Optimal Sampling of Genetic Algorithms on Polynomial Regression

Tian-Li Yu
Taiwan Evolutionary Intelligence Laboratory
Department of Electrical Engineering
National Taiwan University
No.1, Sec. 4, Roosevelt Rd., Taipei, Taiwan
tianliyu@cc.ee.ntu.edu.tw

Wei-Kai Lin
Taiwan Evolutionary Intelligence Laboratory
Department of Electrical Engineering
National Taiwan University
No.1, Sec. 4, Roosevelt Rd., Taipei, Taiwan
b92203051@ntu.edu.tw

ABSTRACT

This paper investigates the utility of sampling as an evaluation-relaxation technique in genetic algorithms (GAs). In many real-world applications, sampling can be used to generate a less accurate, but computationally inexpensive fitness evaluator to speed GAs up. This paper focuses on the problem of polynomial regression as an example of problems with positive dependency among genes. Via statistical analysis of the noise introduced by sampling, this paper develops facet-wise models for the optimal sampling size, and these models are empirically verified. The results show that when the population is sized properly, small sampling sizes are preferred for most applications. When a fixed population size is adopted, which is usually the case in real-world applications, an optimal sampling size exists. If the sampling size is too small, the sampling noise increases, and GAs would perform poorly because of an insufficiently large population. If the sampling size is too large, the GA would spend too much time in fitness calculation and cannot perform well either within limited run duration.

Categories & Subject Descriptors

G.1.6 [Mathematics of Computing]: Global Optimization–Analyze.

General Terms

Algorithms, Theory.

Keywords

Optimal Sampling, Genetic Algorithms, Polynomial Regression, Fitness Relaxation, Function Mapping, Speedup Technique.

1. INTRODUCTION

Over last few decades, significant progress has been made in the theory, design and application of genetic and evolutionary algorithms. A decomposition design theory has been proposed and several *competent* genetic algorithms (GAs) have been developed [11], which aim to solve boundedly difficult problems within a sub-quadratic number of function evaluations.

However, in real-world problems, even a sub-quadratic number of function evaluations can be intractable. Therefore, a number of efficiency enhancement techniques (EETs) have been proposed to alleviate the computational burden. One such technique is evaluation relaxation [22], where an accurate, costly fitness evaluator is replaced by an inexpensive, less accurate one. Partial evaluation through sampling is an example of evaluation relaxation, which has been empirically shown to yield a significant speed-up [13]. Evaluation relaxation through sampling has also been analyzed by developing facet-wise and dimensional models [18, 7, 25]. Apparently contradicting with Grefenstette and Fitzpatrick's results [13], those models indicate that sampling does not yield speed-up in terms of number of samples for problems with uniform salience. Later, Yu *et. al* [24] proposed an adaptive sampling scheme which yields speed-up for problems with nonuniform salience. However, Grefenstette and Fitzpatrick's image registration problem is with uniform salience, and hence the speed-up in their results are still not yet understood.

The objective of this paper is to resolve the above paradox by investigating the effect of sampling on both convergence time and population sizing of GAs. Specifically, the problem of polynomial regression is studied. The remainder of this paper is composed of four primary parts: (1) background knowledge including previous work related to sampling for GAs, introductions to population sizing, convergence time, and fitness relaxation, (2) facet-wise model development for sampling in polynomial regression, (3) empirical results and discussions, and (4) extensions and conclusions of this work.

2. PREVIOUS WORK

Grefenstette and Fitzpatrick achieved a great success in applying sampling techniques to the image registration problem [13]. Their goal was to map a slightly distorted image to its original. The mapping function they used was a simple

[†] Research supported by the National Science Council in Taiwan under grant no. NSC96-3114-P-002-012-Y and NSC97-2218-E-002-020-MY3.

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
Copyright 2008 ACM 978-1-60558-130-9/08/07…$5.00.

2D non-linear function:

$$\begin{cases} x' = & a_0 + a_1 x + a_2 y + a_3 xy \\ y' = & a_4 + a_5 x + a_6 y + a_7 xy \end{cases} \tag{1}$$

The dimension of the images is 100 by 100, and hence 10000 pixels in total. They used a GA to optimize the eight parameters $a_0, a_1, \cdots, a_7$. The fitness function was the summation of the differences in each pixel. Instead of calculating the differences for all 10000 pixels, they randomly sample some pixels and calculate their differences only:

$$\Sigma_{(x,y)\in S}\ |m_1(x,y) - m_2(x',y')|, \tag{2}$$

where S is the set of those samples, m_1 and m_2 are the original and the distorted images respectively, and (x', y') is the mapped pixel using the above mapping function. Their experiments indicated that the GA performed best when $|S| = 8 \sim 10$.

Grefenstette and Fitzpatrick's success motivated the investigation of the use of sampling techniques in both the fields of GAs and evolutionary strategies [2, 14]. Aizawa and Wah studied sampling techniques as scheduling problems in GAs [1]. Specifically, they investigated how many samples that GAs should spend on each generation and each individual in a noisy environment. They also proposed an adaptive scheme when the computational cost and noise are unknown.

