Don't Evaluate, Inherit

Kumara Sastry
David E. Goldberg
Martin Pelikan

IlliGAL Report No. 2001013
January, 2001

Illinois Genetic Algorithms Laboratory (IlliGAL)
Department of General Engineering
University of Illinois at Urbana-Champaign
117 Transportation Building
104 S. Mathews Avenue, Urbana, IL 61801

Don't Evaluate, Inherit

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

Abstract

This paper studies fitness inheritance as an efficiency enhancement technique for genetic and evolutionary algorithms. Convergence and population sizing models are derived and compared with experimental results. These models are optimized for greatest speed-up and the optimal inheritance proportion to obtain such a speed-up is derived. Results also show that when the inheritance effects are considered in the population sizing model, the number of function evaluations are reduced by 20% with the use of fitness inheritance. Results indicate that for a fixed population size, the number of function evaluations can be reduced by 70% using a simple fitness inheritance technique.

1 Introduction

A key challenge in genetic and evolutionary computation (GEC) research is the design of *competent* genetic algorithms (GAs). By *competent* we mean GAs that can solve *hard problems*, quickly, reliably, and accurately, and much progress has been made along these lines (Goldberg, 1999). In essence competent GA design takes problems that were intractable with first generation GAs and renders them tractable, oftentimes requiring only a subquadratic number of fitness evaluations. But in 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. This places a premium on a variety of efficiency enhancement techniques. In this paper, one such efficiency enhancement technique called *fitness inheritance* is modeled and optimized for greatest speedup. In fitness inheritance, an offspring inherits a fitness value from its parents rather than through function evaluation.

The objective of this study therefore is to model fitness inheritance and to employ this model in predicting the convergence time and population size required for the successful design of a GA. We start by modeling fitness inheritance and deriving the convergence time and population sizing models. Subsequently, we derive an optimal proportion of inheritance and comment on the actual speed-up obtained through inheritance. The speed-up that could be obtained for a fixed population sizing is also discussed.

2 Literature Review

Smith, Dike, and Stegmann (1995) proposed fitness inheritance in GAs. They proposed two ways of inheriting fitness, one by taking the average fitness and the other by taking weighted average of the fitness of the two parents. They showed some theoretical justification for their approach. Their results indicated that GAs with fitness inheritance outperformed those without inheritance. However, they did not investigate the effect of fitness inheritance on convergence time and population sizing. Also, the questions as to how many children should have inherited fitness, and how much speed-up one can get remained unanswered. Though the study by Smith, Dike, and Stegmann (1995) showed very encouraging results, unfortunately there have been very few studies on fitness inheritance. Zheng, Julstrom, and Cheng (1997) used fitness inheritance for the design of vector quantization codebooks.

3 Modeling Fitness Inheritance

In the proposed approach, a proportion, p_i, of randomly selected individuals, receive inherited fitness and the rest are assigned the true (evaluated) fitness. In the remainder of this paper, actual fitness refers to the fitness that a individual would have had if it was evaluated, i.e, if its fitness was not inherited. In this section, we assume that the inherited fitness is taken to be the average of the building block (BB) fitness. We assume this to develop a theory behind fitness inheritance and in the implementation the inherited fitness is taken to be the average fitness of the two parents. The building block fitness is taken to be the average fitness of all the individuals in the population that possess the schemata under consideration. Also, in the remainder of the paper, unless otherwise mentioned, all the experimental results are with crossover probability of 1.0.

The model derived is applicable to uniformly scaled problems of fixed string length and known BB size. Specifically OneMax (counting of bits) is employed but the model can be extended to other problems in a straightforward manner. We further assume that the actual fitness distribution, F, is Gaussian with mean $\mu_{f,t}$ and variance $\sigma^2_{f,t}$.

$$F = N\left(\mu_{f,t}, \sigma^2_{f,t}\right),$$

and that the distribution of fitness with inheritance, F$'$ is Gaussian with mean $\mu_{f',t}$ and variance $\sigma^2_{f',t}$.

