i is the weighting for building block i. For the traditional OneMax domain, the weighting w_i is always 1, and is thus not considered in the derivation given below. The building block fitness bb_i is simply the value of the ith bit of the binary chromosome. The optimal chromosome is the chromosome containing all one bits. The population mean fitness and variance for the OneMax domain are given by $\mu_t = lp_t$ and $\sigma_t^2 = lp_t(1 - p_t)$, respectively (Mühlenbein & Schlierkamp-Voosen, 1993; Thierens & Goldberg, 1994), where l is the chromosome length, and p_t is the percentage of correct alleles in the population at generation t.

In Mühlenbein and Schlierkamp-Voosen (1993), a convergence equation for the OneMax domain and ordinal selection schemes is derived and is reproduced here and expanded on later in this paper. Given that the general selection intensity Equation 1 can be represented as

$$\mu_{t+1} - \mu_t = I\sigma_t$$

the rate of change of the percentage of correct alleles can be determined by

$$p_{t+1} - p_t = \frac{I}{\sqrt{l}}\sqrt{p_t(1 - p_t)}$$

Approximating the above difference equation with a differential equation yields

$$\frac{dp}{dt} = \frac{I}{\sqrt{l}}\sqrt{p_t(1 - p_t)} \tag{2}$$

For a randomly initialized population, $p_0 = 0.5$ is a reasonable approximation for the initial percentage of correct alleles. Using this, Equation 2 can be solved exactly to yield

$$p_t = \frac{1}{2}\left(1 + \sin\left(\frac{I}{\sqrt{l}}t\right)\right) \tag{3}$$

Equation 2, along with the assumption that $p_0 = 0.5$, can also be used to determine the time until convergence t_{conv}:

$$t_{\mathrm{conv}} = \frac{\pi\sqrt{l}}{2I} \tag{4}$$

The above derivation by Mühlenbein and Schlierkamp-Voosen (1993) holds for cases where the selection intensity I is independent of the percentage of correct alleles p_t. However, for proportionate selection, the selection intensity is directly dependent on p_t, so that the derivation is different. For proportionate selection see Thierens and Goldberg (1994),

$$p_{t+1} - p_t = \frac{1}{l}I\sigma_t = \frac{\sigma_t^2}{l\mu_t} = \frac{lp_t(1 - p_t)}{l^2 p_t} = \frac{1}{l}(1 - p_t)$$

Approximating the above equations with the differential dp/dt and again assuming that $p_0 = 0.5$ gives

$$p_t = 1 - 0.5e^{-t/l} \tag{5}$$

for the special case of proportionate selection.

It is important to note that, unlike the ordinal-based selection mechanisms, proportionate selection, as shown by Equation 5, will never reach absolute convergence ($p = 1$). Thierens and Goldberg (1994) calculated the amount of time until the population converges to within an arbitrary amount ϵ of $p = 1$:

$$t_{\mathrm{conv}} = -l\ln(2\epsilon)$$

For the purposes of this paper, ϵ is chosen to be $\epsilon = 1/2l$, so that the convergence time for proportionate selection is calculated by

$$t_{\mathrm{conv}} = l\ln(l) \tag{6}$$

3. Noise and Selection Intensity

This section extends the selection intensity equation, given by Equation 1, to accurately predict the selection pressure in the presence of noise. Next, we generalize the derivation of the noisy tournament selection model presented in Miller and Goldberg (1996) to work for all selection scheme models that are based on selection intensity. Then, the corresponding convergence models for the OneMax environment are derived.

3.1 Noisy Selection Intensity Model Derivation

The model derivation in this section has three major steps. First, the relation between an individual's noisy fitness and true fitness values is determined, so that the expected true fitness value of an individual can be estimated from the noisy fitness evaluation. Next, the relation between true and noisy fitness is extended to handle subsets of individuals, so that the mean true fitness of a subset of the population can be estimated from the mean noisy fitness of the subset. Last, we use the general selection intensity equation, Equation 1, to estimate the mean noisy fitness value of the mating pool, where the mating pool is selected on the basis of the noisy fitness values. This mean noisy fitness value is then inserted into the formula found in the second step to estimate the mean true fitness of the mating pool. The selection pressure, based on the expected mean true fitness value of the mating pool, is thus determined. The result is a predictive model for selection schemes that can handle varying noise levels.

In a noisy environment, the noisy fitness f' of an individual is given by $f' = f$ + noise, where f is the real fitness of the individual, and noise is the noise inherent in the fitness function evaluation. The real fitness of the population F is assumed to be normally distributed: $N(\mu_{F,t}, \sigma^2_{F,t})$. This section further assumes that the noise is unbiased and normally distributed: $N(0, \sigma^2_N)$. This facilitates modeling the effects of the noise and is a reasonable assumption for many domains. Using these assumptions, along with the additive property of normal distributions, we find that the noisy fitness F' of the population is also normally distributed: $N(\mu_{F,t}, \sigma^2_{F,t} + \sigma^2_N)$.