Under the assumption that the computational cost and noise are given, Miller *et al.* investigated the optimal number of sampling. The problem is OneMax where the fitness function is clouded by a zero-mean Gaussian noise [19]:

$$f'(\vec{x}) = f(\vec{x}) + N(0, \sigma_N^2), \tag{3}$$

where σ_N^2 is the variance of the noise. They considered calculating fitness by multiple sampling to reduce the external noise:

$$f'_s(\vec{x}) = \frac{1}{s}\Sigma_{i=1}^{s}\ f'(\vec{x}), \tag{4}$$

where s is the number of samples. Assume the cost of the evaluation of $f'(x)$ is β. Sampling reduces the variance of the noise while increases the evaluation cost.

$$\text{variance: } \frac{\sigma_N^2}{s} \tag{5}$$

$$\text{cost: } s\beta \tag{6}$$

Assuming an overhead, optimal sampling can be derived:

$$s^* = \sqrt{\frac{\alpha}{\beta}\frac{\sigma_N^2}{\sigma_F^2}}, \tag{7}$$

where α is the overhead, *i.e.*, the total cost of s samples is $(\alpha + s\beta)$, and σ_F^2 is the variance of the original fitness. This result is later theoretically verified by Sastry and Goldberg [22].

The problem models in both [1] and [19] are different from Grefenstette and Fitzpatrick's image registration problem. In the image registration problem, the sampling noise was endogenous and its variance came to be zero when the sampling size is so long as the chromosome length. In both [1] and [19], the sampling noise was exogenous. The variance of noise will never be zero no matter how large the sampling size is.

To better understand the behavior of endogenous noise, Giguère and Goldberg [7] and Yu *et al.* [25] investigated a problem called the sampling OneMax (SOM):

$$f_s(\vec{x}) = \frac{\ell}{s}(\Sigma_{i\in S}\ x_i), \tag{8}$$

where s is the sampling size $(0 < s \leq \ell)$, S is a subset of $\{1, 2, \cdots, \ell\}$ with a restriction that $|S| = s$, and ℓ is the chromosome length. The term $\frac{\ell}{s}$ is for normalization so that the expectation of the sampled fitness $f_s(\vec{x})$ is the same as the original fitness $f(\vec{x})$.

SOM has an special property: partial-string-partial-evaluation (PSPE), which is essential for studying endogenous noise. Although the assumption of PSPE may not be suitable for problems like deceptions or trap [8, 6], it is true for image registration, regression (both numeric or symbolic [16]), and function mapping.

It has been shown that if population is sized properly according to the gambler's ruin model [15], sampling does not make much difference; it gives speed-up about only $\frac{1}{\ell}$ in the best case [25]. When the population size is fixed, GAs prefer a sampling size as small as possible $(s = 1)$ on SOM. Both results did not reveal an optimal sampling size somewhere in between 1 and ℓ, and Grefenstette and Fitzpatrick's results were still not yet understood.

We believe that the investigation on SOM did not explain Grefenstette and Fitzpatrick's results because SOM is still different from the image registration problem in terms of the dependency among samples. In SOM, every sample is independent, *e.g.*, a bit being one or zero does not affect another bit being one or zero. However, images are usually smooth, and hence the values of pixels usually depend on each other. For example, in a typical 8-bit grayscale image, the neighbors of a pixel of value 255 are rarely 0.

To reduce the difference between modeling and the image registration problem, we focus on the problem of polynomial regression, where data points are believed to be related. We will describe the problem in detail later.

3. BACKGROUND

This section provides background knowledge for readers to easily go through this paper. Specifically, it gives introductions to population sizing, convergence time, and fitness relaxation.

3.1 Population Sizing

In the real world, the running time of a GA is bounded. In this situation, finding a proper population size is important. Because an undersized population offers poor supply, while an oversized population consumes more time for each generation. It has been shown that both undersized and oversized population reduce the solution quality [19]. Therefore, having an accurate population-sizing model becomes urgent for real-world applications.

Goldberg *et al.* [12] proposed a population-sizing model based on decision-making arguments. Their decision-making model can be expressed as follows:

$$N = \Gamma \cdot \sigma_F^2, \tag{9}$$

where N is the population size, Γ is the population coefficient as defined in [19], and σ_F^2 is the fitness variance.

The decision-making model assumes that if an incorrect decision is made in the first generation, GAs are unable to recover from the error. Harik *et al.* [15] refined the decision-making model by incorporating cumulative effects of decision making over time rather than in first generation only. They modeled the decision making between the correct schema and its strongest competitor in a partition as a gambler's ruin problem. Miller expressed their gambler's ruin model as the following [17]:

$$N = \Gamma' \cdot \sigma_F \ln(\psi), \tag{10}$$

where N and σ_F^2 are the same as those in equation (9), Γ' is another coefficient, and ψ is the failure rate, defined as the probability that a particular partition in the chromosome fails to converge to the correct alleles. In other words, $(1 - \psi)$ is the expected proportion of the correct alleles in an individual.

In many theoretical analyses, the failure rate in OneMax domain is usually set to be $\frac{1}{\ell}$ ($\frac{1}{2\ell}$ in [7]), where ℓ is the chromosome length. In this case, the expected solution quality of the OneMax problem is $(\ell - 1)$.

3.2 Convergence Time

Mühlenbein and Schlierkamp-Voosen [20] gave the following convergence-time model for OneMax by assuming an infinite population size and perfect mixing.

$$t_{conv} = \left(\frac{\pi}{2} - \arcsin(2p_0 - 1)\right)\frac{\sqrt{\ell}}{I}, \tag{11}$$

where p_0 is the initial proportion of ones for OneMax problem, I is selection intensity. The convergence-time model can be derived directly from the definition of selection intensity [20, 23]:

$$I = \frac{f_{t+1} - f_t}{\sigma_{f_t}}, \tag{12}$$

where f_t is the average fitness at generation t, and $\sigma_{f_t}^2$ is the fitness variance at generation t. The above equation can be written as

$$f_{t+1} - f_t = I \cdot \sigma_{f_t}. \tag{13}$$

Blickle and Thiele [3] showed that for the tournament selection with a fixed selection pressure, the selection intensity is independent of t. Therefore, the fitness improvement depends mainly on the fitness variance.

