

# **Noise, Sampling, and Efficient Genetic Algorithms**

Brad L. Miller  
Dept. of Computer Science  
University of Illinois at Urbana-Champaign

IlliGAL Report No. 97001  
May 1997

Department of General Engineering  
University of Illinois at Urbana-Champaign  
117 Transportation Building  
104 South Mathews Avenue  
Urbana, IL 61801

© Copyright by Brad L. Miller, 1997

# NOISE, SAMPLING, AND EFFICIENT GENETIC ALGORITHMS

BY

BRAD L. MILLER

B.S., Rensselaer Polytechnic Institute, 1988

THESIS

Submitted in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy in Computer Science  
in the Graduate College of the  
University of Illinois at Urbana-Champaign, 1997

Urbana, Illinois

## **ABSTRACT**

As genetic algorithms (GA) move into industry, a thorough understanding of how GAs are affected by noise is becoming increasingly important. Noise affects a GA's population sizing requirements, performance characteristics, and computational requirements. This research develops quantitative models for determining the effects of noise on the operation of a GA. Furthermore, the question of how to best optimize the performance of a GA in a noisy environments is investigated. Sampling fitness functions are explored, and techniques for determining the optimal sample size that maximizes the performance of a GA within a fixed computational time bound are presented.

For my wife, Susan,  
for her love, support, and patience;  
For my daughter, Kaia,  
for showing me what is truly important.

# Acknowledgments

There are several people I would like to thank for their efforts on my behalf:

- First, my advisor, Dr. David E. Goldberg, for his insight, intelligence, and knowledge of the Zone. Our discussions back in 1994 provided this original impetus for this research effort, and our weekly meetings have been a continuing source of guidance ever since.
- The other members of my committee, Dr. Larry Rendell and Dr. Sylvian Ray, whose insights helped shape key parts of this research effort.
- Dr. Michael J. Shaw, for his willingness to support my exploration of different research areas.
- My peers at the Illinois Genetic Algorithm Laboratory (IlliGAL), for their assistance, humor, and endless arguments (you know who you are ...).
- And of course, my parents, who always let me know that anything was possible.

This work was supported under NASA Grant NGT 9-4, as well as by the University of Illinois Research Board. This study was also sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grants F49620-94-1-0103, F49620-95-1-0338, and F49620-97-1-0050. 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 author 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.

# Preface

In an era where genetic algorithms (GAs) are increasingly being used by industry to solve difficult, and noisy, problems, a thorough understanding of the ramifications of noise on GA performance is becoming critical. This dissertation focuses on the effects of noise on the performance of a genetic algorithms. It develops models that quantify the effects of noise on several GA mechanisms, including population sizing, convergence, and computational requirements.

This dissertation is organized so that the reader can read the chapters relevant to their interests, independently of the rest of the dissertation. The main thrusts of this research are investigations into the effect of noise on convergence properties (Chapter 2), population sizing requirements (Chapter 3), and how to optimize GA performance in noisy environments through the use of sampling techniques (Chapter 4). Another chapter (Chapter 5) introduces several real-world areas where this research can be directly applied, and an example is worked through to illustrate how this research would be applied to real-world domains. Care has been taken so that these chapters can be read independently of one another; if a chapter utilizes research results obtained in other chapters, the results are first summarized.

The reader is assumed to have a basic understanding of genetic algorithms, which can be obtained through a variety of sources, including books by Goldberg (1989a), Davis (1991), or Mitchell (1996). Furthermore, the USENET newsgroup `comp.ai.genetic` can be of assistance in learning about genetic algorithms. A genetic algorithm FAQ (Frequently Asked Questions), also known as *The Hitch Hikers Guide To Evolutionary Computation*, which is primarily aimed at newcomers to the GA community, is periodically posted to the `comp.ai.genetic` newsgroup. A useful starting point for a web search is <http://www-illigal.ge.uiuc.edu/>, which contains a fair amount of GA information as well as links to many other GA web sites. Papers referenced in this dissertation that are (co-)authored by either myself or David E. Goldberg may also be downloaded from this site. My e-mail address is `bmiller@illigal.ge.uiuc.edu`.

# Contents

<b>1</b>	<b>Introduction . . . . .</b>	<b>1</b>
1.1	Research Framework . . . . .	1
1.2	Dissertation Overview . . . . .	2
<b>2</b>	<b>The Effect of Noise on Selection Pressure . . . . .</b>	<b>4</b>
2.1	Background . . . . .	4
2.1.1	Selection Schemes . . . . .	5
2.1.2	Selection Intensity . . . . .	6
2.1.3	Noise . . . . .	6
2.1.4	Convergence Model for Onemax Domain . . . . .	7
2.2	How Noise Affects Selection Pressure . . . . .	10
2.2.1	Noisy Selection Intensity Model Derivation . . . . .	10
2.2.2	Noisy Onemax Convergence Model . . . . .	12
2.2.3	Experimental Validation . . . . .	13
2.3	From Alleles to Building Blocks . . . . .	16
2.3.1	Building Block Model . . . . .	17
2.3.2	Royal Road Level-0 Domain . . . . .	19
2.3.3	Onemax Building Block Domain . . . . .	20
2.3.4	Maxtrap Domain . . . . .	20
2.4	Conclusions . . . . .	21
<b>3</b>	<b>Population Sizing and Noise . . . . .</b>	<b>23</b>
3.1	Building Block Population Sizing . . . . .	23
3.1.1	Building Block Supply Models . . . . .	23
3.1.2	Building Block Decision Models . . . . .	24
3.2	Random Walk Population Sizing . . . . .	26
3.2.1	Random Walk Overview . . . . .	26
3.2.2	Random Walk Population Sizing . . . . .	27
3.2.3	Experimental Verification . . . . .	28
3.3	How Selection Pressure Affects Population Sizing . . . . .	28
3.3.1	Modeling Selection Pressure . . . . .	28
3.3.2	Experimental Verification . . . . .	30
3.4	How Noise Affects Population Sizing . . . . .	31
3.4.1	Modeling Noise . . . . .	31
3.4.2	Experimental Verification . . . . .	31
3.5	Random Walk Population Sizing Generalization . . . . .	33
3.5.1	General Population Sizing Model . . . . .	33
3.5.2	General Population Sizing Model Approximation . . . . .	33
3.6	Conclusions . . . . .	35



<b>4</b>	<b>Sampling</b>	<b>37</b>
4.1	Introduction	37
4.1.1	Overview	38
4.1.2	Related Research	38
4.1.3	Background Research	39
4.2	Optimal Sample Size Determination	41
4.2.1	GA Computational Model	41
4.2.2	Performance as a Function of Sample Size	42
4.2.3	Performance Curve Shape	43
4.3	Sample Size Determination Heuristics	44
4.3.1	Domain Independent Lower Bound	45
4.3.2	Domain Dependent Upper Bound	46
4.3.3	Domain Independent Pareto Pruning	48
4.3.4	Sampling Invariant Parameters	50
4.4	Experimental Verification	50
4.4.1	Methodology	51
4.4.2	Results	51
4.4.3	Sampling Invariant Results	52
4.5	Conclusions	56
<b>5</b>	<b>Research Applications</b>	<b>58</b>
5.1	Introduction	58
5.2	Practical Guide to Applicability	58
5.2.1	Population Sizing	58
5.2.2	Convergence Model Development Heuristics	59
5.2.3	Optimal Selection of Noisy Fitness Function	60
5.3	Noise and Sampling Dichotomies	60
5.3.1	Noise Classification	61
5.3.2	Sampling Classification	61
5.4	Applicable Domains	62
5.4.1	Fitness Independent Noise	62
5.4.2	Fitness Dependent Noise	62
5.5	Conclusions	66
<b>6</b>	<b>Conclusions</b>	<b>67</b>
6.1	Summary	67
6.2	Future Research Areas	69
6.2.1	Building Block Scaling	69
6.2.2	Population Sizing	69
6.2.3	Use of Noise to Model Building Block Disruption	69
6.2.4	Dependent Noise Modeling	70
6.3	Conclusions	70

<b>Appendix: Genetic Algorithm Overview.</b>	<b>72</b>
A.1 Simple Genetic Algorithm	72
A.2 Chromosomes and Building Blocks	72
A.3 Population Sizing and Initialization	73
A.4 Fitness Functions	74
A.4.1 Fitness Noise Definition	74
A.4.2 Fitness Noise Ramifications	75
A.4.3 Sampling Fitness Functions	75
A.5 Selection	75
A.6 Recombination and Mixing	76
A.7 Mutation	76
A.8 Stopping Criteria	76
<b>Bibliography</b>	<b>77</b>

# List of Tables

2.1	Selection Intensity of Common Selection Schemes . . . . .	7
2.2	Time of Convergence Approximations for Noisy Onemax Domain . . . . .	13
2.3	Building Block Domain Convergence Times . . . . .	19
4.1	Lower Bound on Optimal Sample Size . . . . .	46

# List of Figures

1.1	The Effect of Sample Size on Performance . . . . .	2
2.1	Tournament Selection Results for the Onemax Domain . . . . .	15
2.2	$(\mu, \lambda)$ Results for the Onemax Domain . . . . .	16
2.3	Linear Ranking Results for the Onemax Domain . . . . .	17
2.4	Stochastic Universal Selection Results for the Onemax Domain . . . . .	18
3.1	Fitness Distribution of Tournament Competitors . . . . .	24
3.2	Gambler's Ruin Problem . . . . .	26
3.3	Random Walk Population Sizing . . . . .	27
3.4	Population Sizing for Onemax Domain . . . . .	29
3.5	Building Block Trap Function for $k = 4$ . . . . .	29
3.6	Population Sizing for Maxtrap Domain . . . . .	30
3.7	Population Sizing as Function of Selection Pressure . . . . .	31
3.8	Population Sizing as Function of Noise . . . . .	32
3.9	Population Size Approximation . . . . .	34
3.10	Population Size Approximation Correctional Coefficient . . . . .	35
4.1	The Effect of Sample Size on Performance . . . . .	40
4.2	Performance as a Function of Sample Size . . . . .	43
4.3	Effectiveness of Bounding Sample Size . . . . .	44
4.4	Coefficient of Optimal Sample Size Lower Bound . . . . .	46
4.5	Continuous and Discrete Ending Generations . . . . .	49
4.6	Sampling Performance of Onemax Domain . . . . .	50
4.7	Sampling Results for Low Noise Variances . . . . .	53
4.8	Sampling Results for High Noise Variances . . . . .	54
4.9	Performance as a Function of Selection Pressure . . . . .	55
4.10	Performance for Time $T = 1000, 2000$ . . . . .	56
5.1	Experimental Results for Image Matching Domain . . . . .	64
A.1	Genetic Algorithms in a Nutshell . . . . .	73
A.2	Populations, Chromosomes, and Building Blocks . . . . .	74

## Chapter 1

# Introduction

As genetic algorithms (GA) move into industry, a thorough understanding of how GAs are affected by noise is becoming increasingly important. This research explores how noise affects the basic mechanisms of a GA, including convergence, population sizing requirements, and computational resource requirements. It utilizes this newfound understanding of the effects of noise to determine the optimal sample size for GAs employing sampling fitness functions in order to reduce the noise.

This research effort grew out of an attempt to determine the optimal fitness function sample size for GAs operating in noisy environments. To reduce fitness function noise, a GA can use the mean of sampled fitness function evaluations. A sampling model was quickly developed that related the sampling size to the overall fitness noise for the genetic algorithm. However, it quickly became apparent that existing models of GA performance were insufficient in quantifying the effects of the fitness noise on the GA. The research then turned from determining optimal sample sizes to quantifying the effect of noise on many different facets of genetic algorithm performance, including convergence, population sizing, and computational resource requirements. Only after GA models that handled noise were developed could the larger question of determining the optimal sample size for a noisy environment be explored.

This chapter will present a research framework that motivates the different areas of this research effort from the perspective of determining the optimal sample size. The last section outlines the dissertation structure. It is assumed that the reader has a basic grasp of genetic algorithms; if not, Appendix A gives a brief overview of genetic algorithms, with emphasis on the internal GA mechanisms relevant to this research effort. If more detailed knowledge is needed, consult any of the several sources listed in the preface.

## 1.1 Research Framework

This research focuses on answering the difficult question: Given a fixed amount of computational time, for what sample size is the GA performance is maximized? Performance is measured as the average fitness of the ending generation of the GA, as it reflects the total progress made by the GA. For the trivial case of environments with unlimited computational resources, the optimal sample size is simply the size of the sample space, which can be infinite. However, in the real world a GA will be expected to return a solution in a bounded amount of time, and thus will only be able to spend a limited amount of time sampling.

Figure 1.1 outlines the ramifications of the sample size on the overall performance of a time-bounded GA. Performance is determined by both the convergence rate of the GA and the number of generations the GA can run within the computational time bound. To determine how long a GA will be able to run, both the sampling overhead and the population size need to be known in order to calculate the ending generation. The selection noise, which is directly

affected by the sample size through the fitness noise for a given sample size, influences both the GA convergence rate and the required population size. For increasing selection noise, the convergence rate of the GA is slowed due to the decreased effectiveness of the selection mechanism. In addition, increased selection noise requires a larger population size in order to prevent premature convergence due to bad building block decisions being made by the selection mechanism. The optimal sample size is where the performance advantage from faster convergence rates and decreased population sizes balances the performance detriment as a result of increased sampling overhead. These are the the main relationship that will be investigated throughout this research effort.

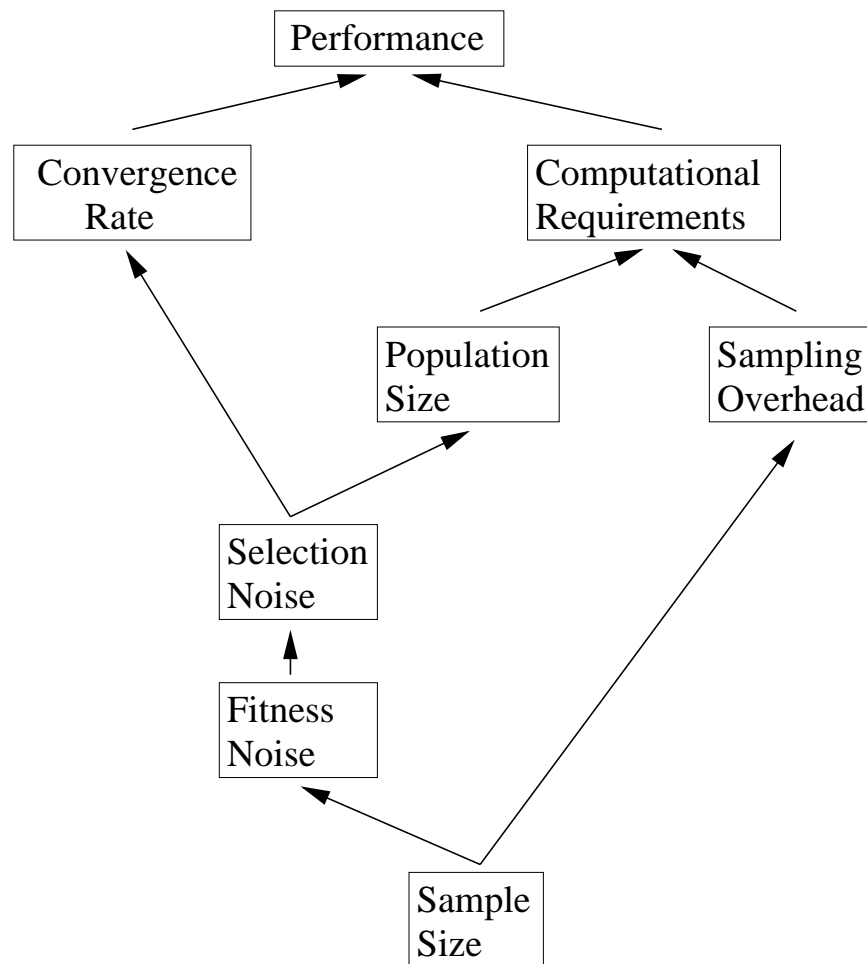


Figure 1.1: The Effect of Sample Size on Performance

## 1.2 Dissertation Overview

This research effort investigates the effects of noise, and how to optimize GAs in noisy environments through the use of sampling. Chapter 2 discusses the effects of noise on the convergence rates of a GA. Next, in Chapter 3 the issue of how noise increases the population sizing re-

quirements is explored. Chapter 4 ties it all together by introducing a computational resource model of GA performance, as well as identifying techniques for identifying the optimal sample size for GAs operating within a fixed computational time limit. Chapter 5 outlines some areas of applicability for this research, and illustrates how this research can be applied for real-world problems. Chapter 6 presents some conclusions drawn from this research, as well as proposing several promising areas for further research.

## Chapter 2

# The Effect of Noise on Selection Pressure

Selection schemes primarily determine the convergence characteristics of genetic algorithms. 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 chapter seeks to model the convergence characteristics of several selection schemes for noisy environments for a simple uniformly scaled domain. It then applies the results of an allele-wise analysis for more complex domains where the building blocks are uniformly scaled. This chapter focuses on the effect of noise on the selection pressure (and hence convergence properties) of a GA; it does not address the equally interesting issue of how building block disruption, caused by recombination and mutation operators, affects the convergence properties of a GA.

The purpose of this chapter 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,  $(\mu, \lambda)$  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 uniformly scaled building block domains.

Section 2 provides the basic background needed to understand this research, including 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 the preceding section to the building block level, and presents experiments demonstrating the accuracy of this approach for several uniformly scaled building block domains, including the Royal Road level-0 domain, the Maxtrap domain, and a domain denoted as the Onemax building block domain. Lastly, section 5 presents some general conclusions from this research.

## 2.1 Background

In this section, a brief overview of the basic background information needed to understand this chapter 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.

### 2.1.1 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 be comprised of ‘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 sub-optimal solution. In addition to providing selection pressure, selection schemes should also preserve population diversity, as this helps avoid premature convergence.

There are two basic types of selection schemes commonly used today: proportionate selection and ordinal-based selection. Proportionate-based selection selects individuals based upon 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 upon their fitness, but based upon 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 upon the relative ordering (ranking) of the population. Some common ordinal-based selection schemes are tournament selection (Brindle, 1981; Goldberg, 1989b),  $(\mu, \lambda)$  selection (Schwefel, 1981), truncation selection (Mühlenbein & Schlierkamp-Voosen, 1993), and linear ranking (Baker, 1985; Baker, 1987; Grefenstette & Baker, 1989). Ordinal selection schemes are normally preferred over proportional selection schemes for a variety of reasons, the most prominent of which is the scaling problem (Forrest, 1985; Goldberg, 1989a; Whitley, 1989).

Convergence models for 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 genetic algorithm 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 (1996b) applied order statistics to extend the tournament selection model to handle tournament sizes larger than two. Bäck also used order statistics to develop a model for  $(\mu, \lambda)$  selection (Bäck, 1995). A convergence model for linear ranking was presented by Blickle and Thiele (1995).

### 2.1.2 Selection Intensity

The *selection intensity*,  $I$ , measures the magnitude of the selection pressure provided by a selection scheme. The selection intensity of genetic algorithms, as defined by Mühlenbein and Schlierkamp-Voosen (1993), is the expected average fitness of a population after selection is performed upon 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:

$$\mu_{t+1} = \mu_t + I\sigma_t \quad (2.1)$$

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, as recombination and mutation operators have a normalizing effect on the population fitness distribution.

Table 2.1 gives the selection intensity for several common selection schemes. Bäck (1995) and Miller and Goldberg (1996b) 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  $i$ th biggest sample out of 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  $(\mu, \lambda)$  selection. In  $(\mu, \lambda)$  selection, the best  $\mu$  individuals are selected out of a random sample of  $\lambda$  individuals. The selection pressure is simply the mean of the top  $\mu$ th order statistics of sample size  $\lambda$ . 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 copies of the worst individual. Mühlenbein and Schlierkamp-Voosen (1993) derived the selection intensity for proportionate selection, which directly depends on the current mean  $\mu_t$  and standard deviation  $\sigma_t$  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 chapter 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 fitness distribution.

### 2.1.3 Noise

While there are many different definitions of noise, this research focuses on the factors that prevent the accurate evaluation of the fitness of individuals. This noise results in the fitness functions being inaccurate, so the fitness function in turn are referred to as noisy fitness functions.

Selection Scheme	Parameters	Selection Intensity $I$
Tournament Selection	$s$	$\mu_{s:s}$
$(\mu, \lambda)$ Selection	$\mu, \lambda$	$\frac{1}{\mu} \sum_{i=\lambda-\mu+1}^{\lambda} \mu_{i:\lambda}$
Linear Ranking	$n^+$	$(n^+ - 1) \frac{1}{\sqrt{\pi}}$
Proportionate Selection	$\sigma_t, \mu_t$	$\sigma_t / \mu_t$

Table 2.1: Selection Intensity of Common Selection Schemes

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.

As the selection process is based upon 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 chapter, it is assumed 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.

#### 2.1.4 Convergence Model for Onemax Domain

This research uses the Onemax domain, also known as the counting ones or bit-counting problem, to verify the accuracy of the selection scheme models. The fitness of a chromosome in the Onemax domain is simply the number of “1” bits in the binary-valued chromosome of length  $l$ :

$$f(x_i) = \sum_{i=1}^l x_i, \text{ where } x_i \in \{0, 1\}$$

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, as it has several

strengths, listed below.

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.

The first item denotes that the alleles independently contribute to the chromosome fitness, and the second item indicates that the alleles contribute equally to the chromosome fitness. The alleles each have an equal weight of one, so it is easy to see that the second item is true. 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 one. The fourth item enables the prediction of the convergence performance of the Onemax domain as a function of  $p$ . The last item is a result of the building block length for the Onemax domain being of length one. As the length is one, standard recombination operators can not 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, as 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 blocks in GAs. As the Onemax domain has a building block of length one, it is a good starting point for the analysis of larger building blocks in more complex domains. Later in this chapter, the results of the Onemax allele-wise convergence analysis is extended 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:  $\sum_{i=1}^m bb_i$ , where  $m$  is the number of building blocks in the chromosome, and  $bb_i$  is the fitness of building block  $i$ . The building block fitness  $bb_i$  is simply the value of the  $i$ th 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 as it is expanded upon later in this chapter. Given that the general selection intensity equation 2.1 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

$$\frac{dp}{dt} = \frac{I}{\sqrt{l}} \sqrt{p_t(1-p_t)} \quad (2.2)$$

For a randomly initialized population,  $p_0 = 0.5$  is a reasonable approximation for the initial percentage of correct alleles. Using this, Equation 2.2 can be solved exactly to yield

$$p_t = \frac{1}{2} \left( 1 + \sin\left(\frac{I}{\sqrt{l}}t\right) \right) \quad (2.3)$$

Equation 2.2, along with the assumption that  $p_0 = 0.5$ , can also be used to determine the time until convergence  $t_{conv}$ :

$$t_{conv} = \frac{\pi\sqrt{l}}{2I} \quad (2.4)$$

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 proportional selection, the selection intensity is directly dependent on  $p_t$ , so the derivation is different. For proportionate selection (Thierens & Goldberg, 1994),

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

Approximating the above with the differential  $\frac{dp}{dt}$ , and again assuming that  $p_0 = 0.5$ , gives

$$p_t = 1 - 0.5e^{-\frac{t}{l}} \quad (2.5)$$

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 2.5, 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  $\epsilon$  of  $p = 1$ :

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

For the purposes of this research,  $\epsilon$  is chosen to be  $\epsilon = \frac{1}{2l}$ , so the convergence time for proportionate selection is calculated by

$$t_{conv} = l \ln(l) \quad (2.6)$$

## 2.2 How Noise Affects Selection Pressure

This section extends the selection intensity equation, given by equation 2.1, to accurately predict the selection pressure in the presence of noise. This section generalizes the derivation of the noisy tournament selection model presented in Miller and Goldberg (1996b) to work for all selection scheme models that are based upon selection intensity. This section then derives the corresponding convergence models for the Onemax environment.

### 2.2.1 Noisy Selection Intensity Model Derivation

The model derivation in this section has three major steps. First, the relationship 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 relationship 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. Lastly, the general selection intensity equation, equation 2.1, is used to estimate the mean noisy fitness value of the mating pool, where the mating pool is selected based on the noisy fitness values. This mean noisy fitness value is then plugged 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

$$f' = f + \text{noise},$$

where  $f$  is the real fitness of the individual, and  $\text{noise}$  is the noise inherent in the fitness function evaluation. The real fitness of the population  $F$  is assumed to be normally distributed  $N(\mu_{F,t}, \sigma_{F,t}^2)$ . This section further assumes that the noise is unbiased and normally distributed  $N(0, \sigma_N^2)$ . 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, gives that the noisy fitness  $F'$  of the population is normally distributed  $N(\mu_{F,t}, \sigma_{F,t}^2 + \sigma_N^2)$ .