$$F' = N\left(\mu_{f',t}, \sigma^2_{f',t}\right).$$

The above assumptions are justified since crossover has a normalizing effect. We can write

$$\mu_{f',t} = \mu_{f,t}(1-p_i) + \mu_{i,t}p_i, \tag{1}$$
$$\sigma^2_{f',t} = (1-p_i)\sigma^2_{f,t} + p_i\sigma^2_{i,t}, \tag{2}$$

where $\mu_{i,t}$, and $\sigma^2_{i,t}$ are the mean and variance of fitness respectivley, of individuals whose fitness is inherited. Since the inherited fitness, f_i, is equal to the average of BB fitness we can write

$$f_i = \frac{1}{\ell}\sum_{j=1}^{\ell} \hat{f}\left(BB_j\right), \tag{3}$$

where, ℓ is the string length, and $\hat{f}\left(BB_j\right)$ is the estimated BB fitness which can be written

$$\hat{f}\left(BB_j\right) = f\left(BB_j\right) + (\ell-1)p, \tag{4}$$

where, $f\left(BB_j\right)$ is the actual BB fitness, p is the proportion of correct BBs, and the term $(\ell-1)p$ incorporates the noise arising from other BBs. Using the above relation, f_i for uniformly scaled problems can be written as

$$f_i = \frac{f}{\ell} + (\ell-1)p. \tag{5}$$

The mean inherited fitness, $\mu_{i,t}$ is given by

$$\begin{aligned}
\mu_{i,t} &= \frac{1}{n}\sum_{j=1}^{n} f_{i,j}, \\
&= \frac{1}{n}\sum_{j=1}^{n}\left[\frac{f_j}{\ell} + (\ell-1)p\right], \\
&= \frac{1}{\ell}\left[\frac{1}{n}\sum_{j=1}^{n} f_j\right] + (\ell-1)p, \\
&= \ell p = \mu_{f,t}, \qquad (6)
\end{aligned}$$

where n is the population size. Using the above relation in eqn. (1), we get

$$\mu_{f',t} = \mu_{f,t}. \qquad (7)$$

The inherited fitness variance, $\sigma^2_{i,t}$ can be derived as follows,

$$\begin{aligned}
\sigma^2_{i,t} &= \frac{1}{n}\sum_{j=1}^{n} f^2_{i,j} - (\ell p)^2, \\
&= \frac{1}{n}\sum_{j=1}^{n}\left[\frac{f_j}{\ell} + (\ell-1)p\right]^2 - (\ell p)^2, \\
&= \frac{p(1-p)}{\ell}. \qquad (8)
\end{aligned}$$

Using the above relation in eqn. (2) we get,

$$\begin{aligned}
\sigma^2_{f',t} &= (1-p_i)\sigma^2_{f,t} + p_i\frac{p(1-p)}{\ell} \\
&\approx (1-p_i)\sigma^2_{f,t} \qquad (9)
\end{aligned}$$

Using the notion of *selection intensity*, I (Bulmer, 1980), we can write the expected average fitness with inheritance after selection as

$$\begin{aligned}
\mu_{f',t+1} &= \mu_{f',t} + I\sigma_{f',t}, \\
&= \mu_{f',t} + I\sqrt{1-p_i}\,\sigma_{f,t}. \qquad (10)
\end{aligned}$$

Since both the actual fitness and inherited fitness distributions are normally distributed, bivariate normal distribution can be used to obtain the expected actual fitness value of F at generation t+1, given $\mu_{f',t+1}$,

$$E(F/\mu_{f',t+1}) = \mu_{f,t+1} = \mu_{f,t} + \frac{\sigma_{F,F'}}{\sigma^2_{f',t}}(\mu_{f',t+1} - \mu_{f',t}).$$

It can be easily seen that the covariance, $\sigma_{F,F'}$ is $(1-p_i)\sigma^2_f$. Using this relation and eqn. (10),

$$\begin{aligned}
\mu_{f,t+1} &= \mu_{f,t} + (\mu_{f',t} + I\sqrt{1-p_i}\,\sigma_{f,t} - \mu_{f',t}), \\
&= \mu_{f,t} + I\sqrt{1-p_i}\,\sigma_{f,t}. \qquad (11)
\end{aligned}$$

We now proceed to derive the convergence and population sizing models.

