Modeling Idealized Bounding Cases of Parallel Genetic Algorithms

Erick Cantú-Paz and David E. Goldberg
Illinois Genetic Algorithms Laboratory
University of Illinois at Urbana-Champaign
104 South Mathews Avenue, Urbana, IL 61801
cantupaz,deg@illigal.ge.uiuc.edu

Abstract

This paper presents models to predict the quality of convergence of idealized bounding cases of parallel genetic algorithms (GAs). The first bounding case is a parallel GA with completely isolated subpopulations (demes). We show how the probability that the parallel GA finds a solution of the minimum desired quality increases as more demes are used. Our second bounding case considers that each deme communicates with every other deme with a maximal migration rate. Our models predict the probability that a locus converges to the correct value and are based on a previous model for simple GAs. For each of the bounding cases, we derive equations to determine the deme size that is required when the quality of the solution and the number of demes are fixed.

1 Introduction

The practical benefits of parallel genetic algorithms (GAs) have been recognized for a long time, but their design has many difficult and interrelated problems. The three main problems are to determine (1) the size and the number of demes (subpopulations), (2) the topology that interconnects the demes, and (3) the migration rate that controls how many individuals migrate at any given time.

Most studies of parallel GAs are empirical investigations that concentrate on the choices of topology and migration rates and treat the population sizing issue as a secondary problem. With this work we take a different approach and seek to contribute to the theoretical understanding of parallel GAs by attacking first the deme sizing problem. Previous studies with serial GAs show that the population size is a major factor in determining the quality of the solution (Goldberg, Deb, & Clark, 1992; Harik, Cantú-Paz, Goldberg, & Miller, 1996), so it is only natural that our study of parallel GAs begins addressing this issue.

To make progress in our study of parallel GAs we must restrict in some ways the choices for topologies and migration rates, but we must not ignore them. To this effect, we study deme sizing by considering two idealized bounding cases for topologies and migration rates. In this way, the models derived show the limits on quality and cost that can be expected from parallel GAs.

The first bounding case is a completely isolated topology where the demes do not communicate with each other. The model for this case is a straightforward extension of a model developed recently that predicts the convergence quality of a simple GA (Harik, Cantú-Paz, Goldberg, & Miller, 1996). The second bounding case is a fully connected topology where each deme is connected to all the others, and the migration rate is set to a maximum value. In our analysis we ignore the effect of the communication time, as it does not affect the quality of the solution. This simplification makes our bounding cases (specifically the second) be idealizations of the real algorithms, but our analysis of these idealized cases discovers important relationships among population sizes, deme counts, and the quality of solutions.

This study begins with a brief review of relevant research in parallel GAs in the next section. Then, in section 3 we present a model that predicts the performance of a simple GA and that constitutes the base of our bounding models. Section 4 extends the single-population model to consider the first bounding case with isolated demes. In section 5 we derive a model that considers completely connected demes with a maximal rate of migration. Finally, we present suggestions for further research and the conclusions of this work.

2 Background

An important concept for the study of GAs is the notion of building blocks (BBs). Following Goldberg, Deb, and Thierens (Goldberg, Deb, & Thierens, 1993), we restrict the definition of BBs to the minimal-order schemata that contribute to the global optimum. For example, for fully deceptive functions with deception length k, the BBs are the optimal schemata of length k. In this view the juxtaposition of two BBs at a particular string does not lead to a BB of length $2k$, but instead we regard them as two separate BBs of length k.

This paper considers a class of parallel GAs with multiple discrete subpopulations or demes. Normally, there is some exchange of individuals between the demes at fixed intervals, but sometimes the demes may run in complete isolation. This class of parallel GAs is often referred to as "coarse-grained" parallel GAs or "distributed GAs", because it is often implemented on distributed memory computers and the communication to computation ratio is low.

Since it is impossible to cite here all the relevant publications on coarse-grained parallel GAs, we must restrict our review to a few representative papers. Further examples of coarse-grained implementations of parallel GAs can be found in the review by Cantú-Paz (1995), along with descriptions and examples of other types of parallel GAs.

Grefenstette was probably the first to discuss how to implement parallel GAs on existing computer architectures (Grefenstette, 1981), and he proposed four prototypes. The first three prototypes only had one population and they consisted on variants of a master-slave scheme. The purpose of these prototypes was to distribute the work of evaluating the population among several processors. Grefenstette's fourth prototype is a coarse grained parallel GA like the ones we study in this paper.