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. As both the true fitness and the 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  $\rho_{XY}$  is the correlation coefficient for  $X$  and  $Y$ . The correlation coefficient  $\rho_{XY}$  can be calculated by  $\rho_{XY} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}$ , where  $\sigma_{XY}$  is the covariance of  $X$  and  $Y$ . The covariance between

$F$  and  $F'$  is simply  $\sigma_F^2$ , thus

$$\begin{aligned}
E(F|f') &= \mu_F + \frac{\sigma_F^2}{\sigma_F \sigma_{F'}} \frac{\sigma_F}{\sigma_{F'}} (f' - \mu_{F'}) \\
&= \mu_F + \frac{\sigma_F^2}{\sigma_{F'}^2} (f' - \mu_{F'}) \\
&= \mu_F + \frac{\sigma_F^2}{\sigma_F^2 + \sigma_N^2} (f' - \mu_{F'})
\end{aligned} \tag{2.7}$$

As the above formula is linear, the expected value of  $F$  for any subset  $R$  of the population can be calculated using equation 2.7, with  $f'$  set to the noisy fitness mean  $\mu_R$  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 2.1, can be used to obtain the expected mean noisy fitness of the mating pool when the selection process is itself based upon noisy fitness values. The expected noisy fitness mean of the mating pool subset can be calculated using equation 2.1, as the mean  $\mu_{F',t}$  and variance  $\sigma_{F',t}^2$  of the noisy population are known:

$$\begin{aligned}
\mu_{F',t+1} &= \mu_{F',t} + I \sigma_{F',t} \\
&= \mu_{F',t} + I \sqrt{\sigma_{F,t}^2 + \sigma_N^2}
\end{aligned}$$

Setting  $f'$  to  $\mu_{F',t+1}$  in equation 2.7 produces the expected true fitness value of the mating pool:

$$\begin{aligned}
E(F_{t+1}|\mu_{F',t+1}) &= \mu_{F,t+1} \\
&= \mu_{F,t} + \frac{\sigma_{F,t}^2}{\sigma_{F,t}^2 + \sigma_N^2} (\mu_{F',t} + I \sqrt{\sigma_{F,t}^2 + \sigma_N^2} - \mu_{F',t}) \\
&= \mu_{F,t} + I \frac{\sigma_{F,t}^2}{\sqrt{\sigma_{F,t}^2 + \sigma_N^2}}
\end{aligned} \tag{2.8}$$

As expected, equation 2.8 reduces to equation 2.1, the formula for the deterministic case, when the noise variance  $\sigma_N^2$  equals zero. Equation 2.8 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 2.8 is that proportionate selection is noise invariant: increased noise levels do not affect the selection pressure of genetic algorithms using proportionate selection! Plugging in  $I = \frac{\sigma_{F',t}}{\mu_{F',t}}$  from Table 2.1 to equation 2.8, and realizing that  $\mu_{F',t} = \mu_{F,t}$  and  $\sigma_{F',t} = \sqrt{\sigma_{F,t}^2 + \sigma_N^2}$  for unbiased normally distributed noise, yields

$$\mu_{F,t+1} = \mu_{F,t} + \sigma_{F,t} \frac{\sigma_{F,t}}{\mu_{F,t}} \tag{2.9}$$

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

### 2.2.2 Noisy Onemax Convergence Model

In section 2.1.4, the basic convergence model for the deterministic domain was derived. This section derives the converge model for the noisy Onemax domain, but 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  $\Delta F_{bb}$ , making the population mean fitness and variance for the Onemax domain be  $\mu_{F,t} = F_{bb,min} + l \Delta F_{bb} p_t$  and  $\sigma_{F,t}^2 = l \Delta F_{bb}^2 p_t (1 - p_t)$ , respectively. Using the same approach outlined in section 2.1.4, along with equation 2.8, it is easy to see that

$$\begin{aligned} p_{t+1} - p_t &= \frac{I}{l \Delta F_{bb}} \frac{\sigma_{F,t}^2}{\sqrt{\sigma_{F,t}^2 + \sigma_N^2}} \\ &= I \frac{p_t(1 - p_t)}{\sqrt{l p_t(1 - p_t) + \sigma_{N'}^2}} \end{aligned} \quad (2.10)$$

where  $\sigma_{N'}^2 = \frac{\sigma_N^2}{\Delta F_{bb}^2}$ . The  $\sigma_{N'}^2$  is effectively the noise scaled relative to the magnitude of the signal size of the building blocks. For the domains considered, the signal size of the building blocks can be calculated by  $\Delta F_{bb} = F_{bb,max} - F_{bb,min}$ .

Approximating equation 2.10 with a differential equation yields

$$\frac{dp}{dt} = I \frac{p_t(1 - p_t)}{\sqrt{l p_t(1 - p_t) + \sigma_{N'}^2}}. \quad (2.11)$$

Although equation 2.11 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 2.3).

While equation 2.11 is not directly solvable for  $p_t$ , it can be solved for  $t$  as a function of  $p$ :

$$\begin{aligned} t(p) &= \frac{1}{I} \left[ \sqrt{l} \arctan \left( \frac{\sqrt{l}(2p - 1)}{2\sqrt{\sigma_{N'}^2 + l p(1 - p)}} \right) + \right. \\ &\quad \left. \sigma_{N'} \ln \left( \frac{p}{p - 1} \frac{-l - 2\sigma_{N'}^2 + l p - 2\sigma_{N'} \sqrt{\sigma_{N'}^2 + l p(1 - p)}}{2\sigma_{N'}^2 + l p + 2\sigma_{N'} \sqrt{\sigma_{N'}^2 + l p(1 - p)}} \right) + c \right] \end{aligned} \quad (2.12)$$

For binary alleles, at time  $t = 0$  it can be assumed that half of the alleles are initially correct  $p = 0.5$ . Using this to solve for  $c$  in equation 2.12 gives that  $c = 0$ . Of particular interest is the time  $t_{conv}$  it takes for all alleles to converge ( $p = 1$ ). For the deterministic case, equation 2.12 reduces to

$$t_{conv} = \frac{\pi \sqrt{l}}{2I} \quad (2.13)$$

which is equivalent to the result obtained in equation 2.4.

While  $t_{conv}$  can not 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 2.12 in Miller and Goldberg (1996b), 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 2.2.

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 $

Table 2.2: Time of Convergence Approximations for Noisy Onemax Domain

### 2.2.3 Experimental Validation

In this subsection, the accuracy of the noisy selection models derived above is verified for four different selection schemes: tournament selection,  $(\mu, \lambda)$  selection, linear ranking, and stochastic 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 is then compared to the results predicted by the models derived above.

#### 2.2.3.1 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  $N = 8(\sigma_F^2 + \sigma_N^2)$ . This is a conservative population sizing model, in that it returns population sizes that are larger than the minimum needed to achieve a certain level of accuracy. However, as population sizing is not being explored in this chapter, conservatively large population sizes were used so as to avoid population sizing effects. The alleles use the standard weighting of one, so  $\Delta F_{bb} = 1$ . For a binomially distributed population, the initial population variance is  $lp(1-p)$ , and the initial the 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 combination 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. As the Onemax domain has a building block length of one, the recombination operator can not 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 as it thoroughly mixes the existing building blocks in the population. For the uniformly scaled domains of interest to this research, 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 one, two, or N-point crossover also have normalizing effects, although less pronounced. In addition, mutation can also has 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.

#### 2.2.3.2 Results

This subsection details the results of the experiments. The experimental results are plotted along with the predicted performance for each selection scheme in Figures 2.1-2.4. In each plot, the dotted line is the predicted performance, and the solid lines are the experimental performance (avg. of 10 runs). In most cases, the model accurately predicts the experimental performance, so that the predicted performance (dashed line) is mostly obscured by the experimental results (solid line). The small difference between predicted and experimental performance is due to building block covariance build-up (“hitchhiking”); this results in a population whose fitness deviates from the normal fitness distribution assumed by the predictive model. In (Thierens & Goldberg, 1994; Miller & Goldberg, 1996b), 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 can be attributed to building block covariance build-up.

For the ordinal-based selection mechanisms (tournament selection,  $(\mu, \lambda)$  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 noises levels of  $\sigma_N^2 = \{0, 12.5, 25, 50, 100\}$ . Equations 2.3 and 2.11 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 selection schemes is calculated according to Table 2.1, and is described in more detail below. The results for proportionate selection are presented in the section on stochastic universal selection below.

#### 2.2.3.3 Tournament Selection

For tournament selection, a tournament size of  $s = 2$  was used, with a corresponding selection intensity (from Table 2.1) of  $I = \mu_{2:2} = 0.5642$ . The maximal order statistic value was obtained from Harter (1970).

The results, plotted in Figure 2.1, demonstrate that the convergence models were very accurate at predicting the convergence of tournament selection.

#### 2.2.3.4 $(\mu, \lambda)$ Selection

For  $(\mu, \lambda)$  selection,  $\mu = 4$  and  $\lambda = 8$  was used. The corresponding selection intensity was calculating using Table 2.1, along with the order statistic values obtained from Harter (1970).

$$I = \frac{1}{4}(\mu_{5:8} + \mu_{6:8} + \mu_{7:8} + \mu_{8:8})$$

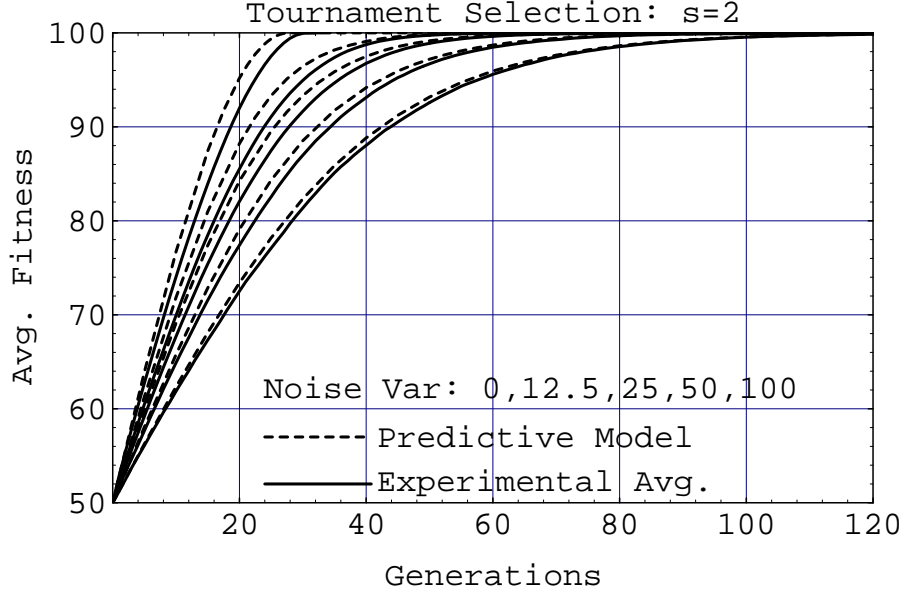


Figure 2.1: Tournament Selection Results for the Onemax Domain

$$\begin{aligned}
 &= \frac{1}{4}(0.1525 + 0.4728 + 0.8522 + 1.4236) \\
 &= 0.7253.
 \end{aligned}$$

The results, plotted in Figure 2.2, demonstrate that the convergence models were very accurate at predicting the convergence of  $(\mu, \lambda)$  selection. Note that this experiment had the same  $\alpha = \frac{\mu}{\lambda}$  ratio value of 0.5 as the binary tournament selection experiment, yet had a higher selection pressure. The tradeoff of effectively having higher values for  $\mu$  and  $\lambda$  for the same  $\alpha$  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  $(\mu, \lambda)$  selection, if there are several possible  $\mu$  and  $\lambda$  combinations that produce a desired selection pressure, the combination with the smallest  $\lambda$  should be preferred in order to maximize population diversity.

#### 2.2.3.5 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 2.1) is

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

which is equivalent to the binary tournament selection case.

The results, plotted in Figure 2.3, demonstrate that the convergence models were very accurate at predicting the convergence of linear ranking.

#### 2.2.3.6 Stochastic Universal Selection

As equation 2.9 demonstrates that the convergence equation for noisy environments is equivalent to the deterministic case for proportionate selection, the deterministic model, equation 2.5,

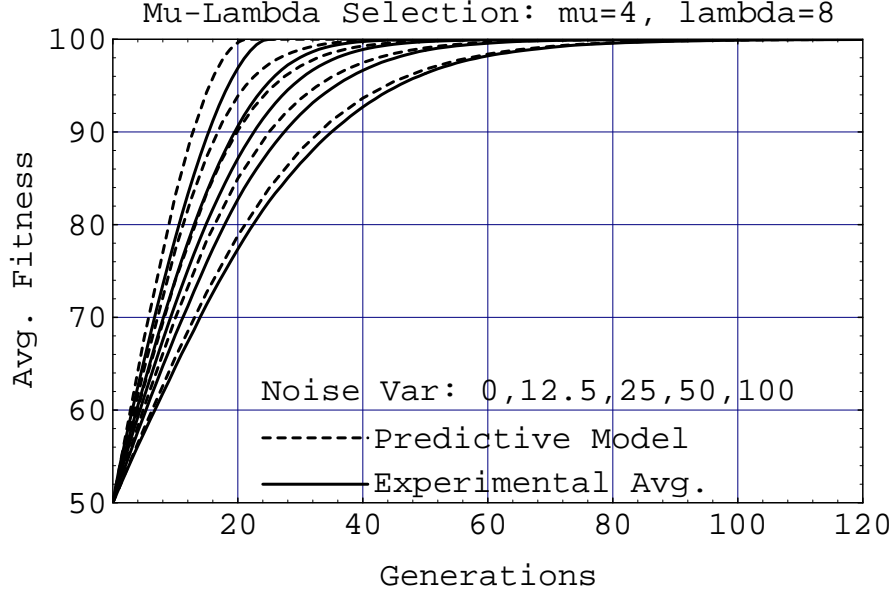


Figure 2.2:  $(\mu, \lambda)$  Results for the Onemax Domain

is used to predict the performance for stochastic universal selection for all noise levels. The predictive accuracy of the model for stochastic universal selection, equation 2.5, is shown in Figure 2.4. This figure is shown at the same scale as the others so as to ease comparisons between figures. As the model is noise invariant, the runs at the five different noise level produced roughly the same plot, and therefore overlay each other in the figure. This accounts for the one dark 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.

While Figure 2.4 demonstrates that the 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. While the number of generation 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 generations to 990 generations due to 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 size of the population. However, the genetic drift is not free to let the population drift any which way, as 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 towards the optimum. Accurate modeling of this one-way genetic drift in the presence of noise is an interesting topic that merits further research.

### 2.3 From Alleles to Building Blocks

In the previous section, an allele-wise model for how noise affects selection pressure in the Onemax domain was given. This section demonstrates how to apply this allele-wise model at

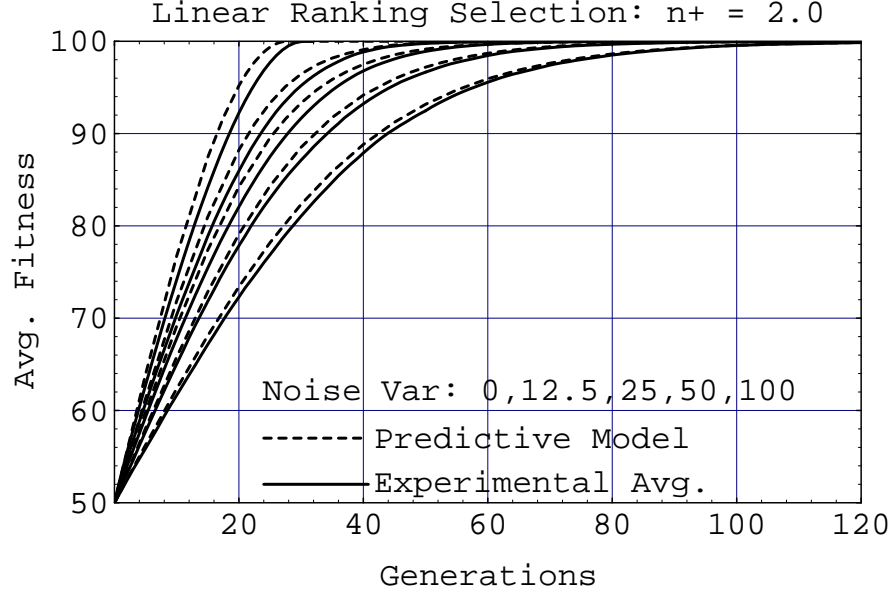


Figure 2.3: Linear Ranking Results for the Onemax Domain

the building block level to determine how noise affects the selection pressure in more complex domains.

### 2.3.1 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. The first item 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 of 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 due to the fact that the building blocks are independent and uniformly scaled.

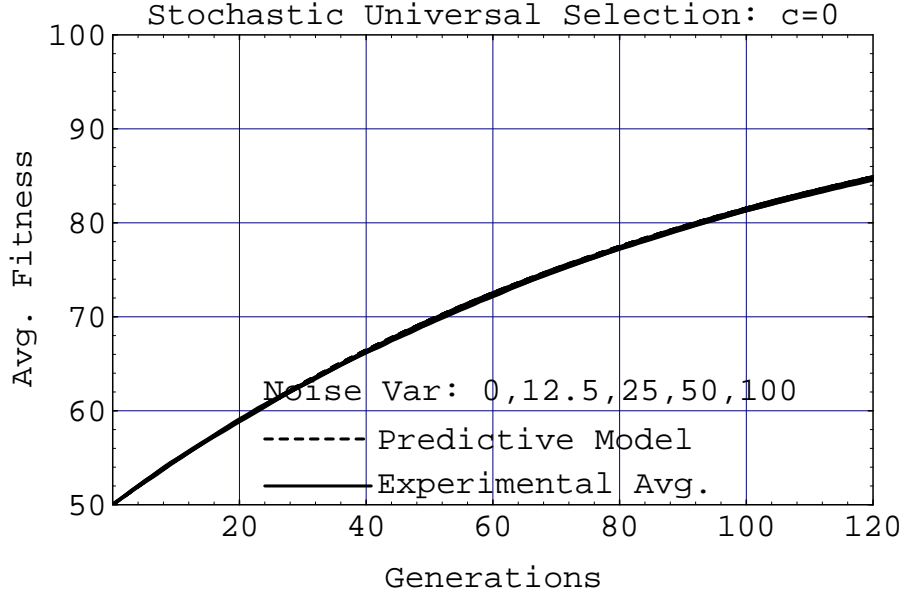


Figure 2.4: Stochastic Universal Selection Results for the Onemax Domain

While 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, Deb, & Thierens, 1993; Mühlenbein & Schlierkamp-Voosen, 1993; Mühlenbein & Schlierkamp-Voosen, 1994; Aizawa, 1996). This research 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 affect 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: a uniform crossover operator (Syswerda, 1989) at the building block level is used, and is denoted as the *uniform building block crossover operator*. This operator thus swaps only whole building blocks, and as such, neither destroys or 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 one or two point crossover. However, the uniform building block operator was preferred as it had no building block disruption, and also did a better job at reducing the covariance between building blocks. Using this uniform building block crossover operator has the effect of bringing the assumption of the modeling closer to those of the empirical study. It also has the effect of moving the modeling somewhat further away from conditions faced in actual practice. Nonetheless, I 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 subsections demonstrate a rational adaptation of 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  $l = 100$ , building block size  $k = 4$ , and 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, and thus the optimal chromosome had a fitness value of  $m = 25$ . As 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  $p_{unif} = 0.5$ . The mutation rate was zero. The GA was run until the proportion of optimal building blocks in the population was  $p_{bb,conv} = (m - 0.5)/m = 0.98$ . These experiments were run using binary tournament selection ( $s = 2$ ), but any of the selection mechanisms listed in Table 2.1 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  $k = 4$ , the population sizing equation is conservatively set 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.

### 2.3.2 Royal Road Level-0 Domain

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

$$F_{bb}(x) = \begin{cases} 1 & \text{if } x = k \\ 0 & \text{otherwise} \end{cases} \quad (2.14)$$

where  $x$  is the number of ones 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 is  $p_{bb,0}$  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  $\sigma_{F,t}^2 = m p_{bb,t} (1 - p_{bb,t})$ . This can be solved as in the Onemax case to yield equation 2.12, but with  $m$  and  $p_{bb}$  replacing  $l$  and  $p$  respectively. This model is then used to calculate the time until  $p_{bb,conv} = 0.98$ , and the results are compared in Table 2.3. For each noise level, the constant  $c$  in equation 2.12 is calculated from knowing  $m$ ,  $p_{bb,0}$  and the noise level.

	0		0.5 $\sigma_{F,0}^2$		1 $\sigma_{F,0}^2$		2 $\sigma_{F,0}^2$		4 $\sigma_{F,0}^2$	
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

Table 2.3: Building Block Domain Convergence Times

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 build-up (“hitchhiking”); this results in a population whose fitness deviates from the normal fitness distribution assumed by the predictive model. In (Thierens & Goldberg, 1994; Miller & Goldberg, 1996b), 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 can be attributed to building block covariance build-up.

### 2.3.3 Onemax Building Block Domain

For the next test domain, a domain that had two main characteristics was desired. 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 maximal (optimal) building block fitness value. To do this, we designed the Onemax building block domain, whose building block fitness function is given below.

$$F_{bb}(x) = \frac{x}{k} \quad (2.15)$$

The variable  $x$  is again the number of ones 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 approximately have 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 then the allele Onemax model is used to solve the convergence time. The key relationship to link the building block to allele model is  $p_{bb} = p^k$ , so  $\sqrt[k]{p_{bb}} = p$ . Thus to use the allele model, set the initial condition to  $p_0 = \sqrt[k]{p_{bb,0}}$ , where  $p_{bb,0} = .5^k$ , and the desired convergence proportion to  $p_{conv} = \sqrt[k]{p_{bb,conv}}$ . 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  $p_{bb,conv}$ . The chromosome length for equation 2.12 applied at the allele level is  $l$ , and  $\Delta F_{bb} = 1/k$ . For each noise level, the constant  $c$  in equation 2.12 is calculated from knowing  $l, p_0$ , and the noise level. Table 2.3 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 build-up of covariance of building blocks, and can be eliminated by multiple rounds of uniform building block crossover per generation (Thierens & Goldberg, 1994).

### 2.3.4 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 the disruption is turned off 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:

$$F_{bb}(x) = \begin{cases} 1 & \text{if } x = k \\ (1 - d)^{\frac{k-1-x}{k-1}} & \text{otherwise} \end{cases} \quad (2.16)$$

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 the 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 easily be calculated by enumerating each of the  $2^k$  possible building blocks and calculating their fitness. For  $d = .25$  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 2.3. 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 domain due to 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, as 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 affect of noise on selection pressure and the resulting convergence performance for uniformly-scaled building block domains.

While the analysis works well for the domains studied, the use of the uniform building block crossover operator as an experimental tool necessitates that it 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) will perform similarly to uniform building block crossover.

While an understanding on how the selection pressure influences the convergence characteristics of a GA has been reached, it is only part of the picture, albeit the most important. More work still has to be done to fully understand how building block disruption from recombination and mutation operators affect the convergence properties of a GA, but that is outside the intended scope of this research.

## 2.4 Conclusions

Now that GAs are increasingly being used by industry, the effects of noise on performance are becoming more important. This chapter'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 is derived that accurately predicts the performance

of several selection schemes in noisy environments, including tournament selection,  $(\mu, \lambda)$  selection, linear ranking and stochastic universal selection. Furthermore, any selection scheme model based upon the deterministic selection intensity model can be similarly adapted to handle noise.

The noisy selection intensity model developed in this chapter 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 solution quality after a certain number of generations, and 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 an optimal sample size for the fitness function. An optimum 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 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 genetic algorithms.

This research also has several long term ramifications. The basic approach of this research 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 different levels of noise will have on GA performance.

## Chapter 3

# Population Sizing and Noise

An important decision in implementing a GA is determining the correct population size for the GA. The population size directly affects the expected solution quality of the GA: larger population sizes have better population diversity, and are thus less prone to premature convergence. Another way of stating this is that large population have an adequate supply of building blocks. The tradeoff for using large population sizes is the additional computational overhead needed to process large populations. The function of a population sizing model is to successfully predict the convergence performance of a GA for a particular population size. Population sizing models can then be used to determine the smallest population size that will provide an acceptable level of performance from the GA. Unfortunately, there are only a few studies available to guide the decision making process of determining the optimal population size (Goldberg, 1989b; Goldberg & Rudnick, 1991; Goldberg, Deb, & Clark, 1992; Goldberg, Deb, & Clark, 1993).

### 3.1 Building Block Population Sizing

Traditionally, population sizing models have been derived along two different approaches. The first approach, Building Block Decision Models, models the supply of building blocks in the initial population. The second approach analyzes the ability of the GA to retain good building blocks over time by focusing on the decision quality of the selection process. These two approaches are discussed below.

#### 3.1.1 Building Block Supply Models

Building block supply models attempt to estimate the supply of optimal building blocks in the initial population. These models use the initial optimal building block estimate to estimate the premature convergence likelihood (Goldberg, 1989b; Goldberg, Deb, & Clark, 1992). The main premise behind these models are that the GA needs an adequate supply of BBs in order to optimally converge. If there is not an adequate supply of BBs, optimal BBs will be “lost”, thus preventing optimal convergence of the population. An important result of these models is the estimate for the initial supply  $x_0$  of an optimal BB that is of length  $k$  in a binary domain is:

$$x_0 = \frac{N}{2^k} \quad (3.1)$$

where  $N$  is the population size. This follows from the fact that given a random initialization in a binary domain, a specific BB of length  $k$  has a  $\frac{1}{2^k}$  chance of being randomly created.

These models are important in that they can be used as rough heuristic for determining whether the starting supply of BBs is adequate. However, the model does not take into account the selection pressure, and thus predicts that the required population size is invariant of the selection pressure. This can result in population sizing errors for low or high selection

pressures.

### 3.1.2 Building Block Decision Models

The decision process refers to the ability of the GA to make good decisions as to which population members should be retained as the basis of the next generation. The selection mechanism is the GA component that is responsible for making the correct decisions. The selection mechanism has two main functions. First, it should retain an adequate supply of optimal building blocks so that the optimal solution can be constructed. Second, it must remove bad building blocks so that the overall fitness of the population improves over successive generations. Unfortunately, the selection scheme does not make decisions at the building block level, but at the chromosome level. By selecting at the chromosome level, the selection process can inadvertently make bad decisions at the building block level. Building block decision models measure the likelihood of retaining better building blocks when making building block decisions based on the chromosome fitness values.