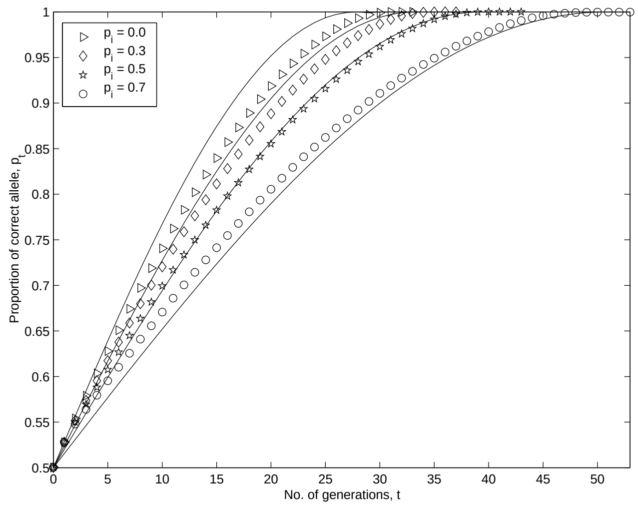

Figure 1: Verification of Proportion of correct BBs predicted by eqn. (13) with empirical results plotted as a function of generation number for different values of inheritance proportion. The experimental results are average over 50 runs. The discrepancy between the theoretical and experimental results are due to hitch-hiking and can be eliminated by ensuring a good mixing of BBs.

3.1 Time to Convergence

In this section we derive convergence model for the OneMax problem with fitness inheritance. For OneMax domain, we can write

$$\mu_{f,t} = \ell p_t, \quad \sigma^2_{f,t} = \ell p_t(1-p_t),$$

where, ℓ is the string length, and p_t is the proportion of correct alleles in the population at generation t. Since the initial population is generated with uniform distribution, $p_0 = 0.5$. Using the above relation in eqn. (11),

$$\begin{aligned} p_{t+1} &= p_t + I\sqrt{\frac{(1-p_i)}{\ell}}\sqrt{p_t(1-p_t)}, \\ p_{t+1} - p_t &= I\sqrt{\frac{(1-p_i)}{\ell}}\sqrt{p_t(1-p_t)}. \end{aligned}$$

Approximating the above equation as a differential equation yields

$$\frac{dp}{dt} = \frac{I\sqrt{1-p_i}}{\sqrt{\ell}}\sqrt{p(1-p)}. \tag{12}$$

Integrating the above equation and using the initial condition $p|_{t=0} = 0.5$ we get,

$$p_t = \sin^2\left(\frac{\pi}{4} + \frac{I\sqrt{(1-p_i)}t}{2\sqrt{\ell}}\right). \tag{13}$$

The above equation is compared to the experimental results in fig. 1 at different inheritance proportions p_i. The proposed convergence model slightly overestimates the proportion of correct BBs for GAs with tournament selection and uniform crossover. This discrepancy between the

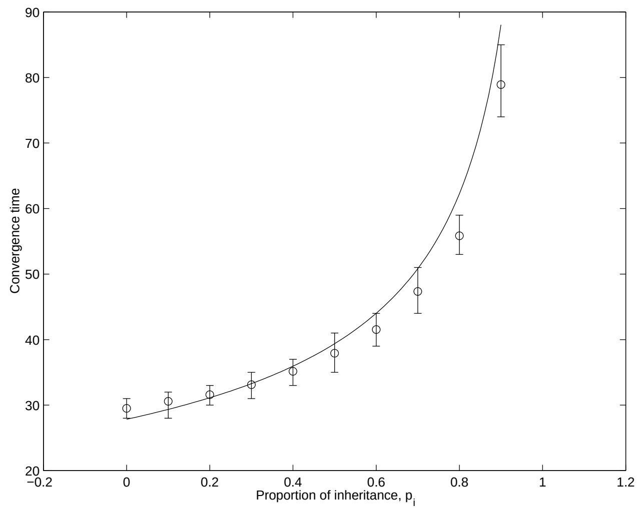

Figure 2: Convergence time for a 100-bit OneMax problem for different proportion of inheritance predicted by eqn. (14) compared to experimental results. The empirical results are averaged over 50 runs.

theoretical and empirical results can be eliminated by employing recombination procedures than ensure that the BBs are well mixed (Thierens, 1995).