Although the real fitness value for an individual is unknown, the expected value of the real fitness can be determined from the individual's noisy fitness value, which is generated by a noisy fitness function evaluation. Because both true fitness and noisy fitness are normally distributed, the bivariate normal distribution can be used to obtain the expected true fitness value of F for a given noisy fitness value f' of F'. For normal random variables X and Y, the bivariate normal distribution states that the expected value of Y for a specific value x of X is

$$E(Y|x) = \mu_Y + \rho_{XY}\frac{\sigma_Y}{\sigma_X}(x - \mu_X)$$

where ρ_{XY} is the correlation coefficient for X and Y. The correlation coefficient ρ_{XY} can be calculated by $\rho_{XY} = \sigma_{XY}/(\sigma_X\sigma_Y)$, where σ_{XY} is the covariance of X and Y. The covariance between F and F' is simply σ^2_F; thus,

$$\begin{aligned} E(F|f') &= \mu_F + \frac{\sigma^2_F}{\sigma_F\sigma_{F'}}\frac{\sigma_F}{\sigma_{F'}}(f' - \mu_{F'}) \\ &= \mu_F + \frac{\sigma^2_F}{\sigma^2_{F'}}(f' - \mu_{F'}) \\ &= \mu_F + \frac{\sigma^2_F}{\sigma^2_F + \sigma^2_N}(f' - \mu_{F'}) \qquad (7) \end{aligned}$$

Because Equation 7 is linear, the expected value of F for any subset R of the population can be calculated using Equation 7, with f' set to the noisy fitness mean μ_R of the subset. Of course, the subset we are interested in is the mating pool selected by the noisy selection process. The general selection intensity equation, Equation 1, can be used to obtain the expected mean noisy fitness of the mating pool when the selection process is itself based on noisy fitness values. The expected noisy fitness mean of the mating pool subset can be calculated using Equation 1 because the mean $\mu_{F',t}$ and variance $\sigma_{F',t}$ of the noisy population are known:

$$\mu_{F',t+1} = \mu_{F',t} + I\sigma_{F',t} = \mu_{F',t} + I\sqrt{\sigma^2_{F,t} + \sigma^2_N}$$

Setting f' to $\mu_{F',t+1}$ in Equation 7 produces the expected true fitness value of the mating pool:

$$\begin{aligned} E(F_{t+1} \mid \mu_{F',t+1}) &= \mu_{F,t+1} \\ &= \mu_{F,t} + \frac{\sigma^2_{F,t}}{\sigma^2_{F,t} + \sigma^2_N}\left(\mu_{F',t} + I\sqrt{\sigma^2_{F,t} + \sigma^2_N} - \mu_{F',t}\right) \\ &= \mu_{F,t} + I\frac{\sigma^2_{F,t}}{\sqrt{\sigma^2_{F,t} + \sigma^2_N}} \end{aligned} \tag{8}$$

As expected, Equation 8 reduces to Equation 1, the formula for the deterministic case, when the noise variance σ^2_N equals zero. Equation 8 is significant in that it extends the basic selection intensity convergence model to handle noise. By doing so, accurate convergence rate prediction in the presence of noise is now possible for selection schemes that can be modeled using the selection intensity approach.

An interesting ramification of Equation 8 is that proportionate selection is noise invariant: Increased noise levels do not affect the selection pressure of GAs using proportionate selection! Inserting $I = \sigma_{F',t}/\mu_{F',t}$ from Table 1 into Equation 8, and realizing that $\mu_{F',t} = \mu_{F,t}$ and $\sigma_{F',t} = \sqrt{\sigma^2_{F,t} + \sigma^2_N}$ for unbiased normally distributed noise, yields

$$\mu_{F,t+1} = \mu_{F,t} + \sigma_{F,t}\frac{\sigma_{F,t}}{\mu_{F,t}} \tag{9}$$

which is identical to the deterministic equation of selection pressure for proportionate selection.

3.2 Noisy OneMax Convergence Model

In Section 2.4, the basic convergence model for the deterministic domain was derived. This section derives the convergence model for the noisy OneMax domain but also expands it to deal with alleles with an arbitrary weighting, as opposed to the standard weight of "1" for alleles in the OneMax domain. For this case, the absolute difference between the minimum and maximum values of the allele is denoted as ΔF_{bb}, making the population mean fitness and variance for the OneMax domain be $\mu_t = F_{bb,\min} + l\,\Delta F_{bb} p_t$ and $\sigma^2_t = l\,\Delta F_{bb}{}^2 p_t(1 - p_t)$, respectively. Using the same approach outlined in Section 2.4, along with Equation 8, it is easy to see that

$$p_{t+1} - p_t = \frac{I}{l\Delta F_{bb}}\frac{\sigma^2_{F,t}}{\sqrt{\sigma^2_{F,t} + \sigma^2_N}} = I\frac{p_t(1 - p_t)}{\sqrt{lp_t(1 - p_t) + \sigma^2_{N'}}} \tag{10}$$

where $\sigma^2_{N'} = \sigma^2_N / \Delta F_{bb}{}^2$. The $\sigma^2_{N'}$ is effectively the noise scaled relative to the magnitude of the signal size of the building blocks. For the domains under consideration, the signal size of the building blocks can be calculated by $\Delta F_{bb} = F_{bb,\max} - F_{bb,\min}$.

Approximating Equation 10 with a differential equation yields

$$\frac{dp}{dt} = I \frac{p_t(1 - p_t)}{\sqrt{l p_t(1 - p_t) + \sigma^2_{N'}}} \tag{11}$$

Although Equation 11 is integrable, it does not reduce to convenient form in the general case for p_t; however, it can be easily solved numerically for p_t, and for the noiseless case ($\sigma_N = 0$), p_t can be solved exactly (see Equation 3).

