Fitness Inheritance in Multi-Objective Optimization

Jian-Hung Chen
Dept. of Information Engineering
Feng Chia University
Taichung 407, Taiwan
ROC
jh.chen@ieee.org

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

Shinn-Ying Ho
Dept. of Information Engineering
Feng Chia University
Taichung 407, Taiwan
ROC
syho@fcu.edu.tw

Kumara Sastry
Dept. of General Engineering
University of Illinois
at Urbana-Champaign
Urbana, IL 61801
ksastry@uiuc.edu

Abstract

In real-world multi-objective problems, the evaluation of objective functions usually requires a large amount of computation time. Moreover, due to the curse of dimensionality, solving multi-objective problems often requires much longer computation time than solving single-objective problems. Therefore, it is essential to develop efficiency enhancement techniques for solving multi-objective problems. This paper investigates fitness inheritance as a way to speed up multi-objective genetic and evolutionary algorithms. Convergence and population-sizing models are derived and compared with experimental results in two cases: fitness inheritance without fitness sharing and fitness inheritance with fitness sharing. Results show that the number of function evaluations can be reduced with the use of fitness inheritance.

1 INTRODUCTION

For many large-scale and real-world problems, the fitness evaluation in genetic and evolutionary algorithms may be a complex simulation, model or computation. Therefore, even this subquadratic number of function evaluations is rather high. This is especially the case in solving multi-objective problems. It is not only because the number of the objectives to be evaluated is increased, but also the curse of dimensionality may increase the convergence time of genetic algorithms (GAs). As a result, it is beneficial to utilize efficiency enhancement techniques (EETs) in multi-objective GAs.

In practice EETs have improved the performance of GAs. Many real-world applications of GAs usually use EETs to improve the speed, ranged from parallel computing, distributed computing, domain-specific knowledge, or cheaper fitness functions. Recently, Sastry (2001) proposed an analytical model for analyzing and predicting behavior of single-objective GAs with EETs. However, due to the popularity of multi-objective GAs, there is a need to investigate multi-objective GAs with EETs. In this paper, one EET called fitness inheritance is modeled and optimized for greatest speedup. In fitness inheritance, an offspring sometimes inherits a fitness value from its parents rather than through function evaluations.

The objective of this paper 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 multi-objective GA. This paper is organized in the following manner. Section 2 briefly reviews the past works on EETs and fitness sharing. Section 3 describes the bicriteria OneMax problem and fitness inheritance, and derives convergence-time and population-sizing models for multi-objective GAs with EETs, as well as the optimal proportion of inheritance, the speed-up. The experimental results on fitness inheritance with and without fitness sharing are presented in Section 4. The paper is concluded in Section 5.

2 BACKGROUND

As background information, a brief review of the fitness inheritance literature is first presented. Then, a brief summary on how to incorporate fitness inheritance in multi-objective GAs is provided. Since fitness inheritance with and without fitness sharing will be discussed in this paper, section 2.2 presents a brief summary on fitness sharing.

2.1 LITERATURE REVIEW

Smith, Dike and Stegmann (1995) proposed two ways of inheriting fitness, one by taking the average fitness of the two parents and the other by taking a weighted average of the fitness of the two parents. Their results indicated that GAs with fitness inheritance outperformed those without inheritance in both the OneMax and an aircraft routing problem. However, theoretical analysis in this paper was limited to considering a flywheel effect that arises in the schema theorem. Zheng, Julstrom, and Cheng (1997) used fitness inheritance for the design of vector quantization codebooks. A recent study by Sastry (2001, 2001a) developed a theoretical framework for analyzing fitness inheritance, and discussed how to determine the optimal proportion of fitness inheritance and speed-up of using fitness inheritance in single-objective GAs. However, until now, there is no study on using fitness inheritance for multi-objective GAs.