We can derive an equation for convergence time, $t_{\rm conv}$, by equating $p_t = 1$, and inverting eqn. (13),

$$t_{\rm conv} = \frac{\pi}{2I}\sqrt{\frac{\ell}{(1-p_i)}}. \tag{14}$$

If p_i is taken as 0 then the above relation reduces to $\pi\sqrt{\ell}/(2I)$ which agrees with existing convergence time models (Muhlenbein & Schlierkamp-Voosen, 1993; Miller & Goldberg, 1996a; Miller & Goldberg, 1996b). The convergence time observed experimentally is compared to the above prediction for a 100-bit OneMax problem in fig. 2. Again the discrepancy between the empirical and analytical results occurs due to hitch-hiking and can be reduced by ensuring a good mixing of building blocks.

3.2 Population Sizing

It is well known that population size is a major determinant of the quality of the solution obtained. Therefore it is essential to appropriately size the population to incorporate the effects of fitness inheritance. Goldberg, Deb, and Clark (1992) proposed population sizing models for different selection schemes. Their model is based on deciding correctly between the best and the second best BBs in the same partition. They incorporated noise arising from other partitions into their model. However, they assumed that if wrong BBs were chosen in the first generation, the GAs would be unable to recover from the error. Harik, Cantu-Paz, Goldberg, and Miller (1997) refined the above model by incorporating cumulative effects of decision making over time rather than in first generation only. They modeled the decision making between the best and second best BBs in a partition as a gambler's ruin problem. This model is based on the assumption that the selection process used is tournament selection without replacement. Miller (Miller, 1997) extended this model to predict population sizing in the presence of external noise. The population sizing model

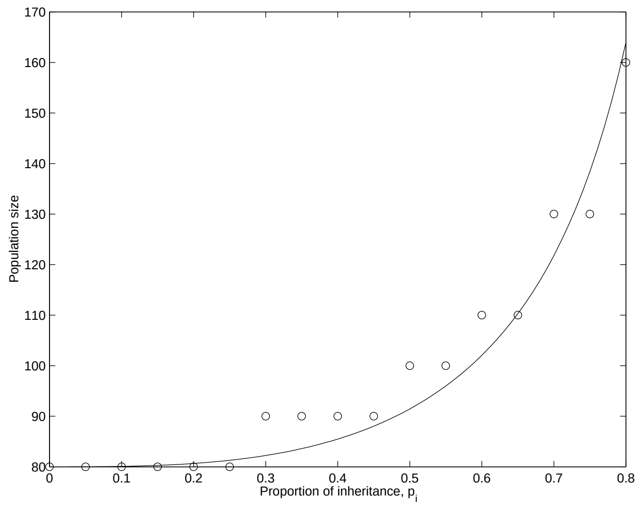

Figure 3: Verification of the population sizing model (eqn. 16) for various inheritance proportions with empirical results. Experimental results depict the population size required for optimal convergence with failure rate of 0.001 and are averaged over 50 runs.

derived by Miller is reproduced below.

$$n = -\frac{2^{k-1}\log(\psi)\sqrt{\pi}}{d_{\min}}\sqrt{\sigma^2_{f'}},$$

where n is the population size, k is the BB length, ψ is the failure rate, $d_{\min}$ is the distance between the best BB and the second best BB (Goldberg, Deb, & Clark, 1992), and $\sigma^2_{f'}$ is the variance of the noisy fitness function. Not only $\sigma^2_{f'}$, but also $d_{\min}$ depends on p_i. For OneMax problems $d_{\min}$ was empirically determined to be

$$d_{\min} = (1-p_i^3)\sqrt{1-p_i}. \tag{15}$$

The population sizing equation can now be written as

$$n = -\frac{2^{k-1}\log(\psi)\sqrt{\pi}}{(1-p_i^3)}\sqrt{\sigma^2_{f}}. \tag{16}$$

The above population sizing is compared to the results obtained for a 100-bit OneMax problem in fig. 3. From the plot we can easily see that our population sizing model fits the experimental result accurately. Using the convergence time and population sizing model derived in this section, we evaluate the inheritance proportion that requires least number of function evaluations (or equivalently, yields greatest speed-up) in the next section.