Although Equation 11 is not directly solvable for p_t, it can be solved for t as a function of p:

$$\begin{aligned} t(p) &= \frac{1}{I}\left[\sqrt{l}\arctan\left(\frac{\sqrt{l}(2p - 1)}{2\sqrt{\sigma^2_{N'} + l\,p(1 - p)}}\right)\right. \\ &\qquad \left. + \sigma_{N'} \ln\left(\frac{p}{p - 1} - \frac{l - 2\sigma^2_{N'} + lp - 2\sigma_{N'}\sqrt{\sigma^2_{N'} + l\,p(1 - p)}}{2\sigma^2_{N'} + l\,p + 2\sigma_{N'}\sqrt{\sigma^2_{N'} + l\,p(1 - p)}}\right) + c\right] \end{aligned} \tag{12}$$

For binary alleles, at time $t = 0$ we can assume that half of the alleles are initially correct ($p = 0.5$). Using this to solve for c in Equation 12 gives $c = 0$. We are particularly interested in the time t_{conv} it takes for all alleles to converge ($p = 1$). For the deterministic case, Equation 12 reduces to

$$t_{\text{conv}} = \frac{\pi\sqrt{l}}{2I} \tag{13}$$

which is equivalent to the result obtained in Equation 4.

Although t_{conv} cannot be solved exactly for the noisy OneMax environment, it can be easily solved numerically. Useful approximations for small, medium, and large levels of noise are derived from Equation 12 in Miller and Goldberg (1996), where the convergence approximations derived for tournament selection can be generalized to the selection intensity model by simply substituting I for $\mu_{s:s}$. These approximations for small, medium, and large amounts of noise can be used to quickly estimate the convergence time of GAs in noisy environments, and are presented in Table 2.

3.3 Experimental Validation

In this section we verify the accuracy of the noisy selection models derived above for four different selection schemes: tournament selection, (μ, λ) selection, linear ranking, and stochas-

Table 2. Time of convergence approximations for the noisy OneMax domain.

Noise Case	Noise Level	t_{conv} Approximation
Small	$\sigma_{N'} \approx 0$	$\frac{1}{I}\left[\frac{\pi\sqrt{l}}{2} + 2\sigma_{N'} \ln(2\sigma_{N'})\right]$
Medium	$\sigma_{N'} \approx \sigma_F$	$\frac{1}{I}\left[\sqrt{l}\arctan\left(\frac{\sqrt{l}}{2\sigma_{N'}}\right) + 2\sigma_{N'} \ln(\sqrt{2}\sigma_{N'})\right]$
Large	$\sigma_{N'} \approx \infty$	$\frac{1}{I}\left[\frac{l}{2\sigma_{N'}} + \sigma_{N'} \ln(l - 1)\right]$

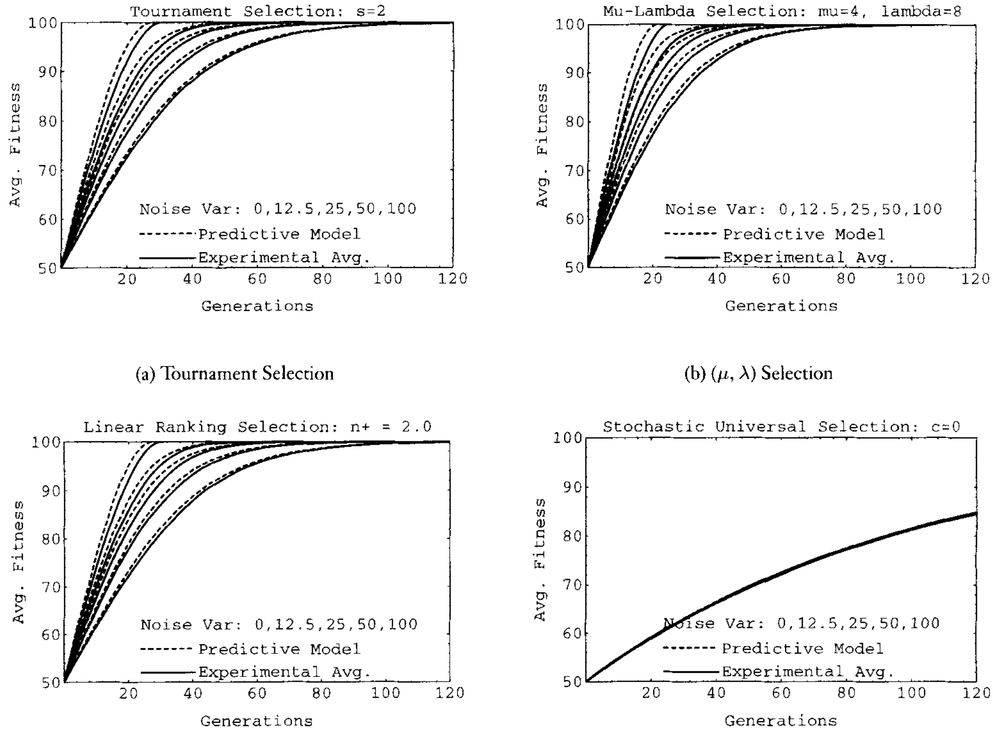
tic universal selection. The experiments are run using the noisy OneMax domain, where the fitness function noise is simulated by adding a random noisy value to the true fitness of each individual. The noise is randomly drawn from an unbiased normal distribution, where the variance of the normal distribution is given for each experiment. The experimental results of running GAs with the different selection schemes in various noise levels are then compared with those results predicted by the models derived above.