2.2 FITNESS INHERITANCE

In fitness inheritance, the fitness of all the individuals in the initial population are evaluated. Thereafter, the fitness of some proportion of individuals in the subsequent population is inherited. This proportion is called the *inheritance proportion*, p_i. The remaining individuals receive evaluated fitness. If none of the individuals receive inherited fitness (p_i = 0), all the individuals are evaluated as usual, then no speed-up will be obtained. On the other hand, if all the individuals receive inherited fitness (p_i = 1), it means that none of the individuals are evaluated. Thereafter, the fitness diversity in the population will vanish rapidly and the population will premature converged, so that GAs will fail to search the global optimum. As a result, it is important to choose an optimal inheritance proportion, so that maximum speed-up will be yielded. The flowchart of multi-objective GAs with fitness inheritance is shown in figure 1.

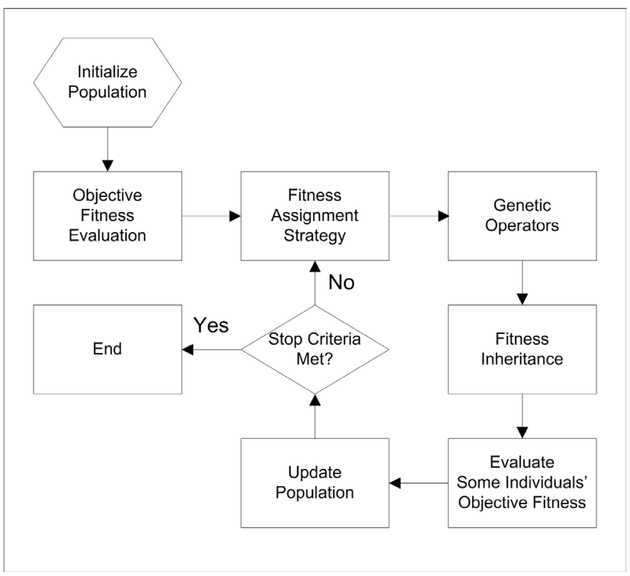

Figure 1: Fitness inheritance in multi-objective GAs.

There are several different ways to inherit fitness (objective fitness values), such as weighted-sum. For a multi-objective problem with z objective, fitness inheritance in multi-objective GAs can be defined as

$$f_z = \frac{w_1 f_{z,p1} + w_2 f_{z,p2}}{w_1 + w_2}, \quad (1)$$

where f_z is the fitness value in objective z, w_1, w_2 are the weights for the two parents p_1, p_2, and $f_{z,p1}$, $f_{z,p2}$ is the fitness values of p_1, p_2 in objective z, respectively. In practice, fitness inheritance can be performed on all the objectives or just several objectives.

In this paper, we assume that all the objective receives inherited fitness from the parents, and the inherited fitness (objective values) is taken to be the average of the two parents. Therefore, w_1 and w_2 are set to 1.

2.3 FITNESS SHARING REVISITED

Most multi-objective problems have multiple Pareto-optimal solutions. This usually causes difficulties to any optimization algorithm in finding the global optimum solutions. In prior GA literature, there have been many niching methods on how to promote and maintain population diversity. Fitness sharing, proposed by Goldberg and Richardson (1987), may be the most widely used niching method in solving multi-modal and multi-objective problems. The basic idea of fitness sharing is to degrade the fitness of similar solutions that causes population diversity pressure. The shared fitness of an individual i is given by

$$F_{sh,i} = \frac{F_i}{m_i}, \quad (2)$$

where F_i is the fitness of the individual, and m_i is the niche count, which defines the amount of overlap (sharing) of the individual i with the rest of the population. The niche count is calculated by summing a sharing function over all individuals of the population:

$$m_i = \sum_{j=1}^{n} sh(d_{i,j}). \quad (3)$$

The distance $d_{i,j}$ represents the distance between individual i and individual j in the population, determined by a similarity metric. The similarity metric can be based on either phenotype or genotype similarity. If the sharing function determines that the distance is within a fixed radius σ_{sh}, it returns a value, as equation (4).

$$sh(d_{i,j}) = \begin{cases} 1 - (\frac{d_{i,j}}{\sigma_{sh}})^{\alpha} & if\ d_{i,j} < \sigma_{sh}, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

The parameter α is usually set to 1. σ_{sh} is often conservatively estimated.

3 FITNESS INHERITANCE IN MULTI-OBJECTIVE OPTIMIZATION

In this section the bicriteria OneMax problem is extended from OneMax problem for analyzing multi-objective GAs with fitness inheritance. In this section, a brief summary of fitness inheritance is also presented.