Note that although these convergence-time models are derived under the assumption of infinite population size, experiments have shown that they are good approximations for a large enough population [22]. Readers who are interested in convergence time with a finite population are referred to [21, 4].

3.3 Evaluation Relaxation

Sastry and Goldberg [22] investigated evaluation relaxation for problems with uniform salience. They derived models of the number of function evaluations separately for two noise sources: variance and bias. Since sampling techniques usually only introduce variance-type noises, the work on bias-type noises will not be mentioned here. The idea of their work is to determine which one to use between two fitness functions with different costs and difference variances of noises. The decision, of course, should be made in the favor of shorter run duration under a fixed solution quality.

In their work, they adjusted the population-sizing and convergence time models for fitness relaxation. When the fitness function is relaxed, the fitness variance increases. For unbiased noises, the fitness variance can be modeled as

$${\sigma'}_F^2 = \sigma_F^2 + \sigma_N^2, \tag{14}$$

where σ_F^2 is the variance of original fitness, and σ_N^2 is the variance of the noise introduced by fitness relaxation.

Recall that both the population size and convergence time increase with the fitness variance increasing. The population size and convergence time can be modeled as:

$$N' = N\sqrt{\frac{\sigma_F^2 + \sigma_N^2}{\sigma_F^2}}, \tag{15}$$

$$t'_{conv} = t_{conv}\sqrt{\frac{\sigma_F^2 + \sigma_N^2}{\sigma_F^2}}, \tag{16}$$

where N and t_{conv} are the original population size and convergence time respectively, and N' and t'_{conv} are those after relaxation.

4. SAMPLING ON POLYNOMIAL REGRESSION

In this paper, we investigate the problem of polynomial regression as an example for problems with uniform salience and positive dependencies among samples. A typical polynomial regression can be described as follows [5]. Consider a polynomial function

$$f(x) = \Sigma_{i=0}^m a_i x^i, \tag{17}$$

where m is the degree of the polynomial. Given a set of data points $D = (x_i, y_i)$, the objective of regression is to find appropriate polynomial coefficients $a_0, a_1, \cdots, a_m$ such that

$$\Sigma_{(x,y)\in D} |f(x) - y| \tag{18}$$

is minimal. Note that every sample contributes equally to the summation, which validates the assumption of uniform salience. It is also not difficult to see the samples are positively depend on each other. Assume that these data points come from some underlying polynomial of degree m: $y_i = g(x_i)$. When $|f(x_i) - y_i|$ are small for many i's, it is highly probable that $f \simeq g$. In other words, $d(x) = g(x) - f(x)$ has small coefficients. Therefore, for any other samples j, $d(x_j)$ should also be small. Formal proofs are omitted since they are beyond the scope of this paper.

For the ease of analysis, the mean squared error (MSE) criterion is more often used than the absolute value: $\frac{1}{|D|}\Sigma_{(x,y)\in D} (f(x)-y)^2$. To alleviate the computational burden, one may consider computing the squared errors over a small, sampled data set instead of the whole one:

$$f_S = \frac{1}{s}\Sigma_{(x,y)\in S} (f(x) - y)^2, \tag{19}$$

where S is the sampled data set, $S \subseteq D$, and $|S| = s$.

With the sampling technique, computing the sampled fitness for one individual becomes faster. However, the population size and convergence time would be elongated accordingly (Equations 15 and 16). To determine if it is really beneficial to adopt sampling, we need to investigate the distributions of the original and the sampled fitness, which is addressed in the next section.

5. VARIANCE OF THE ORIGINAL AND THE SAMPLED FITNESS

The main goal of this section is to derive the variances of the original and the sampled fitness.

Here we make some assumptions to simplify the derivations. First we assume that the x_i's in the given data set are uniformly distributed within the range of $(0, R]$ with a resolution of r. In other words, $x_i = \frac{iR}{r}$ for $i = 1, 2, \cdots, r$. We then further assume that y_i's comes from some underlying polynomial: $y_i = g(x_i) = \Sigma_{j=0}^{m} b_j x_i^j$. This scenario is similar to function mapping with finite observations. With this setting, we know that the optimal solution is $a_i = b_i$, $\forall i \in \{0, 1, \cdots, m\}$.

Define $d(x) = f(x) - g(x) = \Sigma_{j=0}^{m} d_j x^j$, where $d_j = a_j - b_j$. MSE can be written as

$$MSE = \frac{1}{r} \Sigma_{i=1}^{r} d^2(x_i). \tag{20}$$

The sampled MSE can be written as

$$MSE_S = \frac{1}{|S|} \Sigma_{i \in S} d^2(x_i), \tag{21}$$

where $S \subseteq \{1, 2, \cdots, r\}$. MSE is a random variable where d_j's are uncertain, and MSE_S is also a random variable where both d_j's and S are uncertain.

Recall that the sampled fitness variance can be modeled as the summation of the original fitness variance and the variance of sampling noise (Equation 14). We have the following relations:

$$\begin{aligned} V[MSE] &= \sigma_F^2, & (22) \\ V[MSE_S] &= \sigma_F^2 + \sigma_N^2, & (23) \end{aligned}$$

where V[.] denotes the variance of the parenthesized random variable.