3.3.1 Methodology For each different selection scheme, the GA is run 10 times in the noisy OneMax domain for each noise level, and the results are averaged. The results are then plotted against predicted performance for each noise level.

For the OneMax domain, the chromosome length is $l = 100$, and the population size is set according to Goldberg, Deb, and Horn (1992), which works out to $N = 8(\sigma_F^2 + \sigma_N^2)$. The alleles use the standard weighting of 1, so that $\Delta F_{bb} = 1$. For a binomially distributed population, the initial population variance is $lp(1 - p)$, and the initial percentage of correct alleles in the population p is assumed to be 0.5. Thus, the population sizing equation works out to be $N = 8(25 + \sigma_N^2)$. A generational GA is used, with the crossover rate set to 1.0. The crossover mechanism is uniform crossover (Syswerda, 1989), with the probability of swapping alleles set to 0.5. To isolate the effects of selection, no mutation mechanism is used. The tested noise levels are taken to be 0, 0.5, 1, 2, and 4 times the initial function variance of $\sigma_{F,0}^2 = 25$. Thus, the noise levels used in verifying the selection mechanisms are $\sigma_N^2 = \{0, 12.5, 25, 50, 100\}$. In each GA run, the run is stopped only after the population has fully converged to a homogeneous population.

The recombination operator in GAs has two main functions—mixing and building block disruption. A recombination operator mixes existing building blocks in the population to form chromosomes with new combinations of building blocks. Building block disruption is caused by a recombination operator combining building blocks to create new building blocks, thereby destroying existing building blocks. Because the OneMax domain has a building block length of 1, the recombination operator cannot create or destroy building blocks in the population, so there is no building block disruption from the recombination operator. Uniform crossover was chosen as the recombination operator because it thoroughly mixes the existing building blocks in the population. For the uniformly scaled domains of interest in this paper, mixing has a "normalizing" effect on the population fitness distribution, making the predictive model's assumption of a normally distributed population more valid. Other recombination operators, such as 1-, 2-, or N-point crossover operators, also have normalizing effects, although less pronounced. In addition, mutation can also have a normalizing effect on the population fitness distribution but is not used here so as to isolate the convergence effects of noise on the selection pressure.

3.3.2 Results This section details the results of our experiments. The experimental results are plotted along with the predicted performance for each selection scheme in Figure 1. In each plot, the dashed lines are the predicted performance, and the solid lines are the experimental performance (average of 10 runs). In most cases, the model accurately predicts the experimental performance, so that the predicted performance (dashed lines) is mostly obscured by the experimental results (solid lines). The small difference between predicted and experimental performance is due to building block covariance buildup ("hitchhiking"); this results in a population whose fitness deviates from the normal fitness distribution assumed by the predictive model. In Thierens and Goldberg (1994) and Miller and Goldberg (1996), the building block covariance was removed by performing multiple rounds of crossover in



(a) Tournament Selection

(b) (μ, λ) Selection

(c) Linear Ranking Selection

(d) Stochastic Universal Selection

Figure 1. Experimental results for the OneMax domain.

a generation; this had the effect of eliminating any difference between predicted and observed performance. Those studies verified that the slight difference between observed and predicted performance in these experiments can be attributed to building block covariance buildup.

For the ordinal-based selection mechanisms [tournament selection, (μ, λ) selection, and linear ranking], there are five sets of lines corresponding to the five different noise levels. The lines, from left to right, correspond to the five different noise levels of $\sigma_N^2 = \{0, 12.5, 25, 50, 100\}$. Equations 3 and 11 are used to predict the performance of the ordinal-based selection schemes in the deterministic and noisy OneMax domains, respectively. The selection intensity for each of the ordinal-based selection schemes is calculated according to Table 1 and is described in more detail below. The results for proportionate selection are presented in Section 3.3.6.

3.3.3 Tournament Selection

For tournament selection, a tournament size of $s = 2$ was used, with a corresponding selection intensity (from Table 1) of $I = \mu_{2:2} = 0.5642$. The maximum-order statistic value was obtained from Harter (1970). The results, plotted in Figure 1a, demonstrate that the convergence models were very accurate in predicting the convergence of tournament selection.

3.3.4 (μ, λ) Selection For (μ, λ) selection, $\mu = 4$ and $\lambda = 8$ were used. The corresponding selection intensity was calculating using Table 1, along with the order statistic values obtained from Harter (1970):

$$\begin{aligned} I &= \tfrac{1}{4}(\mu_{5:8} + \mu_{6:8} + \mu_{7:8} + \mu_{8:8}) \\ &= \tfrac{1}{4}(0.1525 + 0.4728 + 0.8522 + 1.4236) \\ &= 0.7253. \end{aligned}$$

The results, plotted in Figure 1b, demonstrate that the convergence models were very accurate at predicting the convergence of (μ, λ) selection. Note that this experiment had the same $\alpha = \mu/\lambda$ ratio value of 0.5 as the binary tournament selection experiment, yet had a higher selection pressure. The trade-off of effectively having higher values for μ and λ for the same α ratio is that the resulting selected population has a lower variance. In general, selection mechanisms should maximize the resulting population variance for a given level of selection pressure in order to avoid premature convergence. This entails that for (μ, λ) selection, if there are several possible μ and λ combinations that produce a desired selection pressure, the combination with the smallest λ should be preferred so as to maximize population diversity.