Grosso (1985) was the first to observe some of the phenomena that continue to be reported in the literature. He found that the rate of fitness improvement was faster in the divided population than in a single large population. But, when the demes are isolated, this rapid rise in fitness stops at a lower fitness value than with the single large population. Grosso experimented with a wide range of migration levels and observed that with intermediate migration levels the partitioned population reached the same fitness value as the panmictic population. In his scheme, migrations occurred every generation and the migrants and their destinations were selected randomly. Grosso also experimented with a "delayed" migration scheme that consists in running the simulation without communication until the demes have a significant convergence and then using a high migration rate. This scheme reduces communication costs and results in solutions of the same quality as those obtained with the first migration method.

Tanese's dissertation (Tanese, 1989) has become an important reference in the study of parallel GAs mainly because she asked very important questions. Tanese experimented using many small populations and found that the quality of the solution may be similar to that found using few large populations, even in the case of isolated runs. She also addressed the questions of migration rates and frequency of migrations, and discovered that these two factors influence the performance of the algorithm.

3 Single population GAs

The models for predicting the convergence quality of parallel GAs presented in this paper are based on a model for serial GAs developed recently (Harik, Cantú-Paz, Goldberg, & Miller, 1996). This section reviews briefly the results of that analysis.

3.1 Deciding well between two building blocks

An important piece of the model for single-population GAs is the probability of deciding well between two competing building blocks in a partition. Goldberg, Deb, and Clark (1992) cast this process as a statistical decision-making problem in the presence of noise, and find a domain-dependent probability of deciding in favor of the correct BB. We summarize that result in the next paragraphs.

Consider two BBs H_1 (with mean fitness f_{H_1} and fitness variance $\sigma^2_{H_1}$) and H_2 (with mean fitness f_{H_2} and fitness variance $\sigma^2_{H_2}$). Assuming that H_1 has a higher fitness, we hope to select more representatives of H_1 than of H_2. However, in a one-on-one competition there is a chance of choosing erroneously a representative of H_2, because there is noise resulting from the signal of the other $m-1$ partitions.

Defining the signal difference $d = f_{H_1} - f_{H_2}$, the probability of making the correct decision on a single trial may be calculated as:

$$p = \mathbb{N}\left(\frac{d}{\sqrt{2(m-1)}\sigma_{bb}}\right), \tag{1}$$

where $\mathbb{N}$ is the cumulative distribution function (CDF) of a normal distribution with zero mean and unit standard deviation, and σ_{bb} is the average BB variance.

3.2 The gambler's ruin model

Random walks are mathematical tools that can be used to predict the outcome of certain stochastic processes, and, in particular, can be used to model the selection phase of a GA. To model selection we concentrate on only one allele and assume that decisions are independent across alleles. The number of BBs correct is represented by the position, x, of a particle on a one-dimensional space. Absorbing barriers at $x = 0$ and $x = n$ represent convergence to the wrong and right solutions, respectively. The initial position of the particle, x_0, is the expected number of copies of the best BB in a population initialized randomly, that is equal to $x_0 = n/2^k$, where k is the order of the BB. The probability p that the GA gets an additional copy of the correct BB is the probability of deciding well between BBs as given by equation 1.

The probability that a locus converges to the correct value of the BB can be approximated as follows (Harik, Cantú-Paz, Goldberg, & Miller, 1996):

$$P_{bb} \approx 1 - \left(\frac{q}{p}\right)^{n/2^k}, \tag{2}$$

where p is the probability that the correct BB is selected and $q = 1 - p$ is the probability of making the wrong decision. From this form of P_{bb} it is straightforward to compute the population size. Making $P_{bb} = b/m$ and solving for n results in the population size needed to obtain a solution with an average of b out of m BBs right,

$$n = \frac{\ln(1 - \frac{b}{m})2^k}{\ln\left(\frac{q}{p}\right)}. \tag{3}$$

In the next section we extend the gambler's ruin problem to account for the first bounding case in our study of parallel GAs.