Two variables would come to be handy and are defined here.

$$\begin{aligned} O_1 &= \Sigma_{i=1}^{r} h^2(x_i), & (24) \\ O_2 &= \left(\Sigma_{i=1}^{r} h(x_i)\right)^2, & (25) \end{aligned}$$

where $h(x) = d^2(x)$.

The remainder of this section is organized as follows: (1) we first show that both the fitness variance and the variance of sampling noises variance can be expressed by these two variables, and then (2) we derive O_1 and O_2 by assuming that d_j are uniformly distributed over a small range of $[-\delta, \delta]$. Combing (1) and (2), the derivation of the variances is completed, and it is then empirically verified.

5.1 Expressing Variances by O_1 and O_2

Now we derive σ_F^2 and σ_N^2. Since MSE and MSE_S have different dimensions of uncertainties, we use subscriptions under $V[.]$ and $E[.]$ to indicate the variance and expectation over some particular uncertainty, respectively.

It is easily seen that $MSE = \frac{1}{r}\sqrt{O_2}$. By definition, $V_{d_j}[MSE] = E_{d_j}[(MSE - E_{d_j}[MSE])^2] = E_{d_j}[MSE^2] - (E_{d_j}[MSE])^2$. Therefore, σ_F^2 can be then expressed as

$$\sigma_F^2 = V_{d_j}[MSE] = \frac{1}{r^2}\left(E_{d_j}[O_2] - (E_{d_j}[\sqrt{O_2}])^2\right). \tag{26}$$

Note that the expectation values are the arithmetic means over all possible d_j's.

The derivation of σ_N^2 is slightly more complicated since MSE_S is a random variable where both d_j's and S are uncertain. We derive σ_N^2 in a 2-stage manner. In the first stage, assuming d_j's are given, we compute the variance of MSE_S overall possible S's. Then in the second stage, we average the variances over all possible d_j's to retrieve σ_N^2. In other words, instead of using Equation 23 to compute σ_N^2, we use the following relation:

$$\sigma_N^2 = E_{d_j}\left[V_S[MSE_S]\right]. \tag{27}$$

This subsection focuses on the first stage, and the next subsection will complete the second stage.

We know that the expectation of MSE_S over all possible S's should be unbiased: $E_S[MSE_S] = MSE = \frac{\sqrt{O_2}}{r}$. There are totally C_s^r sets of size s for a given s. Number them as $S_1, S_2, \cdots, S_{C_s^r}$. [1] The variance of MSE_S can be expressed as

$$\begin{aligned} V_S[MSE_S] &= E_S[MSE_S^2] - (E_S[MSE_S])^2 \\ &= \frac{\Sigma_i \, (MSE_{S_i})^2}{C_s^r} - \frac{O_2}{r^2}. & (28) \end{aligned}$$

Take a closer look at the term $\Sigma_i \, MSE_{S_i}^2$. $MSE_{S_i}^2$ is of the formation: $\left[\frac{1}{s}(\cdots + h(x_u) + h(x_v) + \cdots)\right]^2$, where x_u and x_v are two distinct sampling points in set S_i. By power expansion, the above formula can be written as $\frac{1}{s^2}\left[\cdots + h^2(x_u) + h^2(x_v) + \cdots + 2h(x_u)h(x_v) + \cdots\right]$. If we consider all possible S_i's, there are C_{s-1}^{r-1} sets containing the term $h^2(x_u)$ for any specific u; similarly, there are C_{s-2}^{r-2} sets containing the term $h(x_u)h(x_v)$ for any specific u and v where $u \neq v$. Therefore, $\Sigma_i \, MSE_{S_i}^2$ can be expressed as

$$\Sigma_i \, MSE_{S_i}^2 = \frac{1}{s^2}\left(C_{s-1}^{r-1}\Sigma_{u=1}^{r} h^2(x_u) + C_{s-2}^{r-2}\Sigma_{u \neq v} h(x_u)h(x_v)\right). \tag{29}$$

With algebra manipulations, we can rewrite the above equation as:

$$\begin{aligned} \Sigma_i \, MSE_{S_i}^2 = \; & \tfrac{1}{s^2}\big((C_{s-1}^{r-1} - C_{s-2}^{r-2})\Sigma_{u=1}^{r} h^2(x_u) \\ & + C_{s-2}^{r-2}\Sigma_{u=1}^{r} \Sigma_{v=1}^{r} h(x_u)h(x_v)\big). & (30) \end{aligned}$$

Since $C_{s-1}^{r-1} - C_{s-2}^{r-2} = C_{s-1}^{r-2}$ and the double summation is actually a perfect square, we can express $\Sigma_i \, MSE_{S_i}^2$ as:

$$\Sigma_i \, MSE_{S_i}^2 = \frac{1}{s^2}\left(C_{s-1}^{r-2}O_1 + C_{s-2}^{r-2}O_2\right). \tag{31}$$

By substituting Equation 31 into Equation 28, we can simplify the variance of MSE_S as:

$$V_S[MSE_S] = \frac{r-s}{rs(r-1)}(O_1 - \frac{1}{r}O_2). \tag{32}$$

5.2 Computing O_1 and O_2

Recall that $h(x) = d^2(x) = \left(\Sigma_{j=0}^{m} d_j x^j\right)^2$ and that x is within the range of $(0, R]$. For a large R and large x, the leading term dominates. Therefore, $h(x)$ and $h^2(x)$ can be approximated as follows.

$$\begin{aligned} h(x) &\simeq d_m^2 x^{2m}. & (33) \\ h^2(x) &\simeq d_m^4 x^{4m}. & (34) \end{aligned}$$

[1] C_s^r is the notation of combination, which is read as "r choose s."

