
Genetic Algorithms, Efficiency Enhancement, and Deciding Well with Differing Fitness Bias Values

Kumara Sastry and **David E. Goldberg**
Illinois Genetic Algorithms Laboratory (IlliGAL)
Department of General Engineering
University of Illinois at Urbana-Champaign
104 S. Mathews Ave, Urbana, IL 61801
{ksastry,deg}@uiuc.edu

Abstract

This study develops a decision-making strategy for deciding between fitness functions with differing bias values. Simple, yet practical facetwise models are derived to aid the decision-making process. The decision making strategy is designed to provide maximum speed-up and thereby enhance the efficiency of GA search processes. Results indicate that bias can be handled temporally and that significant speed-up values can be obtained.

1 Introduction

Since the inception of genetic algorithms (GAs) (Holland, 1975), significant progress has been made in designing and analyzing them. A design decomposition has been proposed for the development of *competent* GAs and much progress has been made along these lines (Goldberg, 1999). Competent GAs take problems that were intractable with first generation GAs and render them tractable, oftentimes requiring only a subquadratic number of function evaluations.

However, for large-scale problems, the task of computing even a subquadratic number of function evaluations can be daunting. This is especially the case if the fitness evaluation is a complex simulation, model, or computation. Therefore, one usually resorts to approximate fitness functions that are less expensive to compute. However, approximations introduce error in assessing the solution quality. Also, we may have to choose from many fitness functions with differing error and cost values, and that choice has a large impact on the computational resources and the solution quality.

At present, practitioners make the choice among fitness function alternatives on an ad hoc basis. Therefore, we need to investigate which fitness function

should be used under what scenarios. Furthermore, one has to recognize that error introduced through approximations comes in two flavors: Bias, and variance (Keijzer & Babovic, 2000). The decision-making strategy depends on whether variance or bias dominates the error. We have considered the presence of bias and variance in isolation to demonstrate this difference and to ease the analytical burden.

This paper investigates decision making under the presence of bias, while the decision making under the presence of variance is developed elsewhere (Sastry, 2001). Specifically, we investigate the decision making between two fitness functions with differing bias values. A fitness function with higher bias value will yield a more inaccurate solution when compared to the function with a lower bias value. This inaccuracy can be eliminated temporally (not spatially). That is, using the spatial approach—sampling the high-bias fitness function—does not eliminate the effect of bias and yields an inaccurate solution.

On the other hand, a high-bias, low-cost function can be used during the initial few generations of the evolutionary process to obtain a crude solution. The low-bias, high-cost fitness function can then be used (later part of genetic search) to refine the genetic search and to obtain a more accurate solution. The generation at which the fitness functions are switched, called the *switching time* is an important factor in determining the speed-up. The objective of this study is to utilize facetwise models to predict the optimal switching time that yields greatest speed-up and to develop a decision-making strategy to handle bias in fitness functions.

This paper is organized as follows. Section 2 briefly discusses some previous work on handling error in fitness functions. The specific problem that we solve is defined in section 3. Section 4 defines the test problem used for developing models. A convergence-time model that incorporates bias in fitness functions is derived in

section 5. Section 6 develops models for predicting the optimal switching time and the speed-up. Finally, section 7 presents key conclusions of the study.

2 Related Work

Efficiency enhancement is essential for solving large-scale, complex search problems. One such technique is evaluation relaxation, in which the computation burden is reduced by utilizing inexpensive, but error-prone fitness assignment procedures instead of an expensive, but accurate fitness function.

Grefenstette and Fitzpatrick (1985) studied the use of approximate evaluations for an image registration problem. Follow-up studies (Fitzpatrick & Grefenstette, 1988; Mandava, Fitzpatrick, & Pickens, 1989) have further analyzed the utility of approximate fitness evaluations. However, these studies were largely empirical, and a design methodology for handling external noise was developed only recently (Miller & Goldberg, 1995; Miller, 1997). These studies consider only the effects of variance alone, and effects of bias, albeit to a limited extent has also been investigated (Jin, Olhofer, & Sendhoff, 2000; Albert, 2001). For further details on these and other studies on approximate fitness functions in GAs, the interested reader should consult the review presented elsewhere (Sastry, 2001).

3 Problem Definition

Consider two fitness function, f_1 and f_2 for a search problem with bias values of b_1 and b_2 respectively. That is, the optimal solution when f_1 is used is $\mathbf{x}^* + b_1$ and that when f_2 is used is $\mathbf{x}^* + b_2$. Here \mathbf{x}^* is the true optimal solution. The computational costs of f_1 and f_2 are c_1 and c_2 respectively. Furthermore, $b_1 < b_2$ and $c_1 > c_2$. An illustration of the fitness functions with different bias values is shown in figure 1. The figure shows a single variable unimodal fitness functions with and without bias. Note that the optimal value of the fitness functions need not be the same.