4 Isolated demes

The first case of parallel GAs that we examine considers that the demes evolve in parallel with no communication between them. This case bounds the topology because there is no connectivity between the demes, and also bounds the migration rate as it is set to zero, its lowest value. We start this section presenting the criterion for success for the case of multiple demes and we compute the probability of achieving it. Next, we derive an estimate of the population size needed to find the desired solution. Finally, the section concludes with experiments on a simulated parallel environment to verify the theoretical predictions.

4.1 Probability of success

In this section we use a framework set forth by Nakano, Davidor, and Yamada (1994) and consider that the parallel GA succeeds when at least one of the demes finds the solution with the minimum required quality. To compute the probability of success let P_m be the probability that one deme finds the desired solution, then $(1 - P_m)^r$ is the probability that none of the demes succeeds (in the paper by Nakano, Davidor, and Yamada (1994) the r stands for the number of independent runs, in this paper we use it to denote the number of demes). Thus, the overall success probability for multiple demes is

$$P_s = 1 - (1 - P_m)^r. \tag{4}$$

To compute the probability, P_m, that a GA will converge to a solution with *at least* b of m BBs correct we may use the CDF of a binomial distribution ($\mathbb{B}$) with parameters m and P_{bb},

$$P_m = 1 - \mathbb{B}(m, P_{bb}), \tag{5}$$

because the success probability in each of the m independent partitions is P_{bb}.

In the next section, we use the probability of success P_s to find the population size needed to find a solution of a minimum quality with r isolated demes.

4.2 Isolated population size

Recall that to compute the average population size needed by the simple GA to find a solution with b out of m BBs correct, we need to find a n that makes $P_{bb}(n) = b/m$. Alternatively, we could find an n such that $P_m(b, m, n) = 0.5$ since the binomial distribution is symmetrical around the mean. Note, however, that we can set P_m to any value, δ, between 0 and 1 in order to find the population size that with probability δ results in a solution of quality b/m.

In this section, our objective is to find the population size that a parallel GA with isolated demes needs to find a solution of the same quality as the serial GA. Of course, we also have to pay attention to the probability of finding the solution, and, for our results to be meaningful, we need to compare the two algorithms at the same level of reliability. Since we already derived the population size for serial GAs that makes $P_m = 0.5$, we use that same reliability level ($\delta = 0.5$) in the following discussion for parallel GAs.

To obtain the deme size in the isolated case, we need to solve $P_s = \delta = 0.5$ for n_d. We do this in several steps, since the deme size is buried deep inside P_s. First, using equation 4 we compute the value of P_m that makes $P_s = 0.5$, and call it P_m^*

$$P_m^* = 1 - \exp\left(\frac{\ln 0.5}{r}\right). \tag{6}$$

Recall that P_m has a binomial distribution with mean mP_{bb} and variance $mP_{bb}(1 - P_{bb})$. To make any progress we need to approximate the binomial distribution with a normal distribution. The number of BBs correct is normalized as $z = \frac{b - mP_{bb}}{\sqrt{mP_{bb}(1-P_{bb})}}$. Therefore, $P_m \approx 1 - \mathbb{N}(z)$ where $\mathbb{N}$ is the CDF of a normal distribution with zero mean and unit standard deviation. Furthermore, we can approximate the CDF of the normal distribution (Abramovitz & Stegun, 1972) to get

$$P_m = \frac{1}{2} - \frac{(1 - \exp(-2z^2/\pi))^{1/2}}{2}. \tag{7}$$

Now, using this equation we can find the ordinate (z) of a unit normal distribution that makes $P_s = 0.5$ as

$$z^* = \left[-\frac{\pi}{2} \ln\left(1 - (1 - 2P_m^*)^2\right)\right]^{1/2}. \tag{8}$$

Since we know that $z = \frac{b - mP_{bb}}{\sqrt{mP_{bb}(1-P_{bb})}}$ we can now solve for the value of P_{bb} that makes $P_s = 0.5$ and call it P_{bb}^*

$$P_{bb}^* = \frac{z^{*2}m + 2bm - \sqrt{z^{*2}m(-4b^2 + z^{*2}m + 4bm)}}{2m(z^{*2} + m)}. \tag{9}$$

Finally, making $P_{bb}^* = P_{bb} = 1 - \left(\frac{p}{q}\right)^{n_d/2^k}$, we can solve for the multiple-run population size

$$n_d = \frac{2^k \ln(1 - P_{bb}^*)}{\ln(q/p)}. \tag{10}$$

Before proceeding to examine the other bounding case where the demes communicate with all others, we present empirical evidence that validates our results for the isolated demes case.