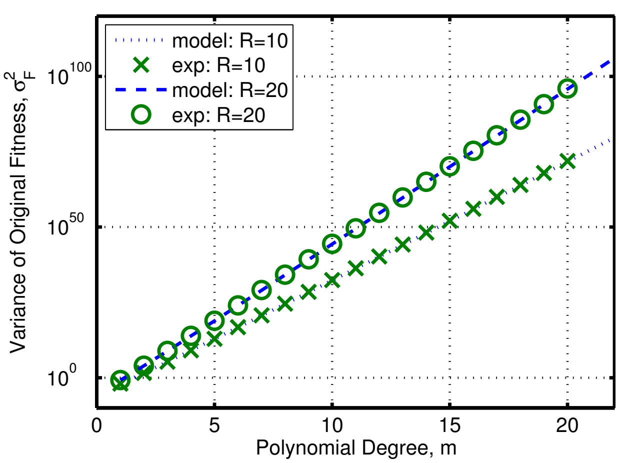

Figure 1: Variance of the original fitness versus different polynomial degrees (m) for $R = 10$ and $R = 20$. δ is set at 0.1, and resolution (r) is set at 100.

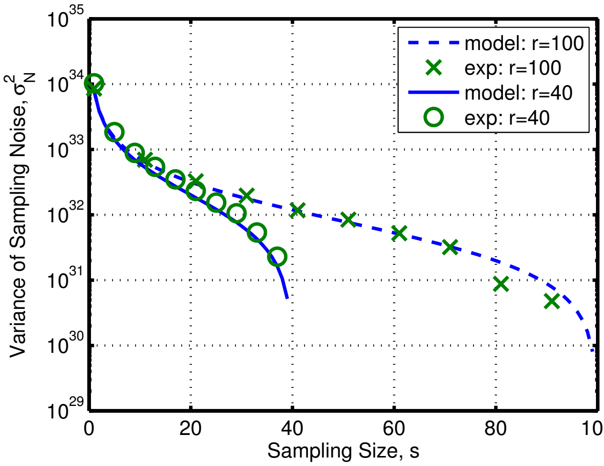

Figure 2: Variance of sampling noise versus different sampling size for $r = 40$ and 100. δ is set at 0.1, and R is set at 100.

Given these approximations, O_1 and O_2 can be approximated as

$$O_1 \quad = \Sigma_i\, h^2(x_i) \quad \simeq d_m^4 \Sigma_i\, x_i^{4m}. \tag{35}$$

$$O_2 \quad = (\Sigma_i\, h(x_i))^2 \quad \simeq d_m^4 \left(\Sigma_i\, x_i^{2m}\right)^2. \tag{36}$$

Both approximations involve the calculation of $\Sigma_i\, x_i^t$. Recall that $x_i = \frac{iR}{r}$. Here we use integral to approximate the summation:

$$\Sigma_i\, x_i^t \simeq \frac{rR^t}{t+1}. \tag{37}$$

Finally, we get

$$O_1 \simeq d_m^4 \frac{rR^{4m}}{4m+1}. \tag{38}$$

$$O_2 \simeq d_m^4 \frac{r^2 R^{4m}}{(2m+1)^2}. \tag{39}$$

$$\sqrt{O_2} \simeq d_m^2 \frac{rR^{2m}}{2m+1}. \tag{40}$$

Assuming that d_j's are uniformly distributed over a small range of $[-\delta, \delta]$, we then calculate their expectations:

$$E_{d_j}[O_1] \simeq \frac{\delta^4}{5} \frac{rR^{4m}}{4m+1}. \tag{41}$$

$$E_{d_j}[O_2] \simeq \frac{\delta^4}{5} \frac{r^2 R^{4m}}{(2m+1)^2}. \tag{42}$$

$$E_{d_j}[\sqrt{O_2}] \simeq \frac{\delta^2}{3} \frac{rR^{2m}}{2m+1}. \tag{43}$$

5.3 Putting It All Together

Given the approximations of the expectations of O_1 and O_2, we can derive the variances of fitness and sampling noises.

$$\begin{aligned} \sigma_F^2 &= \frac{1}{r^2}\left(E_{d_j}[O_2] - (E_{d_j}[\sqrt{O_2}])^2\right) \\ &\simeq \frac{4}{45} \frac{\delta^4 R^{4m}}{(2m+1)^2}. \end{aligned} \tag{44}$$

$$\begin{aligned} \sigma_N^2 &= E_{d_j}\left[V_S[MSE_S]\right] \\ &= \frac{r-s}{rs(r-1)}(E_{d_j}[O_1] - \frac{1}{r}E_{d_j}[O_2]) \\ &\simeq \frac{(r-s)\delta^4 R^{4m}}{5s(r-1)}\left(\frac{1}{4m+1} - \frac{1}{(2m+1)^2}\right). \end{aligned} \tag{45}$$

Experiments are conducted to verify the above derivations. In these experiments, every data point is averaged over 10^5 randomly generated chromosomes. Figure 1 shows the variance of the original fitness, σ_F^2, versus different polynomial degrees, m, for $R = 10$ and $R = 20$. δ is set at 0.1, and resolution, r, is set at 100.

Figure 2 shows the variance of sampling noises, σ_N^2, for different resolutions, r, and sampling sizes, s. As expected, σ_N^2 tends to zero when $s = r$. Basically, both models match the experimental data pretty well.