Goldberg and Rudnick (1991) derived a population sizing model based on the fitness variance of the population. More recently, Goldberg, Deb, and Clark (1992) developed a decision-based model to conservatively bound the convergence quality of a GA based on the population size. Their model calculates the probability  $p$  of retaining better building blocks for the binary tournament selection mechanism. Their model for determining the good decision probability  $p$  is discussed below, as it forms an important part of the random walk population sizing model presented later.

The probability of binary tournament selection making the correct decision at the BB level can be analyzed in the following manner (Goldberg, Deb, & Clark, 1992). Consider a tournament, based on the chromosome fitness values, between two individuals where one individual,  $H1$ , contains the optimal building block for the partition  $i$  of interest, and the second individual,  $H2$ , contains the second best building block in partition  $i$ . Assuming that the optimal building block in  $H1$  contributes a fitness signal of magnitude  $d$  better than the second best building block in  $H2$ , the fitness distributions of  $H1$  and  $H2$  can be estimated: see Figure 3.1.

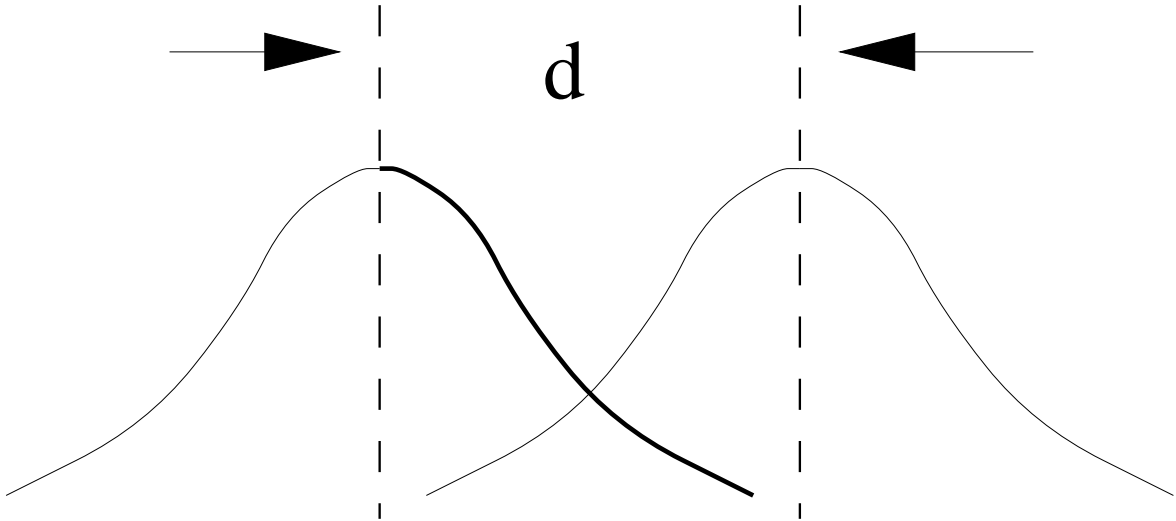


Figure 3.1: Fitness Distribution of Tournament Competitors

As the fitness signal difference between the optimal and second best building blocks in partition  $i$  is  $d_i$ , the distance between the means of the fitness distributions for  $H1$  and  $H2$  is also  $d_i$ . To calculate the variance of the distributions, assume that the total population fitness variance  $\sigma_F^2$  is equal to the sum of each partitions' variance in the chromosome. Then the variance of  $H1$  and  $H2$  is simply  $\sigma_F^2 - \sigma_i^2$ , as the building blocks in partition  $i$  are fixed (optimal and second best building blocks respectively) and thus the variance  $\sigma_i^2$  normally contributed by partition  $i$  to the overall fitness variance must be dropped. The partition fitness distributions are assumed to be normal. Now the question becomes given the fitness distributions for chromosomes  $H1$  and  $H2$ , what is the probability that a binary tournament between competitors randomly chosen from the  $H1$  and  $H2$  distributions will result in the individual from the  $H1$  distribution having the higher fitness? This gives the probability that the chromosome containing the optimal building block for partition  $i$  will be chosen by the tournament selection mechanism, and thus the probability of tournament selection correctly choosing the optimal building block.

To compute this probability, all possible values of the area of the region where the fitness distributions of the two schemas overlap must be accumulated. This computation is called a *convolution* and, in the case of normal variables, the convolution itself is normally distributed and has known properties. The mean of the convolution is the difference of the means of the two individual distributions, and the variance of the convolution is the sum of the individual variances.

Defining the *signal difference* in partition  $i$  as  $d_i = \overline{f_{H1}} - \overline{f_{H2}}$ , and the mean variance of the two BB schemas as  $\sigma_M^2 = (\sigma_{H1}^2 + \sigma_{H2}^2)/2$ . The probability of binary tournament selection making an error by choosing the chromosome containing BB  $H_2$  can be calculated by finding the probability  $p$  such that  $z^2(p) = d^2/(2\sigma_M^2)$ , where  $z(p)$  is the ordinate of a unit normal distribution. To calculate this, the schema fitness variance for both  $H_1$  and  $H_2$  in partition  $i$  is estimated. The chromosome fitness variance is the sum of partition fitness variances:  $\sigma_F^2 = \sum_{i=1}^m \sigma_i^2$ . The fitness variance for a particular BB in partition  $i$  (i.e. schema  $H_1$  or  $H_2$ ) can be represented as the total chromosome fitness variance minus the fitness variance of the partition:  $\sigma_F^2 - \sigma_i^2$ . This indicates the the collateral noise is due to the fitness variance of all other partitions but the one we are interested in, partition  $i$ . Using this results in  $\sigma_M^2 = \sigma_F^2 - \sigma_i^2$ . The probability of making the correct choice  $p_i$  in partition  $i$  between competing individuals when the selection mechanism is binary tournament selection can be calculated by the convolution of the two fitness distributions:

$$p_i = \mathbb{N}\left(\frac{d_i}{\sqrt{2(\sigma_F^2 - \sigma_i^2)}}\right) \quad (3.2)$$

where  $\mathbb{N}(x)$  is the cumulative distribution function from negative infinity to  $x$  of the unit normal distribution.

For domains where the  $m$  partitions are equally weighted (uniformly scaled domains), the fitness variance can be represented as  $\sigma_F^2 = m\sigma_i^2$ , where all the partition variances  $\sigma_i^2$  are equivalent. This results in  $\sigma_M^2 = \frac{m-1}{m}\sigma_F^2$ , and thus the good decision probability can be represented as:

$$p = \mathbb{N}\left(\frac{d}{\sigma_F \sqrt{\frac{2(m-1)}{m}}}\right) \quad (3.3)$$



Figure 3.2: Gambler's Ruin Problem

The signal difference and correct decision probability are not denoted with a  $i$  subscript as they are equivalent for all of the building blocks in a uniformly scaled domain.

Goldberg, Deb, and Clark (1992) apply the good decision probability  $p$  to derive a population sizing model. The resulting model gave a good rough heuristic for the population size. However, their population sizing model can still be improved upon. Their population sizing model is based upon the above good decision probability, which only holds for binary tournament selection. The population sizing model should be extended to other selection intensities. However, a strength of their model is that it demonstrated how to incorporate the effects of noise into the population sizing equation. In Harik, Cantú-Paz, Goldberg, and Miller (1997) the strengths and weaknesses of their original population sizing model are analyzed in more detail. The random walk model, introduced in the next section, incorporates the strengths of both the BB supply and decision population sizing models to create a more accurate population sizing model.

## 3.2 Random Walk Population Sizing

### 3.2.1 Random Walk Overview

In (Harik, Cantú-Paz, Goldberg, & Miller, 1997), a breakthrough was made for population sizing by relating it to the gambler's ruin problem, which has been extensively studied. The basic gambler's ruin problem estimates how long it will take for a gambler to either go broke or win all of the money. A simple form of the problem relies on the probability of a gambler winning  $p$ , the probability of the gambler losing a hand  $q$ , the gambler's starting stake represented as  $x_0$ , and the total amount of money available  $N$ . Each hand can win or lose 1 unit, and the game stops if the gambler either goes broke or wins all the money (absorbing states). Figure 3.2 displays the basic gambler's ruin problem.

The probability that the gambler wins all of the money is given by:

$$P_N = \frac{1 - (q/p)^{x_0}}{1 - (q/p)^N} \quad (3.4)$$

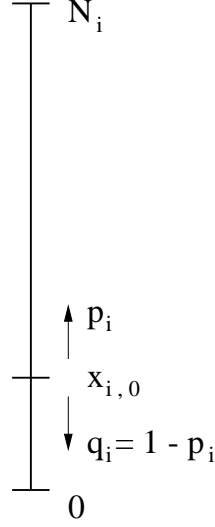


Figure 3.3: Random Walk Population Sizing

### 3.2.2 Random Walk Population Sizing

The random walk population model draws an analogy between the gambler's ruin problem and population sizing; see Figure 3.3. This model, like other population sizing models based on building blocks, determines the population sizing requirement relative to a single partition. The overall population size can be obtained by selecting the maximum population size from the population sizes calculated for each partition. However, note that for uniformly scaled domains, all of the partition population size requirements are equivalent, and thus the random walk population sizing model can be used to predict the overall population sizing requirement.

The population size requirements relative to each partition  $i$  can be calculated in the following manner. The optimal building blocks are the currency that can be won in each tournament decision, and there are at most  $N_i$  optimal buildings to be won where  $N_i$  is the population size. This model only considers tournaments where an optimal building block can be won or lost, as this is the only mechanism by which selection can change the proportion of optimal building blocks in the population. The starting number of optimal building blocks in partition  $i$  can be represented as  $x_{i,0}$ , which can be estimated using a building block supply model:  $x_{i,0} = N_i/2^{k_i}$ , where  $N_i$  is the population size and  $k_i$  is the building block size for partition  $i$ . If  $p_i$  is the probability that the optimal building block is kept by tournament selection, the probability of a bad decision is simply  $q_i = 1 - p_i$ .

The probability of a successful tournament,  $p_i$ , can be calculated using the building block decision models, as given by equation 3.2. For the population sizing model,  $p_i > q_i$ , so the denominator of equation 3.4 can be ignored as it approaches 1 very quickly. The probability for a partition to converge to the optimal building blocks can then be given as:

$$P_N \approx 1 - \left( \frac{1 - p_i}{p_i} \right)^{N_i/2^{k_i}} \quad (3.5)$$

Let the probability of premature convergence  $\psi$  (also known as the failure rate) be repre-

sented as  $\psi = 1 - P_N$ , and solving for the population size  $N_i$ :

$$N_i = 2^{k_i} \ln(\psi) / \ln\left(\frac{1 - p_i}{p_i}\right) \quad (3.6)$$

This equation is a significant advance over previous population sizing equations in that it takes into account both the initial supply of building blocks  $x_{i,0}$ , and the decision quality of the selection scheme  $p_i$ . For uniformly scaled domains, such as the Onemax or Maxtrap domains, Equation 3.3 can be used to calculate  $p_i$ . For uniformly scaled domains, the probability that a particular partition converges to the optimal building block (equivalent to  $1 - \psi$ ) is equal to the expected proportion of optimally converged partitions in the final converged population. This is utilized in verifying the models in the next section.

### 3.2.3 Experimental Verification

Experimental results for the predicted proportion of optimal building blocks in the converged population are given for the Onemax domain, Figure 3.4, and for the Maxtrap domain, Figure 3.6. These experiments are reproduced from the study done by Harik, Cantú-Paz, Goldberg, and Miller (1997). These plots demonstrate the accuracy of the random walk population sizing model. The proportion of optimal building blocks in the converged population are plotted over a range of population sizes. In both of these plots the prediction of the random walk model is in bold, the experimental results are the dotted line, and the previous decision-based model presented in Goldberg, Deb, and Clark (1992) is the thin line.

For the Onemax domain, the chromosome length was 100 bits, with a building block length  $k$  of one. No mutation operator is used. The crossover operator was uniform crossover (Syswerda, 1989), with a uniform crossover swap probability of 0.5. This crossover operator was chosen as it does a good job of mixing the building blocks, and as the case with all crossover operators in domains with one bit building blocks, does not cause any building block disruption.

The second experiment is for the Maxtrap, also known as the trap function, domain, which is presented by Goldberg, Deb, and Clark (1992). The building block length  $k$  is four, with a signal difference  $d$  of one, and a resulting building block fitness variance of 1.215. The fitness function for the four bit crossover operator is shown in Figure 3.5. There are twenty building blocks in the chromosome, making a chromosome length of  $l = 80$ . A two-point crossover operator was chosen to avoid the excessive BB disruption of uniform crossover. The results are shown in Figure 3.6.

For both of these domains, the random walk does a very good job at predicting the proportion of optimal building blocks in the converged population for a wide range of population sizes. The previous population sizing model is shown to be conservative, as the experimental performance is much better than the performance predicted the previous population sizing model.

## 3.3 How Selection Pressure Affects Population Sizing

### 3.3.1 Modeling Selection Pressure

In Harik, Cantú-Paz, Goldberg, and Miller (1997), the building block decision model developed in Goldberg, Deb, and Clark (1992) for binary tournament selection is extended to tournament



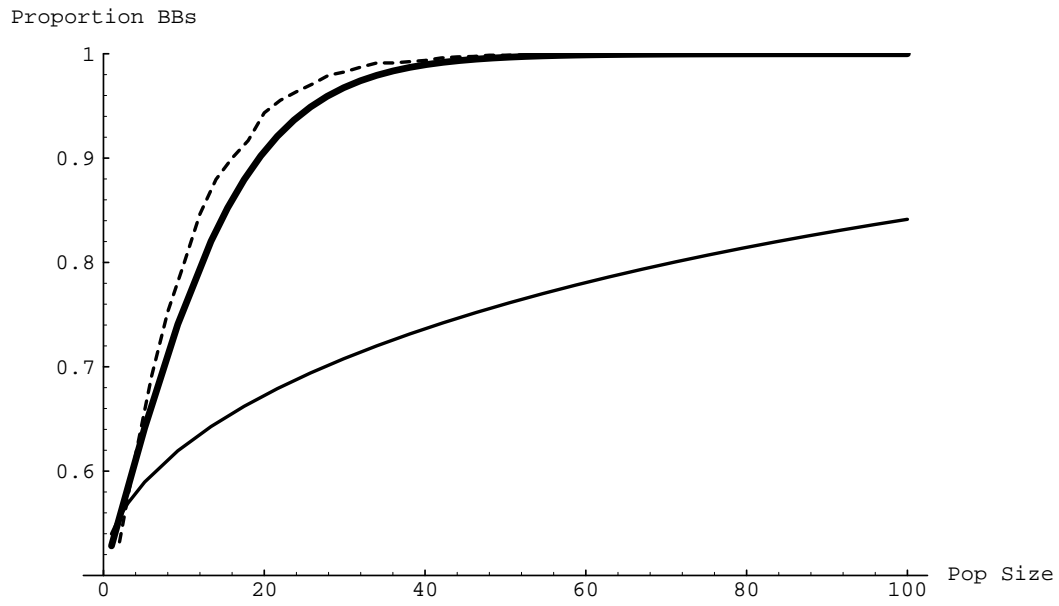


Figure 3.4: Population Sizing for Onemax Domain (Harik, Cantú-Paz, Goldberg, & Miller, 1997)

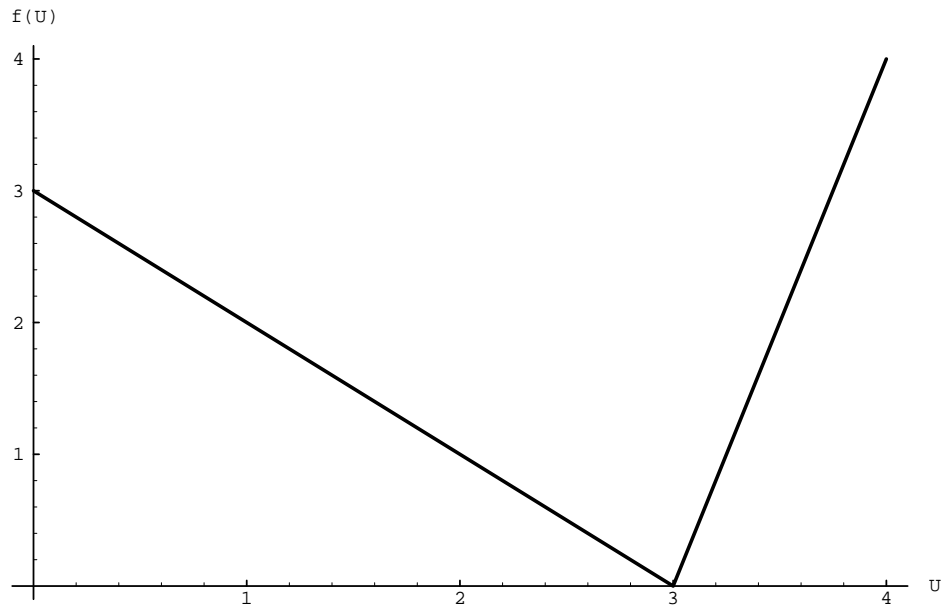


Figure 3.5: Building Block Maxtrap Function for  $k = 4$  (Harik, Cantú-Paz, Goldberg, & Miller, 1997)

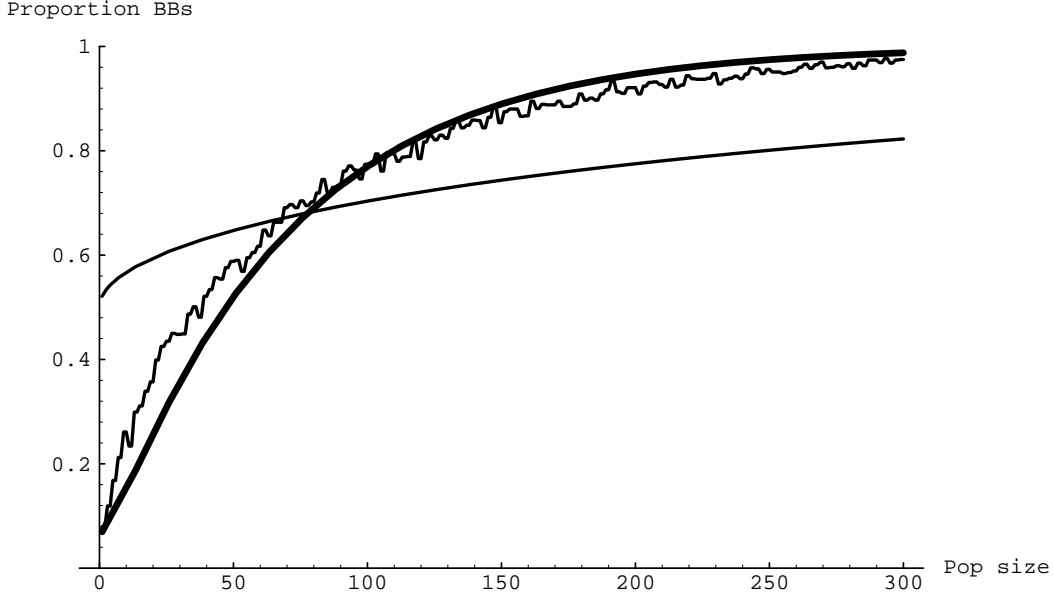


Figure 3.6: Population Sizing for Maxtrap Domain (Harik, Cantú-Paz, Goldberg, & Miller, 1997)

sizes greater than two. Given a tournament of size  $s$ , assume that there is one competitor containing the optimal building block, and  $s - 1$  competitors that do not. The tournament can be reduced to a binary tournament between the competitor containing the optimal BB and the *best* competitor of the remaining  $s - 1$  individuals. As the competition is between the optimal and the best, and not the average, of the remaining  $s - 1$  individuals, this has the effect of reducing the difference  $d$  of the convolution. One can use order statistics to estimate the new convolution difference  $d'$ , or can use the approximation given below:

$$d' = d + z(1/s)\sigma_{BB} \quad (3.7)$$

where  $z(1/s)$  is the ordinate of a unit normal distribution where the CDF equals  $1/s$ .

The term  $z(1/s)$  is increasingly negative for larger tournament sizes, thus reducing the effective convolution distance  $d'$ . The new convolution distance  $d'$  is then used instead of  $d$  in calculating  $p$  when using equations 3.2 or 3.3. While this model is based on tournament selection, the results for other selection schemes can be obtained by extrapolating from tournament selection by selecting the tournament size with equivalent selection intensity.

### 3.3.2 Experimental Verification

Experiments were run to verify this model, again using the Onemax domain. Experiments were run for tournament sizes 2, 4, and 8. The experimental results are plotted in Figure 3.7, with the dotted lines representing the theoretical predictions and solid lines the experimental results. The leftmost plot corresponds to a tournament size  $s = 2$ , the next is  $s = 4$ , and the rightmost is  $s = 8$ . Once again, the model is a good predictor for the proportion of BBs correct at the end of the run, even for a wide range of tournament sizes. The chromosome length was  $l = 100$ .

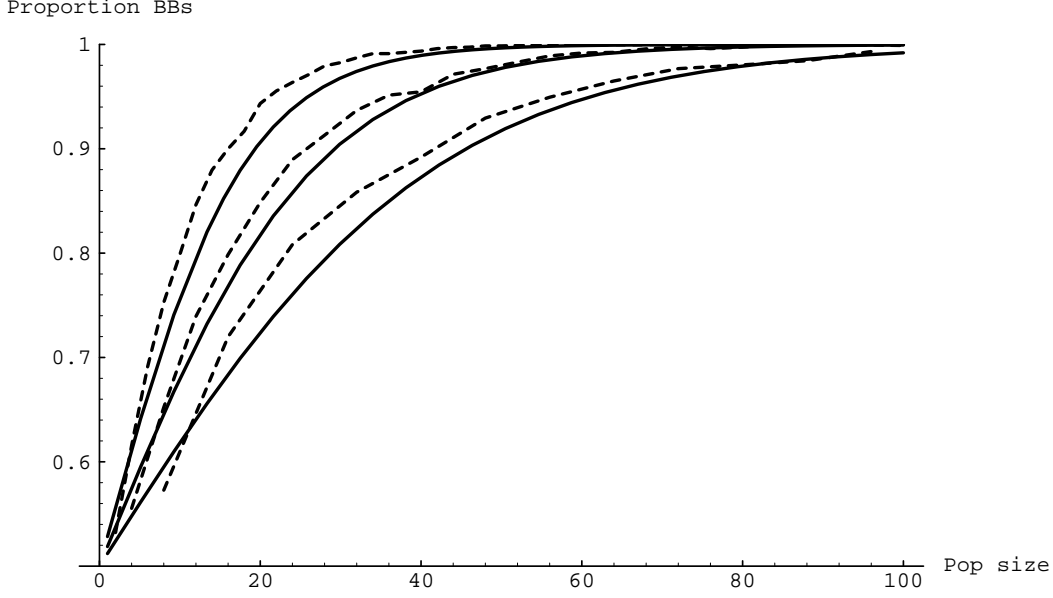


Figure 3.7: Population Sizing as Function of Selection Pressure(Harik, Cantú-Paz, Goldberg, & Miller, 1997)

## 3.4 How Noise Affects Population Sizing

### 3.4.1 Modeling Noise

Noise also affects the ability of a selection mechanism to make correct selection decisions at the building block level. In Goldberg, Deb, and Clark (1992), their population model was adapted to handle the effects of noise by determining the effect of noise on the optimal building block selection probability  $p$ . This section will apply a similar derivation to analyze the effects of noise for the random walk population sizing model.

Noise increases the fitness variance of the population. If the population fitness is normally distributed  $N(\mu, \sigma_F^2)$ , and the noise is normally distributed  $N(0, \sigma_N^2)$ , then the noisy fitness is distributed  $N(\mu, \sigma_F^2 + \sigma_N^2)$ . Given this, the noisy fitness distribution of  $H1$  and  $H2$ , used in determining the optimal building block selection probability  $p_i$  for partition  $i$ , can be calculated as in section 3.1.2, but using a variance of  $\sigma_F^2 + \sigma_N^2$  instead of  $\sigma_F^2$ . This changes the convolution for calculating the win probability (equation 3.2 to

$$p_i = \mathbb{N}\left(\frac{d_i}{\sqrt{2(\sigma_F^2 + \sigma_N^2 - \sigma_i^2)}}\right) \quad (3.8)$$

Similar to the deterministic case for uniformly scaled domains (equation 3.3), this can be simplified to:

$$p = \mathbb{N}\left(\frac{d_i}{\sqrt{2\left(\frac{m-1}{m}\sigma_F^2 + \sigma_N^2\right)}}\right) \quad (3.9)$$

### 3.4.2 Experimental Verification

The accuracy of equation 3.9 is verified by assessing its ability to accurately predict the performance of different population sizes at different noise levels for the Onemax domain. In Figure 3.8, equation 3.9 is used in conjunction with equation 3.6 to predict the required

population sizes for a given level of performance. For uniformly scaled building blocks, the probability that a given building block will optimally converge ( $1 - \psi$ ) is equivalent to the expected proportion of optimally converged building blocks in the final (converged) population. The experiments were run using a 120-bit Onemax problem with uniform crossover and no mutation for a variety of different noise levels. For a 120-bit Onemax problem, the initial fitness variance is  $\sigma_F^2 = l/4 = 30$ , and the experiments were run for  $(0, 1, 2) * \sigma_F^2$  noise levels. Figure 3.8 displays the results, with the dashed line showing the experimental averages over 50 runs, and the solid line being the predicted performance.