3.3.5 Linear Ranking For linear ranking selection, the number of desired copies of the best individual n^+ was set to 2, which made $n^- = 0$. The corresponding selection intensity (from Table 1) is

$$I = \frac{1}{\sqrt{\pi}} = 0.5642$$

which is equivalent to the binary tournament selection case.

The results, plotted in Figure 1c, demonstrate that the convergence models were very accurate in predicting the convergence of linear ranking.

3.3.6 Stochastic Universal Selection Because Equation 9 demonstrates that the convergence equation for noisy environments is equivalent to the deterministic case for proportionate selection, the deterministic model, Equation 5, is used to predict the performance of stochastic universal selection for all noise levels. The predictive accuracy of our model for stochastic universal selection, Equation 5, is shown in Figure 1d. Figure 1d is shown at the same scale as Figure 1, a–c so as to ease comparisons between figure parts. Since the model is noise invariant, the runs at the five different noise levels produced roughly the same plot and therefore overlay each other in the figure. This accounts for the single solid line, which is also the predicted performance for stochastic universal selection. This bears out the model's prediction of noise invariance. Note that the number of generations until convergence is much larger than the rank-based selection schemes.

Although Figure 1d demonstrates that our model accurately predicts the selection pressure of fitness proportionate selection, it does not show that the absolute convergence times differ for different noise levels and population sizes. Whereas the number of generations until the population reached 99.5% convergence is roughly equivalent for all five noise levels (approximately at generation 460), the time for absolute convergence ranged from 773 to 990 generations because of genetic drift. This can be explained by the following: As the population approaches near convergence for fitness proportionate selection schemes, the resulting selection pressure becomes negligible. At this point, the effect of genetic drift dominates the convergence process. The time until convergence under genetic drift is influenced both by the noise and the size of the population. However, the genetic drift is not free to let the population drift any which way, because the further away the population drifts from the optimum, the more selection pressure is applied. So in effect, there is a one-way genetic drift tendency of the population toward the optimum. Accurate modeling of this one-way genetic drift in the presence of noise is an interesting topic that merits further research.

4. From Alleles to Building Blocks

In Section 3 we presented an allele-wise model for how noise affects selection pressure in the OneMax domain. This section demonstrates how to apply this allele-wise model at the building block level to determine how noise affects the selection pressure in more complex domains.

4.1 Building Block Model

To apply the allele-wise analysis at the building block level, the assumptions made at the allele level for the OneMax domain also have to be true or approximately true at the building block level:

1. The building blocks are independent of each other.
2. The building blocks are uniformly scaled.
3. The initial proportion of optimal building blocks $p_{bb,0}$ can be determined.
4. The mean and variance of the population fitness can be represented as functions of p_{bb}.
5. There is minimal creation and deletion of building blocks from other GA operators, such as recombination and mutation.

The explanation for these items is similar to the explanation for the OneMax domain given in Section 2.4 but at the building block level instead of the allele level. Item 1 denotes that the building blocks independently contribute to the fitness. For this analysis, we concentrate on domains where the building blocks are uniformly scaled, but a more complex analysis could be used for other types of scaling. The initial proportion of building blocks $p_{bb,0}$ can either be obtained directly from knowledge of the domain or estimated through sampling initial GA populations. Determining the mean and variance as functions of p_{bb} can be difficult for many domains but can often be roughly approximated because the building blocks are independent and uniformly scaled.

Although selection pressure can explain a large part of the convergence characteristics of a GA, the convergence is also affected by the building block disruption of recombination and mutation operators. These operators create and destroy building blocks (building block disruption) and as such can significantly affect the population convergence. These effects are important and are being studied elsewhere (Goldberg et al., 1993; Mühlenbein & Schlierkamp-Voosen, 1993, 1994; Aizawa, 1996). This paper focuses on how the selection pressure resulting from a certain noise level will affect the convergence, and as such it is beneficial to remove the convergence effects of building block disruption so as to isolate the effect of the selection pressure. The mutation effect on convergence is removed by simply setting the mutation rate to zero. The building block disruption effect of the recombination operator is removed by utilizing a priori knowledge of the building block locations for the domains of interest: We use a uniform crossover operator (Syswerda, 1989) at the building block level, which we call the *uniform building block crossover operator*. This operator thus swaps only whole building blocks, and as such neither destroys nor creates new building blocks. In effect, the uniform building block crossover operator has a building block disruption rate of zero. Another approach would have been to use crossover operators that only have a small building block disruption rate, such as a one- or two-point crossover operator. However, the uniform building block operator was preferred because it had no building block disruption and also did a better job of reducing the covariance between building blocks. Using this uniform building block crossover operator has the effect of bringing the assumption of our modeling closer to that of the empirical study. It also has the effect of moving our modeling somewhat further away from conditions faced in actual practice. Nonetheless, we feel this first step is a necessary one if we are ever to subsequently be able to analyze these kinds of results with heritability or other operator adjustment coefficients.