Implicitly, we assume that some building blocks (BBs) of f_1 and f_2 are different and others are the same. We recognize that this assumption might not hold true if the biased fitness function introduces multiple false optima. However, this study is the first step toward developing a decision making strategy for handling bias in fitness functions and it serves as a starting point for the analysis of more complex cases. It is important to note that the proposed models can be extended and applied to real-world problems (Albert, 2001).

Since f_1 and f_2 share some BBs, f_2 can be used for

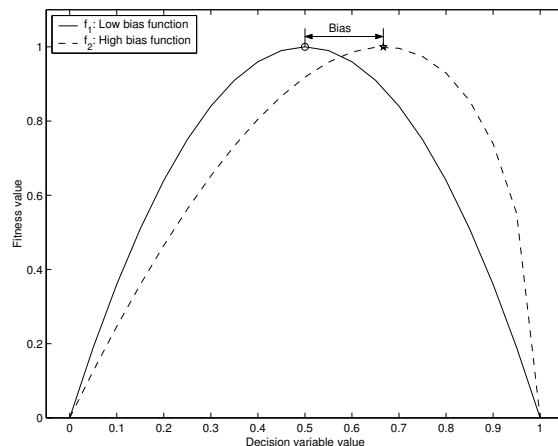


Figure 1: Fitness functions with different bias values.

the first few generations to obtain good convergence on the BBs shared by both the fitness functions. Fitness function f_1 can then be used to obtain a solution of better accuracy (lower bias). The time, t_s , at which we change from f_2 to f_1 is called the *switching time*. The objective of this study is to optimize the switching time to maximize speed-up and thus develop a decision-making strategy for choosing the correct fitness function. To develop models for solving the problem defined above, we need to first construct a test function. One such test function used in this study is described in the following section.

4 Test Function

The test function used in this study is the *weighted OneMax* defined as:

$$f = \sum_{i=1}^{\ell} w_i x_i, \quad (1)$$

where, x_i is the value of the i th allele and w_i is the weight associated with it. Similar to the OneMax function, the weighted OneMax is a linear unimodal function and the BBs are independent of each other. Therefore, the weighted OneMax function reduces the analytical burden for developing models considerably. Furthermore, fitness functions with differing bias values can be considered as weighted OneMax functions with different weights.

The BBs are uniformly scaled—that is, contribution of every BB to the fitness is equal in magnitude—if the weights, w_i , are restricted to be either ± 1 . Then, the fitness variance of a randomly generated population is equal to that for an OneMax problem. This further eases the analytical burden and the required popu-

lation size does not change with differing bias values. Therefore, we only need to develop a convergence-time model, which is presented in the next section.

5 Convergence-Time Model

Understanding time in GAs is one of the critical factors for a successful design of GAs (Goldberg, in press). Convergence-time model helps us in predicting the scale-up behavior of GAs. Existing studies on understanding time in GAs can be broadly classified into three approaches: (1) *Takeover-time models*, where the growth of the best individual in the population is analyzed (Goldberg & Deb, 1991), (2) *Selection-Intensity models*, where the dynamics of average fitness of the population is analyzed (Mühlenbein & Schlierkamp-Voosen, 1993; Bäck, 1995; Miller & Goldberg, 1995), and (3) *Higher-Order-Cumulant models*, where the dynamics of the average and higher order cumulants of fitness of the population are analyzed (Blickle & Thiele, 1995; Prügel-Bennet & Shapiro, 1994).

In contrast to selection-intensity models, higher-order-cumulant models do not yield closed-form solutions. Therefore, a selection-intensity-based convergence-time model is developed in this paper. For this purpose consider two weighted OneMax functions f_1 and f_2 :

$$f_1 = \sum_{i=1}^{\ell} w_i x_i, \quad (2)$$

$$f_2 = \sum_{i=1}^{\ell} w'_i x_i. \quad (3)$$

Without loss of generality assume that the fitness function f_1 has zero bias and that the weights w_i and w'_i are assigned as follows:

$$w_i = \begin{cases} 1 & 1 \leq i \leq \ell_1 \\ -1 & \ell_1 + 1 \leq i \leq \ell \end{cases}, \quad (4)$$

$$w'_i = \begin{cases} 1 & 1 \leq i \leq \ell_1 + b \\ -1 & \ell_1 + b + 1 \leq i \leq \ell \end{cases}, \quad (5)$$

where, b is the bias. That is, f_1 and f_2 share $\ell - b$ BBs and differ only in b alleles (in this case BBs). For example, the correct BB in any one of the b alleles for f_1 is 1 and for f_2 it is 0.