6. OPTIMAL SAMPLING

In this section, we investigate the effect of sampling noise on GA running time for different population-sizing scenarios. Specifically, we investigate the optimal sampling size (1) when population is properly sized, and (2) when a fixed population size is used. We firstly adopt the population sizing model and convergence time model under fitness relaxation [22], and investigate the optimal sampling size such that the overall GA running time is minimal. Then we investigate the same thing but with a fixed population size. The second scenario is more realistic for most real-world applications since many parameters are unknown in the population-sizing model. Finally, we explain Grefenstette and Fitzpatrick's results from the findings of our models and experiments.

6.1 Optimal Sampling When Population Is Properly Sized

From Equations 15 and 16, we can model the total number

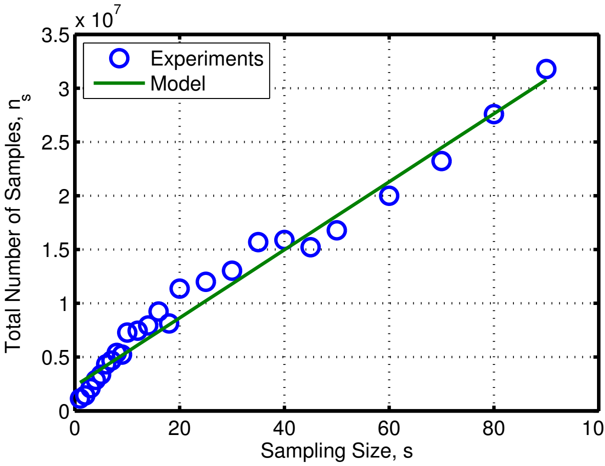

Figure 3: Total number of samples versus different sampling size. The underlying objective function is $x^3 + x^2 + x + 1$. The number of data points is 100. The optimal sampling size is one.

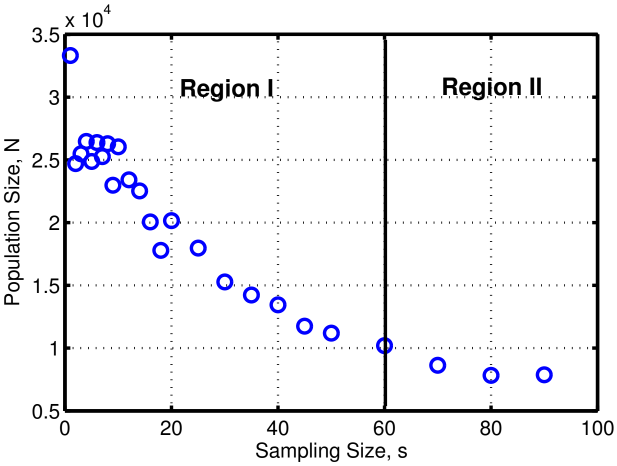

Figure 4: Population size versus sampling size. $R = 1000$, $r = 100$, $m = 3$. $N \simeq 10000$ for $s = 60$. If a fixed population size $N = 10000$ is used, GAs in region I ($s < 60$) would suffer from insufficient population size, and GAs in region II ($s > 60$) would be less efficient than $s = 40$ since small sampling sizes are preferred.

of samples that a GA requires as:

$$n_s = sNt_{conv}\left(1+\frac{\sigma_N^2}{\sigma_F^2}\right). \qquad (46)$$

Note that the minimal of s is 1 for GAs to work. With Equations 44 and 45, we can approximate the ratio:

$$\begin{aligned}\frac{\sigma_N^2}{\sigma_F^2} &\simeq \frac{9}{4}\frac{r-s}{s(r-1)}\frac{4m^2}{4m+1} \\ &\simeq \frac{9}{4}\frac{r-s}{s(r-1)}m. \qquad (47)\end{aligned}$$

The second approximation is valid for a large m. By substituting Equation 47 into Equation 46, and discarding irrelative terms to the sampling size, we obtain the following relation:

$$n_s \propto s + \frac{9}{4}\frac{r-s}{r-1}m, \qquad (48)$$

which is a simple linear relation. Therefore, the optimum occurs at the boundaries, either $s = 1$ or $s = r$. We can check when the optimum occurs at the minimal sampling size:

$$\begin{aligned} & n_s(s=1) < n_s(s=r) \\ \Rightarrow \quad & m < \tfrac{4}{9}(r-1). \qquad (49)\end{aligned}$$

This is the case for most polynomial regression applications, where usually $m \ll r$ to prevent over-fitting since there are only r data points. The exception is when the objective polynomial is not full. For example, one might want to fit $ax^{50}+bx^{13}+c$ to 20 data points. In this case, the polynomial degree, m, is 50, and r is only 20, which breaks the above relation. However, in this case, our derivations are no longer valid since our assumptions are violated.

A series of experiments are conducted to verify the conclusion drawn from the analytical model. To get the appropriate population size, a procedure of bisection is invoked. Given the minimal and maximal limits, the procedure seeks the minimal population size such that the GA are able to converge for 30 out of 30 times. In the experiment, $R = 1000$, $r = 100$, m is set at 3, and the objective b_j's are set at 1. In other words, the underlying objective function is $x^3 + x^2 + x + 1$. A chromosome is a vector of 4 real numbers. Binary tournament selection and extended line crossover [20] are adopted, and there is no mutation. The termination criterion is that all alleles in the best chromosome are within the range $[1-10^{-3}, 1+10^{-3}]$. The result is shown in Figure 3. As predicted, the optimal sampling size is $s = 1$.

6.2 Optimal Sampling For A Fixed Population Size

The previous section indicated that if population is sized properly, a small sampling size is preferred. In reality, when facing an unknown problem, it is virtually impossible for one to determine an appropriate population size. Instead, a population with fixed size is usually used.