4 Optimal Inheritance Proportion

We can intuit that given a problem there should be a value (or a range) of inheritance proportions that are more efficient than the others. Too low a p_i or too high a p_i would not reduce the number of function evaluations. Our aim is to determine the inheritance proportion such that the total number of function evaluation required is minimized. Here we assume that the cost of inheritance is insignificant. This is justified by the fact that inherited fitness is just an average of the fitness

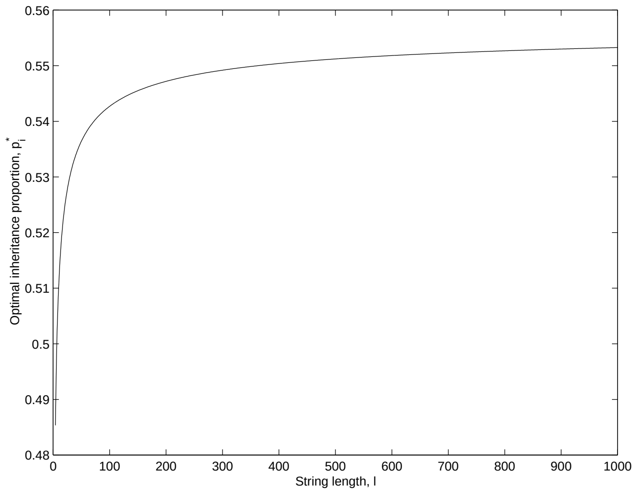

Figure 4: Optimal inheritance proportion, p_i^* as a function of string length ℓ obtained by numerically solving eqn. (19). p_i^* is independent of ℓ for moderate to large values of ℓ.

values of the two parents, a computationally trivial task when compared to the usual function evaluation. We reiterate that fitness inheritance is needed in cases where function evaluation takes a long time (eg., a large real-world problem), or when only some individuals can be evaluated (eg, interactive GAs). Total number of function evaluations required is given by

$$\begin{aligned} N_{fe} &= n\left[(t_{\text{conv}}-1)(1-p_i)+1\right], \\ &= n\left[t_{\text{conv}}(1-p_i)+p_i\right]. \end{aligned} \tag{17}$$

From previous sections, for a given problem,

$$\begin{aligned} t_{\text{conv}} &= \frac{c_2}{\sqrt{1-p_i},} \\ n &= \frac{c_3}{1-p_i^3,} \end{aligned}$$

where c_2 and c_3 are $\pi\sqrt{\ell}/(2I)$ and $-2^{k-1}\log(\psi)\sqrt{\pi\sigma_f^2}$ respectively. Using the above equations, the total number of function evaluations is given by,

$$N_{fe} = \frac{c_3}{1-p_i^3}\left[c_2\sqrt{1-p_i}+p_i\right]. \tag{18}$$

The optimal proportion of inheritance is then given by solving,

$$\begin{aligned} \frac{\partial N_{fe}}{\partial p_i} &= 0, \\ 3p_i^2\left[c_2(1-p_i)+p_i\sqrt{1-p_i}\right]+ & \\ (1-p_i^3)\left[-0.5c_2+\sqrt{1-p_i}\right] &= 0. \end{aligned} \tag{19}$$

The above equation can be solved for two asymptotic cases: (1) the string length, $\ell = 0$, then $c_2 = 0$, and the optimal evaluates to $p_i^* = 0$. (2) The problem is of infinite string length, then

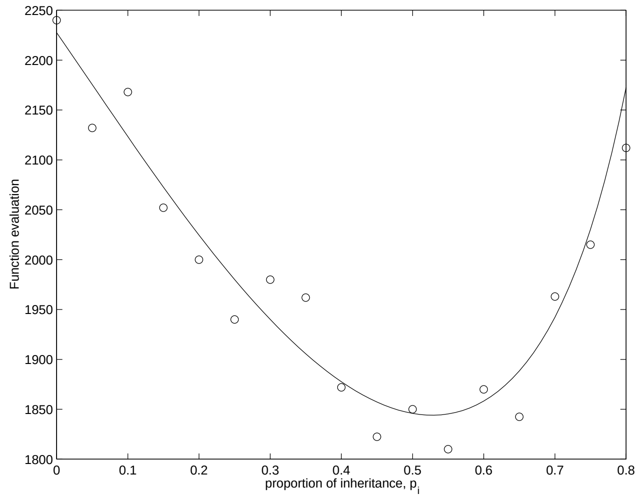

Figure 5: Total number of function evaluations predicted by eqn. (17) is compared to empirical results as a function of inheritance proportion. The experimental results depict the total number of function evaluations required for optimal convergence of a 100-bit OneMax problem with a failure rate of 0.0001. The empirical results are averaged over 50 runs.