Note that initially, fitness function f_2 is used in the initial phase ($t < t_s$) of the genetic search. Assuming a uniform BB convergence, and a Gaussian fitness distribution, the expected average fitness of the population after selection is given by (Mühlenbein & Schlierkamp-Voosen, 1993):

$$\mu_{t+1} = \mu_t + I\sigma_t, \quad (6)$$

where, I is the selection intensity and is defined as the expected increase in the average fitness of a population after selection is performed upon a population whose fitness is distributed according to a unit normal distribution. Selection intensity is constant for tournament selection and is approximately given as a function of tournament size s by the following relation (Blickle & Thiele, 1995):

$$I = \sqrt{2 \left(\log(s) - \log \left(\sqrt{4.14 \log(s)} \right) \right)}. \quad (7)$$

Since fitness function f_2 is used in the first phase ($t \leq t_s$) of the run, the mean ($\mu_{f_2,t}$) and variance ($\sigma_{f_2,t}^2$) of fitness are given by

$$\mu_{f_2,t} = \ell p_t - (\ell - \ell_1 - b), \quad (8)$$

$$\sigma_{f_2,t}^2 = \ell p_t (1 - p_t), \quad (9)$$

where, p_t is the proportion of ones at time t . Using the mean and variance values in equation 6, we obtain

$$p_{t+1} - p_t = \frac{I}{\sqrt{\ell}} \sqrt{p_t(1 - p_t)}. \quad (10)$$

Approximating the above difference equation by a differential equation and integrating it yields

$$p_t = \frac{1}{2} \left[1 - \cos \left(\frac{It}{\sqrt{\ell}} + 2 \sin^{-1} \sqrt{p_0} \right) \right]. \quad (11)$$

Assuming that the initial population is randomly generated, we have $p_0 = 0.5$, and we get the following expression for the proportion of correct BBs as a function of time:

$$p_t = \frac{1}{2} \left[1 + \sin \left(\frac{It}{\sqrt{\ell}} \right) \right]. \quad (12)$$

The proportion of correct BBs at switching time t_s is therefore given by

$$p_{t_s} = \frac{1}{2} \left[1 + \sin \left(\frac{It_s}{\sqrt{\ell}} \right) \right]. \quad (13)$$

At the switching time t_s , the low bias fitness function f_1 is used instead of the high bias fitness function f_2 . Hence, the proportion of correct BBs changes. Since both f_1 and f_2 share $\ell - b$ BBs, the proportion of correct BBs for those BBs remains the same. That is the proportion of correct BBs for the $\ell - b$ is p_{t_s} . However, since f_1 and f_2 do not share b BBs, the proportion of correct BBs, for the b alleles is $1 - p_{t_s}$. This implies that there are two proportions of correct BBs one for $(\ell - b)$ alleles and the other for b alleles. The adjusted