3.1 BICRITERIA ONEMAX PROBLEM

The OneMax or bit-counting problem is well-known and well-studied in the context of GAs. The OneMax problem is a bit-counting problem where fitness value of each binary string is equal to the number of one bits in it. Accordingly, the optimum binary string is an all 1s string. The simplicity of the OneMax problem makes it a prime candidate to study the effect of fitness inheritance on the performance of GAs. In order to investigate the performance of multi-objective GAs with fitness inheritance, we develop the bicriteria OneMax problem for analyzing multi-objective GAs with fitness inheritance. The bicriteria OneMax problem is defined by

$$Maximize \begin{cases} f_1(s,x_1) = l - d(s,x_1) \\ f_2(s,x_2) = l - d(s,x_2) \end{cases}, \qquad (5)$$

where string s is the string to be evaluated, x_1, and x_2 are two fixed string, the string length is l, and $d(s, x)$ is the hamming distance of two string. If the fixed string x is all 1s string, then the corresponding objective function will be the OneMax problem. The number of Pareto-optimal solutions, m, in the bicriteria OneMax problem can be calculated by

$$m = 2^{d(x_1,x_2)}. \qquad (6)$$

In this paper, unless otherwise mentioned, x_1 is all 1s string, and x_2 is all 1s string except the first four bits of x_2 is 0s.

3.2 TIME TO CONVERGENCE

In this section we derive convergence-time model for the bicriteria OneMax problem with fitness inheritance. For OneMax domain, the convergence model can be derived by using the response to selection equation (Mühlenbein and Schlierkamp-Voosen ,1993),

$$\Delta f = f_{t+1} - f_t = I\sigma_f. \qquad (7)$$

This equation was derived by calculating the difference in mean fitness of two populations using the selection intensity I, the population's fitness variance σ_f^2 at time t.

Sastry (2001) extended this model for fitness inheritance in single-objective GAs. This population-sizing model derived by Sastry is reproduced below:

$$\Delta f = f_{t+1} - f_t = I\sqrt{1-p_i}\,\sigma_f \qquad (8)$$

Now, we can proceed to derive the convergence model for the bicriteria OneMax problem by extending equation (8). Based on the concept of fitness sharing, assumed that the population were divided into several subpopulations (niches), and each niche optimizes its own separate OneMax problem. Therefore, the optimizing process for the bicritiera OneMax problem can be regarded as optimizing several OneMax problems simultaneously. Since niches are from the same population, each niche will receive external noise from other niches. As a result, we can use the OneMax model with noisy fitness functions (Miller, 1997) to predict convergence time in the presence of external noise caused by niches. For each niche, the convergence model for the bicriteria OneMax problem can be expressed as

$$\Delta f = f_{t+1} - f_t = I\sqrt{1-p_i}\,\frac{\sigma_f^2}{\sqrt{\sigma_f^2+\sigma_N^2}}, \qquad (9)$$

where σ_N^2 is the noise variance from other niches.

Let M be the number of niches in the population, and

$$\rho_e = \frac{\sqrt{\sigma_f^2+\sigma_N^2}}{\sigma_f^2}.$$

Assumed that each niche has same proportion of correct BBs, let p_t be the proportion of correct BBs in the niche at generation t. For the OneMax domain, the mean fitness at generation t equals lp_t, the fitness variance can be approximated by $lp_t(1-p_t)$, and the noise variance from other niches can be approximated by $(M-1)\,p_t(1-p_t)$. The population is converged to optimal when $p_t = 1$. Equation (9) now yields

$$p_{t+1} - p_t = \frac{I}{\rho_e}\sqrt{\frac{(1-p_i)}{l}}\sqrt{p_t(1-p_t)}. \qquad (10)$$

Approximating the above equation with a differential equation and integrating this equation using the initial condition $p|_{t=0} = 0.5$, we get

$$p_t = \sin^2\left(\frac{\pi}{4} + \frac{It\sqrt{(1-p_i)}}{2\rho_e\sqrt{l}}\right). \qquad (11)$$