Figure 4 shows the population sizes of the experiment in Figure 3, where $R = 1000$, $r = 100$, $m = 3$. Note that $n \simeq 10000$ for $s = 60$. Now suppose that a fixed population size of 10000 is used. Those GAs with a sampling size smaller than 60 (region I) would suffer from insufficient information to make good decision [12], and hence would perform poorly. On the other hand, those GAs with a sampling size greater than 60 (region II) would require more number of samples since we know that small sampling sizes are preferred when population is sufficiently large.

The above argument can also be explained via the population-sizing models. The gambler's ruin model is of the following form [15] (Equation 10):

$$N = c_N \ln \psi, \qquad (50)$$

where c_N represents the rest terms, and ψ is the probability that GAs fail to converge as mentioned in Section 3.1. In our previous experiment, the requirement was successful

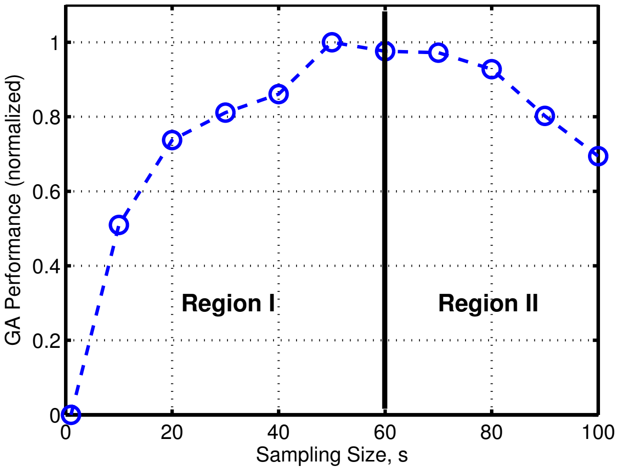

Figure 5: Performance versus sampling size. $R = 1000$, $r = 100$, $m = 3$. The population size, N, is fixed at 10000. It can be seen that the GA performs well around $s = 60$ as predicted.

convergence for 30 times. Therefore, we can estimate an upper bound of ψ by $(1-\psi)^{30} \geq 0.5$, which yields $\psi < 0.023$. Suppose that the fixed population size is N'. Given that c_N is constant for the same problem, we have the following relation:

$$N' = c_N \ln \psi' \tag{51}$$

$$\Rightarrow \quad \psi' = \psi^{\frac{N'}{N}}. \tag{52}$$

In the above equation, we see that the smaller N' is, the greater ψ' becomes, and hence the GA is unlikely to satisfy the convergence criterion, resulting in worse performance.

Experiments are conducted to support the above argument. All other parameters are the same, but with a fixed population size of 10000. We let the GA continue until the total number of samples exceeds 10^8. Then we use maximal distance between the objective and the current best chromosome as a measurement of GA performance. The results are shown in Figure 5, where every data point is averaged over 200 independent runs. Basically, the GA performs best around $s = 60$ as we expected, although the actual optimum occurs at $s = 50$, which is slightly off from our prediction. We believe that the reason lies on hill-climbing behavior. When the population size is not large enough, the behavior of crossover is not far from mutation. Given a maximal number of samples, the smaller the population size is, the more generations that the GA can execute. Therefore, even though the population size may be slightly too small to make good decisions, the GA spends longer time on hill climbing and performs well.

6.3 Resolving the Paradox

Now we revisit Grefenstette and Fitzpatrick's image registration results [13]. The sample space was 10000, and maximal number of samples is limited at 200000. They used a fixed population size of 80, and found the optimal sampling size is around $8 \sim 10$.

Given the modeling and experiments in this paper, we believe that the minimal population size requirement of their image registration problem at $s = 8 \sim 10$ should be around 80. The reason that the GA did not perform well for too small and too large sampling sizes is described above.

Note that if our explanation is true, the optimal sampling size depends on the fixed population size. In the previous experiment, if we fixed the population size at 20000 (the minimal population size requirement around $s = 20$) instead of 10000, the optimal sampling size would then occur at $s = 20$ instead of $s = 60$. Generally speaking, a larger fixed population size favors a smaller sampling size, and a smaller fixed population size favors a larger sampling size.

7. CONCLUSIONS

This paper investigates optimal sampling for GAs on polynomial regression problems. Specifically, the variances of the original fitness and sampling noises are modeled. Based on [22], facet-wise models for the total number of samples as a function of sampling size are derived. Then the derived models are applied to two different population-sizing scenarios, namely, (1) the gambler's ruin model [15] adjusted for fitness relaxation [22], and (2) fixed population sizing.

The results indicate that a small sampling size is preferred if the population is properly sized. When a fixed population size is adopted, an optimal sampling size, s^*, exists. For sampling sizes less than s^*, GAs suffer from insufficient population size; for sampling sizes greater than s^*, GAs suffer from inefficient fitness calculation. Generally speaking, a larger fixed population size favors a smaller sampling size, and a smaller fixed population size favors a larger sampling size. This paper also well explains the results of Grefenstette and Fitzpatrick's image registration problem [13].

Although this paper focuses on polynomial regression, the indications of the results are not limited to this specific type of problem. We expect that the indications can be applied to any problem with uniform salience and substantial positive dependency among samples. By positive dependency, we mean that if the fitness contribution of a sample is high, it is likely that the fitness contributions of other samples are also high. Combined with previous work, we conclude

1. That sampling techniques are useful when overhead is taken into consideration, and an optimal sampling size exists [19].