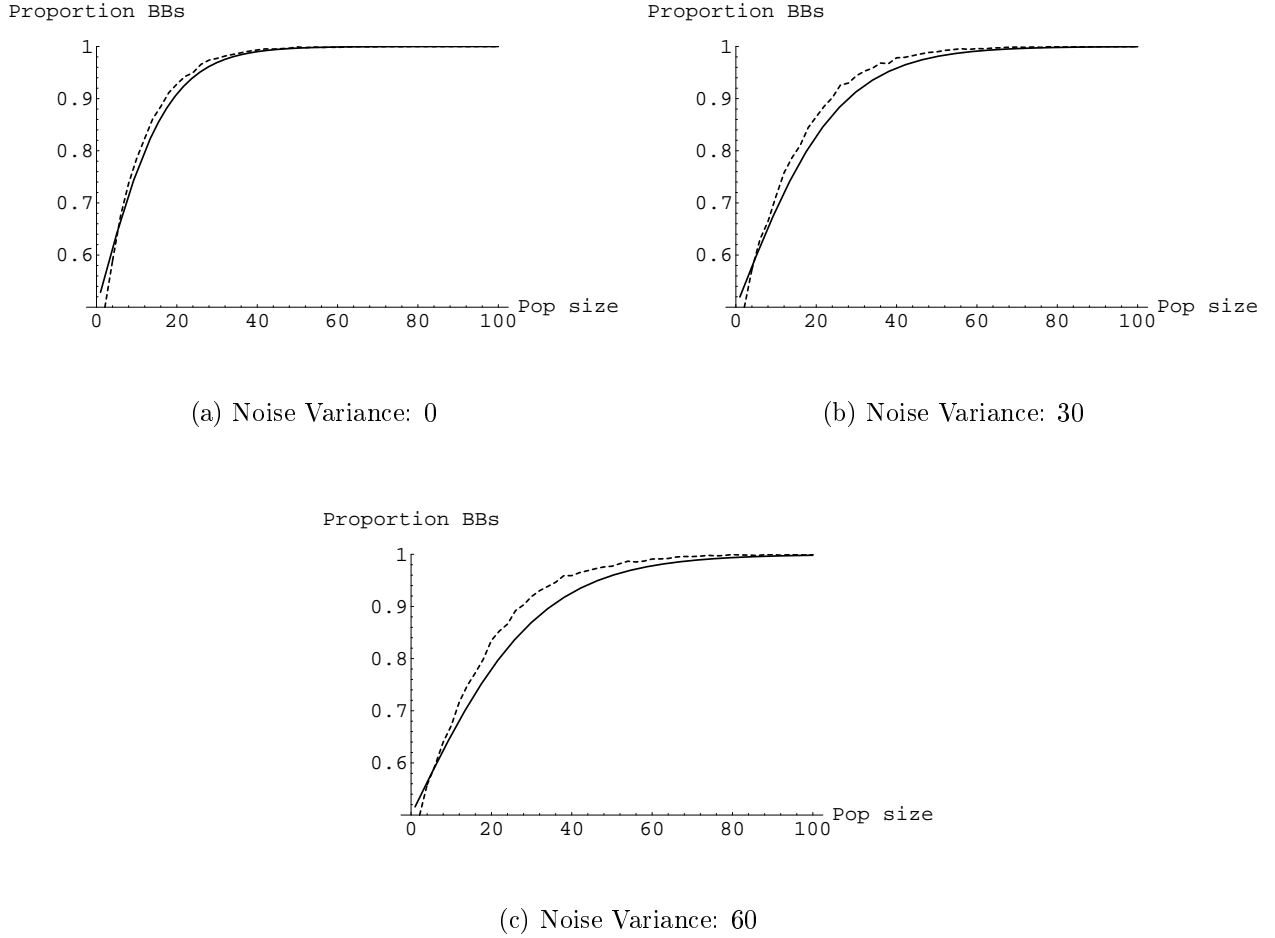


Figure 3.8: Population Sizing as Function of Noise

Figure 3.8 clearly demonstrates that the random walk population sizing model can be used to accurately model the effects of noise. However, the figure also clearly shows that the model is increasingly less accurate for higher levels of noise. Further research is needed to explore why additional noise decreases the accuracy of the model. However, the model is still conservative, and thus can safely be used to model the effects of noise.

### 3.5 Random Walk Population Sizing Generalization

The above random walk population sizing model depends on domain knowledge at the building block level ( $d_i$  and  $\sigma_i^2$ ). For domains where this building block information is available, the population size can be conservatively set equal to the largest population size required by the partitions. For the special case of uniformly scaled domains, this is equivalent to the population size calculated for any of the equivalent partitions.

However, in many domains the partitioning of the chromosome into its independent building blocks constituents is not known, making it impossible to use building block population sizing models. This section seeks to identify a conservative population sizing model that does not depend on the building block knowledge, but only general knowledge of the fitness space. Next, an approximation for this general population sizing model is derived. This approximation is shown to be accurate, and useful for analyzing the effects of different variables on the population sizing requirements.

#### 3.5.1 General Population Sizing Model

Equations 3.6 and 3.8 form an accurate population sizing model, but they are based on building block knowledge. However, the population size can be conservatively estimated by making conservative assumptions for the building block variables. In equation 3.6,  $k_i$  can be replaced by  $k_{max}$ , which is estimate of the maximum length of a building block in the domain. Likewise, in equation 3.8,  $d_i$  is replaced by  $d_{min}$ , which is an estimate of the minimum signal difference between the first and second best building block in the domain. The building block noise variance  $\sigma_i^2$  is conservatively set to zero, as it reduces the noise signal in the denominator. Taken together, the new equations for the domain population sizing model are:

$$N = 2^{k_{max}} \ln(\psi) / \ln\left(\frac{1-p}{p}\right) \quad (3.10)$$

and

$$p = \mathbb{N}\left(\frac{d_{min}}{\sqrt{2(\sigma_F^2 + \sigma_N^2)}}\right) \quad (3.11)$$

This model only requires general knowledge about the domain, in terms of the  $d_{min}$ , and  $k_{max}$  variables, and does not require the GA implementor to calculate the population size required for each partition in the domain.

#### 3.5.2 General Population Sizing Model Approximation

This section derives an accurate approximation of the population sizing model that is easy to calculate, conservative, and allows for better insight into the effect of different variables on the population sizing requirements. This approximation is partly based on an approximation given by Harik, Cantú-Paz, Goldberg, and Miller (1997).

First,  $p$  can be approximated using the first two terms of the power series expansion for the normal distribution (Abramovitz & Stegun, 1972):

$$p = \frac{1}{2} + \frac{x}{\sqrt{2\pi}} \quad (3.12)$$

where  $x = d_{min} / \sqrt{2(\sigma_F^2 + \sigma_N^2)}$ .

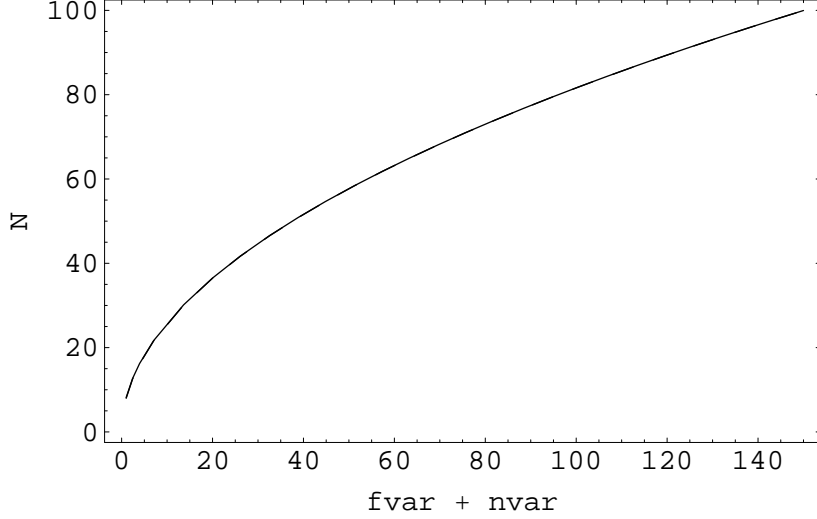


Figure 3.9: Population Size Approximation

Substituting the approximation for  $p$  into equation 3.10 we get:

$$N = 2^{k_{max}} \ln(\psi) / \ln\left(\frac{1 - \frac{x\sqrt{2}}{\sqrt{\pi}}}{1 + \frac{x\sqrt{2}}{\sqrt{\pi}}}\right) \quad (3.13)$$

Since  $x$  is a small number,  $\ln(1 \pm \frac{x\sqrt{2}}{\sqrt{\pi}})$  can be approximated by  $\pm \frac{x\sqrt{2}}{\sqrt{\pi}}$ . Substituting these approximations and the value of  $x$ , we get the approximation population size  $N'$ :

$$N' = -\frac{2^{k_{max}-1} \ln(\psi) \sqrt{\pi}}{d_{min}} \sqrt{\sigma_F^2 + \sigma_N^2} \quad (3.14)$$

To get an idea of the accuracy of this population sizing approximation, the approximation (equation 3.14 represented by the solid line) is plotted against the original population sizing model (equations 3.10 and 3.11 represented by the dashed line) for the Onemax domain in Figure 3.9. The approximation is very accurate, so the two lines appear as a single line in the below plot. The minimum building block fitness signal,  $d_{min}$  is set to one, the chromosome length to  $l = 120$ , with a corresponding initial fitness variance  $\sigma_F^2 = 30$ . The failure rate is set to  $\psi = 0.01$ . The model is plotted over the range of the total variance  $\sigma_F^2 + \sigma_N^2$  going from 0 to 145.

While this is fairly accurate for the Onemax domain, the accuracy for other domains can be verified by the use of dimensional analysis to derive a correctional coefficient for the approximate population sizing model. The ratio  $N/N'$  determines the correctional coefficient for a given set of population sizing parameters. Defining the signal to noise ratio as  $r = \frac{d_{min}}{\sqrt{2(\sigma_F^2 + \sigma_N^2)}}$  gives:

$$\frac{N}{N'} = \frac{-r2\sqrt{2}}{\sqrt{\pi} \ln((1-p)/p)} \quad (3.15)$$

Note that  $p$ , defined by equation 3.11, can be rewritten as a function of  $r$ . Thus the ratio

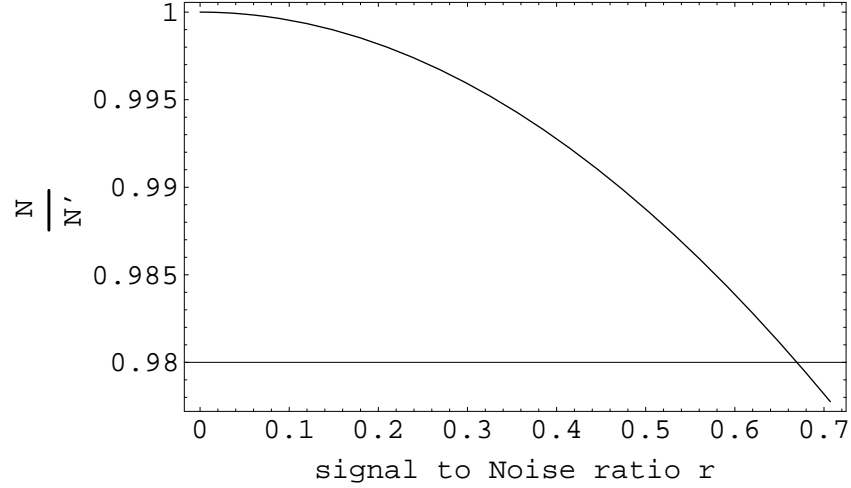


Figure 3.10: Population Size Approximation Correctional Coefficient

$N/N'$  is totally a function of  $r$ . The lower bound of  $r$  is zero, and the upper bound for  $r$  can be calculated by the conservative assumption that  $d_{min} \leq \sigma_F$ , resulting in an upper bound for  $r$  of  $1/\sqrt{2}$ . Figure 3.10 shows the plot of  $N/N'$  (the correctional coefficient) for  $r$  ranging from zero to  $1/\sqrt{2}$ . The correctional coefficient ranges from approximately 1.00 to 0.98. This demonstrates that the approximation given by equation 3.14 should be accurate within two percent for all population sizing approximations for any domain. Indeed, since the signal to noise ratio  $r$  is significantly less than  $1/\sqrt{2}$  for most domains, the population sizing approximation generally does much better than two percent inaccuracy.

This approximation serves two purposes. First, it is a good rule of thumb for a GA implementor to estimate the population sizing requirements of a domain. It eliminates the complexity of calculating the CDF for  $p_i$  (equation 3.11). Next, and perhaps more importantly, the simpler approximation allows for the effects of different variables on the population sizing requirements to be better analyzed, as will be explored in the next chapter.

### 3.6 Conclusions

This chapter introduced an important new population sizing model, the random walk population sizing model. The random walk population sizing model incorporates theory from previous population sizing models based on building block supply theory and building block decision theory. The random walk population sizing model was shown to accurately predict the performance as a function of population size for uniformly scaled domains. Equivalently, the random walk population sizing model can be used to determine the population sizing requirements of a GA for non-uniformly scaled domains when sized using the least significant partition (the partition which has the greatest population sizing requirements). Alternatively, if the individual partitions in a domain are not known, a conservative approximation of the random walk population sizing model was derived, and shown to be accurate. This approximation requires minimal domain knowledge (just the maximal building block length and the minimum fitness signal for the domain), and should be very useful for GA implementors as a rule-of-thumb population sizing model. The population sizing approximation also clearly

revealed the relationship between various parameters in the random walk population sizing model and their corresponding effect on population sizing requirements. Lastly, the random walk population size was extended to handle varying noise and selection intensity levels, and the accuracy of the extended model experimentally verified. This is the first time that any population sizing model has been able to predict the effect of selection intensity on the population sizing requirements.



## Chapter 4

# Sampling

### 4.1 Introduction

Genetic algorithms (GAs) are increasingly being used to solve real-world problems (Davis, 1991; Winter, Périaux, Galán, & Cuesta, 1995). Implementing a GA for a real-world environment is difficult for several reasons. First, the GA must identify a solution within a bounded amount of computational time. For many domains, this entails that the GA will not be allowed to run until convergence, but instead will be halted after a fixed amount of time and the best individual in the ending population selected as the final solution. Second, noise is often present in the fitness functions. For many domains, a genetic algorithm practitioner is faced with a choice of several fitness functions, each with different noise, speed, and cost characteristics. This research focuses on determining which fitness function should be chosen so as to maximize performance of the GA within a given amount of computational time. A model is derived that predicts the resulting performance of a GA within a given time bound for a fitness function with known speed and accuracy characteristics. This allows for performance comparisons of different fitness functions for a given domain.

An important subclass of noisy fitness functions are *sampling fitness functions*, which utilize sampling to reduce the noise from fitness evaluations. An entire range of possible fitness functions can be generated by simply changing the sample size of a sampling fitness function, with each sample size producing a fitness function with unique speed and accuracy characteristics. It is difficult for a GA implementor to determine which sample size will maximize GA performance *a priori*, as each different sample size produces a fitness function that is Pareto-optimal in terms of speed and accuracy (Hans, 1988; Srinivas & Deb, 1995; Goldberg, 1989a).

This chapter develops lower and upper bounds for the optimal sample size, which are used to bound the range of sample sizes considered for sampling fitness functions. A pruning method is developed based on the effect that the ending generation is truncated in order to stay within the time-bound allotted to the GA. This pruning method can be used to remove a large portion of the sample sizes between the lower and upper bounds from consideration. The sample sizes remaining between the bounds are then searched using a predictive performance model based upon the noise resulting from the different sample sizes. In this manner, the sample size that produces the highest GA performance within the fixed computational bound can be identified. Through this procedure, the optimal sample size can be quickly and reliably identified without the need for costly trial and error experimentation by the GA implementor.

The purpose of this study is to develop a predictive model that determines the (optimal) sample size that maximizes GA performance within a fixed amount of time for the current domain. This will alleviate the time and effort of determining optimal sample sizes through experimental trial and error.

#### 4.1.1 Overview

The remainder of this section discusses related research covering genetic algorithms and sampling, as well as summarizes previous research relevant to this analysis of sampling and GAs. Section 2 applies the basic GA convergence models for noisy environments in order to determine the optimal sample size. In section 3, heuristics for limiting the search for the optimal sample size are derived. Equations for both the lower and upper bound of the optimal size are derived, and are used to limit the search for the optimal sample size. A pruning method for reducing the sample size search space is also developed. Section 4 presents experimental results that are used to verify the models developed in the previous sections. Lastly, section 5 presents some general conclusions drawn from this research.

#### 4.1.2 Related Research

The impact of noise on bounded GA performance has not been extensively studied. An early study by Grefenstette and Fitzpatrick (1985) explored the use of sampling fitness functions. It empirically showed that smaller sample sizes, which cause noisier sampling fitness function evaluations, could actually increase the performance of a GA. This study used a constant population size for the GA, so the population size was independent of the noise level. A later, more rigorous study (Fitzpatrick & Grefenstette, 1988) developed a computational model of GA time requirements, and utilized the model to determine the performance tradeoffs of different ratios of the GA processing time per individual per generation (excluding fitness evaluation time) and time per fitness evaluation (sample time). This study again empirically showed clear performance gains for a GA using an optimal sample size. In their study, the total amount of processing time for each generation was held constant, and the sample and population sizes were adjusted accordingly.

There has been a limited amount of research into sampling strategies for GAs. In the study by Aizawa and Wah (1994b), the process of how to best distribute a fixed amount of samples among a fixed size population is studied. In Aizawa and Wah (1994a), they investigate the *duration-sizing problem*, which deals with determining when a current generation has used a sufficient number of samples so as to adequately evaluate each individual. The sample size can vary for each generation, with each member of the population being sampled equally. The rationale behind this approach is that the population variance is initially large, and slowly decreases until it is zero at convergence. As less fitness accuracy is needed for earlier generations, the number of samples per individual can be decreased. In that study, the population size is held constant, and the emphasis is on determining adequate sample sizes and the corresponding amount of processing time to spend per generation.

More recent research has improved our understanding of how noise affects GA performance. In the paper by Goldberg, Deb, and Clark (1992), it was shown that for GAs to avoid premature convergence in noisy domains, the population size must be increased. A population sizing model was derived based upon schema analysis that predicts that population sizes for a GA operating in an environment with a given noise level. A newer, more accurate population sizing model was presented by Harik, Cantú-Paz, Goldberg, and Miller (1997), and expanded upon in the previous chapter, in which the population size was also shown to be dependent on the noise level. This paper uses this population sizing equation to determine the correct population size for a given level of fitness function noise. A weakness with the sampling studies above (Grefenstette & Fitzpatrick, 1985; Fitzpatrick & Grefenstette, 1988; Aizawa & Wah,

1994b; Aizawa & Wah, 1994a) is that they did not vary the population size based on the different noise levels resulting from different sampling sizes. Instead they either used a fixed population size, or sized the population so that the total time spent per generation remained constant for different sample sizes. This can result in inadequately sized populations, as the population size does not depend on the noise variance resulting from the different sample sizes. If the population is sized too small, premature convergence to a suboptimal solution is more likely to occur; if it is too big, valuable time is wasted processing redundant individuals.

Recent research has investigated how fitness function noise affects the performance of GAs. In Miller and Goldberg (1996b) and Miller and Goldberg (1996a), predictive models of GA performance in noisy environments were developed for a variety of different selection schemes. These predictive models were shown to accurately predict GA performance in noisy environments for selection methods that could be modeled using selection intensity (Mühlenbein & Schlierkamp-Voosen, 1993). These selection methods include, but are not limited to, tournament selection, linear ranking,  $(\mu, \lambda)$  selection, and proportionate selection. A determination of the lower bound for the optimal sample size, based on the population sizing model developed by Goldberg, Deb, and Clark (1992), was presented by Miller and Goldberg (1996c).

### 4.1.3 Background Research

The previous chapters define how different noise levels affect the convergence rate and population sizing requirements of the GA. Figure 4.1 summarizes the research framework of this dissertation. Chapter 2 presented the effect of noise on convergence, and Chapter 3 explored how population sizing requirements change with varying noise and selection intensity levels. The major remaining piece is to determine the computational requirements resulting from different samples sizes so that the performance can be optimized with respect to a computational time bound.

This section summarizes the important equations upon which the sampling research, presented in the next section, is based. The sampling fitness function is defined, and the role of the central limit theorem in reducing the noise variance is explained. Next, the random walk population sizing equation that takes into account the noise level of the fitness function, developed in the previous chapter, is presented. The effect of noise on the convergence of a GA is then shown (derived in Chapter 2), along with the corresponding convergence model for the noisy Onemax domain, which is later used to verify the accuracy of the developed sampling models.

#### 4.1.3.1 Sampling Fitness Functions

A sampling fitness function seeks to reduce the amount of noise by taking the mean of multiple noisy fitness evaluations of an individual. Evaluation of an individual  $i$  by a sampling fitness function using a sample size of  $n$  can be expressed as:

$$f_{i,n}^* = \frac{1}{n} \sum_{j=1}^n f'_{i,j}, \quad (4.1)$$

where  $f'_{i,j}$  is the  $j$ th noisy fitness evaluation of individual  $i$ . According to the Central Limit Theorem, the sampling fitness  $f_{i,n}^*$  will have a distribution of  $N(f_i, \sigma_N^2/n)$ , as it is distributed as the mean of  $n$  samples from the normal distribution  $f'_{i,j}$ , which is distributed  $N(f_i, \sigma_N^2)$ .

The speed and accuracy of the sampling fitness function therefore directly depends on the

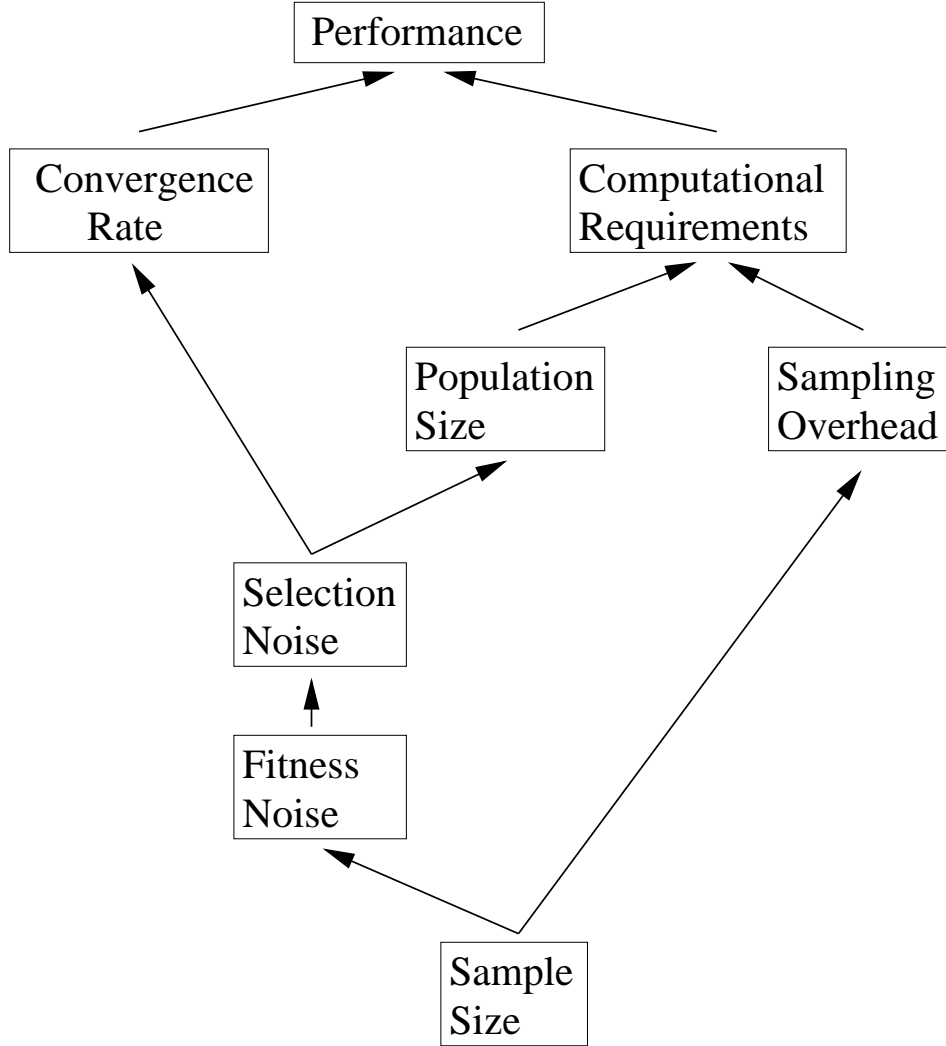


Figure 4.1: The Effect of Sample Size on Performance

sample size. For a noisy fitness evaluation  $f'$  with noise variance  $\sigma_N^2$  and computational cost  $\beta$ , the sampling fitness function  $f^*$  has a noise variance and computational cost of  $\sigma_N^2/n$  and  $\beta n$ , respectively. Thus by simply changing the sample size, an entire range of fitness functions can be generated, each of which is Pareto-optimal (Hans, 1988; Srinivas & Deb, 1995; Goldberg, 1989a) in terms of speed and accuracy. Thus any of the sampling sizes could result in a GA performing optimally within a given time-bound, as none of the sample sizes are absolutely better than any of the other sample sizes in terms of speed and accuracy of the resulting sampling fitness function.

#### 4.1.3.2 Population Sizing and Noise

A useful population sizing approximation for the random walk population sizing model (Harik, Cantú-Paz, Goldberg, & Miller, 1997) is:

$$N' = -\frac{2^{k_{max}-1} \ln(\psi) \sqrt{\pi}}{d_{min}} \sqrt{\sigma_F^2 + \sigma_N^2} \quad (4.2)$$

This is a conservative population sizing model for determining the population sizing requirements of a GA for various noise levels. For domains utilizing sampling fitness functions, the noise variance  $\sigma_N^2$  can be replaced with  $\sigma_N^2/n$ .