$c_2 = \infty$. For this case eqn. (19) reduces to

$$3p_i^2 (1 - p_i) - \frac{1}{2}\left(1 - p_i^3\right) = 0, \tag{20}$$

$$p_i^2 - \frac{1}{5}p_i - \frac{1}{5} = 0. \tag{21}$$

The above quadratic equation can be easily solved, and the optimal proportion for this case comes out to be $p_i^* = 0.558$. For other values of string length, eqn. (19) cannot be solved analytically, and hence it has been solved numerically for different problem sizes. The optimal proportion of inheritance obtained by solving the above equation numerically is plotted as a function of string length is shown in fig. 4. We can see that for moderate to large sized problems the optimal proportion of inheritance, p_i^* lies between 0.54-0.558, i.e.,

$$0.54 \leq p_i^* < 0.558 \tag{22}$$

The above result (eqn. 22) suggests that p_i is independent of problem size for problems of moderate to large size. The predicted number of function evaluations is compared with experimental results for a 100-bit OneMax in fig. 5, for a 40-bit trap function with BB size 4 with a crossover probability of 0.9 in fig. 6(a), and for a 40-bit trap function with BB size 4 with tournament size of 8 and crossover probability of 1 in fig. 6(b). Even though our model was derived for OneMax problems, it holds even for other problems with different parameter settings as shown by the plots. This exemplifies the robustness and usefulness of the proposed model. Another point to be noted is that the optimal inheritance proportion is between 0.54-0.558 in all cases.

Another interesting fact to note is that the number of function evaluations with inheritance is only around 20% less than that without inheritance. In other words the speed-up defined as the ratio of number of function evaluations with $p_i = 0$ to the number of function evaluations at optimal p_i is around 1.2. This implies that we get a moderate advantage by using fitness inheritance. The existence of an optimal p_i and the moderate value of speed-up are in contrast with the earlier studies on inheritance. A detailed discussion on this is presented in the next section.

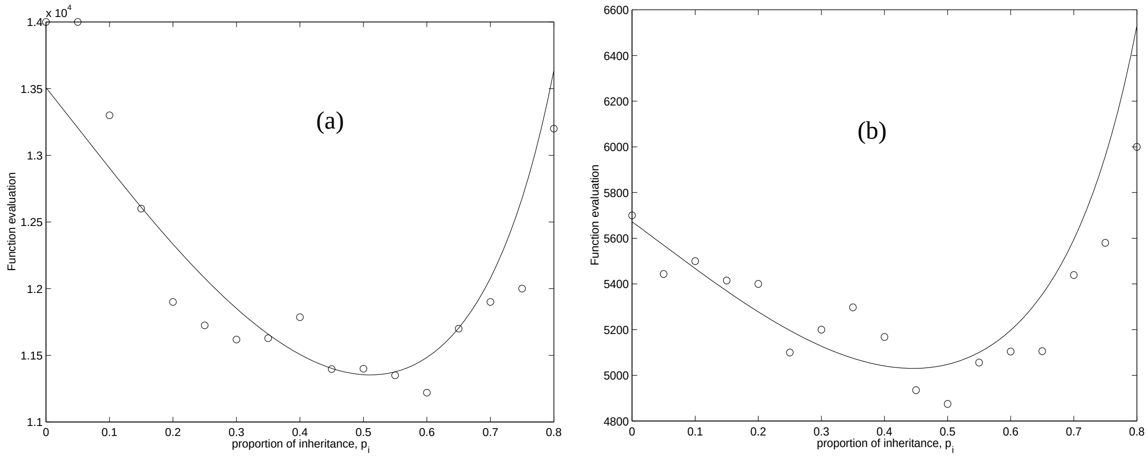

Figure 6: Total number of function evaluations predicted by eqn. (17) is compared to empirical results as a function of inheritance proportion. The experimental results depict the total number of function evaluations required for optimal convergence of a 40-bit trap function with a BB size of 4, and a failure rate of 0.0001. The empirical results are averaged over 50 runs. Crossover probability is (a) 1.0, and (b) 0.8. The results indicate that the optimal proportion of inheritance given by eqn. (22) is in fact approximately valid for other uniformly scaled problems with BB size greater than one and different GA parameter values.