Then we can derive an equation for convergence time, t_{conv}, by equating $p_t = 1$, and inverting equation (11),

$$t_{conv} = \frac{\pi}{2I}\sqrt{\frac{l}{(1-p_i)}}\rho_e. \qquad (12)$$

Finally, we can yield

$$t_{conv} = \frac{\pi}{2I}\sqrt{\frac{l}{(1-p_i)}}\sqrt{1+\frac{M-1}{l}}. \qquad (13)$$

If p_i is taken as 0, and M is taken as 1, then the above relation reduces to

$$t_{conv} = \frac{\pi\sqrt{l}}{2I}, \qquad (14)$$

which agrees with existing convergence-time models for the OneMax problem.

Generally, M can be set to the number of niches in the population or the number of Pareto-optimal solutions in equation (13). However, it is difficult to determine M, because niches are often overlapped in the real-world problems, and the number of niches in the population is always varied in the real runs of GAs with fitness sharing. The convergence-time model will be examined and compared with experiments in the later section.

3.3 POPULATION SIZING

Selecting a conservative population size reduces the chance of premature convergence, and it also influences the quality of the solution obtained. Therefore, it is important to appropriately size the population to incorporate the effects of fitness inheritance. For the OneMax problem, the Gambler's Ruin population-sizing model (Harik et al., 1997) can be used to determine the population-sizing model. Sastry (2001) extend this model for fitness inheritance. This population-sizing model derived by Sastry is

$$n = -\frac{2^{k-1}\ln(\psi)\sqrt{\pi}}{(1-p_i^3)}\sqrt{\sigma_f^2}\,, \qquad (15)$$

where n is the population size, k is the building block (BB) length, ψ is the failure rate, and σ_f is the variance of the noisy fitness function. For an OneMax with string length 100, k = 1, $\sigma_f^2 = 25$.

Assuming the population were divided into M niches, and each niche optimizes for its own separate OneMax problem. Similar to the population-sizing model for the bicriteria OneMax problem, we can extend this model by using the OneMax model with noisy fitness functions (Miller, 1997) to predict population-sizing in the presence of external noise caused by niches. The population model for the bicriteria OneMax problem can be written as

$$n = -\frac{2^{k-1}\ln(\psi)M\sqrt{\pi}}{(1-p_i^3)}\sqrt{\sigma_f^2+\sigma_N^2}\,, \qquad (16)$$

where σ_N^2 is the noise variance from other niches, and M is the number of niches.

The population-sizing model will be examined and compared with experiments in the later section.

3.4 OPTIMAL INHERITANCE PROPORTION AND SPEED-UP

Given a problem there should be a range of inheritance proportions that are more efficient than the others. An inappropriate inheritance proportions would not reduce the number of function evaluations. For large sized problems, Sastry's study indicates that the optimal inheritance proportion, p_i, lies between 0.54 -0.558. The total number of function evaluations required can be calculated by

$$N_{fe} = n\left[t_{conv}\left(1-p_i\right)+p_i\right]. \qquad (17)$$

From the equation (10) and equation (13), we can the predicted the total number of function evaluations required, as shown in figure 1.

The speed-up of fitness inheritance is defined as the ratio of number of function evaluations with $p_i = 0$ to the number of function evaluation at optimal p_i. From the practical view, a user usually fixes the population size and then optimizes the proportion of fitness inheritance. Therefore, the optimal proportion of fitness inheritance with a fixed number of population size can be obtained by the inverse of equation (16).

$$p_i^* = \sqrt[3]{1-\frac{\kappa}{n}}\,, \qquad (18)$$

where $\kappa = -2^{k-1}\ln(\psi)M\sqrt{\pi(\sigma_f^2+\sigma_N^2)}$. Equation (18) indicates that if the population is larger than κ, the larger the population size, the higher of inheritance proportion can be used.