#### 4.1.3.3 Convergence and Noise

For noisy domains, Miller and Goldberg (1996a) extended the selection intensity convergence model of Mühlenbein and Schlierkamp-Voosen (1993) to handle fitness function noise:

$$\mu_{F,t+1} = \mu_{F,t} + \frac{\sigma_{F,t}^2}{\sqrt{\sigma_{F,t}^2 + \sigma_N^2}} I, \quad (4.3)$$

where  $\sigma_N^2$  is the variance of the unbiased normally distributed fitness function noise. This equation can be adapted for various domains to predict the performance of the GA over time. This convergence model assumes that the population is sized large enough to avoid premature convergence (it does not take into account population sizing effects resulting from very small populations). Again, for domains where sampling is used, the noise variance should be adjusted accordingly for the sample size ( $\sigma_N^2 \rightarrow \sigma_N^2/n$ ).

#### 4.1.3.4 Onemax Domain

In Miller and Goldberg (1996a), a performance model for the Onemax domain with noisy fitness functions was derived:

$$\frac{dp}{dt} = I \frac{p(1-p)}{\sqrt{lp(1-p) + \sigma_N^2}}, \quad (4.4)$$

where  $p$  is the proportion of correct alleles,  $l$  is the chromosome length, and  $\sigma_N^2$  is the fitness function noise variance. Equation 4.4, taken together with the assumption that half of the alleles are correct in the initial population ( $p_0 = 0.5$ ), can be numerically solved to predict the convergence of the Onemax population with noisy fitness functions. This model is used in this paper to predict the performance of a GA utilizing a noisy sampling fitness function for different sample sizes, by simply setting the noise variance of the model to the noise variance of the sampling fitness function for the sample size of interest ( $\sigma_N^2 \rightarrow \sigma_N^2/n$ ).

## 4.2 Optimal Sample Size Determination

### 4.2.1 GA Computational Model

The primary goal of this chapter is to determine the optimal sample size that maximizes GA performance within a fixed amount of time. To do so, the computational requirements of the GA over time must be modeled. Fitzpatrick and Grefenstette (1988) developed a model for the total time  $T$  required by a GA:

$$T = (\alpha + \beta n)GN, \quad (4.5)$$

where  $G$  is the total number of generations,  $N$  is the population size, and  $n$  is the sample size of the sampling fitness function. The variable  $\alpha$  represents the fixed amount of GA overhead time per individual per generation, which includes the costs of selection, recombination and mutation, but excludes the cost of fitness evaluations. The variable  $\beta$  represents the cost of a single fitness evaluation (a sample), so  $\beta n$  is simply the total computational cost of a sampling fitness function with a sample size of  $n$ . This model does ignore the cost of generating the initial population, but this is reasonable as the runtime costs dominate.

This research effort is interested in determining the ending generation of a time-bounded GA for different sample sizes. Each different sample size results in a different sampling fitness function noise variance, as the noise variance resulting from a sample size of  $n$  is  $\sigma_N^2/n$ . This in turn changes the population size  $N$  required by equation 3.6. The different sample sizes thus not only affect the  $\beta n$  component of equation 4.5, but also affect the population size. If the computational bound  $T$  is given, along with the relevant domain dependent variables  $\alpha, \beta, \sigma_N^2, \sigma_F^2$ , and  $\psi$ , the ending generation as a function of the sample size  $g^*(n)$  can be calculated as follows:

$$\begin{aligned} g^*(n) &= \frac{T}{(\alpha + \beta n)N} \\ &= \frac{T \ln(\frac{1-p}{p})}{(\alpha + \beta n)2^k \ln(\psi)} \end{aligned} \quad (4.6)$$

where it is understood that  $p$  and  $k$  are for the partition with the largest population sizing requirements. A simplified, and fairly accurate, approach is to use the general population sizing approximation model to calculate  $N$ , given by equation 3.14, which has the benefit of making the population sizing parameter effects easier to analyze:

$$\begin{aligned} g^*(n) &= \frac{T}{(\alpha + \beta n)N} \\ &= \frac{T d_{min}}{(\alpha + \beta n) 2^{(k_{max}-1)} \ln(\psi) \sqrt{\pi} \sqrt{\sigma_F^2 + \sigma_N^2}} \\ &= \frac{T}{(\alpha + \beta n) ? \sqrt{\sigma_F^2 + \sigma_N^2/n}} \end{aligned} \quad (4.7)$$

where the last approximation uses a  $?$  term to group all of the domain dependent population sizing variables, and sizes the noise variance according to the sample size  $n$ . The  $?$  term is introduced so that this analysis can be applied to any population sizing model that can be expressed as  $N = ? \sqrt{\sigma_F^2 + \sigma_N^2/n}$ . However,  $?$  in this paper is calculated as indicated by equation 3.14.

#### 4.2.2 Performance as a Function of Sample Size

The ending generation is important, as the performance of the GA can be predicted if both the convergence rate and ending generation are known. Taken together, the convergence rate model, equation 4.3, and the ending generation, equation 4.7, can be used to estimate the ending performance of the GA. The predicted performance for different sample sizes can thus be calculated to identify the optimal sample size for a given computational bound  $T$ .

For domains where a convergence rate model is known, a brute force method of identifying the optimal sample size is to calculate the performance of the GA for all sample sizes greater than one. The performance curve as a function of the sample size is unimodal for reasons discussed in detail below. The main question is whether the optimal sample size is less than one (Figure 4.2a) or greater than one (Figure 4.2b). In either case, the GA implementor can identify the optimum sample size through brute force, simply by iterating through all sample sizes (starting from  $n = 1$ ) until the performance clearly starts to decrease for increasing  $n$ . The optimal sample size is simply the sample size that resulted in the highest performance. One caveat of this approach is that since in practice the ending generation is truncated, there can be small “dips” in the performance due to the truncation effect. However, normally these truncation effects are readily apparent, and the GA practitioner can easily identify the larger trend of the GA performance as a function of increasing sample size.

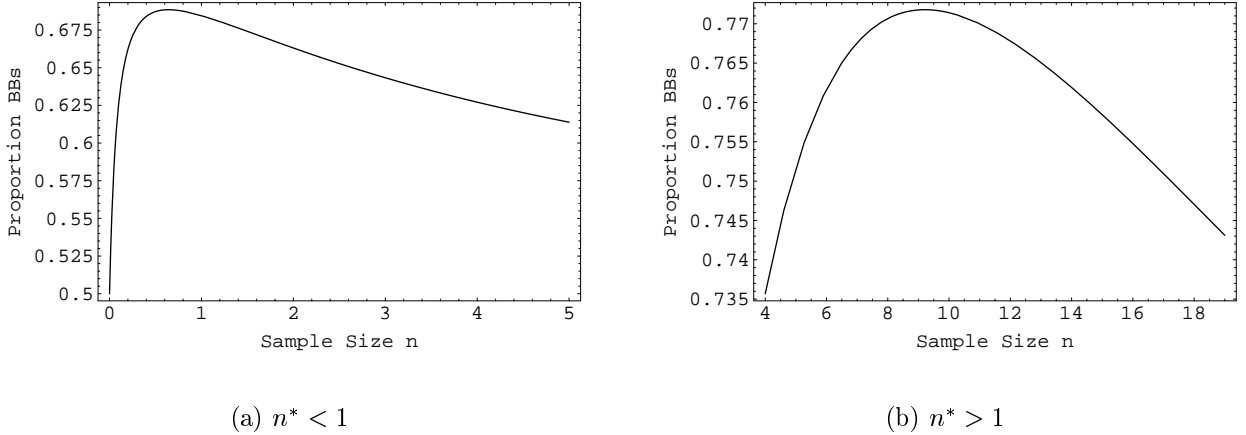


Figure 4.2: Performance as a Function of Sample Size

### 4.2.3 Performance Curve Shape

The noise resulting from a given sample size determines the convergence rate of the GA, as given by equation 4.3. The performance is also affected by how many generations can be run within the time bound, where the ending generation is calculated using equation 4.7. Of particular interest is determining the shape of the performance curve as a function of the sample size. This section explains why the performance curve as a function of sample size must be unimodal.

The shape of the continuous performance curve for a time-bound GA over different sample sizes is unimodal, and allows for the optimal sample size to be identified. To see why it is unimodal, an understanding must be developed of how the sample size affects the critical factors of performance: computational overhead, population sizing, and convergence rate. The performance penalty for increasing the sample size comes from the increased sampling overhead, given by the  $\beta n$  term in equation 4.7. However, this penalty is offset by the reduced noise variance of the sampling fitness function as a result of the increased sample size. The population sizing requirements are reduced (see equation 4.7), and the convergence rate increases (see equation 4.3). However, both of these performance advantages are based on the sampling noise variance  $\sigma_N^2/n$ , which has diminishing returns as the sample size increases.

This entails that the performance penalty from increased computational overhead must eventually overtake the (diminishing) performance advantages from the reduced noise variance. This mandates that the performance curve for a time-bound GA over different sample sizes must be unimodal, with a clearly defined optimal sample size.

The next section will try to improve upon this brute force method of determining the optimal sample size, described above, through the use of various heuristics. A lower and upper bound of the optimal sample size are derived, and used to reduce the size of the sample search space. A pruning method is further developed that can eliminate some sample sizes between the lower and upper bounds.

### 4.3 Sample Size Determination Heuristics

Given the above GA computational model, the optimal sample size that maximizes the performance within in a given time-bound can now be determined. However, there are ways of bounding the search for the optimal sample size that avoid costly enumeration over each different sample size as required by the brute force method described above. This section derives both a lower and upper bound for the optimal sample size. These sample size bounds greatly restrict the search for the optimal sample size; see Figure 4.3. For this figure, the lower and upper bounds are 5.80 and 23.57 respectively, which greatly reduce the search space. In general, the sample sizes contained within the bounds will all have similar performance. This causes the lower and upper bounds to be very useful in identifying an optimal, or near-optimal, sample size. There is another technique, Pareto pruning, that can be used to eliminate many of the points between the lower and upper bounds from consideration; however, its effect is minor compared to that of correctly bounding the optimal sample size in the first place.

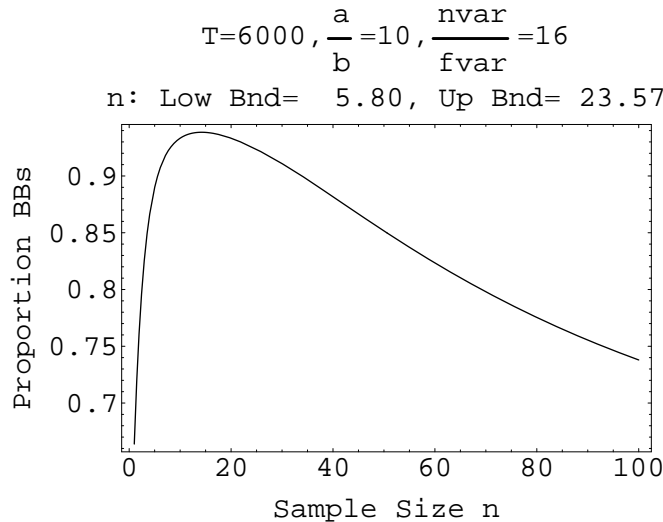


Figure 4.3: Effectiveness of Bounding Sample Size

This section will present the derivation of different sample size determination heuristics. First, a domain independent lower bound for the sample size is determined, so that the search for an optimal sample size can begin at a more appropriate sample size than one, the starting sample size for the brute force method. Next, a domain dependent upper bound for the



optimal sample size is estimated. Lastly, a technique called Pareto pruning is introduced that eliminates many sample sizes from consideration. Taken together, these mechanisms can greatly restrict the search for the optimal sample size.

### 4.3.1 Domain Independent Lower Bound

To determine the lower bound for the optimal sample size, a critical assumption is made: the improvement in the convergence rate resulting from increasing the sample size is ignored. As this entails that GA runs for all sample sizes will be modeled at the same convergence rate, the optimal sample size is the sample size that maximizes the ending generation  $g^*$ . Thus the lower bound is simply the sample size that maximizes the ending generation. In Miller and Goldberg (1996c) this is calculated for the population sizing model developed by Goldberg, Deb, and Clark (1992); this section performs a similar analysis for the random walk population sizing model. This lower bound is considered domain independent in that the GA convergence characteristics for the domain do not need to be known.

#### 4.3.1.1 Calculating the Lower Bound

To calculate the lower bound, the ending generation  $g^*$  is represented as a function of the sample size  $n$ . The derivative  $\frac{dg^*}{dn}$  is calculated, set to 0, and solved for  $n$ , with the resulting  $n$  being the lower bound. However, the simpler approximation for the population sizing equation is used (equation 4.7) so as to be able to obtain a meaningful (and simpler) derivative for the ending generation.

To determine the sample size that maximizes the ending generation, the derivative of the ending generation (equation 4.7) is calculated with respect to  $n$ :

$$\frac{dg^*}{dn} = \frac{T}{(\alpha + \beta n)\sqrt{\sigma_F^2 + \sigma_N^2/n}} \left( -\frac{\beta}{(\alpha + \beta n)} + \frac{\sigma_N^2}{2n^2(\sigma_F^2 + \sigma_N^2/n)} \right) \quad (4.8)$$

Setting  $\frac{dg^*}{dn} = 0$  and solving for  $n$  yields:

$$n_{lb} = \frac{\sqrt{8\alpha\beta\sigma_F^2\sigma_N^2 + \beta^2\sigma_N^4} - \beta\sigma_N^2}{4\beta\sigma_F^2} \quad (4.9)$$

To get a feel for the effect of the variables in this equation, equation 4.9 can be rewritten as:

$$n_{lb} = \frac{\sqrt{a^2 + b^2} - b}{c} \quad (4.10)$$

where  $a = \sqrt{8\alpha\beta\sigma_F^2\sigma_N^2}$ ,  $b = \beta\sigma_N^2$ , and  $c = 4\beta\sigma_F^2$ . The ratio between the  $a^2$  and  $b^2$  terms in the numerator is important for determining the range of the lower bound. Defining the ratio  $r = \frac{a^2}{b^2} = \frac{8\alpha\sigma_F^2}{\beta\sigma_N^2}$ , approximations can be made for  $r \ll 1$ ,  $r = 1$ , and  $r \gg 1$ . A physical interpretation of  $r$  is that it is essentially the ratio of the GA operating parameters ( $\alpha$  and  $\sigma_F^2$ ), to the “bad” noise parameters ( $\beta$  and  $\sigma_N^2$ ). For  $r \ll 1$ , equation 4.10 reduces to  $\frac{b-b}{c} = 0$ . For  $r = 1$ , equation 4.10 reduces to  $\frac{\sqrt{2a^2 - a}}{c} = \frac{\sqrt{2}-1}{\sqrt{2}} \sqrt{\frac{\alpha}{\beta}} \sqrt{\frac{\sigma_N^2}{\sigma_F^2}}$ . For  $r \gg 1$ , and further assuming that  $\sqrt{r} \gg 1$  (that is,  $a \gg b$ ), gives the lower bound equal to  $\frac{a}{c} = \frac{1}{\sqrt{2}} \sqrt{\frac{\alpha}{\beta}} \sqrt{\frac{\sigma_N^2}{\sigma_F^2}}$ . An interesting result of this last approximation is that the lower bound must always be smaller than  $\frac{1}{\sqrt{2}} \sqrt{\frac{\alpha}{\beta}} \sqrt{\frac{\sigma_N^2}{\sigma_F^2}}$ .

These approximations are summarized in Table 4.1.

$r = \frac{8\alpha\sigma_F^2}{\beta\sigma_N^2}$	Approximation	Result
$r \ll 1$	$\frac{b-b}{c}$	0
$r = 1$	$\frac{\sqrt{2b^2-b}}{c}$	$\frac{\sqrt{2}-1}{\sqrt{2}} \sqrt{\frac{\alpha}{\beta}} \sqrt{\frac{\sigma_N^2}{\sigma_F^2}}$
$r \gg 1$	$\frac{a}{c}$	$\frac{1}{\sqrt{2}} \sqrt{\frac{\alpha}{\beta}} \sqrt{\frac{\sigma_N^2}{\sigma_F^2}}$

Table 4.1: Lower Bound on Optimal Sample Size

Another way to view this is that the lower bound on the optimal sample size can be written as:

$$n_{lb} = c(r) \sqrt{\frac{\alpha}{\beta}} \sqrt{\frac{\sigma_N^2}{\sigma_F^2}} \quad (4.11)$$

where  $c(r)$  is a coefficient obtained by dividing  $n_{lb}$  by  $\sqrt{\frac{\alpha}{\beta}} \sqrt{\frac{\sigma_N^2}{\sigma_F^2}}$  for each value of  $r$ . Figure 4.4 shows the coefficient range for different values of  $r$ . As indicated by the table above, the coefficient starts out at zero for low values of  $r$ , and approaches  $\frac{1}{\sqrt{2}}$  as  $r$  approaches infinity.

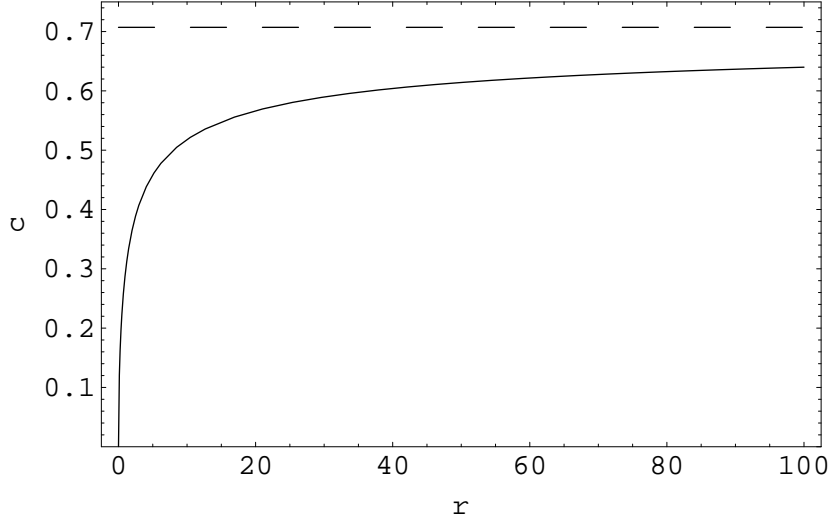


Figure 4.4: Coefficient of Optimal Sample Size Lower Bound

#### 4.3.2 Domain Dependent Upper Bound

Obtaining an upper bound on the sample size is more difficult, as the convergence model for the domain of interest must be known. This analysis can be extremely difficult due to the complexity of the convergence rate model. An approach taken elsewhere (Miller & Goldberg, 1996c) is to use a simpler model to approximate the convergence model. The approximate convergence model must have the property that it provably converges faster than the exact convergence model at the same noise level (sample size). The approximate convergence model is then used in conjunction with equation 4.7 to determine an optimal sample size that maximizes the performance of the computationally bound GA. As the performance advantage of

the convergence is overstated, the optimal sample size determined in this manner must be an upper bound of the optimal sample size ( $n^* < n_{ub}$ ).

The Onemax domain is used to demonstrate the methodology for obtaining an upper bound on the optimal sample size. An approximating model must first be derived for the Onemax exact convergence model, equation 4.4. The approximation must be faster than the exact convergence model, have the convergence rate depend upon the noise variance, and have the property that it can be symbolically solved. For the Onemax domain, this can be achieved by ignoring the population variance in the denominator, the  $lp(1-p)$  term, of the Onemax exact convergence model (equation 4.4). The approximate convergence model for equation 4.4 then becomes:

$$\frac{dp}{dt} = I \frac{p(1-p)}{\sqrt{\sigma_N^2/n}} \quad (4.12)$$

Equation 4.12 can be solved to determine the performance  $p$  as a function of time  $t$ :

$$\begin{aligned} \int \frac{1}{-p^2 + p} dp &= \frac{I\sqrt{n}}{\sigma_N} \int dt, \\ \ln\left(\frac{p}{1-p}\right) &= \frac{I\sqrt{n}}{\sigma_N} t + c, \\ \frac{p}{1-p} &= e^{\frac{I\sqrt{n}}{\sigma_N} t + c} \end{aligned}$$

Substituting  $z$  for the right-hand side of the above equation, and solving for  $p$ :

$$\begin{aligned} \frac{p}{1-p} &= z, \\ p(1+z) &= z, \\ p &= \frac{z}{1+z} \end{aligned}$$

Finally, replacing  $z$  with the original term yields:

$$p(t) = \frac{e^{\frac{I\sqrt{n}}{\sigma_N} t + c}}{1 + e^{\frac{I\sqrt{n}}{\sigma_N} t + c}}$$

Given our assumption that half of the alleles are correct in the initial generation ( $p(0) = 0.5$ ) and solving for  $c$  yields  $c = 0$ . Thus our convergence approximation for the Onemax domain is:

$$p_{approx}(t) = \frac{e^{\left(\frac{I\sqrt{n}}{\sigma_N} t\right)}}{1 + e^{\left(\frac{I\sqrt{n}}{\sigma_N} t\right)}}. \quad (4.13)$$

To determine the optimal sample size for a computationally bound GA that behaves according to equation 4.13, equations 4.7 and 4.13 are used. The ending generation for a particular sample size is calculated using equation 4.7, and equation 4.13 is then used to calculate the performance at time  $t$  equal to the ending generation.

$$p_{ub}(n) = p_{approx}(g^*(n)) \quad (4.14)$$

The sample size that results in the highest performance is optimal (and is also an upper bound for the optimal sample size for the exact convergence model). However, the optimal sample size for the approximate convergence model can be identified in a more efficient manner by solving for  $\frac{dp_{ub}}{dn}$ , setting  $\frac{dp_{ub}}{dn} = 0$ , and solving for  $n$  (using Mathematica to do the hairy differentiation):

$$n_{ub} = \frac{\sqrt{8\alpha\beta\sigma_F^2\sigma_N^2 + \alpha^2\sigma_F^4} + \alpha\sigma_F^2}{2\beta\sigma_F^2} \quad (4.15)$$

This gives the optimal sample size for the approximate convergence model of GA performance, and is thus also the upper bound of the optimal sample size for the exact convergence model. The relationship between the Onemax upper bound and lower bound can be derived through simple algebraic manipulation:

$$n_{ub} \geq 2 n_{lb} + \frac{\alpha}{2\beta}$$

where  $n_{lb}$  is the maximum possible value of lower bound ( $n_{lb} = \frac{1}{\sqrt{2}} \sqrt{\frac{\alpha}{\beta}} \sqrt{\frac{\sigma_N^2}{\sigma_F^2}}$ ). This upper bound model applies only to the Onemax domain, but the methodology used can be applied to develop an upper bound of the optimal sample size for other domains.

### 4.3.3 Domain Independent Pareto Pruning

The lower and upper bounds for the optimal sample size limit the range of sample sizes in which the optimal sample size may be located. This section explores a weaker method that may be used in many domains to eliminate some of the sample sizes within the bounded range from consideration. It is based on the fact that GAs are not typically able to run for a fractional number of generation; instead, the ending generation is truncated so as to not exceed the time bound  $T$ . In practice, truncating the ending generation causes a minor “staircase” fluctuation in the overall performance curve. This fluctuation results in the performance curve not being strictly convex, but the effect is minor enough that the performance from the non-truncated performance curve can be used to guide the search over the truncated performance curve.

Pareto pruning exploits this truncation effect by realizing that only the sample sizes that are Pareto-optimal over two performance criteria, the truncated ending generation and the noise level, need to be considered. The sample sizes that are dominated by another sample size for both performance criteria can be pruned. In other words, only the largest sample size from each set of samples sizes that have the same *truncated* ending generation value needs to be considered. Of the sample sizes that result in the same truncated ending generation, the sample size that causes the GA to converge the fastest will have the best performance. The largest sample size of the set of sample sizes that evaluates to the same truncated ending generation value will have the fastest convergence rate, as it will have the lowest noise variance due to the central limit theorem ( $\sigma_N^2/n$ ). Thus only the largest sample size of all sample sizes that produce the same truncated ending generation needs to be considered. The set of all such largest sample sizes for each distinct truncated ending generation is known as the Pareto-optimal set.

Figure 4.5 shows the continuous ending generation versus the truncated ending generation for a range of sample sizes. The truncated ending generation plot resembles “stairs.” Each stair is a range of sample sizes whose ending generation is truncated to the same value. As the

convergence rate increases with decreasing noise, and noise decreases with increasing sample sizes, the greatest (rightmost) sample size in each stair will have the fastest convergence rate. In this manner, only the greatest sample size of the set of sample sizes that produce the same truncated ending generation needs to be considered, as it provably will have a faster convergence rate and be run for the same number of generations. The Pareto-optimal set for Figure 4.5 would be  $\{1, 2, 5, 8, 11\}$ . This phenomena is used to remove all “dominated” sample sizes from the sample size search space.

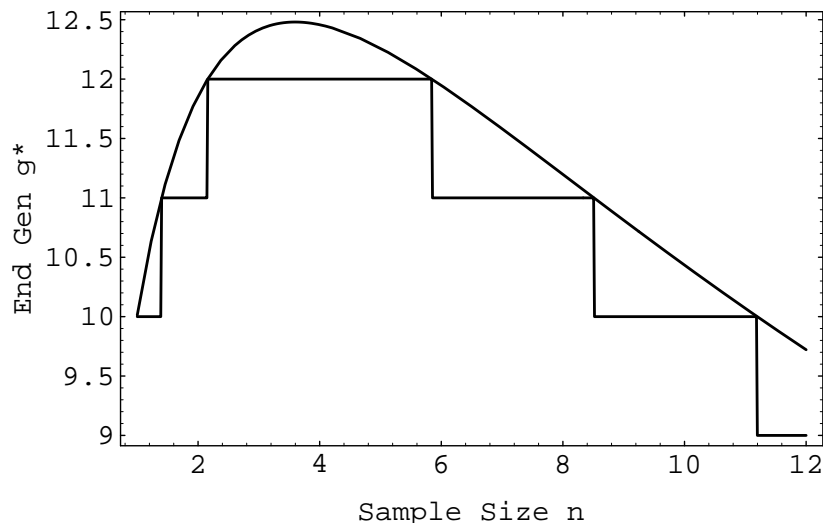


Figure 4.5: Continuous and Discrete Ending Generations

Figure 4.6 compares the predicted performance in the Onemax domain of an “imaginary” GA that can be run for a fractional number of generations, versus a “real” GA whose ending generation must be truncated. The top smooth curve is the continuous ending generation, the solid stair-steps is the truncated ending generation, and the dots are the experimental average of 50 GA runs with truncated ending generations. Truncation of the ending generation produces the “stairs” or scalloped performance shape. When compared with the graph of the truncated ending generations give above, it is easy to see that the rightmost sample size of each ending generation stair results in the best performance of the sample sizes contained in the stair. This validates the Pareto pruning technique.