4.3 Experiments

For all the experiments in this paper we use a simple GA with binary tournament selection and a crossover operator without excessive disruption. As such, the crossover operator was domain-dependent and we specify our choice as we present each problem. In all cases, the crossover probability is set to 1.0 (to promote adequate exchange of BBs) and the mutation probability is set to zero (the model assumes that all variability comes from the initial random population). The results presented are the average over 30 independent runs.

The experiments were performed on a simulated parallel environment consisting of eight IBM RS6000 workstations connected by a 10 Mbps Ethernet. A virtual parallel computer was simulated using PVM 3.3.

The first test problem is a one-max function with $m = 100$ bits. In this domain the fitness of the individuals is simply the number of ones in their strings. We use uniform crossover as it provides good mixing of BBs and we are unconcerned about BB disruption because the BB have length equal to one.

For the one-max function, the signal difference d equals 1 and the variance $\sigma_{bb}^2 = 0.25$ (see (Goldberg, Deb, & Clark, 1992)). This is a very easy problem for GAs since there is no isolation or deception and the BB are short. The supply of BBs is no problem either since in a population initialized randomly we expect to find 50% of correct BBs. In figure 1 we compare the theoretical predictions for the probability of success with experimental results using 1, 2, 4, and 8 demes. In each case, we demanded solutions with 90 BBs correct.

Our second test function is the 4-bit trap function that was also used by Goldberg, Deb, and Clark (1992) in their study of population sizing. As in the one-max, the value of this function depends on the number of bits set to one, but in this case the fitness increases with more bits set to zero until it reaches a local (deceptive) optimum when the four bits are zero. The global maximum of the function occurs precisely at the opposite extreme where all four bits are set to one, so an algorithm cannot use any partial information to find it. The signal difference d (the difference between the optimal and the deceptive maxima) is 1 and the fitness variance (σ_{bb}^2) is 1.215. Our test function is formed concatenating $m = 20$ copies of the trap function for a total string length of 80 bits.

Since the BBs in this function are longer than in the one-max problem, we chose a two-point crossover operator to avoid the excessive disruption of uniform crossover. Figure 2 presents the prediction of the

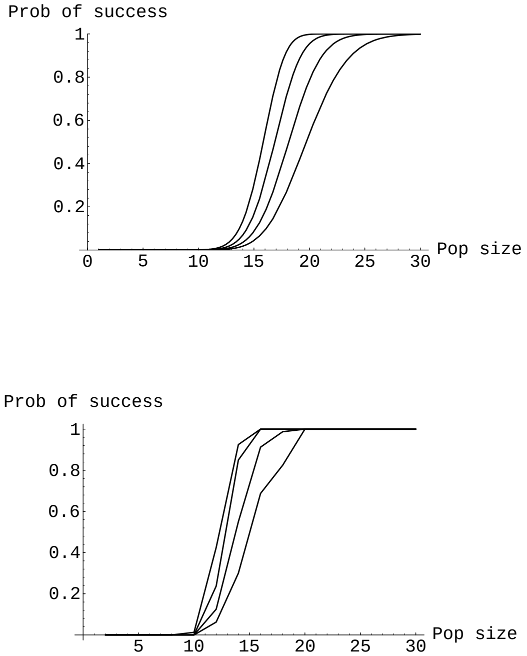

Figure 1: Results comparing theory and experimental results for the probability of finding 90 BBs correct using a one-max function with 100 BBs.

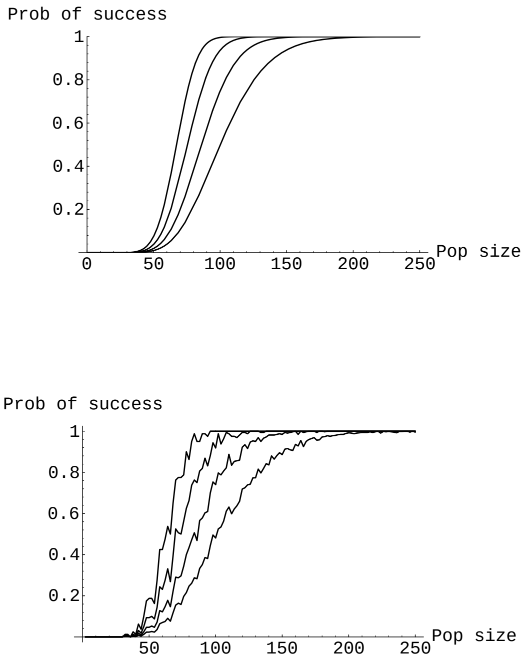

Figure 2: Results comparing theory and experimental results for the probability of finding 16 BBs using a 4-bit deceptive function with 20 BBs.