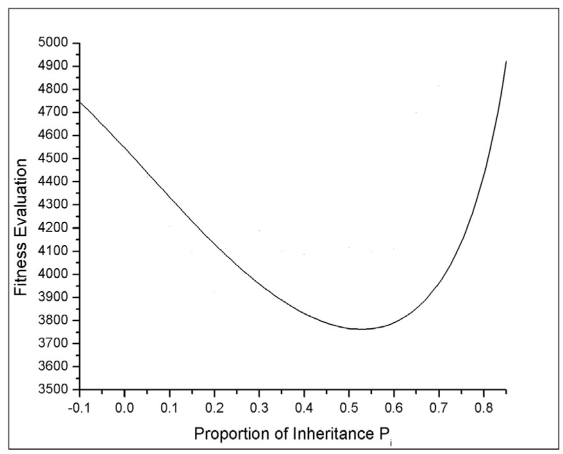

Figure 1: Total number of function evaluations predicted by equation (17) with a failure rate of 0.0001.

4 EXPERIMENTS AND RESULTS

The experiments were performed using *selectorecombinative* GAs with binary tournament selection, and uniform crossover with crossover probability of 1.0. No mutation operator is used. The sharing factor σ_{sh} is set to 50. The fitness assignment strategy we used is proposed by Ho (1999), is defined by

$$F(X) = p - q + c\,, \qquad (19)$$

where p is the number of individuals which can be dominated by the individual X, and q is the number of individuals which can dominate the individual X in the objective space. To ensure a positive fitness value, a constant c is added. Generally, the constant c can be assigned using the number of all participant individuals.

All experiments were performed 30 runs using the 100-bit bicriteria OneMax problem.

As to M in equation (13) and equation (16), considering the bicriteria OneMax problem and assuming perfect niching, M can be set to 2. Because better mixing of BBs is able to generate other Pareto-optimal solutions from x_1 and x_2. It should be an approximated lower-bound for the comparison with experimental results. However, it is noted that, in the real runs of GAs with fitness sharing, M is varied in the population. Therefore, equation (13) and equation (17) is also varied.

In order to investigate multi-objective GAs with fitness inheritance, two kind of experiments, ftiness inheritance without fitness sharing and fitness inheritance with fitness sharing, were performed and compared with analytical results. However, since multi-objective GAs without fitness sharing may lead to only some niches. Therefore, for fitness inheritance without fitness sharing, the algorithm used an external non-dominated set to store the non-dominated solutions during its search process.

4.1 FITNESS INHERITANCE WITHOUT FITNESS SHARING

The convergence time observed experimentally is compared to the above prediction for a 100-bit bicriteria OneMax problem in figure 2. Although fitness sharing was not used, the results indicate fitness inheritance is able to find all the Pareto-optimal solutions during the search process. The discrepancy between the empirical and analytical results may due to some niches disappear out of the population. Therefore, multi-objective GAs will focus the search on the remaining niches. When there is only one niche left, it lead to that all the population is optimizing an OneMax problem.

The population-sizing model is compared to the results of 100-bit OneMax problem and the results obtained for a 100-bit bicriteria OneMax problem and in figure 3. From the plot it can be easily seen that when the proportion of fitness inheritance is smaller than 0.4, our population-sizing model fits the experimental result accurately. However, when the proportion of fitness inheritance is bigger than 0.4, the experiments results get closer to the analytical results of the OneMax problem. It is because when the proportion of inheritance is higher, the diversity of population becomes lesser. So that the search was focused on the remaining niches when some niches disappeared during the search process. As a result, the convergence time of fitness inheritance without fitness sharing is varied and may be lower then the analytical results predicted by equation (13).

By using an appropriate population size and proportion of fitness inheritance and from the equation (13) and equation (16), we can the predicted the total number of function evaluations required and compared with experimental results, as shown in figure 4. The above results indicates the optimal inheritance proportion lies between 0.6 – 0.8 for fitness inheritance without fitness sharing. The speed-up is around 1.4. In other words, the number of function evaluations with inheritance is around 40% less than that without inheritance. This implies that we can get a moderate advantage by using fitness inheritance. The discrepancy between our results and Sastry's study occurs due to the disappearance of niches.

Considering the fixed population size, the speed-up is different to the speed-up obtained above. From figure 5, it can be seen that if the population size is 2000, then fitness inheritance can yield a speed-up of 3.4. The result agrees with that obtained by Sastry (2001).

Figure 2: Convergence time for a 100-bit bicriteria OneMax problem for different proportion of inheritance predicted by equation (13) compared to experimental results.