5 Apparent Speed-up

In the previous section we presented the speed-up we can obtain if we choose the population size appropriately. This is the speed-up that we can obtain from a GA algorithmist point of view. A GA practitioner, unlike a GA algorithmist, views GAs as means to reach an end. He usually fixes the population size and then opts for fitness inheritance. From a GA practitioner's point of view, the speed-up obtained through fitness inheritance can be much higher. We call this speed-up, that is obtained through a fixed population size as *apparent* speed-up.

Function evaluations taken for different population sizes plotted as a function of p_i for a 100-bit OneMax problem is shown in fig. 7. The plot indicates only those points for which the population converged to the optimal solution in all 50 runs. The apparent optimal inheritance proportion, $p_i^{\rm app}$, is given by the inverse of the population sizing model, eqn. (16).

$$p_i^{\rm app} = \sqrt[3]{1 - \frac{\kappa}{n}}, \tag{23}$$

where κ is a constant dependent on the problem type, and the solution quality desired, and is related by $\kappa = -2^{k-1}\log(\psi)\sqrt{\pi}$. There are two asymptotic cases for the above result. One, when the population size is less than κ, then fitness inheritance does not yield any speed-up and in fact can result in premature convergence. The other case is when the population size is very large when compared to κ. In this case a very high inheritance proportion can be used and high speedup values can be obtained.

The apparent optimal inheritance proportion predicted by eqn. (23) is compared to experimental results for a 100-bit OneMax in fig. 8. The value κ for this problem is 81.63. The experimental results indicate the inheritance proportion that required lowest number of average function evaluations to converge to optimal solution in all 50 runs. From fig. 7, it can be seen that if we choose an arbitrarily high population size, say 300, then fitness inheritance can yield a speed-up of around

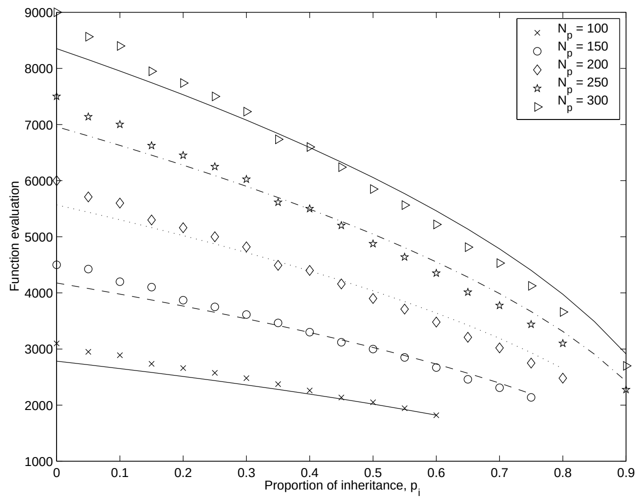

Figure 7: Total number of function evaluations for various proportion of inheritance at different population sizes. The experimental results are average of 50 runs and are compared to the results predicted using eqn. (17). Experimental results include only those points for which all 50 runs converged to the optimal solutions.

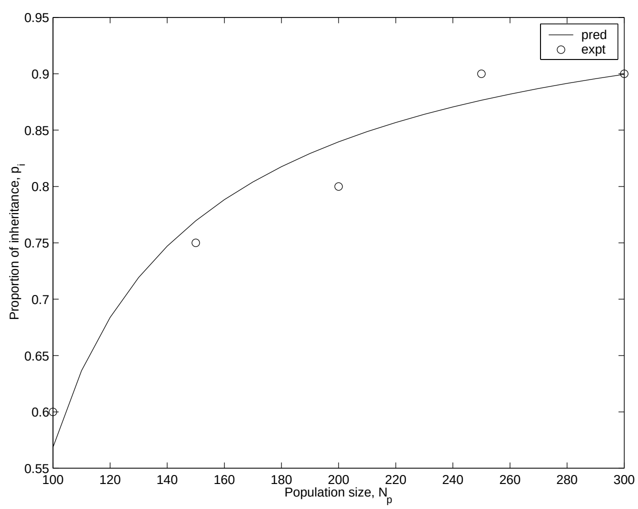

Figure 8: Apparent optimal inheritance proportion, p_i^{app}, predicted by eqn. (23) compared to empirical results. The empirical results depict the inheritance proportion that requires minimum number of function evaluations to converge to the optimal solution for a 100-bit OneMax problem. The experimental results are averaged over 50 runs. The value of κ is 81.63.