The following sections demonstrate how we rationally adapted the OneMax allele model for use at the building block level for different domains. These experiments again used a generational GA. In all cases, the GA runs in these domains used a chromosome length $l = 100$ and building block size $k = 4$; thus, the number of building blocks is $m = 25$. The uniformly scaled building blocks were scaled so that the optimal building block had a fitness value of 1; thus, the optimal chromosome had a fitness value of $m = 25$. Because the minimum building block in all cases had a fitness value of zero, $\Delta F_{bb} = 1$. The crossover rate was 1.0, and the probability of swapping building blocks for the uniform building block crossover operator was $p_{\text{unif}} = 0.5$. The mutation rate was zero. The GA was run until the proportion of optimal building blocks in the population was $p_{bb,\text{conv}} = (m - 0.5)/m = 0.98$. These experiments were run using binary tournament selection ($s = 2$), but any of the selection mechanisms listed in Table 1 could have been used with similar results. To avoid population sizing effects, the population was sized conservatively large. Since there is no building block disruption due to no mutation and the use of the uniform building block crossover operator, the initial population has to have an adequate supply of optimal building blocks. Thus, the population has to be sized on the order of 2^k for size k building blocks. For $k = 4$, we conservatively set the population sizing equation to be $N = 2000(1 + \sigma_N^2/\sigma_{F,0}^2)$. The initial fitness variance of the population $\sigma_{F,0}^2$ is domain dependent. For each domain, a set of four experiments was run corresponding to noise levels of $\sigma_N^2 = \sigma_{F,0}^2 * \{0, 0.5, 1, 2, 4\}$. For each experiment, the results were averaged over 10 trials. Each trial was stopped when the population converged to a homogeneous population.

4.2 Royal Road Level-0 Domain

The Royal Road level-0 fitness function (Forrest & Mitchell, 1993) at the building block level is defined by

$$F_{bb}(x) = \begin{cases} 1 & \text{if } x = k, \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where x is the number of 1's in the building block. The Royal Road level-0 fitness function is simply the sum of the building block fitness evaluations for each building block in the chromosome. This domain is a good example of domains where all building blocks but the optimal building block have a near-minimum fitness value.

For this domain, the initial proportion of building blocks $p_{bb,0}$ is p_0^k, where the initial allele proportion $p_0 = 0.5$. The fitness and variance of the population can be calculated just as in the OneMax case but with m instead of l and p_{bb} instead of p: $\mu_t = m\, p_{bb,t}$, and

Table 3. Building block domain convergence times.

	0		$0.5\ \sigma^2_{F,0}$		$1\ \sigma^2_{F,0}$		$2\ \sigma^2_{F,0}$		$4\ \sigma^2_{F,0}$	
Domain	Pred	Exp	Pred	Exp	Pred	Exp	Pred	Exp	Pred	Exp
Royal Rd 0	20.8	23.8	23.5	25.0	25.8	28.1	29.7	31.9	36.3	38.0
OneMax BB	25.3	28.5	43.0	46.7	54.5	57.5	72.0	75.0	97.9	100.5
MaxTrap	25.3	28.0	46.8	42.3	60.3	52.3	80.6	66.5	110.4	88.0

$\sigma^2_{F,t} = m\ p_{bb,t}\ (1 - p_{bb,t})$. This can be solved as in the OneMax case to yield Equation 12 but with m and p_{bb} replacing l and p, respectively. This model is then used to calculate the time until $p_{bb,\text{conv}} = 0.98$, and the results are compared in Table 3. For each noise level, the constant c in Equation 12 is calculated from knowing m, $p_{bb,0}$, and the noise level.

The predicted results for each noise level closely matched the experimental results. This is a result of the Royal Road level-0 domain being exactly modeled by the OneMax allele-wise convergence model. The small difference between predicted and experimental performance is due to building block covariance buildup ("hitchhiking"); this results in a population whose fitness deviates from the normal fitness distribution assumed by the predictive model. In Thierens and Goldberg (1994) and Miller and Goldberg (1996), the building block covariance was removed by performing multiple rounds of crossover in a generation; this had the effect of eliminating any difference between predicted and observed performance. Those studies verified that the slight difference between observed and predicted performance in these experiments could be attributed to building block covariance buildup.

4.3 OneMax Building Block Domain

For the next test domain, we wanted a domain that had two main characteristics. First, the initial fitness distribution of the building blocks should be normally distributed. Second, the proportion of converged building blocks in the population should be reflected by how far the building block fitness distribution had converged to the maximum (optimal) building block fitness value. To do this, we designed the OneMax building block domain, whose building block fitness function is as follows:

$$F_{bb}(x) = x/k \tag{15}$$

The variable x is again the number of 1's in the building block, and k is the building block size. This is essentially the OneMax function, with the difference that the alleles are scaled by $1/k$ and that the uniform building block crossover operator is used. The building blocks have approximately the desired normal fitness distribution (actually, they are binomially distributed), and as the building block fitness distribution approaches convergence, the number of converged building blocks in the population increases. These traits are characteristic of many domains, giving promise that this analysis can be used as a rough guide for convergence performance of many domains.