2. That sampling techniques are not useful for problems with uniform salience and no dependency among samples, unless overhead is taken into consideration [7, 25].

3. Small sampling sizes are preferred for problems with uniform salience and substantial positive dependency among samples. Optimal sampling size exists when a fixed population is used.

4. That sampling techniques are useful for problems with non-uniform salience when the adaptive sampling scheme is adopted [24].

There is still plenty of room for the sampling issue. For example, although this paper indicates that an optimal sampling size exists as a function of the fixed population size, the location of the optimal sampling is still yet unpredictable for real-world applications. It is desirable to develop some adaptive sampling scheme similar to [24] for problems with uniform salience. Also, the use of sampling techniques enforces two phenomena: genetic drift—genes converge without good reason [10] and premature convergence. Genetic drift did not cause any problem in the experiments in this paper maybe the selection pressure was not too low; however, we need to quantify this enforcement so that a lower bound of the selection pressure can be suggested. For a higher selection pressure, on the other hand, we will need some mechanism to keep diversity. Niching techniques [9] seem to be a good choice, but the interaction between niching and sampling needs to be investigated. Finally, we would also like to quantify the dependency among samples so that the representativeness of the polynomial regression problem can be measured. The main challenge would be assuming an underlying distribution of the data points that is general enough.

8. REFERENCES

[1] A. N. Aizawa and B. W. Wah. Scheduling of genetic algorithms in a noisy environment. *Evolutionary Computation*, 2:97–122, 1994.

[2] H. G. Beyer. Toward a theory of evolution strategies: Some asymptotical results from the (1,+λ)-theory. *Evolutionary computation*, 1:165–188, 1993.

[3] T. Blickle and L. Thiele. A mathematical analysis of tournament selection. *Proceedings of an International Conference on Genetic Algorithms and Their Applications (ICGA 1995)*, pages 9–16, 1995.

[4] A. Ceroni, M. Pelikan, and D. E. Goldberg. Convergence-time models for the simple genetic algorithm with finite population. IlliGAL Report No. 2001028, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, 2001.

[5] J. Cohen, C. P., S. West, and L. Aiken. *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, Hillsdale, NJ, second edition, 2003.

[6] K. Deb and D. E. Goldberg. Analyzing deception in trap functions. *Foundations of Genetic Algorithms*, 2:93–108, 1993.

[7] P. Giguère and D. E. Goldberg. Population sizing for optimum sampling with genetic algorithms: A case study of the Onemax problem. *Genetic Programming 98*, pages 496–503, 1998.

[8] D. E. Goldberg. Simple genetic algorithms and the minimal, deceptive problem. In *Genetic Algorithms and Simulated Annealing*, chapter 6, pages 74–88. Pitman Publishing, London, 1987.

[9] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, 1989.

[10] D. E. Goldberg. The race, the hurdle, and the sweet spot: Lessons from genetic algorithms for the automation of design innovation and creativity. *Evolutionary Design by Computers*, pages 105–118, 1999.

[11] D. E. Goldberg. *The design of innovation: Lessons from and for competent genetic algorithms*. Kluwer Academic Publishers, Norwell, MA, 2002.

[12] D. E. Goldberg, K. Deb, and J. H. Clark. Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6:333–362, 1992.

[13] J. J. Grefenstette and J. M. Fitzpatrick. Genetic search with approximate function evaluations. *Proceedings of an International Conference on Genetic Algorithms and Their Applications (ICGA 1985)*, pages 112–120, 1985.

[14] U. Hammel and T. Back. Evolution strategies on noisy functions: How to improve convergence properties. *Parallel Problem Solving from Nature (PPSN-III)*, pages 159–168, 1994.

[15] G. Harik, E. Cantú-Paz, D. E. Goldberg, and B. L. Miller. The gambler's ruin problem, genetic algorithms, and the sizing of populations. *Proceedings of 1997 IEEE International Conference on Evolutionary Computation*, pages 7–12, 1997.

[16] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992.

[17] B. L. Miller. *Noise, sampling, and efficient genetic algorithms*. doctoral dissertation, University of Illinois at Urbana-Champaign, Urbana, 1997.

[18] B. L. Miller and D. E. Goldberg. Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary Computation*, 4(2):113–131, 1996.

[19] B. L. Miller and D. E. Goldberg. Optimal sampling for genetic algorithms. *Proceedings of the Artificial Neural Networks in Engineering Conference (ANNIE 1996)*, 6:291–297, 1996.

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

[21] M. Rattray and J. L. Shapiro. Noisy fitness evaluations in genetic algorithms and the dynamics of learning. *Foundations of Genetic Algorithms*, 4:117–139, 1997.

[22] K. Sastry and D. E. Goldberg. Genetic algorithms, efficiency enhancement, and deciding well with differing fitness variances. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*, pages 528–535, 2002.

[23] D. Thierens and D. E. Goldberg. Convergence models of genetic algorithm selection schemes. In *Parallel Problem Solving fron Nature (PPSN III)*, pages 119–129, 1994.

[24] T.-L. Yu, Y.-p. Chen, D. E. Goldberg, and J.-H. Chen. An adaptive sampling scheme for genetic algorithms on the sampled onemax problem. *Proceedings of Artificial Neural Networks in Engineering 2003 (ANNIE 2003)*, pages 39–44, 2003.

[25] T.-L. Yu, D. E. Goldberg, and K. Sastry. Opitmal sampling and speed-up for genetic algorithms on the sampled onemax problem. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2003)*, pages 1554–1565, 2003.