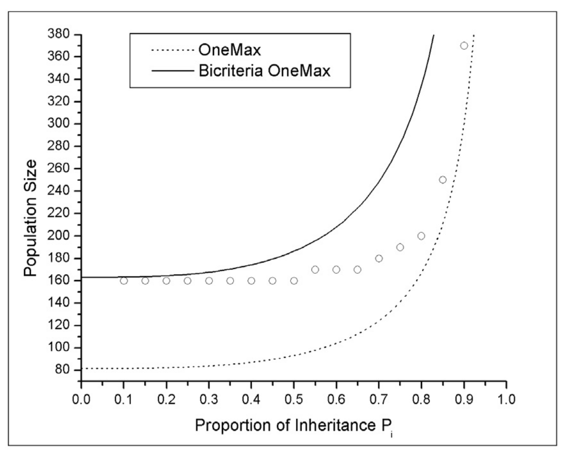

Figure 3: Verification of the population-sizing model for various inheritance proportions with empirical results. The curves are analytical results of Onemax problem and bicriteria OneMax problem, respectively. Experimental results depict the population size required for optimal convergence with failure rate of 0.0001.

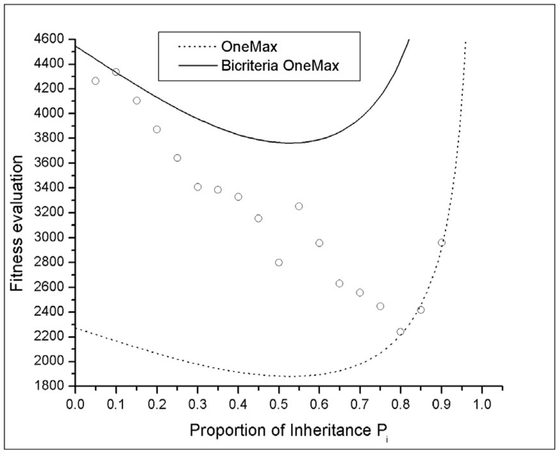

Figure 4: Total number of function evaluations predicted by equation (17) compared to experimental results. The curves are the analytical results of 100-bit Onemax problem and 100-bit bicriteria OneMax problem, respectively.

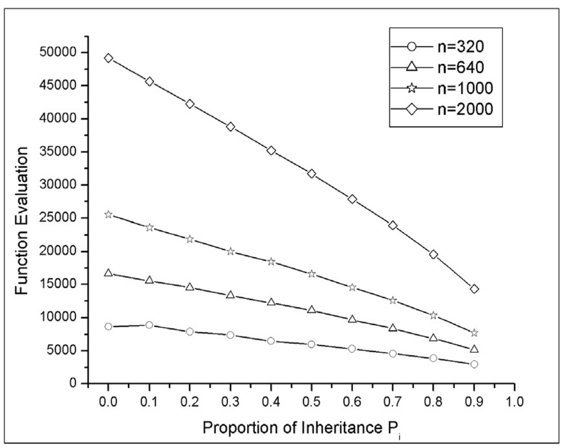

Figure 5: Total number of function evaluations for various proportion of fitness inheritance at different population sizes..

4.2 FITNESS INHERITANCE WITH FITNESS SHARING

In section 4.2, the experiments were performed using fitness inheritance with fitness sharing. The external non-dominated set was not used.

Recalling the definition of fitness sharing in section 2.3, we know that fitness sharing will degrade the fitness of similar individuals, so that these individuals will have smaller opportunity to be selected into the next generation. However, considering fitness inheritance with fitness sharing, an individual inherits fitness (objective value) from its parents. So the objective values are approximated. Then the dummy fitness is assigned according to the approximated objective values. Therefore, the dummy fitness is also approximated. Apparently, if some individuals are over-estimated and receive better fitness than their actual fitness, fitness sharing will also maintain these individuals. As a result, when fitness inheritance is used with fitness sharing, we expect that over-estimated individuals are likely to survive in the population and affect other solutions as the proportion of inheritance increased.