This pruning technique can be used to vastly reduce the search space of sample sizes, even when the convergence model for a given domain is not known. It can be considered domain independent as only the truncated ending generation values for a domain need to be determined, and not the more difficult question of quantifying the effect of increasing sample sizes on the convergence rate in the domain of interest (domain convergence). This does assume that the domain convergence model must have strictly non-decreasing convergence rates for decreasing noise levels (GA performance increases as the noise level decreases). However, this assumption holds for all domains currently known to the author, and thus considering only the sample sizes at ending generation stair-edges is a valid pruning technique for most domains. Note again that this technique of pruning the search space holds because the GA is not able to run for a fractional number of generations.

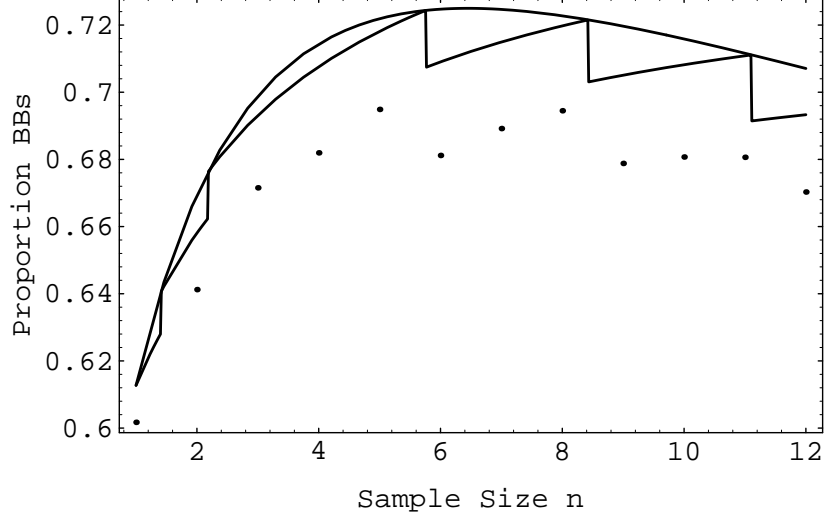


Figure 4.6: Sampling Performance of Onemax Domain

#### 4.3.4 Sampling Invariant Parameters

Now that the lower and upper bound on the sample size for the Onemax domain have been derived, it is interesting to note that they are independent of several variables. First, the lower bound is a function of  $\alpha, \beta, \sigma_F^2$ , and  $\sigma_N^2$ , and for the Onemax domain the upper bound is also function of these same variables; this is believed to be the case for most domains. Thus the lower and upper bounds are independent of the total time allocated  $T$ , the population sizing coefficient  $\gamma$ , and the selection intensity  $I$ . This is interesting in that it denotes that these variables must have at most a minor effect on the optimal sample size, as they can not change that the optimal sample size must lie between the sample size bounds. The selection intensity  $I$ , time bound  $T$ , and population sizing coefficient  $\gamma$  primarily change the quality of the solution at the optimal sample size; they do not change the location of the optimal sample size itself. One caveat is that in GA implementations where ending generation truncation takes place (see above), the time bound  $T$  and population sizing coefficient  $\gamma$  can change the location of the truncation transition points. This can have a relatively minor effect on the location of the optimal sample size. The invariance of these parameters will be experimentally verified in the next section.

### 4.4 Experimental Verification

This section assesses the predictive accuracy of the exact convergence model, and determines if the predicted optimal sample size matches the actual optimal sample size obtained from experimental results obtained using the Onemax domain. In addition, the validity of the lower and upper bounds of the optimal sample size is tested by determining if the actual optimal sample size obtained from experimentation is between the predicted lower and upper bounds. The experiments also verify that Pareto pruning does not prune any optimal sample sizes. The experiments are run for a wide range of different  $\frac{\alpha}{\beta}$  and  $\frac{\sigma_N^2}{\sigma_F^2}$  ratios so as to thoroughly test the validity of the lower and upper bound models. Some additional experiments are also run to demonstrate the relatively minor effect that the time allocated,  $T$ , and population sizing

coefficient,  $\beta$ , have on the optimal sample size, due to the affect on the ending generation truncation transition points. It is also shown that the selection intensity  $I$  has no bearing on the optimal sample size.

#### 4.4.1 Methodology

To evaluate the effect of sample size on GA performance, the GAs are run at a variety of  $\frac{\alpha}{\beta}$  and  $\frac{\sigma_N^2}{\sigma_F^2}$  ratios. For each different combination of  $\frac{\alpha}{\beta}$  and  $\frac{\sigma_N^2}{\sigma_F^2}$  ratios, the lower and upper bound for the optimal sample size are calculated. The GA is then run 50 times (described below) for each sample size between the two bounds, and the results are averaged for each sample size. The experimental results are then plotted against the results predicted by the exact sampling model. The empirically determined optimal sample size is simply the sample size that results in the maximum performance of the GA. The performance metric is the average population fitness of the ending generation for each sample size. This metric was also used in the study by Fitzpatrick and Grefenstette (1988), as the fitness of the final population can be considered to represent the knowledge obtained by the GA.

For each GA run, the following parameters are fully specified for each experiment  $(\frac{\alpha}{\beta}, \frac{\sigma_N^2}{\sigma_F^2}, T)$ . To make the analysis machine independent,  $T$  is measured in terms of the minimum evaluation time per individual, which happens when the sample size is one:  $(\alpha + \beta * 1)$ . Thus  $T = 5000$  includes the time for 5000 fitness evaluations (sample size of one), as well as the corresponding GA overhead time needed to process the individuals. The domain is the Onemax domain, with a chromosome length  $l = 120$ , building block length  $k = 1$ , and minimum signal detection  $d = 1$  (fitness signal of one bit). The convergence model is given by equation 4.4, with the initial proportion of alleles  $p_0 = 0.5$ . The population size is calculated using equation 4.2, with the failure rate  $\psi$  set to half of one bit of the chromosome ( $\psi = \frac{1}{2l} = 1/240$ ). The stopping criterion for the GA is the ending generation calculated by equation 4.7, rounded down to the nearest generation.

The recombination operator is binary tournament selection, with a corresponding selection intensity of  $I = 0.5642$  (Thierens & Goldberg, 1994). The mutation rate is zero, crossover rate is one, and the crossover mechanism is uniform crossover with a swap probability of  $p_{unif} = 0.5$  (Syswerda, 1989).

#### 4.4.2 Results

Below, in Figures 4.7 and 4.8, are the experimental averages from 50 runs in the Onemax domain. The experiments were run for a wide range of  $\frac{\alpha}{\beta}$  and  $\frac{\sigma_N^2}{\sigma_F^2}$  values, as these values are the main influences on the optimal sample size determination (see equations 4.9 and 4.15). The  $\frac{\alpha}{\beta}$  ratio varied from  $\{0.1, 1.0, 10.0\}$ , and are the top, middle, and bottom rows respectively of Figures 4.7 and 4.8. The noise to fitness variance ratio  $\frac{\sigma_N^2}{\sigma_F^2}$  varied, from left column to right column respectively, from  $\{0.5, 1, 2\}$  for Figure 4.7, and from  $\{4, 8, 16\}$  for Figure 4.8.

In these plots, the solid smooth lines are the predicted performance using continuous ending generations, the scalloped smooth line is the predicted performance using truncated ending generations, and the points represent the experimentally determined performance for the given sample sizes, averaged over 50 runs. The performance metric  $p$  is the proportion of correct alleles (building blocks for  $k = 1$ ) in the population at the ending generation. The performance is plotted for all sample sizes  $N$  between the calculated lower and upper bounds

for the onemax domain, which are determined by equations 4.9 and 4.15 respectively and displayed in the plot keys. However, so that the performance curve tendency can be seen even when the lower and upper bound are close, the minimum lower and upper bound plotted are one and five, respectively. As the plotting program does not support greek characters, a quick key of the variables given in each plot is in order:  $a = \alpha, b = \beta, nvar = \sigma_N^2, fvar = \sigma_F^2$ .

In looking at the plots, it is important to note that only the sample sizes between the lower and upper bounds are plotted. This is somewhat misleading, in that it results in there being only a small performance difference for different sample sizes. Figure 4.3 is a good example of plotting over the entire range of  $n$ ; it demonstrates the “bigger picture” of the large effect that sample size has on GA performance.

In general, the experimentally determined performance was very close to the predicted performance. In all cases the optimal sample size was contained within the range predicted by the lower and upper bounds. Indeed, for the Onemax domain tested, a randomly picked sample size between the lower and upper bound is likely to perform within a few percentage points of the optimal performance point. A GA practitioner may use the bounds to guide the selection of a sample size, and be assured that the sample size will produce good, if not optimal, performance.

The Pareto pruning technique was also validated by the experimental results. The experimentally determined optimal sample size in all plots was a Pareto-optimal sample size located between the calculated lower and upper bound; the exception is when the upper bound was calculated to be less than one, in which case the optimal sample size is of course one. For domains with relatively low sampling costs ( $\beta$ ) and high noise ( $\sigma_N^2$ ), the lower and upper bound covered a large range. Fortunately, it is in just such domains that Pareto pruning is the most useful, as it can vastly reduce the number of sample points that need to be considered. As an example, look at the bottom right graph in Figure 4.8, where there are 20 sample sizes inclusively between the minimum and maximum sample sizes, but only 5 sample points have to be considered, as there are only 5 stair edges (at  $n = 9, 13, 17, 21$ , and 24). The Pareto pruning technique is more powerful (greater reduction of sample size search space) for noisier domains, characterized by higher  $\frac{\sigma_N^2}{\sigma_F^2}$  ratios, and for inexpensive sampling functions, characterized by higher  $\frac{\alpha}{\beta}$  ratios. For low noise ratios or expensive sampling functions, it is likely that each sample has a unique ending generation, thus preventing Pareto pruning from reducing the search space by much. However, these conditions also cause the lower and upper bound to be fairly close, so it is not as important to be able to prune when these conditions hold.

#### 4.4.3 Sampling Invariant Results

While the optimal sampling size is highly dependent on the  $\frac{\alpha}{\beta}$  and  $\frac{\sigma_N^2}{\sigma_F^2}$  parameters, it is relatively independent of many others, including  $I$ ,  $T$ , and  $?$ . The selection intensity’s main effect is to scale the performance gained in each generation. As it is just a scaling factor, the optimum sample size  $n^*$  does not change for different selection intensities (or selection mechanisms). The selection intensity also does not change the ending generation (and the corresponding truncation points), so the optimal sample remains the same for different selection intensities. Figure 4.9 show the average of 50 runs for tournament size of 2 ( $I = 0.5642$ ), and 5 ( $I = 1.1630$ ) (Miller & Goldberg, 1996b). The top line in the plot shows the predicted and experimental results for a tournament size of 5, and the bottom line for a tournament



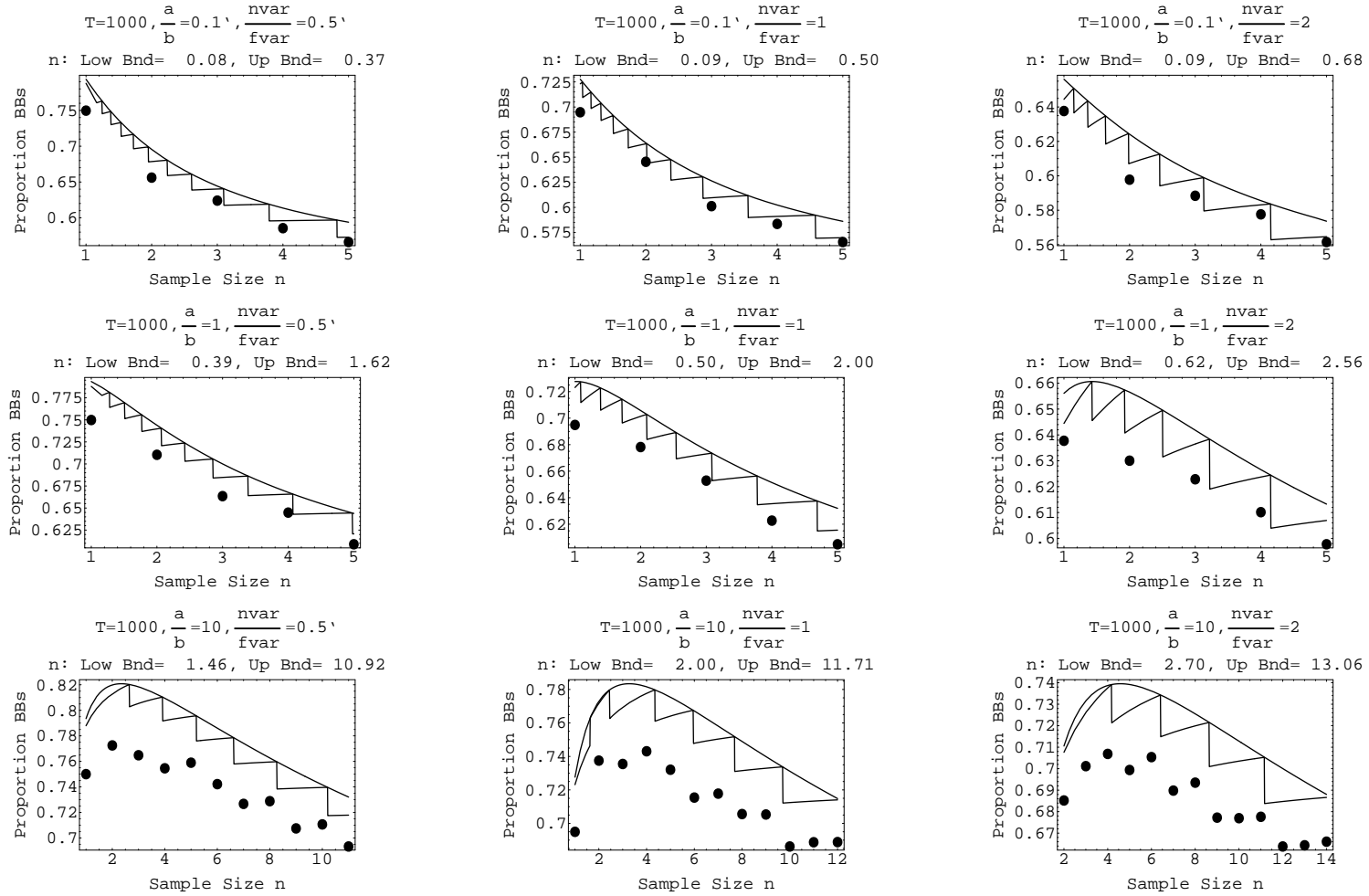


Figure 4.7: Sampling Results for Low Noise Variances

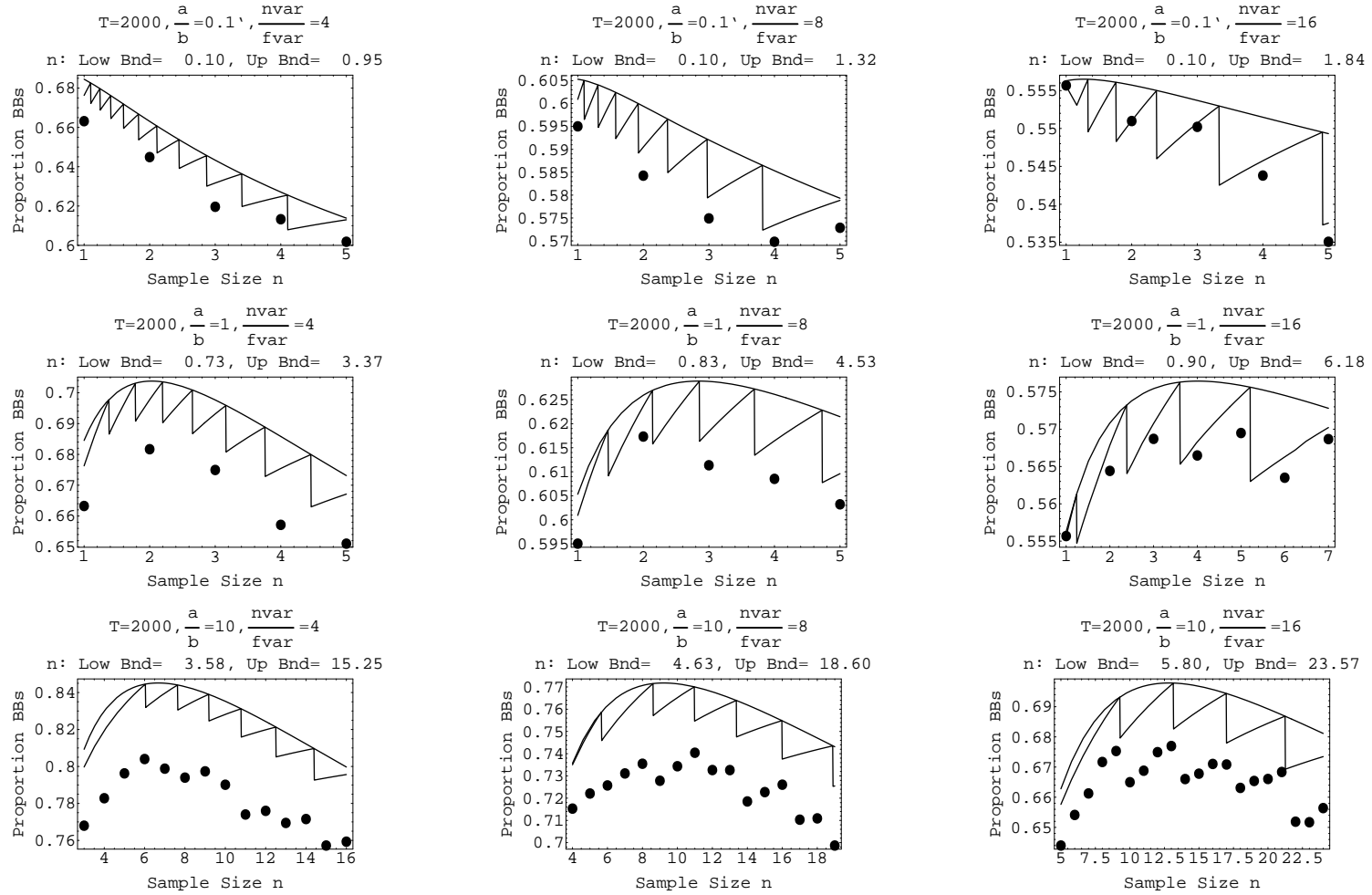


Figure 4.8: Sampling Results for High Noise Variances

size of 2. It is easy to see that changing the selection intensity does not alter the optimal sample size, but just results in a different performance level at the optimal sample size. This is of great practical interest to the GA practitioner, as it eliminates the need to determine the optimal sample size for different selection intensities. Instead, the GA practitioner can rely on the fact that the optimal sample size by one selection mechanism (with its corresponding selection intensity) will be optimal for all other selection mechanisms (selection intensities) also.

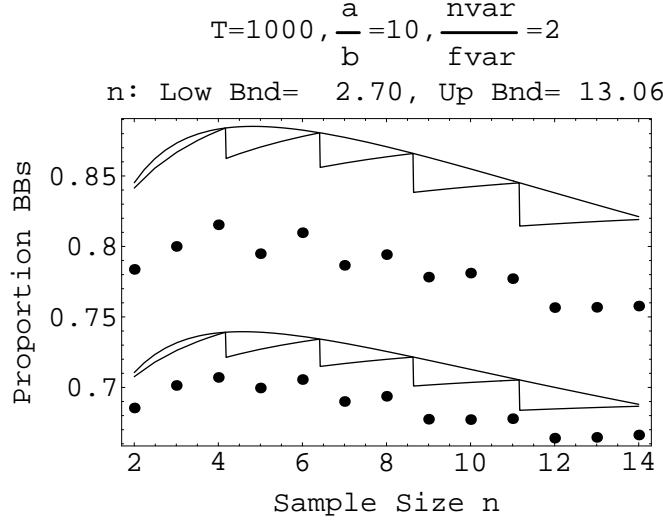


Figure 4.9: Performance as a Function of Selection Pressure

Similarly, the time bound  $T$  and population sizing coefficient  $\psi$  do not have an effect on the optimal sample size  $n^*$ , if a continuous ending generation is used. However, there is a minor effect if the GA is only able to run for a whole number of generations (the ending generation is truncated so that the GA will not exceed the time bound). This is because these variables affect the magnitude of the ending generation, in that the ratio  $\frac{T}{F}$  is directly proportional to the size of the ending generation (see equation 4.7). Changing this ratio changes the number and position of truncation transition points (stairs) between the lower and upper bounds of the sample size. This truncation effect can result in minor differences between the obtained optimal sample sizes  $n^*$  for different  $\frac{T}{F}$  ratios, as an optimal sample point for one  $\frac{T}{F}$  value might be dominated (and thus pruned) for another  $\frac{T}{F}$  value. In Figure 4.10, the experimental are run for different values of  $T$ , and show how the truncation transition points are different. Similar figures can be generated by changing the  $\psi$  parameter, as the transition points are determined by the ratio of  $\frac{T}{F}$ . Note that while  $\psi$  represents the domain specific population sizing variables, for this research the  $\psi$  term is a function of  $k_{max}$ ,  $d_{min}$ , and  $\psi$ . Thus changing the values for any of these parameters only has a minor effect due to the change in the truncation transition points.

In most domains, the initial generation of the GA reflects the fitness landscape. For domains where the fitness landscape is non-normally distributed, it normally takes several generations for the GA operators to “shape” the population distribution into a near normal distribution. For these domains, ratios of  $\frac{T}{F}$  that cause the ending generation to be very small can have significantly different optimal sample sizes, due to the non-normal population

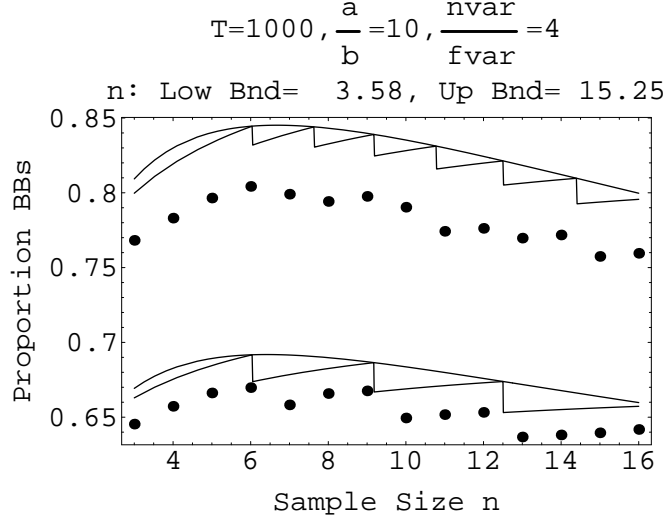


Figure 4.10: Performance for Time  $T = 1000, 2000$

distribution.

## 4.5 Conclusions

This chapter developed methods for determining the optimal sample size for a GA operating in a noisy environment. While this analysis is based on the relationship between the sample size  $n$  and the noise variance  $\sigma_N^2$  (sampling noise variance is  $\sigma_N^2/n$ ), a similar analysis can be conducted for any domain where the noise level is a known function of the sample size. This research derived a model to calculate the number of generations that can be run within a time bound for a given sample size. This, combined with the convergence model for noisy environments derived earlier, allowed for the performance resulting from different sample sizes to be calculated. Furthermore, heuristics for limiting the size of the sample size search space were derived, including lower and upper bounds, along with a pruning technique (Pareto pruning). The usefulness of these heuristics was experimentally verified.

The lower and upper bounds for the optimal sample size were very good at bounding the location of the optimal sample size. An intriguing part of these bounds is that they were independent of many important GA parameters, including the selection intensity  $I$ , the computational bound  $T$ , and the domain dependent population sizing coefficient ? (consisting of the failure rate  $\psi$ ,  $k_{max}$ , and  $d_{min}$  for this effort). The independence of the selection intensity  $I$  was experimentally verified, and is of great use to the GA practitioner. It shows that the optimal sample size obtained for one selection mechanism (with its resulting selection intensity) will be the optimal sample size for any other selection mechanism. In practice, this means that a simple selection mechanism, such as binary tournament selection, can be used in determining the optimal sample size for all selection mechanisms. The other variables mentioned at most have only a very minor affect on the optimal sample size, as they adjust the location of the truncation transition points for the ending generation. For GAs that run for continuous numbers of generations, these variables have no effect on the optimal sample size. It is important to realize that the independence only denotes that they do not have an effect on the location of the optimal sample size; however, these variables do influence the

quality of the solution for an optimal sample size.

## Chapter 5

# Research Applications

### 5.1 Introduction

This chapter discusses how to apply the research presented in the previous chapters to real-world applications. The research results, including population sizing for noisy domains, convergence modeling for noisy domains, and optimal sample size determination, are all utilized in presenting techniques for GA practitioners to optimize the performance of time-bound GAs. While this research has concentrated on how to optimize GA performance within a given time bound, the results are more broadly applicable to GA practitioners who simply desire to improve the efficiency of their GA implementations.