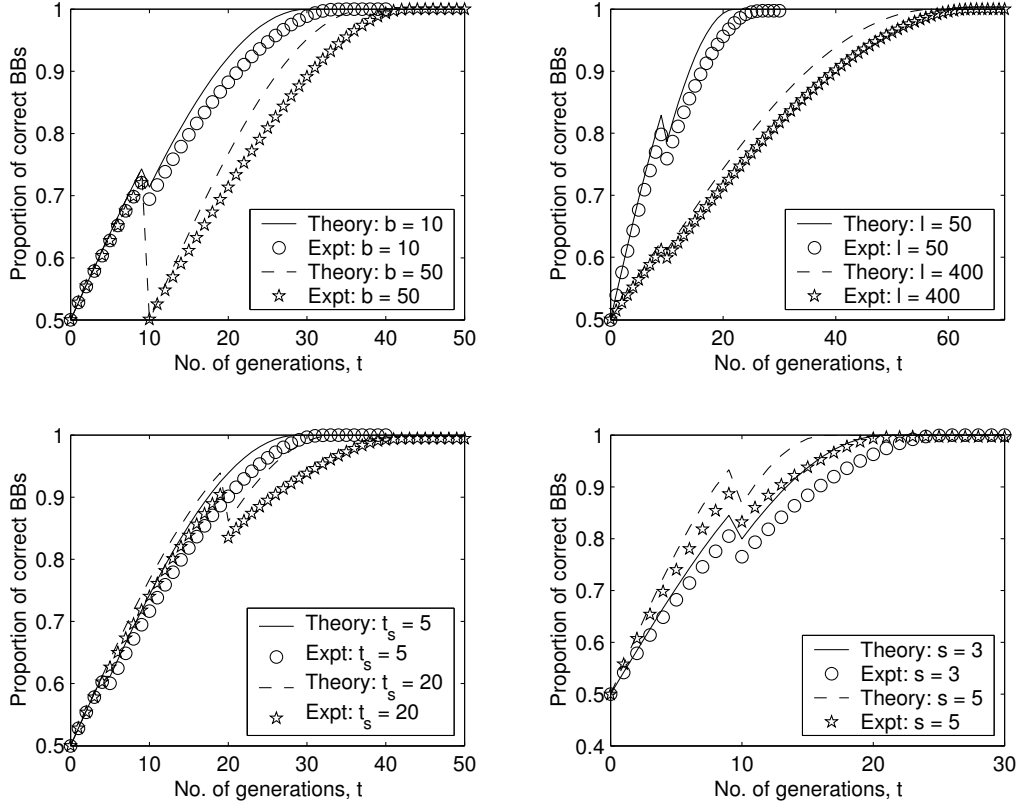


Figure 2: Empirical verification of the proportion of correct building blocks predicted by equations 12, 14, and 16 for different values of b , ℓ , t_s , and s .

proportion of correct BBs for the overall string, p'_{t_s} is given by

$$\begin{aligned} p'_{t_s} &= \frac{1}{\ell} [(\ell - b)p_{t_s} + b(1 - p_{t_s})], \\ &= \left(1 - 2\frac{b}{\ell}\right)p_{t_s} + \frac{b}{\ell}. \end{aligned} \quad (14)$$

From the selection-intensity model assumption, we know that the number of correct BBs in both $\ell - b$ and b portion are distributed normally. Since these two portions are statistically independent of each other, the number of correct BBs for the overall string, and similarly the fitness is also normally distributed. The mean and variance of fitness at time t ($t \geq t_s$) is given by $\ell p'_t - (\ell - \ell_1)$, and $\ell p'_t(1 - p'_t)$ respectively. Proceeding in the same way as we did for $t < t_s$, results in the following difference equation

$$p'_{t+1} - p'_t = \frac{I}{\sqrt{\ell}} \sqrt{p'_t(1 - p'_t)}. \quad (15)$$

Solving the above equation with the initial condition that at $t = t_s$, $p'_t = p'_{t_s}$, we get

$$p'_t = \frac{1}{2} \left[1 - \cos \left(\frac{I(t - t_s)}{\sqrt{\ell}} + 2 \sin^{-1}(\sqrt{p'_{t_s}}) \right) \right]. \quad (16)$$

From the above relation for the proportion of correct BBs, we can derive an expression for the convergence time, by equating $p'_t = 1$:

$$t_{\text{conv}} = t_s + \frac{\sqrt{\ell}}{I} \left[\pi - 2 \sin^{-1} \left(\sqrt{p'_{t_s}} \right) \right]. \quad (17)$$

The models developed above are verified with empirical results. A selectorecombinative GA with tournament selection without replacement, and uniform crossover scheme is employed for this purpose. The probability of crossover is taken to be 1.0 and mutation is not used. The value of ℓ_1 is kept constant at 25 for all the runs. The population size is determined by the relation $8\sigma_f^2$ (Goldberg, Deb, & Clark, 1992). This population-sizing model overestimates the population size and is used to remove any population-sizing effects. Unless otherwise mentioned the following parameters are used: $\ell = 100$, $s = 2$, $b = \frac{\ell}{10}$, and $t_s = 10$. The empirical results are averaged over 100 independent runs.

The proportion of correct BBs predicted by equation 12, 14, and 16 is validated by empirical results. The figures plot the proportion of correct BBs as a