Figure 6 and figure 7 present the convergence model and population-sizing model observed for 100-bit bicriteria OneMax problem using fitness inheritance with fitness sharing. When the inheritance proportion is smaller than 0.7, the experimental results fit the predicted convergence model and population-sizing model. However, when the inheritance proportion is bigger than 0.8, GAs with fitness inheritance and fitness sharing cannot converge to all the Pareto-optimal solutions.

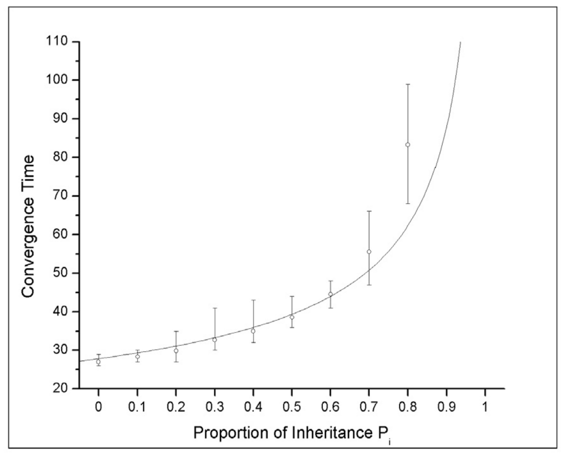

Figure 6: Convergence time for different proportion of inheritance predicted by equation (13) compared to experimental results using fitness inheritance with fitness sharing.

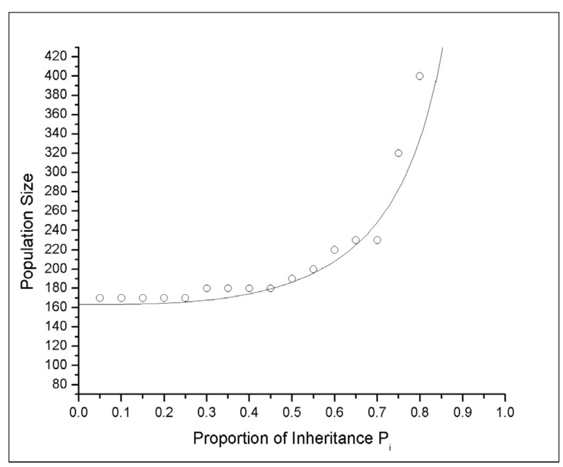

Figure 7: Verification of the population-sizing model for fitness inheritance with fitness sharing compared with empirical results. Experimental results depict the population size required for optimal convergence with failure rate of 0.0001.

Figure 8 presents the distance to Pareto front of both actual and inherited fitness for the experimental results with inheritance proportion 0.9. It indicates that the search process was divided into two phases. In this first phase, fitness inheritance proceeded well. The second phase started around the 40^{th} generation. Some individuals were approximated to better fitness and maintained by fitness sharing. Due to the high inheritance proportion, these inferior individuals mixed with other individuals. Finally the population was filled with incorrect individuals. This phenomenon explains the discrepancy between empirical and analytical results in figure 6.

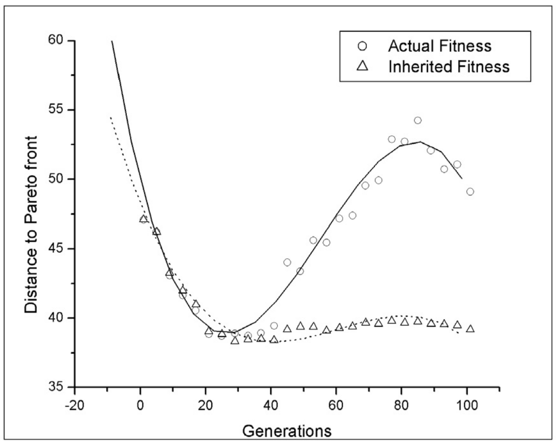

Figure 8: The distance to the Pareto front of actual fitness and inherited fitness for the experimental results with inheritance proportion 0.9. The empirical results are averaged over 30 runs.