probability of success along with the results from the experiments. For this function, the quality that we demanded was 16 BBs.

The experiments with the onemax and the 4-bit trap functions show that the predictions from the model approximate closely the performance of the parallel GA. The differences that can be observed in the figures are due to small inaccuracies of the gambler's ruin model that are accentuated when one demands multiple correct BBs, as is our case. In the next section we use the gambler's ruin model again to predict the convergence quality with completely connected demes.

5 Completely connected demes

In the previous section we computed the probability of success and the population size needed to find a solution of a certain quality when multiple isolated demes are used. Executing demes in complete isolation is clearly a bounding case for parallel GAs, as the migration rate is zero and there is no connectivity between the demes. In this section we shift focus to the opposite extreme of the connectivity and migration rate spectrum, and let every deme exchange individuals with all the others with a maximum migration rate.

In this study we consider that migration occurs after all the demes have converged (i.e., all the individuals in each deme are identical). Then, every deme exchanges n/r individuals with all the other demes, and after migration the GA continues to execute until it converges again. We refer to the time before the first convergence as the first *iteration* of the GA. The second iteration starts after migration and finishes when the algorithm converges again.

This method of migration is more efficient in terms of communication costs than other methods that migrate individuals every few generations. Also, migrating after (or near) convergence has been shown empirically to have a similar effect on the solution quality as the frequent migration methods. For example, Grosso (1985) describes an experiment where migration is not activated until late in the run, and the quality of the results is similar to other experiments where migration is activated from the beginning. Also, note that communication costs can be dramatically reduced if migration is done after convergence because all individuals are the same: it is sufficient to send just one individual and replicate it at the receiving deme any number of times.

We follow the same methodology as in the section for isolated demes: first, we compute the probability of success and from there we derive a population sizing equation. Also, we validate experimentally the results of this section.

5.1 Success probability and population size

Recall that the probability that a locus converges to the correct value in a GA with a single population is given by the solution to the gambler's ruin problem:

$$P_{bb} = 1 - \left(\frac{q}{p}\right)^{x_0},$$

where $x_0 = n/2^k$ is the expected number of copies of the correct BB in the initial population. After convergence, a given locus either has n copies of the correct value with probability P_{bb}, or has n copies of the wrong value with probability $1-P_{bb}$. When every deme sends n/r individuals to each of the other $r-1$ demes, each locus has *on average* nP_{bb} copies of the correct value. This means that the second iteration of the parallel GA starts from a much better starting point than the first iteration.

We use the solution to the gambler's ruin problem again to compute the probability that a locus converges to the correct value after the second iteration. We only need to replace x_0 by the new initial count of correct BBs,

$$P_{bb_2} = 1 - \left(\frac{q}{p}\right)^{nP_{bb}}. \qquad (11)$$

Using P_{bb_2} instead of P_{bb} in equation 5 gives the probability that a deme finds a solution with the minimum desired quality. The probability that at least one deme finds the solution is given by equation 4.

We would like to compute the deme size for the completely connected topology using equation 11, but note that P_{bb} depends on n and so it impossible to obtain a closed-form expression for n. Instead, we approximate equation 11 and derive the population size from the approximation.