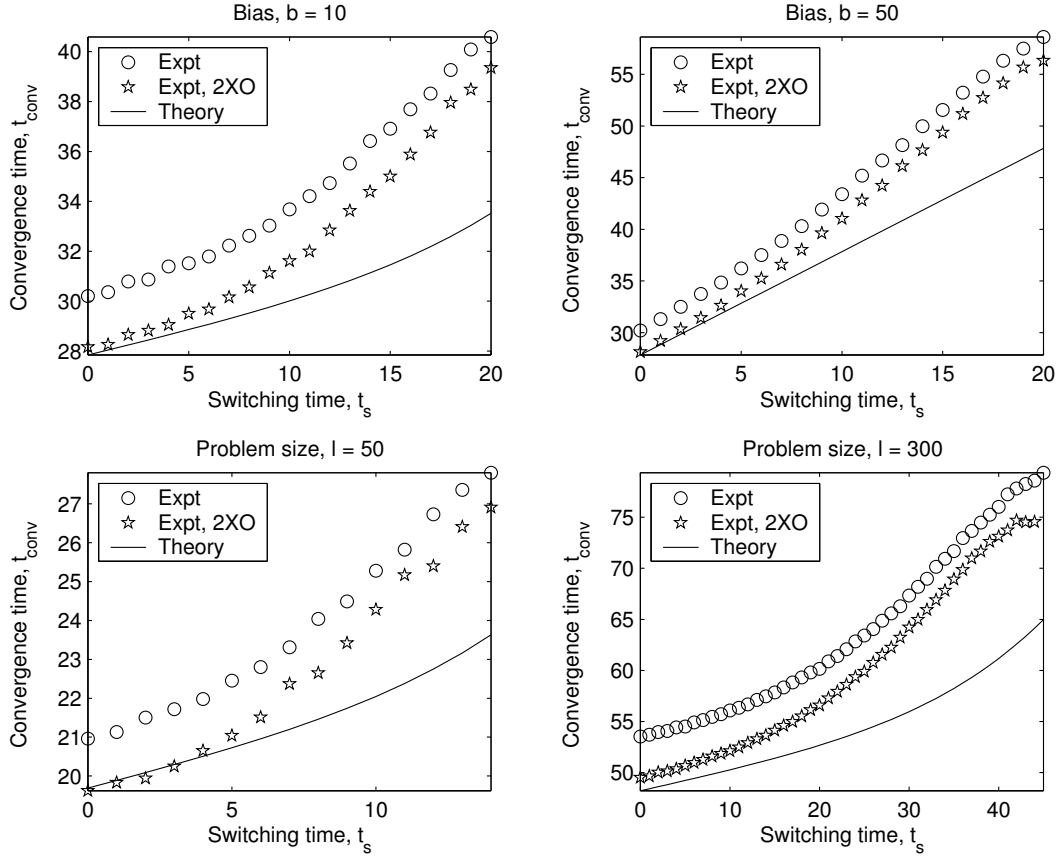


Figure 3: Empirical verification of convergence-time models (equation 17) for different bias values.

function of time. Different values of b , ℓ , t_s , and s are used to validate the model and are shown in figure 2. The results show that the model captures the dynamics accurately over a considerable range of parameter values. The discrepancy between the model and empirical results is due to hitch-hiking and can be further decreased by using multiple crossovers or using a population-wise crossover (Thierens & Goldberg, 1994).

The convergence-time model (equation 17) is compared to empirical results for different bias and problem-size values as shown in figure 3. The figure plots the convergence time as a function of switching time. The empirical results for the case where recombination is applied twice every generation are also shown in the figures. As expected, the agreement between the theoretical and experimental results increases when multiple crossover is applied. Note that the compressed convergence-time scale in figure 3 exaggerates the error and the model accuracy is comparable to existing models for other problem domains.

With the convergence-time model at hand, we will now

proceed to derive an expression for the optimal switching time. The speed-up that can be obtained by using the optimal switching time is also estimated in the next section.

6 Optimal Switching Time

From the problem definition and the convergence-time model (equation 17), total cost of function evaluation is then given by

$$\begin{aligned} n_{fe} &= n(c_2 t_s + c_1(t_{\text{conv}} - t_s)), \\ &= n c_2(t_s + c_r(t_{\text{conv}} - t_s)), \end{aligned} \quad (18)$$

where, $c_r = c_1/c_2$ is the ratio of cost of the high-cost fitness function to the cost of the low-cost fitness function. Employing model 2 (equation 17) for the convergence time, the above equation can be written as

$$n_{fe} = n c_2 \left(t_s + c_r \frac{2\sqrt{\ell}}{I} \left[\frac{\pi}{2} - \sin^{-1} \left(\sqrt{p'_{t_s}} \right) \right] \right). \quad (19)$$

We can define the total number of function evaluations in terms of time units by dividing the above equation