To analyze the OneMax building block domain, the building block terms are represented by their allele equivalents, and the allele OneMax model is then used to solve the convergence time. The key relation to link the building block to the allele model is $p_{bb} = p^k$, so that $\sqrt[k]{p_{bb}} = p$. Thus, to use the allele model, we set the initial condition to $p_0 = \sqrt[k]{p_{bb,0}}$, where $p_{bb,0} = 0.5^k$ and the desired convergence proportion to $p_{\text{conv}} = \sqrt[k]{p_{bb,\text{conv}}}$. The time for the allele model to converge to p_{conv} is equivalent to the time it takes for the building blocks to converge to $p_{bb,\text{conv}}$. The chromosome length for Equation 12 applied at the allele level is l, and $\Delta F_{bb} = 1/k$. For each noise level, the constant c in Equation 12 is calculated from knowing l, p_0, and the noise level. Table 3 compares the predicted convergence times with the experimentally determined convergence times. Again, there is very good agreement between the predicted and experimental results. The small difference between predicted and observed performance is due to the buildup of covariance of building blocks and can be eliminated by multiple rounds of uniform building block crossover per generation (Thierens & Goldberg, 1994).

4.4 MaxTrap Domain

The MaxTrap domain, also called fully deceptive trap functions (Deb & Goldberg, 1993; Thierens & Goldberg, 1993) is a domain designed to show how building block disruption can be deceptive for a GA, reducing the overall GA performance. However, since we are turning off disruption by using uniform building block crossover, for our purposes the MaxTrap domain is being explored as an example of a domain that is difficult to precisely analyze. The MaxTrap building block fitness function is

$$F_{bb}(x) = \begin{cases} 1 & \text{if} \quad x = k \\ (1 - d)^{\frac{k-1-x}{k-1}} & \text{otherwise} \end{cases} \tag{16}$$

For the MaxTrap domain, it is difficult to determine the fitness and variance of the population fitness as a function of the proportion of optimal building blocks in the population. One approach is to note that the convergence of the population has roughly the same characteristics as the OneMax building block domain described above: The building block distribution is roughly normally distributed, and the proportion of converged building blocks p_{bb} is roughly a factor of how far the building block fitness distribution has converged to the optimal building block. Because of these factors, a reasonable approach is to use the OneMax building block model as a heuristic to estimate the convergence times of the MaxTrap domain.

The initial variance of the MaxTrap domain can be easily calculated by enumerating each of the 2^k possible building blocks and calculating their fitness. For $d = 0.25$ and $k = 4$, the initial variance is $\sigma_F^2 = 2.025$. Using this to set the noise levels for the various experiments, and otherwise proceeding exactly as in the OneMax building block domain described above, yields the performance results predicted in Table 3. The predicted performance is actually a pretty good approximation of the experimental convergence performance for the lower levels of noise but is not as good as the previous two domains because of the approximations used for the fitness and variance of the MaxTrap function (the OneMax building block fitness and variance functions were used). The approximation is less accurate for higher levels of noise because the OneMax approximation for the MaxTrap variance as a function of the proportion of building blocks p_{bb} becomes increasingly less accurate for higher noise levels. Even so, the predictive results are in the ballpark for all but exceedingly high levels of noise. The OneMax building block analysis appears to be a promising heuristic for at least approximating the effect of noise on selection pressure and the resulting convergence performance for uniformly scaled building block domains.

Although the analysis works well for the domains studied, use of the uniform building block crossover operator as an experimental tool is only a first step toward understanding how noise affects GA performance characteristics as a whole. However, it is expected that recombination operators with low building block disruption rates (e.g., one- or two-point crossover operator) will perform similarly to the uniform building block crossover operator. More work will need to be done to understand how recombination and mutation operators are affected by noise and domain properties, but that is outside the intended scope of the present paper.

5. Conclusions

Now that GAs are increasingly being used by industry, the effects of noise on performance are becoming more important. This paper's main contribution is that it demonstrates how fitness function noise will affect the selection pressure of GAs utilizing several common selection schemes. A noisy selection intensity model was derived that accurately predicted the performance of several selection schemes in noisy environments, including tournament selection, (μ, λ) selection, linear ranking, and stochastic universal selection. Furthermore, any selection scheme model based on the deterministic selection intensity model can be similarly adapted to handle noise.

The noisy selection intensity model developed in this paper has several immediate practical applications. It can predict the convergence rate of a GA within noisy domains characterized by uniformly scaled building blocks, which is critical for time-sensitive applications. The model can be used to predict the quality of the solutions after a certain number of generations and can therefore be used to determine appropriate stopping criteria for noisy environments. For noisy fitness functions where the noise is due to sampling error, the model can be used to determine the optimal sample size for the fitness function. Optimal sample size will maximize the performance of a GA within a fixed computational time.

Uniform building block crossover could also be used as an important tool for isolating the convergence effects of building block disruption for recombination and mutation operators for different domains. The baseline convergence for a GA in a given domain can be established by using uniform building block crossover and no mutation; this can be compared with the convergence resulting from the recombination operator of interest to quantify the building block disruption effect on convergence. A similar approach could also be used to isolate the convergence effect of mutation for a given domain. Uniform building block crossover is an important new tool for analyzing the inner mechanisms of GAs.

This research also has several long-term ramifications. The basic approach of this paper can be used to study the delaying effects of noise on other selection schemes. It may also be possible to apply the same approach to predict convergence delays resulting from noise inherent within other GA operators, such as recombination or mutation. The approach outlined for the convergence of uniformly scaled building blocks can provide a basis for exploration into badly scaled building block domains. It also has the practical result of providing the GA practitioner with a rational approach for estimating the impact that different levels of noise will have on GA performance.