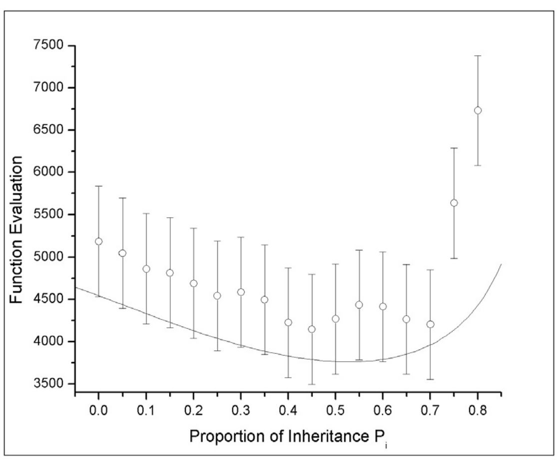

Figure 9: The distribution of function evaluations. The curve is the total number of function evaluations predicted by equation (17) for optimal convergence of a 100-bit bicriteria OneMax problem with a failure rate of 0.0001.

The predicted number of function evaluations is compared with experimental results in figure 9. The speed-up is around 1.25. The discrepancy between our results and analytical results may due to the number of niches, M, is varied in the real runs of GAs with fitness sharing. some inferior individuals are maintained by fitness sharing, and then mixed with other niches. Therefore, more function evaluation times are required. This may be the overhead in using GAs with fitness sharing.

In summary, the experimental results of fitness inheritance with fitness sharing indicate that the proportion of inheritance lies between 0.4 -0.5, so that incorrect niches will have lesser chance to be maintained by fitness sharing. The result is slightly different to the optimal proportion of inheritance derived by Sastry.

5 CONCLUSIONS

In this paper, we have developed a bicriteria OneMax problem and derived models for convergence-time and population-sizing. The models have been analyzed in two cases: fitness inheritance without fitness sharing and fitness inheritance with fitness sharing. In the first case, fitness inheritance yields saving on 40% in terms of the number of function evaluations. While using a fixed number of population size, fitness inheritance can yield a speed up of 3.4. In the second case, fitness inheritance yields saving to 25%.

Though the speed-up of fitness inheritance seems to be modest, it can be incorporated with parallelism, time continuation, and other efficiency enhancement techniques. In such case, a speed up of 1.25 can be important.

Further studies on using complex inheritance techniques and incorporating fitness inheritance with state-of-the-art multi-objective genetic algorithms are still remains to be done.

Acknowledgments

The authors would like to thank to Tian-Li Yu and Ying-Ping Chen for many useful comments. The first author wishes to thank the third author for his encouragement to visit the Illinois Genetic Algorithms Laboratory (IlliGAL). This paper was done during the visit.

The first author is supported by Taiwan Government Funds of Ministry of Education.

This work was partially sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grants F49620-97-0050 AND F49620-00-0163. Research funding for this work was also provided by a grant from the National Science Foundation under grant DMI-9908252. The US 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, that National Science Foundation, or the U.S. Government.

References

Goldberg D. E., & Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In Grefenstette, J. J. (Eds.), *Proceedings of the Second International Conference on Genetic Algorithms*, 41-49. Hillsdale, NJ: Lawrence Erlbaum Associates.

Harik, G., Cantu-Paz, E., Goldberg, D. E., & 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 Conference on Evolutionary Computation*, 7-12, Piscateaway, NJ, USA: IEEE.

Ho, S.-Y. and Chang, X.-I., (1999). An efficient generalized multiobjective evolutionary algorithm. In *Proceeding of Genetic and Evolutionary Computation Conference*, 871-878. Orlando, FL, USA: Morgan Kaufmann.

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

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

Smith, R. Dike, B., & Stegmann, S. (1995). Fitness inheritance in genetic algorithms. In *Proceedings of the ACM Symposium on Applied Computing*, 345-350. New York, Y, USA: ACM.

Sastry, K. (2001). *Evaluation-relaxation schemes for genetic and evolutionary algorithms*. Master dissertation, University of Illinois at Urbana-Champaign, Urbana, IL.

Sastry, K., D. E. Goldberg & M. Pelikan (2001a). Don't evaluate, inherit. In *Proceeding of Genetic and Evolutionary Computation Conference*, 551-558. San Francisco, CA, USA: Morgan Kaufmann.

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*, 525-529. Picataway, NJ, USA: IEEE.