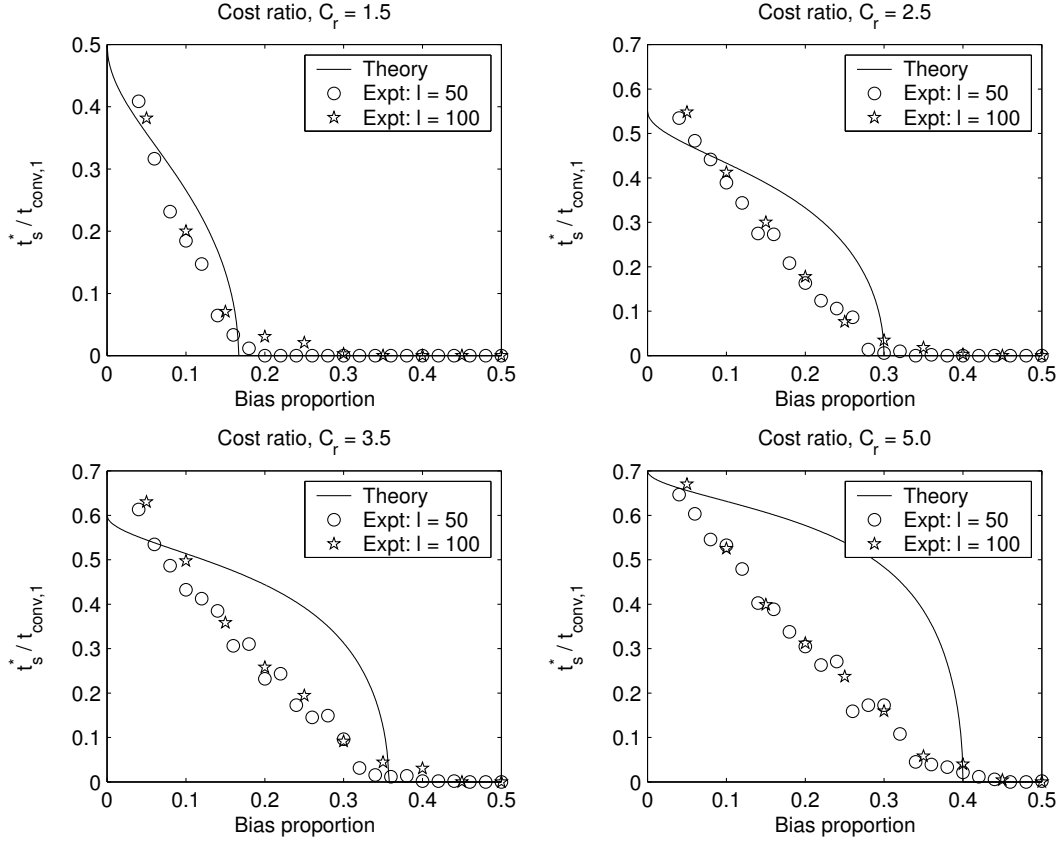


Figure 4: Verification of optimal switching-time model (equation 23).

by nc_2 :

$$n'_{fe} = t_s + c_r \frac{2\sqrt{\ell}}{I} \left[\frac{\pi}{2} - \sin^{-1} \left(\sqrt{p'_{t_s}} \right) \right]. \quad (20)$$

Our objective is to determine t_s that minimizes n'_{fe} (note that this is same as minimizing n_{fe}), which is given by solving

$$\begin{aligned} \frac{\partial n'_{fe}}{\partial t_s} &= 0, \\ 1 - c_r \frac{\sqrt{\ell}}{I} \frac{1}{\sqrt{p'_{t_s}(1-p'_{t_s})}} \frac{\partial p'_{t_s}}{\partial t_s} &= 0. \end{aligned}$$

The optimal switching generation, t_s^* , that minimizes n_{fe} when $c_r \geq \ell/(\ell - 2b)$, comes out to be

$$t_s^* = \frac{\sqrt{\ell}}{I} \cos^{-1} \left[\frac{2\sqrt{\frac{b}{\ell}(1-\frac{b}{\ell})}}{\left(1-\frac{2b}{\ell}\right)\sqrt{c_r^2-1}} \right]. \quad (21)$$

When $c_r < \ell/(\ell - 2b)$, $t_s^* = 0$.

Recognizing that the convergence-time when a low-bias, high-cost fitness function is used is given by

(Bäck, 1995)

$$t_{\text{conv},1} = \frac{\pi\sqrt{\ell}}{2I},$$

and dividing equation 21 with the above quantity, we obtain the a dimension-less expression for the optimal switching time when $c_r \geq 1/(1 - 2\beta)$:

$$\frac{t_s^*}{t_{\text{conv},1}} = \frac{2}{\pi} \cos^{-1} \left[\frac{2\sqrt{\beta(1-\beta)}}{(1-2\beta)\sqrt{c_r^2-1}} \right], \quad (22)$$

where, $\beta = b/\ell$ is the bias proportion. When $c_r < 1/(1 - 2\beta)$, $t_s^* = 0$. Equation 22 can be further reduced using the approximation $\cos^{-1}(x) \approx \frac{\pi}{2} - x$:

$$\frac{t_s^*}{t_{\text{conv},1}} = \left[1 - \frac{4}{\pi} \frac{\sqrt{\beta(1-\beta)}}{(1-2\beta)\sqrt{c_r^2-1}} \right]. \quad (23)$$