Equation 11 can be rewritten exactly as: $P_{bb_2} = 1-(1-c)^{nP_{bb}}$ where $c = 1 - q/p$. In this form P_{bb_2} can be approximated as

$$P_{bb_2} \approx 1 - \exp(-cnP_{bb}).$$

We can approximate roughly the value of P_{bb} as $P_{bb} \approx \frac{cn}{2^k}$. Substituting this form of P_{bb} into the equation above we get

$$P_{bb_2} \approx 1 - \exp\left(\frac{-c^2n^2}{2^k}\right). \qquad (12)$$

Using this form for P_{bb_2}, we can compute the population size needed to get b BBs on average. Recall that to satisfy our success criterion (that requires that at least one of the demes finds the desired solution), the probability of converging correctly on one locus (P_{bb_2}) has to be equal to P^*_{bb}. Making equation 12 equal to P^*_{bb} and solving for n results in a deme sizing equation for a completely connected topology using the maximum migration rate,

$$n = \frac{\sqrt{-2^k \ln(1-P^*_{bb})}}{1-\frac{q}{p}}. \qquad (13)$$

Note that in this case, the deme size depends on the *square root* of $2^k \ln(1-P^*_{bb})$ as opposed to the case for isolated demes where the deme size is directly proportional to this term. By comparing the two sizing equations we observe that migration results in an important change in the way the variables relate.

5.2 Experiments

To test the model for the completely connected topology, we use the one-max function and a 4-bit trap function, as we did in the experiments with isolated demes. We experimented using 4 and 8 demes with their corresponding maximal migration rates of 25% and 12.5%.

Figure 3 shows the predictions of the proportion of BBs correct against the population size for a 100-BB one-max function with 4 and 8 demes. As we discussed before, the one-max problem is very easy for GAs and small population sizes are sufficient to find solutions of high quality. Also, note that when using 8 demes and a migration rate of only 12.5%, we do not expect any migration to occur for population sizes smaller than 12 individuals. This is evident on the bottom portion of figure 3 where the theory predicts a better convergence quality of the GA for low population sizes. Once the population size is large enough to make migration possible, the GA performs in better accordance with the model.

The results for the experiments with the 4-bit trap function are plotted in figure 4. This function requires larger population sizes than the one-max, thus the gap between the prediction and the experimental results at low population sizes is much smaller.

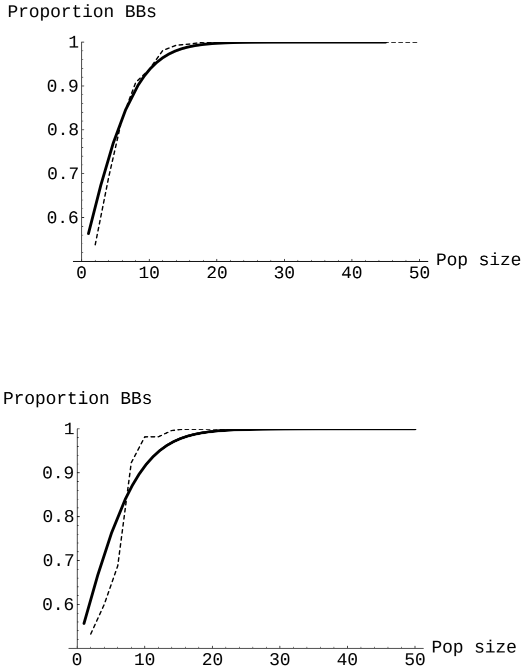

Figure 3: Results comparing theory and experimental results for a one-max function with 100 BBs. On the left are results using 4 demes and a migration rate of 25%, and on the right are results for 8 demes with a migration rate of 12.5%.

6 Future work

This paper lays the foundation for a better understanding of parallel GAs, but there are still many open avenues for further research. As a first step, there is a need for a model that predicts performance in those cases where the migration rate is much lower than the extreme case we presented in this paper. Even though our analysis bounds the expected performance of a parallel GA with smaller migration rates, a specialized model would have important practical applications.

Work is already underway to construct such a model. The analysis is very similar to the derivation of the model for maximum migration rate, as we consider the proportion of BBs found in the first iteration to model performance in the second. The key observation is that when low migration rates are used, only a small portion of each deme is replaced by immigrants. If a particular allele was correct before migration, only a small proportion of incorrect BBs may migrate from other demes, and it is highly probable that the allele will remain correct after the second iteration. However, if the allele had converged to the incorrect value in the first iteration, the immigrant individuals give the allele a chance to converge