The first section presents general techniques, based on research developed in this dissertation, for initially configuring several key GA parameters, including population size requirements, convergence modeling, and optimal fitness function selection from a range of fitness function possibilities. The next section develops a dichotomy for both noise and fitness. Different applications of this research are then discussed, including image registration, Monte Carlo integration, discrete event simulation, rule evaluation, and neural net training.

### 5.2 Practical Guide to Applicability

This section presents heuristics for optimizing GA performance that can be used by GA practitioners to develop efficient genetic algorithm implementations. First, how to determine the appropriate population size for a domain is discussed. Of particular interest is how to determine estimates of the population fitness and noise variances if the domain theory is inadequate. Next, the development of a convergence model is presented. Last, the question of how to choose a fitness function from a range of possibilities is then presented.

#### 5.2.1 Population Sizing

In Chapter 3, a good rule of thumb population sizing model is derived, and is shown below:

$$N' = -\frac{2^{k_{max}-1} \ln(\psi) \sqrt{\pi}}{d_{min}} \sqrt{\sigma_F^2 + \sigma_N^2} \quad (5.1)$$

where  $\psi$  is the failure rate,  $\sigma_F^2$  and  $\sigma_N^2$  are the population (real) fitness variance and noise variance,  $k_{max}$  is the maximum building block length of the domain, and  $d_{min}$  is the minimum building block signal that the GA is to be able to detect. It is important to note that the population fitness and noise variances are measured relative to the initial generation, as the population is sized according to the requirements of the initial distribution.

In practice,  $\psi$  is often set to a fixed low value such as  $\psi = 0.01$ , or  $\psi = 1/l$ , where  $l$  is the chromosome length. The domain dependent variables  $d_{min}$  and  $k_{max}$  are more problematic,

as they are based on domain knowledge that may not be available. The  $d_{min}$  parameter can be used to limit the acceptable performance bound for the solution identified by the GA: any chromosome within  $d_{min}$  of the true optimal is considered an acceptable solution. For fitness functions that return integer values, this can be conservatively set to  $d_{min} = 1$ . Obviously, it must never be set to zero, as this causes an infinite population sizing requirement. A heuristics for setting  $k_{max}$  is to take a guess at the maximum number of variables in the domain that are interdependent, and set  $k_{max}$  to the number of bits needed to represent these variables in the chromosome.

In equation 5.1, the sum of the population fitness variance and the noise variance is equivalent to the initial noisy fitness variance of the population. The initial noisy fitness variance can be directly obtained from the noisy population fitness values of the initial generation. This can be used in place of the sum of the population plus noise variance sum in the population sizing model.

However, it is often useful to be able to estimate the noise variance for a given fitness function. The estimate of the noise variance is used by the convergence and sampling models, as well as in selecting an optimal fitness function. This estimate can be obtained through sampling of the initial population. If there is a fixed random noise component as part of the fitness, randomly pick an individual from the population, and calculate the variance of  $x$  different fitness evaluations of the individual. The variance is entirely due to noise, and can thus be used as the estimate for the noise variance. However, some domains have noise components that are dependent on the fitness of the individual. In this case, the noise variance for the population sizing equation should be set to the average mean noise variance of the population. Select  $x$  individuals, find the noise variance for each by sampling each individual  $y$  times to obtain its noise variance, and then take the mean of the  $x$  obtained noise variances. This can be used to obtain a working estimate of the average mean noise variance of the population. The true fitness distribution of the initial population can be estimated by subtracting the noise variance estimate from the initial noisy population fitness variance,

### 5.2.2 Convergence Model Development Heuristics

A convergence model is useful for estimating the tradeoffs of different GA implementation decisions, such as fitness function selection or sample size determination. Ideally, the domain knowledge should guide the development of a convergence model. The research presented in Chapter 2 dealt with determining the effect of noise on the selection pressure. The selection pressure is the prime component in determining the convergence rate of a GA, and the selection pressure convergence models developed can be used as a heuristic for estimating the overall convergence rate. It is only a rough estimate, as a complete convergence model must account for the building block disruption effect of recombination and mutation, which is very difficult to do even with extensive domain knowledge. As a result of these factors, it is often necessary to estimate the convergence model for a domain.

In the previous chapter on sampling, an upper bound for the sample size was derived by simplifying the selection pressure convergence model. The resulting convergence model had the form:

$$f(t) = \frac{e^{\frac{x}{\sigma_N}t+c}}{1 + e^{\frac{x}{\sigma_N}t+c}}$$

This equation results in the desired “shape” for performance as a function of time, as this

behavior is typical of GA convergence in most domains. It also takes into account the effect of different noise levels on the selection pressure, with  $\sigma_N$  representing the standard deviation of the noise variance (and includes the effects of sampling). Now that the population sizing requirements are known (see previous section), the GA can be run so as to calibrate the  $x$  variable, which controls the convergence rate, and  $c$ , which determines the starting population fitness. These can be calibrated by determining the initial population fitness (uniquely determines  $c$ ), and the ending fitness  $f_{end}$  at time  $t_{end}$ , which is used to calibrate  $x$ .

Again, this produces a very rough model of GA performance, which can then be used to analyze performance tradeoffs. For domains where the GA convergence is not (roughly) described by the above general convergence model, “custom” convergence models must of course be developed.

### 5.2.3 Optimal Selection of Noisy Fitness Function

This section discusses how to select a fitness function from a range of possibilities so as to maximize performance. Many domains have a multitude of different fitness function possibilities (e.g., chess board evaluation), each of which can have different speed and accuracy characteristics. It is assumed that the performance characteristics of the fitness functions, speed and accuracy, are known. If not, the speed and accuracy of a fitness function can easily be estimate through sampling (see the population sizing discussion above).

The first step is to identify the Pareto-optimal set (Hans, 1988) of the fitness functions, based on the speed and accuracy attributes. Simply remove any fitness function that is dominated by another fitness function; a fitness function is defined as dominated if it is worse, in terms of both speed and accuracy, than another fitness function in the set. The remaining fitness functions form the Pareto-optimal set.

Next, calculate the performance for each remaining fitness function by determining the ending generation, and then using the convergence model to predict the performance of the fitness function at the calculated ending generation. Equation 4.6 can be used to calculate the ending generation (using the population size estimate described above), with the  $\beta n$  term reflecting the computational overhead of the fitness function. The accuracy of the fitness function is reflected in the  $\sigma_N$  parameter in the convergence model. The fitness function resulting in the highest performance is selected. In this manner, the optimal fitness function can be determined for a given domain.

This technique can also be used to optimize GA performance in domains where the GA is allowed to run until convergence. In this case, the technique enables the GA to run more efficiently, and thus reduce the amount of time until convergence. For domains where there is no time-bound, the GA practitioner can arbitrarily select one to use for optimization purposes. This is a ramification of the fact that the performance optimization of the GA is relatively independent of the time bound, which was shown in the previous chapter. Using the selected time-bound, the GA practitioner can then perform the above procedure to identify the most efficient fitness function from the range of possibilities.

## 5.3 Noise and Sampling Dichotomies

This section first puts forth a simple dichotomy of noise, based on whether the noise is independent of the fitness distribution. Next, sampling is categorized based on the size of the sample space.



### 5.3.1 Noise Classification

This study has so far concerned itself with a noise source that is independent of the fitness distribution. This type of noise source is a good model for noise resulting from faulty data, sensor noise, or human error, as it introduces noise into the fitness evaluation that is independent of the individual being evaluated. Sampling can be used to reduce the noise level through the use of the central limit theorem. For this type of noise, the magnitude of the noise remains constant throughout the GA run.

Another type of noise results when a fitness function samples data that is used to evaluate an individual. The magnitude of the noise turns out to be dependent on both the sample size (quantity of data), as well as the fitness of the individual being evaluated. This last point is best illustrated by example. If the fitness of an individual is based on the individual's predictive accuracy at classifying the data, an individual that accurately classifies the entire data set will never have any noise. That is, no matter what subset of data is used in evaluating this individual, the fitness function will always return a perfect score (which is correct, and is thus not noisy). Contrast this with an individual that only correctly classifies a portion of the data; the variance of the fitness evaluations for this individual is very much a function of the sample size. This illustrates that the magnitude of the noise resulting from a sampling fitness evaluation for an individual is dependent on the fitness of an individual, as well as the sample size.

For the fitness dependent noise domains considered in this study, the noise variance turns out to be proportional to the fitness variance of the population. As the GA causes the fitness variance to decrease over time, so does the noise variance. The case of an independent (and constant) noise source can be considered to be a limiting case for domains with dependent noise, as the decreasing noise variance results in better performance than predicted by an independent (constant) noise convergence model. Thus the optimal sample size lower bound developed for the constant independent noise source in Chapter 4 can also be considered as the lower bound for a dependent noise source. In this case, the noise variance for the optimal sample size lower bound is the (conservative) noise variance in the initial population.

### 5.3.2 Sampling Classification

Sampling can be categorized on the basis of the sample size search space. If there is an infinite range of possible fitness outcomes for a single fitness evaluation, the sample size is said to be infinite. The noisy Onemax domain (independent noise component) and discrete event simulation (infinite number of input distribution samples - described below) are good examples of domains with infinite sample size search spaces. For this type of sampling search space, the noise variance may be reduced through the central limit theorem, but never eliminated.

The other type of sampling occurs when there is a finite range of fitness evaluation outcomes, normally due to having a finite amount of data with which to evaluate the individual. The true fitness of an individual is defined by evaluating the individual with the entire data set. Then for this type of domain, the noise can be eliminated by simply setting the sample size to the size of the sample size space (size of the data set). In general, the noise variance is a function of the proportion of the data set used in evaluating the individual.

## 5.4 Applicable Domains

The following section discuss several domains where the sampling analysis may be employed. It is divided into domains categorized by noise that is independent or dependent on the fitness distribution.

### 5.4.1 Fitness Independent Noise

This research primarily concentrated on modeling noise as an independent source of the fitness distribution. This type of noise directly models faulty data, sensor noise, or human error, as all of these introduce error into the system that is independent of the individual being evaluated. As such, the central limit theorem can be used to define the resulting noise variance as a function of the sample size. The lower bound and upper bound derivations for the sample size were defined for applications characterized by a fitness independent noise source. The noisy Onemax domain, presented in the previous chapter, demonstrates the application of this research on a domain with an independent noise source. It verified the validity of the optimal sample size lower and upper bounds.

#### 5.4.1.1 Discrete Event Simulation

Discrete-event simulation concerns the modeling of a system as it evolves over time by a representation in which the state variables change instantaneously at separate points in time (Law & Kelton, 1991). It is typically used to model complex systems that react to input (i.e., order arrival), where the input occurs with a known distribution. The discrete event simulation is typically run over  $n$  different trials, with each trial having its input randomly drawn from the input distribution, and the simulation outcomes being monitored to evaluate the given system configuration performance. In addition to the input distributions, there are typically internal processes of the discrete event simulator whose actions are controlled by a given distribution. In either case, this research may be used to determine how many samples of a given discrete event simulation are needed to accurately evaluate a given discrete simulation system for a known input process distribution. As the noise is based on sampling from the known (infinite) input distributions, this can be considered a case of fitness independent noise.

### 5.4.2 Fitness Dependent Noise

For domains characterized by fitness dependent noise, the noise of a fitness evaluation is related to the true fitness of the individual. Generally, the more fit the individual, the less noise. As discussed above, for domains where the noise is proportional to the fitness, the noise will decrease as the population fitness converges. As such, the performance of a domain characterized by this type of noise will be better than the corresponding domain with constant noise. This entails that the lower bound for the independent noise source case will also be a lower bound for the fitness proportional noise source case. Also, the convergence model for the fitness independent noise source can be used as a lower bound for the performance of the fitness proportional noise source. Several different domains characterized by this type of noise are discussed below.

#### 5.4.2.1 Image Registration

Image registration is a domain where sampling can be used to efficiently estimate the difference between two images (Grefenstette & Fitzpatrick, 1985). Image registration has been used in such fields as aerial photography and medical imaging.

Image registration attempts to transform one image to another by performing geometrical transformations. To see if the geometrical transformation has succeeded, the images must be compared on a pixel by pixel basis. Sampling of the pixels may be used to estimate the pixel difference between two images, and may in fact be necessary as the number of pixels can be extremely large.

An image comparison, base on pixel sampling, of two images may be done in the following manner. Randomly sample  $n$  pixel locations. For each pixel location, if the pixel value at each image matches, increments a “match” counter  $m$ . The fitness function  $f$  estimates the total number of matching pixels in the images  $i_1$  and  $i_2$  by multiplying the sample match proportion times the total sample search space  $S$  (the number of pixels in the image):  $f'(i_1, i_2) = \frac{m}{n} * S$ .

According to the central limit theorem, the sampling noise is distributed  $N(0, (S - n) * \sigma_f^2/n)$ , where  $\sigma_f^2$  is the fitness variance of the individual over the entire data set. This is because the mean of the  $n$  pixel samples has a variance of  $\sigma_f^2/n$  (central limit theorem), which is used to predict the remaining  $S - n$  pixels of the individual that were not sampled. Given that the population fitness is distributed  $N(\mu_f, \sigma_f^2)$ , along with the additive property of normal distributions, gives that the noisy fitness of the population is distributed  $N(\mu_f, \sigma_f^2 + \frac{S-n}{n}\sigma_f^2)$ , where  $\mu_f$  is the true mean average fitness. The sum of the population and noise variance can be simplified to  $\frac{S}{n}\sigma_f^2$ .

Image comparison is an example where the noise variance can clearly be shown to be a function of both the population fitness variance and the sample size. As such, it is a good domain to test whether the optimal sample size lower bound, developed for the independent noise source in the previous chapter, is good for a noise source that is dependent (proportional) on the population fitness variance.

A simplified domain, Image Matching, is used to verify this the lower bound on the sample size for the image comparison function. The geometrical transformation requirement of the image registration domain is dropped, and the problem then shifts to evolving an image that matches a fixed image. The fitness value for an image is simply the number of pixels for which it matches the fixed image, with the chromosome length being the number of pixels in the image. The fitness function employs sampling without replacement to estimate the fitness of an image from a sampled subset of pixels (see above image comparison description). In practice, the fixed image is randomly generated at the start of each GA run. This domain is equivalent to the noisy Onemax domain, but with a fitness dependent noise source as opposed to the constant noisy source for the noisy Onemax domain.

The convergence model for the Onemax domain is used to bound the convergence for this domain, with the noise variance in the convergence model set to noise variance in the initial generation of the Image Matching domain. The population sizing requirements are likewise sized using the initial noise variance in the Image Matching domain. As the Image Matching domain noise variance will decrease over time due to its (proportional) dependence on the fitness variance, the Onemax model actually overstates the noise variance of the Image Matching domain for all but the initial generation. This results in the Onemax model being a conservative performance model for the Image Matching domain.

The lower bound for the Onemax domain can be shown to hold for the Image Matching domain. The lower bound of the sample size is the sample size that optimizes the ending generation of the GA. This does not account for the beneficial affects of lowering the noise variance on the convergence rate, and is thus the lower bound for both the Onemax and

the Image Matching domains. However, the upper bound derivation for the Onemax domain relied on a convergence model that converged provably faster than the domain. As discussed above, Image Matching will converge at least as quickly as the Onemax domain, and maybe even quicker than the upper bound convergence model for the Onemax domain. As such, the upper bound result can not be expected to hold for the Image Matching domain.

Figure 5.1 demonstrates that the lower bound for the Onemax domain holds for the Image Matching domain. The solid line is the Onemax convergence model, and the points represent the experimentally determined performance for image matching as a function of the sample size, averaged over 10 runs. These plots are plotted at three difference  $\alpha/\beta$  ratios: (.1, 1, 10). For this domain the Onemax domain convergence model is a reasonable predictor of the optimal sample size: the actual Image Match optimal point is in the neighborhood of the maximum point on the Onemax convergence model. However, while the lower bound should hold for any domain whose noise variance is proportional to the fitness variance, the optimal sample point in general may not be gleaned from the Onemax convergence model.

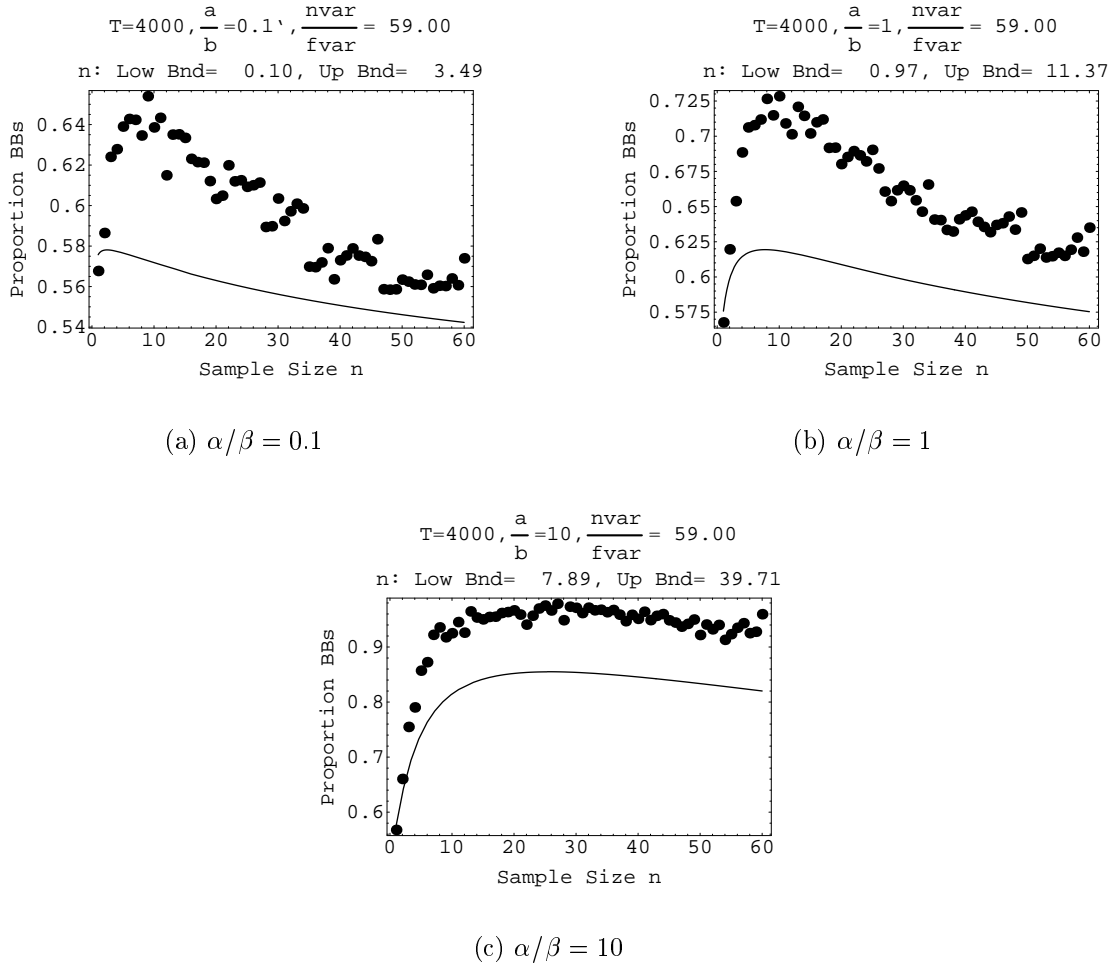


Figure 5.1: Experimental Results for Image Matching Domain

The rest of the domains discussed have noise variances that are dependent on the population fitness distribution, just as in the Image Matching domain. As such, as long as the noise

variance has the property that it decreases over time, the optimal sample size lower bound should be valid.

#### 5.4.2.2 Monte Carlo Integration

Another possible application for genetic algorithms employing sampling is Monte Carlo integration (Hammersley & Handscomb, 1964), where the integral of a function is estimated based upon randomly sampled points. This technique is used for multiple integrals, as better techniques exist for single integral approximations.

If the integral is given as follows:

$$\int_{a_1}^{b_1} \int_{a_2}^{b_2} \cdots \int_{a_n}^{b_n} f(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n$$

then Monte Carlo integration would approximate this value of this integral by multiplying the base area times the average height of the function in the base area. The base area is the product of the integral bounds:  $b = \prod_{i=1}^n (|b_i - a_i|)$ . The average height is simply the mean of  $n$  function evaluations at points uniformly sampled from the base area. The average height can be shown to be distributed  $N(\bar{h}, \sigma_f^2/n)$  for sufficiently large  $n$ , with  $\bar{h}$  being the average height for the defined base area. This results in the Monte Carlo technique being an unbiased estimator of the integral, with a variance of  $\sigma_{mc}^2 = b \frac{\sigma_f^2}{n}$ . It is important to note that  $\sigma_f^2$  is the variance of the function over the sample points defined in the base area; it is not the variance of the function over infinite bounds. Thus the noise variance is directly dependent on the bounds of the area being sampled, as well as the sampling function  $f$ , and sample size  $n$ .

As this domain has a noise variance that is dependent on the fitness variance, the independent noise variance techniques may be used to define the (conservative) lower bound on the optimal sample size.

#### 5.4.2.3 Hypothesis Testing

Hypothesis testing domains are domains where the GA is responsible for evolving an accurate hypothesis (or “rule”). The accuracy of a given hypothesis is verified by testing its predictive accuracy on samples from a given data set, known as the training data set. The noise resulting from evaluating an individual can be shown to be a function of the predictive accuracy of the hypothesis, as well as the sample size. As such, the optimal sample size lower bound derived for independent noise sources should be a conservative lower bound on the sample size for hypothesis testing. Neural net training, discussed below, can be considered to be a special case of this type of domain.

#### 5.4.2.4 Neural Net Training

Genetic algorithms are frequently used to determine the correct weighting for a neural net. A fitness evaluation of an individual consists of evaluating the effectiveness of the given weighting for the neural net on a sample randomly drawn from the training set. An important question is to determine how many samples should be used from the neural net training set so as to accurately (and efficiently) evaluate a given a neural net weighting. The noise from the fitness evaluation is dependent on the fitness of the individual being developed. As such, the optimal sample size lower bound derived for independent noise sources should be a conservative lower bound on the sample size for neural net training.

## 5.5 Conclusions

This chapter focussed on applying this research to real-world problems. A general guide for determining the population sizing requirements for noisy domains was presented, as well as how to estimate various domain parameters through the use of sampling, including the fitness and noise variances. Next, the development of a rough GA convergence model is presented, which is useful for domains where there is insufficient domain knowledge to build an exact convergence model. A technique is then introduced for selecting the optimal fitness function from a range of possibilities, based upon the population sizing and convergence models developed earlier. A simple dichotomy of noise (independent or dependent of fitness distribution) and sampling (infinite or finite sample search space) is then presented. The work done in this dissertation is based on independent noise source in an infinite sample size search space. However, the lower bound of the optimal sample size, which is based solely on the ending generation and computational requirements of the GA, is shown to be domain independent. That is, the lower bound also holds for domains where the noise variance is dependent on the fitness distribution. This is illustrated using the Image Matching domain, which is an abstraction of the Image Registration problem. Several other domains where this research should be applicable, including Monte Carlo integration, discrete event simulation, hypothesis testing, and neural net training, were then discussed.

This research has been shown to be broadly applicable. The effect that varying the noise or selection pressure has on the population sizing requirements is applicable across all domains, not just those employing sampling. The optimal lower bound for the sample size is shown to not only work in domains where the noise source is independent of the fitness, but also in any domain where the noise is proportional to the fitness. This is very useful, as it allows for the determination of the lower bound of the optimal sample size in the majority of domains where sampling is applicable. Another useful result is that the convergence model developed based on an independent noise source can be used as a lower bound on the performance of a fitness proportional noise source domain.

## Chapter 6

# Conclusions

This chapter first summarizes the main results of this research effort. Next, some extensions to this research are proposed, including building block scaling, population sizing, and noise modeling. Lastly, some conclusions are drawn about this research effort.

### 6.1 Summary

Now that GAs are increasingly being used by industry, the effects of noise on performance are becoming more important. The main contribution of Chapter 2 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 is derived that accurately predicts the performance of several selection schemes in noisy environments, including tournament selection,  $(\mu, \lambda)$  selection, linear ranking, and stochastic universal selection. This is extremely useful later when determining the optimal sample size, as it allows for the performance of the GA at a given noise level to be predicted. Furthermore, any selection scheme model based upon the deterministic selection intensity model can be similarly adapted to handle noise.

Chapter 2 also introduces a new technique for isolating the effects of building block disruption, Uniform Building Block Crossover. While this study used it to remove the effects of building block disruption so as to better isolate the effects of selection pressure, it will more useful to the research community studying the building block disruption effects of recombination and mutation. It allows for the effect of BB disruption to be quantified by comparing the performance of a GA employing uniform building block crossover to that of a GA employing the building block disruption operator that is being studied. Uniform building block crossover is an important new tool for analyzing the inner mechanisms of genetic algorithms.

Chapter 3 presented an important population sizing model, the random walk population sizing model, first presented by Harik, Cantú-Paz, Goldberg, and Miller (1997). The random walk population sizing model incorporates theory from previous population sizing models based on building block supply theory and building block decision theory. The random walk population sizing model was shown to accurately predict the performance as a function of population size for uniformly scaled domains. Equivalently, the random walk population sizing model can be used to determine the population sizing requirements of a GA when sized using the least significant partition (the partition which has the greatest population sizing requirements). However, if the individual partitions in a domain are not known, a conservative approximation of the random walk population sizing model was derived, and shown to be accurate. This approximation requires minimal domain knowledge (just the maximal building block length and the minimum fitness signal for the domain), and should be very useful for GA implementors as a rule-of-thumb population sizing model. The population sizing approximation also clearly revealed the relationship between various parameters in the

random walk population sizing model and the corresponding population sizing requirements.