Equation 23 indicates that the strategy of employing the low cost fitness function for the first few generations yields speed-up only if the product of cost ratio, c_r , is above a critical limit which is inversely proportional to the bias proportion. If this is the case, then the optimal switching time is proportional to the

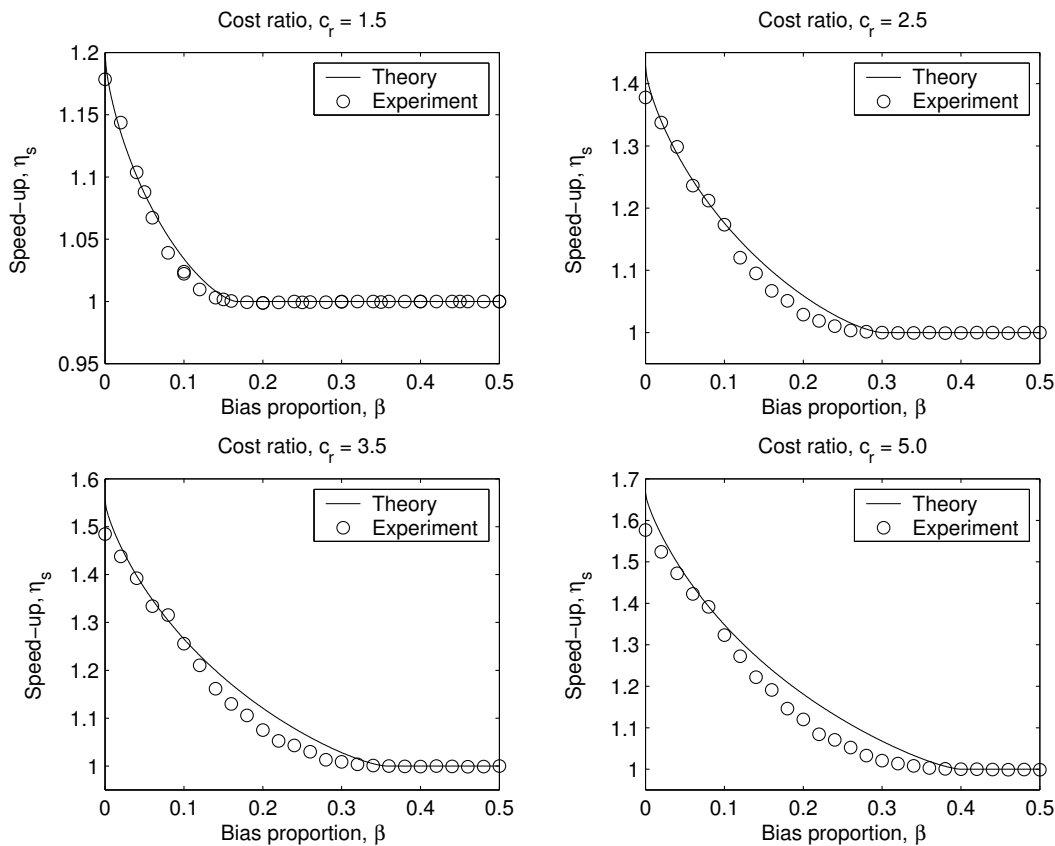


Figure 5: Empirical verification of speed-up prediction (equation 24).

square root of the string length, inversely proportional to the square root of the bias proportion, and inversely proportional to the cost ratio c_r . As expected, if the number of biased bits increases, the switching time decreases, and if the cost ratio increases, the switching time increases. Equation 23 is verified with empirical results in figure 4. The figure plots $t_s^*/t_{\text{conv},1}$ as a function of bias proportion β for different cost-ratio values. A binary tournament selection without replacement, uniform crossover with crossover probability of 1.0 is used. Mutation was not used in obtaining the empirical results. The results are averaged over 50 independent runs.

Using the optimal switching-time given by equation 23, we can compute the speed-up obtained by making the correct decision. Here the speed-up, η_s , is defined as the ratio of the total computational cost incurred if the low-bias fitness function is used to that if the high-bias fitness function is used for t_s^* generations and then the low-bias function is used till the end of the GA run. That is,

$$\eta_s = \frac{n_{fe,1}}{n_{fe,2}} = \frac{nc_1 t_{\text{conv},1}}{n [c_2 t_s^* + c_1 (t_{\text{conv},2} - t_s^*)]},$$

$$= \frac{c_r}{\left[\frac{t_{\text{conv},2}}{t_{\text{conv},1}} - (c_r - 1) \tau_s^* \right]}. \quad (24)$$

Where, $\tau_s^* = t_s^*/t_{\text{conv},1}$. Note that the above equation is valid when $c_r \geq 1/(1 - 2\beta)$. When $c_r < 1/(1 - 2\beta)$, $\eta_s = 1$. The speed-up predicted by equation 24 is verified with empirical results in figure 5. The figure plots η_s as a function of bias proportion β for different cost-ratio values. Tournament selection without replacement with tournament size $s = 2$ is used. Uniform crossover with crossover probability of 1.0 is employed and mutation is not used. The results are averaged over 50 independent runs.