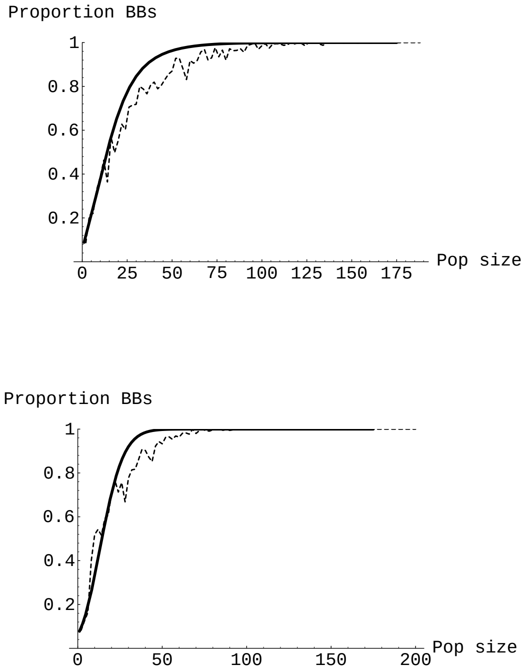

Figure 4: Results comparing theory and experimental results for a 4-bit trap function with 20 BBs. On the left are results using 4 demes and a migration rate of 25%, and on the right are results for 8 demes with a migration rate of 12.5%.

correctly in the second iteration.

Another matter of practical interest is the analysis of the expected parallel speedup. The population sizing equations in this paper can be used directly to predict parallel speedup under idealized conditions where the communication time is ignored. However, general models for the communication time can be constructed and the effect of communications on the parallel speedup should also be investigated.

We also need to extend our work to analyze different topologies. Again, we believe that the bounding cases we presented here cover effectively a full range of topologies, but specialized models may be useful to practitioners.

7 Conclusions

In this paper, we attack the problem of modeling parallel GAs by diverting our attention away from the usual discussion of topologies and migration rates and focusing on the adequate sizing of demes. We recognize that the design of parallel GAs is a complex problem and that the issues of choosing topologies and migration rates are not completely separable from the deme sizing problem. Considering this, we attack the sizing problem using bounding cases for the deme interconnection topologies and migration rates.

We compute the probability that at least one deme finds a solution of a fixed quality for bounding cases of parallel GAs. From this probability of success we derived population-sizing equations that give practitioners a powerful tool to decide adequate population sizes for their problems.

Our models show that the deme size is much smaller when there is migration, than when the demes are completely independent. These smaller population sizes translate directly into savings in computation time.

Acknowledgments

This study was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grants F49620-94-1-0103, F49620-95-1-0338, and F49620-97-1-0050. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the 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 or the U.S. Government.

Erick Cantú-Paz was supported by a Fulbright-García Robles-CONACYT Fellowship.

References

Abramovitz, M., & Stegun, I. e. (1972). *Handbook of mathematic functions with formulas, graphs, and mathematical tables.* Dover Publications.

Cantú-Paz, E. (1995). *A summary of research on parallel genetic algorithms* (IlliGAL Report No. 95007). Urbana, IL: University of Illinois at Urbana-Champaign.

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

Goldberg, D. E., Deb, K., & Thierens, D. (1993). Toward a better understanding of mixing in genetic algorithms. *Journal of the Society of Instrument and Control Engineers*, *32*(1), 10–16.

Grefenstette, J. J. (1981). *Parallel adaptive algorithms for function optimization* (Tech. Rep. No. CS-81-19). Nashville, TN: Vanderbilt University, Computer Science Department.

Grosso, P. B. (1985). *Computer simulations of genetic adaptation: Parallel subcomponent interaction in a multilocus model.* Unpublished doctoral dissertation, The University of Michigan. (University Microfilms No. 8520908).

Harik, G., Cantú-Paz, E., Goldberg, D., & Miller, B. (1996, july). *The gambler's ruin problem, genetic algorithms, and the sizing of populations* (IlliGAL Report No. 96004). Urbana, IL: University of Illinois at Urbana-Champaign.

Nakano, R., Davidor, Y., & Yamada, T. (1994). Optimal population size under constant computation cost. In Davidor, Y., Schwefel, H.-P., & Männer, R. (Eds.), *Parallel Problem Solving fron Nature, PPSN III* (pp. 130–138). Berlin: Springer-Verlag.

Tanese, R. (1989). *Distributed genetic algorithms for function optimization.* Unpublished doctoral dissertation, University of Michigan, Ann Arbor.