Acknowledgments

This work was supported under NASA grant NGT 9-4. This effort was also sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grants F4960-94-1-0103 and F49620-95-1-0338. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

References

Aizawa, A. (1996). Fitness landscape characterization by variance of decompositions. In R. K. Belew and M. Vose (Eds.), *Foundations of genetic algorithms 4* (pp. 390–400). San Mateo, CA: Morgan Kaufmann. Proceedings Preprint.

Bäck, T. (1995). Generalized convergence models for tournament- and (μ, λ)-selection. In L. Eschelman (Ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms* (pp. 2–8). San Francisco, CA: Morgan Kaufmann.

Baker, J. E. (1985). Adaptive selection methods for genetic algorithms. In J. J. Grefenstette (Ed.), *Proceedings of an International Conference on Genetic Algorithms and Their Applications* (pp. 101–111). Hillsdale, NJ: Lawrence Erlbaum.

Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. In J. J. Grefenstette (Ed.), *Proceedings of the Second International Conference on Genetic Algorithms* (pp. 14–21). Hillsdale, NJ: Lawrence Erlbaum.

Blickle, T., & Thiele, L. (1995). *A comparison of selection schemes used in genetic algorithms* (Technical Report No. 11). Gloriastrasse 35, CH-8092 Zurich: Swiss Federal Institute of Technology (ETH) Zurich, Computer Engineering and Communications Networks Lab (TIK).

Booker, L. B. (1982). Intelligent behavior as an adaptation to the task environment. *Dissertation Abstracts International, 43*(2), 469B. (University Microfilms No. 8214966.)

Brindle, A. (1981). *Genetic algorithms for function optimization.* Unpublished doctoral dissertation. University of Alberta, Edmonton, Canada.

Deb, K., & Goldberg, D. E. (1993). Analyzing deception in trap functions. In L. D. Whitley (Ed.), *Foundations of genetic algorithms 2* (pp. 93–108). San Mateo, CA: Morgan Kaufmann.

Fitzpatrick, J. M., & Grefenstette, J. J. (1988). Genetic algorithms in noisy environments. *Machine Learning, 3,* 101–120.

Forrest, S. (1985). *Documentation for PRISONERS DILEMMA and NORMS programs that use the genetic algorithm.* Unpublished manuscript, University of Michigan, Ann Arbor.

Forrest, S., & Mitchell, M. (1993). Relative building-block fitness and the building-block hypothesis. In L. D. Whitley (Ed.), *Foundations of genetic algorithms 2* (pp. 109–126). San Mateo, CA: Morgan Kaufmann.

Goldberg, D. E. (1989a). Genetics-based machine learning: Whence it came, where it's going. In M. S. Elzas, T. I. Oren, & B. P. Zeigler (Eds.), *Modelling and simulation methodology* (pp. 285–300). New York: North-Holland.

Goldberg, D. E. (1989b). Zen and the art of genetic algorithms. In J. D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 80–85). San Mateo, CA: Morgan Kaufmann. (Also TCGA Report 88003).

Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In G. J. E. Rawlings (Ed.), *Foundations of genetic algorithms* (pp. 69–93). San Mateo, CA: Morgan Kaufmann.

Goldberg, D. E., Deb, K., & Horn, J. (1992). Massive multimodality, deception, and genetic algorithms. In R. Männer & B. Manderick (Eds.), *Parallel problem solving from nature 2* (pp. 37–46). Amsterdam: Elsevier.

Goldberg, D. E., Deb, K., & Thierens, D. (1993). Toward a better understanding of mixing in genetic algorithms. *Journal of the Society of Instrument and Control Engineers, 32*(1), 10–16.

Grefenstette, J. J., & Baker, J. E. (1989). How genetic algorithms work: A critical look at implicit parallelism. In J. D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 20–27). San Mateo, CA: Morgan Kaufmann.

Harter, H. L. (1970). *Order statistics and their use in testing and estimation*, Vol. 2. *Estimates based on order* *statistics of samples from various populations.* Washington, DC: U.S. Government Printing Office.

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor, MI: University of Michigan Press.

Miller, B. L., & Goldberg, D. E. (1996). Genetic algorithms, tournament selection, and the varying effects of noise. *Complex Systems, 9*(3), 193–212.

Mühlenbein, H., & Schlierkamp-Voosen, D. (1993). Predictive models for the breeder genetic algorithm: I. Continuous parameter optimization. *Evolutionary Computation, 1*(1), 25–49.

Mühlenbein, H., & Schlierkamp-Voosen, D. (1994). The science of breeding and its application to the breeder genetic algorithm (BGA). *Evolutionary Computation, 1*(4), 335–360.

Schwefel, H.-P. (1981). *Numerical optimization of computer models.* Chichester: Wiley.

Syswerda, G. (1989). Uniform crossover in genetic algorithms. In J. D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 2–9). San Mateo, CA: Morgan Kaufmann.

Thierens, D., & Goldberg, D. E. (1993). Mixing in genetic algorithms. In S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms* (pp. 38–45). San Mateo, CA: Morgan Kaufmann.

Thierens, D., & Goldberg, D. E. (1994). Convergence models of genetic algorithm selection schemes. In Y. Davidor, R. Männer, & H.-P. Schwefel (Eds.), *Parallel problem solving from nature 3* (pp. 119–129). Amsterdam: Elsevier.

Whitley, D. (1989). The Genitor algorithm and selective pressure: Why rank-based allocation of reproductive trials is best. In J. D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 116–123). San Mateo, CA: Morgan Kaufmann.