Figure 5 clearly indicates the improvement in efficiency using the decision-making strategy developed to handle bias in fitness functions. It also validates our hypothesis that bias has to be handled temporally. Furthermore, even though we made some simplifying assumptions the final result for the optimal switching time and the speed-up are in dimensionless quantities and should be easily applicable to other problem domains as well.

7 Conclusions

This paper develops a decision-making strategy for choosing between fitness function with differing bias values. We proposed that bias has to be handled temporarily by switching from a high-bias fitness function to a low-bias fitness function. We also hypothesized that an optimal switching time exists and when the fitness functions are switched at this optimal time, the total computation cost will be the minimum. We developed approximate, but practical convergence-time model, and used it to determine the optimal switching time. Based on the computational cost and the total number of function evaluations taken by each fitness function, a decision-making strategy was presented.

The paper shows that bias has to be handled temporarily. That is, a high-bias fitness function should be used for coarse-grain optimization and then a low-bias fitness function should be used for fine-grain optimization. Although, we considered only two fitness functions, the decision making can be easily extended for more than two fitness functions. Furthermore, the models developed in this study should provide guidance to GA practitioners in choosing key GA parameters and to provide maximum efficiency enhancement.

Acknowledgments

This work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-00-0163, and the National Science Foundation under grant DMI-9908252. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation thereon.

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

References

Albert, L. A. (2001). *Efficient genetic algorithms using discretization scheduling*. Master's thesis, University of Illinois at Urbana-Champaign, General Engineering Department, Urbana, IL.

Bäck, T. (1995). Generalized convergence models for tournament—and (μ, λ) —selection. *Proceedings of the Sixth International Conference on Genetic Algorithms*, 2–8.

Blickle, T., & Thiele, L. (1995). A mathematical analysis of tournament selection. *Proceedings of the Sixth International Conference on Genetic Algorithms*, 9–16.

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

Goldberg, D. E. (1999). The race, the hurdle, and the sweet spot: Lessons from genetic algorithms for the automation of design innovation and creativity. In Bentley, P. (Ed.), *Evolutionary Design by Computers* (Chapter 4, pp. 105–118). San Mateo, CA: Morgan Kaufmann.

Goldberg, D. E. (in press). *Design of innovation: Lessons from and for competent genetic algorithms*. Boston, MA: Kluwer Academic Publishers.

Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms*, 69–93.

Goldberg, D. E., Deb, K., & Clark, J. H. (1992). Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6, 333–362.

Grefenstette, J. J., & Fitzpatrick, J. M. (1985). Genetic search with approximate function evaluations. *Proceedings of the International Conference on Genetic Algorithms and Their Applications*, 112–120.

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

Jin, Y., Olhofer, M., & Sendhoff, B. (2000). On evolutionary optimization with approximate fitness functions. *Proceedings of the Genetic and Evolutionary Computation Conference*, 786–793.

Keijzer, M., & Babovic, V. (2000). Genetic programming, ensemble methods and the bias/variance tradeoff - introductory investigations. *Genetic Programming: Third European Conference*, 76–90.

Mandava, V. R., Fitzpatrick, J. M., & Pickens, III, D. R. (1989). Adaptive search space scaling in digital image registration. *IEEE Transactions on Medical Imaging*, 8(3), 251–262.

Miller, B. L. (1997). *Noise, sampling, and efficient genetic algorithms*. Doctoral dissertation, University of Illinois at Urbana-Champaign, Urbana, IL. (Also IlliGAL Report No. 97001).

Miller, B. L., & Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9(3), 193–212.

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

Prügel-Bennet, A., & Shapiro, J. L. (1994). An analysis of a genetic algorithm using statistical mechanics. *Physics Review Letters*, 72(9), 1305–1309.

Sastry, K. (2001). *Evaluation-relaxation schemes for genetic and evolutionary algorithms*. Master's thesis, University of Illinois at Urbana-Champaign, Urbana, IL. (Also IlliGAL Report no. 2002004).

Thierens, D., & Goldberg, D. E. (1994). Convergence models of genetic algorithm selection schemes. *Parallel Problem Solving from Nature*, 3, 116–121.