In Chapter 3, the random walk population size was also extended to handle varying noise and selection intensity levels, and the accuracy of the extended model was experimentally verified (Harik, Cantú-Paz, Goldberg, & Miller, 1997). This is the first time that any population sizing model has been able to predict the effect of selection intensity on the population sizing requirements. Furthermore, the ability to quantify the population sizing requirements for varying noise levels becomes an important part of determining the optimal sample size, or optimal fitness function, for a given domain.

Chapter 4 builds upon the convergence models and population sizing models for noisy environments developed in the earlier chapters in order to present methods for determining the optimal sample size for a GA operating in a noisy environment. A model is presented for calculating the ending generation of a time-bound GA employing a fitness function with known speed and accuracy characteristics. The performance can be predicted by the convergence model, where the convergence model is given the noise of the fitness function and the calculated ending generation. In this manner, the performance of different sample sizes can be compared, or any fitness function with known speed and accuracy characteristics.

Furthermore, Chapter 4 develops heuristics for limiting the search for the optimal sample size. A domain independent model for calculating a lower bound is presented, based on maximizing the ending generation of the GA. This lower bound model holds for all domains where the convergence rate monotonically increases for decreasing noise levels. Furthermore, an approach for determining the upper bound is developed. This approach requires a convergence model that provably converges faster than the actual convergence, and is thus domain dependent. Another technique, Pareto pruning, is also presented. This technique is very useful for domains characterized by high noise variances and low sampling costs, as it is able to prune a large portion of the sample size space between the lower and upper bound.

Chapter 4 also demonstrates that the optimal sample size is largely a function of two ratios: the ratio of the GA overhead to the sampling overhead, and the ratio of the noise variance to the fitness variance. Of equal importance is what does not influence the optimal sample size: selection intensity, time bound, and domain dependent variables such as the failure rate, maximum building block length, and minimum signal strength. It is important to realize that the independence only denotes that they do not have an effect on the location of the optimal sample size; however, these variables do influence the quality of the solution for an optimal sample size. This independence allows for the GA practitioner to be confident that the optimal sample size determined for one selection intensity will be optimal at other selection intensities.

Chapter 5 focussed on applying this research to real-world problems. A general guide for determining the population sizing requirements for noisy domains was presented, as well as how to estimate various domain parameters through the use of sampling, including the fitness and noise variances. Next, the development of a rough GA convergence model is presented, which is useful for domains where there is insufficient domain knowledge to build an exact convergence model. Next, the population sizing and convergence models are used to identify the optimal fitness function from a pool of fitness functions that differ based on their speed and accuracy characteristics. A simple dichotomy of noise (independent or dependent of fitness distribution) and sampling (infinite or finite sample search space) is then presented. The work done in this dissertation is based on independent noise source in an infinite sample size search



space. However, the lower bound of the optimal sample size, which is based solely on the ending generation and computational requirements of the GA, is shown to be domain independent. That is, the lower bound on the sample size also holds for domains where the noise variance is dependent on the fitness distribution. This is illustrated using the Image Matching domain, which is an abstraction of the Image Registration problem. Several other domains where this research should be applicable are discussed, including Monte Carlo integration, discrete event simulation, hypothesis testing, and neural net training.

## 6.2 Future Research Areas

This research can serve as the basis for future work. Several different promising research areas are listed below.

### 6.2.1 Building Block Scaling

This dissertation mainly dealt with domains where the BBs are uniformly scaled. Another important class of domains is where the BBs are exponentially scaled, and should be investigated. Also of interest, but more difficult, are domains where the BBs are badly (arbitrarily) scaled. These domains might be able to be analyzed using a combination of the uniformly scaled and exponentially scaled analysis. The analysis should be applied to determine the effect of noise on both the population sizing requirements and the convergence performance.

### 6.2.2 Population Sizing

There are several issues left to be explored for population sizing. First, it has been suggested that population sizing for selection schemes other than tournament selection can be obtained by extrapolating from tournament selection with similar selection intensity. This needs to be experimentally verified. Next, the effect of larger tournament sizes on the selection intensity is only approximated in Chapter 3 (equation 3.7). This might be better captured by a model based directly on order statistics, and bears further investigation. While a conservative general population sizing equation is derived, it is a “worst-case” model for the population sizing requirements. A more accurate model might be developed for exponentially scaled domains. Furthermore, the population sizing model is increasingly less accurate at higher noise levels. Work needs to be done to investigate the root cause of this inaccuracy.

### 6.2.3 Use of Noise to Model Building Block Disruption

This research concentrated on how noise reduced the convergence rate of a GA by analyzing the effect of noise on the selection process (the good decision probability calculation). The effect on convergence of other GA operators, such as mutation, selection, or recombination, might be modeled as additional noise sources to the selection process. This appears to be a very promising technique for measuring the convergence effect of building block disruption, which can be caused by either mutation or recombination.

It is also important to realize that this dissertation did not address the effects of building block disruption. This research focussed on determining how noise affects the selection process, and subsequently how the selection process affects the convergence rate. Building block disruption was eliminated in this study by employing the uniform building block crossover operator, or by using domains characterized by one bit building blocks, where no building block disruption was possible. This allowed for the isolation of the effect of selection on conver-

gence, and thus led to models that could quantify the convergence effect of noise. In practice, building block disruption is another important part of the convergence equation. However, it is highly domain dependent, as it is to a large part based on the BB linkage unique to the domain. This is an active area of research for the GA community, and would be very interesting to combine with this research effort to develop a model that could completely predict the convergence effect of noise for any domain. However, this research is extremely useful by itself in that the effect of noise on the selection pressure can be used to roughly estimate the convergence performance of a domain.

#### 6.2.4 Dependent Noise Modeling

This research primarily focused on noise sources that were constant, and thus independent of the population fitness variance. However, it was demonstrated that the optimal sample lower bound holds for any domain where the convergence rate monotonically increases for decreasing noise levels. This is true for domains where the noise variance is proportional to the fitness variance. It would be very useful to build a general convergence model for domains where the noise variance is proportional to the fitness variance. This would allow for an in-depth analysis for many domains employing sampling, including discrete event simulation, hypothesis testing, image registration, and neural net training.

### 6.3 Conclusions

This research has several important ramifications for GA practitioners interested in deploying efficient genetic algorithms.

1. This research demonstrates the relationship between noise, selection intensity, and the corresponding population sizing requirements. An efficient GA must have a population that is sized to correctly account for both the noise in the domain and the selection intensity of the selection mechanism. Failure to account for either effect can result in a GA that prematurely converges, or has an unnecessarily large population that results in inefficient GA operation.
2. Noise has a detrimental affect on the convergence rate of a GA. A model for quantifying the effect of noise on the selection pressure is presented that quantifies the effect of noise on the overall convergence of the GA. This enables the GA practitioner to estimate the impact of different noise levels on the overall performance of the GA. In addition, techniques are presented for developing convergence models for domains where there is little domain knowledge to guide the GA practitioner.
3. Genetic algorithms can deploy sampling fitness functions to reduce the operating noise level of the GA in noisy domains. The tradeoff for decreasing the noise level is the additional overhead of the sampling fitness function. There is an optimal sample size that balances the performance penalty for the additional sampling overhead with the performance advantages gained from having the lower noise variance, which causes faster convergence rates and smaller population sizing requirements. This optimal sample size has a domain independent lower bound, and a technique called Pareto-pruning can eliminate many other sample sizes from optimal sample size consideration. In addition, a technique is presented for developing a domain dependent upper bound on the optimal

sample size. These heuristics can greatly aid a GA practitioner in determining the optimal sample size for a noisy environment.

4. The optimal sample size determined for one selection mechanism will be the optimal sample size for any other selection mechanism. This is a result of the optimal sample size being invariant of the selection pressure. It is also largely invariant of the computational time bound, failure rate, maximum building block length, and minimum signal strength.
5. The same techniques developed for determining the optimal sample size can be used to determine the optimal fitness function from a set of fitness functions with known speed and accuracy characteristics.

Taken together, this research is a strong step forward for easing the design of efficient GAs for industrial (and noisy) applications.

# Appendix: Genetic Algorithm Overview

This section gives a high-level overview of genetic algorithms; for a more detailed description, see (Goldberg, 1989a; Davis, 1991; Mitchell, 1996). A simple genetic algorithm is presented, and then the major components of a GA are discussed.

## A.1 Simple Genetic Algorithm

A simple GA is depicted in Figure A.1. The desired output of a GA is an optimal solution for the given problem. Genetic algorithms attempt to identify optimal solutions by applying the techniques of natural selection to a population of solutions: the solutions are evaluated, the bad solutions are killed, and the remaining solutions are recombined (“mate”) to form a new generation of solutions. The general principle is that of Darwinism - the good traits will survive over time as they are less likely to be on solutions that are killed in a generation. Over a number of generations, the individuals in a population will have fewer “bad” traits, resulting in an overall increase in the quality of solutions in the population. Mutation is a less important operator that can be viewed as an “insurance policy” against premature convergence. The mutation operator randomly alters parts of the chromosome, in order to discover new building blocks (BBs), or to prevent the GA from premature convergence to a non-optimal BB.

## A.2 Chromosomes and Building Blocks

Chromosomes are used to encode the solution. Normally the chromosome is represented as a binary vector (see Figure A.2), but other encodings are also possible (Goldberg, 1989a; Davis, 1991). Combinations of bits (schemata) that acts as a unit to influence the fitness of an individual are known as *building blocks*. In Figure A.2, the building blocks are shown to consist of adjacent bits, but in practice the building block can have bits scattered throughout the chromosome. A GA can be seen as an algorithm that, through the selection process, slowly eliminates bad building blocks from the population. A facet-wise decomposition of a GA based upon building blocks (BBs) (Goldberg & Rudnick, 1991; Goldberg, Deb, & Clark, 1992) is shown below.

1. Know what a GA is processing: building blocks
2. Solve problems tractable by BBs
3. Supply enough BBs in the initial population
4. Ensure the growth of necessary BBs
5. Mix the BBs properly
6. Decide well among competing BBs

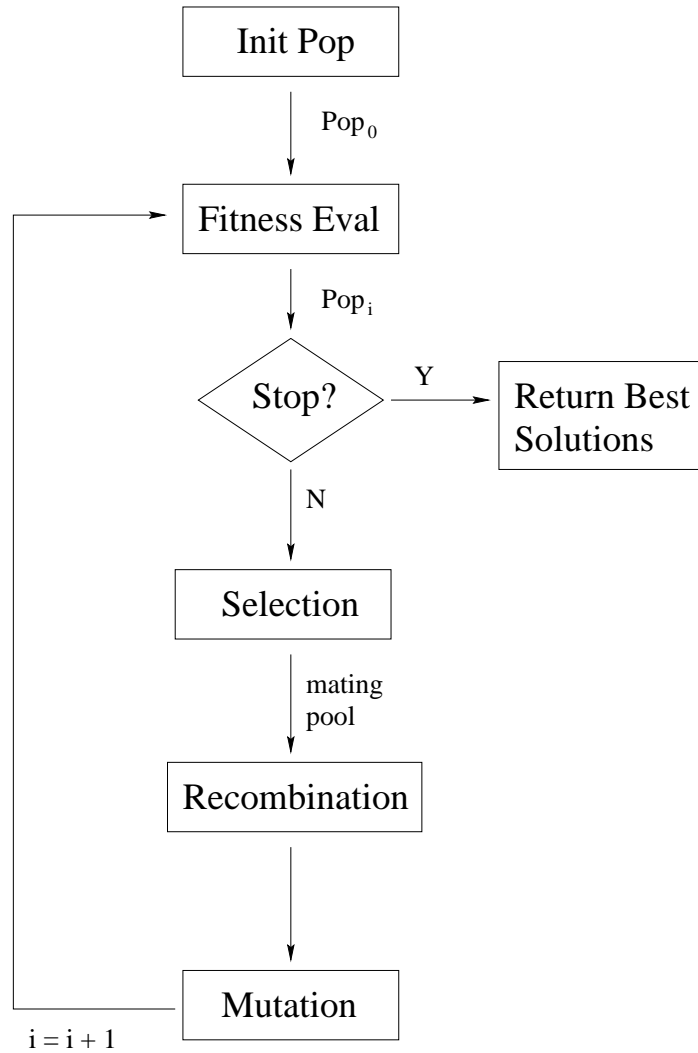


Figure A.1: Genetic Algorithms in a Nutshell

### A.3 Population Sizing and Initialization

The first step in a GA is to produce a population of solutions. If a heuristic is available for producing good solutions for the domain, it can be used in generating the initial population of solutions. Otherwise, the solutions are randomly generated. The number of solutions, known as the population size, is important in that it can influence whether the GA can identify optimal solutions. If the population size is too low, there may be an inadequate supply of building blocks in the population, resulting in a GA converging to a sub-optimal solution (premature convergence). If the population size is set too high, the GA wastes time processing redundant individuals.

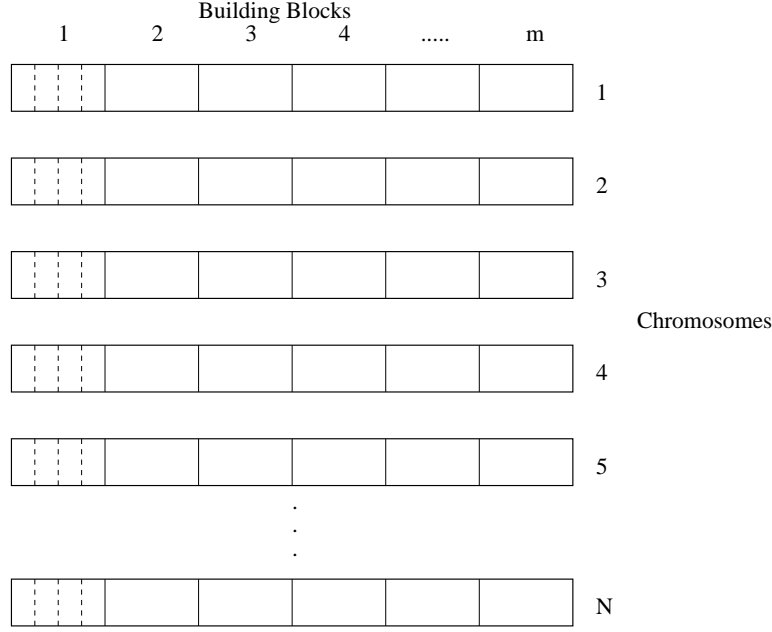


Figure A.2: Populations, Chromosomes, and Building Blocks

## A.4 Fitness Functions

A fitness function returns a fitness value that is used by the selection mechanism in determining which individuals will survive and recombine. Ideally, a fitness function will accurately assess an individual, and return a fitness value that can be used to make correct selection decisions. Most research into GAs has been done using fitness functions that accurately assess an individual. However, especially in industry, it is sometimes necessary to use fitness functions that can only approximate the true fitness of an individual. These approximate fitness functions are known as noisy fitness functions.

### A.4.1 Fitness Noise Definition

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.

For the purposes of this research, a noisy fitness function is defined as a deterministic fitness function plus a random noise component. The noisy fitness value  $f'$  of an individual can be represented as  $f' = f + noise$ , where  $f$  is the real fitness of the individual, and  $noise$  is the noise inherent in the fitness function evaluation. The deterministic portion of the fitness  $F$  of the population is assumed to be normally distributed  $N(\mu_{F,t}, \sigma_{F,t}^2)$ . This research further assumes that the noise portion of the fitness is unbiased and normally distributed  $N(0, \sigma_N^2)$ . 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,

gives that the noisy fitness  $F'$  of the population is normally distributed  $N(\mu_{F,t}, \sigma_{F,t}^2 + \sigma_N^2)$ .

#### A.4.2 Fitness Noise Ramifications

The noise within the fitness evaluations has several important ramifications. First, the noise inherent in the fitness evaluations causes the selection process to be noisy, as “bad” individual may be selected at the expense of “good” individuals. For ordinal-based selection schemes such as tournament selection, linear ranking, and  $(\mu, \lambda)$  selection, this fitness noise slows down the convergence rate of the GA. This is caused by the quality of the selected individuals decreasing with increasing fitness function noise levels. For proportionate-based selection mechanisms, noise does not change the overall convergence rate, as will be shown later. However, proportionate selection mechanisms have other problems that often preclude them from being used, such as a much slower convergence rate than ordinal-based methods, and the scaling problem (Forrest, 1985; Goldberg, 1989a; Whitley, 1989).

The other major effect is that given the increased noise in the selection process, the population needs a larger supply of building blocks to prevent premature convergence. The population size needs to be increased to compensate for the increased noise level in the selection process, which will incorrectly “lose” a lot of good building blocks as a result of the selection criteria being based on the noisy fitness values. The fitness function noise thus affects both the convergence characteristics of the GA and the population sizing requirements.

#### A.4.3 Sampling Fitness Functions

Sampling fitness functions repeatedly evaluate an individual with a noisy fitness function, and return the average fitness score. They are useful when the fitness noise is randomly distributed, and thus repeated evaluations of the same individual by the noisy fitness function will (probably) have different fitness scores. Sampling can be used to reduce the noise in the fitness value for an individual. According to the Central Limit Theorem, the **mean** of  $n$  samples from a distribution with a mean  $\mu$  and variance  $\sigma^2$  is distributed  $N(\mu, \frac{\sigma^2}{n})$ . It is important to note that the original distribution does not have to be normally distributed for the distribution of the mean to be normal. A straight-forward application of the Central Limit Theorem shows that the mean of  $n$  samples of a noisy fitness function distributed  $N(\mu_{F,t}, \sigma_{F,t}^2 + \sigma_N^2)$  is itself distributed

$$N(\mu_{F,t}, \frac{\sigma_{F,t}^2 + \sigma_N^2}{n}).$$

### A.5 Selection

The selection mechanism uses the fitness values to determine which individuals should be kept. The individuals kept by the selection process can be referred to as the mating pool, as they become the parents for the next generation. The individuals not selected as part of the mating pool are “killed,” as they are not used to breed the next generation of individuals, nor are they retained in the next generation. There are many different selection mechanisms, which are discussed in detail in section 2.1.1.

## A.6 Recombination and Mixing

Recombination uses the mating pool as parents of the next generation. Recombination combines the traits of parents, perhaps randomly, to form new individuals, known as offspring. This process is also referred to as “mixing,” as it essentially mixes the traits in the mating pool to form new combinations of traits in the offspring. There are several popular recombination methods, including uniform crossover, one-point crossover, two-point crossover, and N-point crossover; see Goldberg (1989a) for a description of these crossover operators.

## A.7 Mutation

Mutation is the process of randomly altering a part of an individual to produce a new individual. As in nature, most of the time a mutation results in worse solution. However, occasionally a mutation will have a beneficial effect, enabling the individual to be more likely to survive. For most genetic algorithms, the mutation operator has a relatively minor role compared to the recombination operator.

## A.8 Stopping Criteria

There are many different ways to determine when to stop running the GA and return the best solution(s). One method is to stop after a preset number of generations or time-limit. Another is to stop after the GA has converged - that is, all individuals in the population are identical. Note that convergence will only happen if the the mutation operator is not used. The GA can also be halted if the solution quality of the population does not improve by more than a specified amount in a specified number of generations.



# Bibliography

- Abramovitz, M., & Stegun, I. (Eds.) (1972). *Handbook of mathematic functions with formulas, graphs, and mathematical tables*. Dover Publications.
- Aizawa, A. (1996). Fitness landscape characterization by variance of decompositions. In Belew, R. K., & Vose, M. (Eds.), *Foundations of Genetic Algorithms IV* (pp. 390–400). San Mateo, CA: Morgan Kaufmann. Proceedings Preprint.
- Aizawa, A. N., & Wah, B. J. (1994a). Scheduling of genetic algorithms in a noisy environment. *Evolutionary Computation*, 2(2), 97–122.
- Aizawa, A. N., & Wah, B. W. (1994b). A sequential sampling procedure for genetic algorithms. *Computers and Mathematics with Applications (Also in Proceedings of the Fifth International Workshop of the Bellman Continuum '93)*, 27(9/10), 77–82.
- Bäck, T. (1995). Generalized convergence models for tournament- and  $(\mu, \lambda)$ -selection. In Eschelman, L. (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 Grefenstette, J. J. (Ed.), *Proceedings of an International Conference on Genetic Algorithms and Their Applications* (pp. 101–111). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. In Grefenstette, J. J. (Ed.), *Proceedings of the Second International Conference on Genetic Algorithms* (pp. 14–21). Hillsdale, NJ: Lawrence Erlbaum Associates.
- 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*. Doctoral dissertation, The University of Michigan. (University Microfilms No. 8214966).
- Brindle, A. (1981). *Genetic algorithms for function optimization*. Unpublished doctoral dissertation, University of Alberta, Edmonton, Canada.
- Davis, L. (Ed.) (1991). *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold.
- Deb, K., & Goldberg, D. E. (1993). Analyzing deception in trap functions. In Whitley, L. D. (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, MI.
- Forrest, S., & Mitchell, M. (1993). Relative building-block fitness and the building-block hypothesis. In Whitley, L. D. (Ed.), *Foundations of Genetic Algorithms 2* (pp. 109–126). San Mateo, CA: Morgan Kaufmann.

- Goldberg, D. E. (1989a). *Genetic algorithms in search, optimization, and machine learning*. New York: Addison-Wesley.
- Goldberg, D. E. (1989b). Sizing populations for serial and parallel genetic algorithms. In Schaffer, J. D. (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 70–79). San Mateo, CA: Morgan Kaufmann. (Also TCGA Report 88004).
- Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In Rawlins, G. J. E. (Ed.), *Foundations of Genetic Algorithms* (pp. 69–93). San Mateo, CA: Morgan Kaufmann.
- Goldberg, D. E., Deb, K., & Clark, J. H. (1992). Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6, 333–362.
- Goldberg, D. E., Deb, K., & Clark, J. H. (1993). Accounting for noise in the sizing of populations. In Whitley, L. D. (Ed.), *Foundations of Genetic Algorithms 2* (pp. 127–140). San Mateo, CA: Morgan Kaufmann.
- Goldberg, D. E., Deb, K., & Horn, J. (1992). Massive multimodality, deception, and genetic algorithms. In Männer, R., & Manderick, B. (Eds.), *Parallel Problem Solving from Nature*, 2 (pp. 37–46). Amsterdam: Elsevier Science.
- 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.
- Goldberg, D. E., & Rudnick, M. (1991). Genetic algorithms and the variance of fitness. *Complex Systems*, 5(3), 265–278. (Also IlliGAL Report No. 91001).
- Grefenstette, J. J., & Baker, J. E. (1989). How genetic algorithms work: A critical look at implicit parallelism. In Schaffer, J. D. (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 20–27). San Mateo, CA: Morgan Kaufmann.
- Grefenstette, J. J., & Fitzpatrick, J. M. (1985). Genetic search with approximate function evaluations. In Grefenstette, J. J. (Ed.), *Proceedings of an International Conference on Genetic Algorithms and Their Applications* (pp. 112–120). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Hammersley, J. M., & Handscomb, D. C. (1964). *Monte Carlo methods*. London: Methuen & Co.
- Hans, A. E. (1988). Multicriteria optimization for highly accurate systems. In Stadler, W. (Ed.), *Multicriteria Optimization in Engineering and Sciences: Mathematical concepts and methods in science and engineering*, Volume 19 (pp. 309–352). New York: Plenum Press.
- Harik, G., Cantú-Paz, E., Goldberg, D. E., & Miller, B. L. (1997). The gambler’s ruin problem, genetic algorithms, and the sizing of populations. In *1996 IEEE International Conference on Evolutionary Computation* (pp. 7–12). Piscataway, NJ: IEEE Service Center.
- Harter, H. L. (1970). *Order statistics and their use in testing and estimation*, Volume 2: Estimates Based on Order Statistics of Samples from Various Populations. Washington, D.C.: U.S. Government Printing Office.

- Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.
- Law, A. M., & Kelton, W. D. (1991). *Simulation modeling and analysis* (2nd ed.). New York: McGraw-Hill.
- Miller, B. L., & Goldberg, D. E. (1996a). Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary Computation*, 4(2), 113–131.
- Miller, B. L., & Goldberg, D. E. (1996b). Genetic algorithms, tournament selection, and the varying effects of noise. *Complex Systems*, 9(3), 193–212.
- Miller, B. L., & Goldberg, D. E. (1996c). Optimal sampling for genetic algorithms. In Dagli, C. H., Akay, M., Chan, C. L. P., Fernández, B. R., & Ghosh, J. (Eds.), *Intelligent engineering systems through artificial neural networks (ANNIE '96)*, Volume 6 (pp. 291–298). New York: ASME Press.
- Mitchell, M. (1996). *An introduction to genetic algorithms*. Cambridge, MA: MIT Press.
- 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.
- Srinivas, N., & Deb, K. (1995). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3), 221–248.
- Syswerda, G. (1989). Uniform crossover in genetic algorithms. In Schaffer, J. D. (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 Forrest, S. (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 Davidor, Y., Schwefel, H.-P., & Männer, R. (Eds.), *Parallel Problem Solving from Nature, PPSN III* (pp. 119–129). Berlin: Springer-Verlag.
- Whitley, D. (1989). The GENITOR algorithm and selective pressure: Why rank-based allocation of reproductive trials is best. In Schaffer, J. D. (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 116–121). San Mateo, CA: Morgan Kaufmann.
- Winter, G., Périaux, J., Galán, M., & Cuesta, P. (Eds.) (1995). *Genetic algorithms in engineering and computer science*. Chichester: John Wiley and Sons.