3.3. If we have a still higher population size, then the speed-up will be higher. This result agrees with that obtained by Smith, Dike, and Stegmann (1995), in which they had considered a 64 bit OneMax problem and taken the population size of 500. A 100-bit OneMax without inheritance requires a population size of about 80 which implies that a 64 bit problem would need a still lower population size. The reason why Smith, Dike, and Stegmann (1995) did not get an optimal proportion of inheritance was due to the fact that they took a very high population size and did not compare the minimum population size required for different proportions of inheritance. In other words, they did not consider the effect of fitness inheritance on population sizing.

6 Future Work

In the present study we have analyzed fitness inheritance for OneMax problems and the proposed model can be extended to other problems (eg., non-uniformly scaled problems). Further investigation is required for determining analytically the signal to noise-ratio used in the population sizing model. The inheritance procedure used in the present study is a simple one, and a study on more complex inheritance techniques still remains to be done.

7 Conclusions

In this paper, we have developed a theoretical basis for fitness inheritance, and derived models for convergence time and population sizing. This model has been analyzed and optimized for greatest speed-up and that yields savings on 20% in terms of number of function evaluations. Though by itself this speed-up value seems to be modest, it can be coupled with parallelism, time continuation, and other evaluation relaxation schemes. In such a scenario the effective speed-up obtained will be a product of all individual speed-ups and even a speed up of 1.2 can be important. We have been careful to include the effects of fitness inheritance on quality-duration theory in predicting the above results. However, GA practitioners usually fix the population size and then try inheritance. Under these conditions apparent speed-up is much greater (as high as 3.3), a result that agrees with the earlier empirical study of Smith, Dike, and Stegmann (1995).

Acknowledgments

This work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-00-1-0163. Research funding for this work was also provided by the National Science Foundation under grant DMI-9908252. Support was also provided by a grant from the U. S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0003. 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, the U. S. Army, or the U. S. Government.

Thanks are also due to Fernando Lobo, Ravi Srivastava, and Abhishek Sinha for their useful comments.

References

Bulmer, M. (1980). *The mathematical theory of quantitative genetics.* Oxford: Clarendon Press.

Goldberg, D., Deb, K., & Clark, J. (1992). Genetic Algorithms, Noise, and the Sizing of Populations. *Complex Systems*, *6*, 333–362.

Goldberg, D. E. (1999). *Evolutionary design by computers* (Chapter 4. The Race, the Hurdle, and the Sweet Spot: Lessons from Genetic Algorithms for the Automation of Design Innovation and Creativity, pp. 105–118). San Mateo, CA: Morgan Kaufmann.

Harik, G., Cantu-Paz, E., Goldberg, D., & Miller, B. (1997). The Gambler's Ruin Problem, Genetic Algorithms, and the Sizing of Populations. In Back, T., et al. (Eds.), *Proceedings of the IEEE International Convference on Evolutionary Computation* (pp. 7–12). Piscataway, NJ, USA: IEEE.

Miller, B., & Goldberg, D. (1996a). Genetic Algorithms, Selection Schemes, and the Varying Effects of Noise. *Evolutionary Computation*, *4*(2), 113–131.

Miller, B. L. (1997, May). *Noise, sampling, and efficient genetic algorithms.* Doctoral dissertation, University of Illinois at Urbana-Champaign, Urbana, IL.

Miller, B. L., & Goldberg, D. E. (1996b). Genetic Algorithms, Tournament Selection, and the Varying Effects of Noise. *Complex Systems*, *9*(3), 193–212.

Muhlenbein, H., & Schlierkamp-Voosen, D. (1993). Predictive Models for the Breeder Geneteic Algorithm: I. Continous Parameter Optimization. *Evolutionary Computation*, *1*(1), 25–49.

Smith, R., Dike, B., & Stegmann, S. (1995). Fitness Inheritance in Genetic Algorithms. In *Procedings of the ACM Symposium on Applied Computing* (pp. 345–350). New York, NY, USA: ACM.

Thierens, D. (1995). *Analysis and Design of Genetic Algorithms.* Doctoral dissertation, Katholieke Universiteit Leuven, Leuven, Belgium.

Zheng, X., Julstrom, B., & Cheng, W. (1997). Design of Vector Quantization Codebooks Using a Genetic Algorithm. In *Proceedings of the IEEE COnference on Evolutionary Computation, ICEC* (pp. 525–529). Picataway, NJ, USA: IEEE.