
Topologies, Migration Rates, and Multi-Population Parallel Genetic Algorithms

Erick Cantú-Paz

Department of Computer Science and
Illinois Genetic Algorithms Laboratory
University of Illinois at Urbana-Champaign
cantupaz@illigal.ge.uiuc.edu

Abstract

This paper presents a study of parallel genetic algorithms (GAs) with multiple populations (also called demes or islands). The study makes explicit the relation between the probability of reaching a desired solution with the deme size, the migration rate, and the degree of the connectivity graph. The paper considers arbitrary topologies with a fixed number of neighbors per deme. The demes evolve in isolation until each converges to a unique solution. Then, the demes exchange an arbitrary number of individuals and restart their execution. An accurate deme-sizing equation is derived, and it is used to determine the optimal configuration of an arbitrary number of demes that minimizes the execution time of the parallel GA.

1 INTRODUCTION

Parallel genetic algorithms (GAs) with multiple populations are difficult to configure because they are controlled by many parameters that affect their efficiency and accuracy. Among other things, one must decide the number and the size of the populations (demes), the rate of migration, and the destination of the migrants.

This paper extends previous theoretical studies that focused on the number of demes and their sizes (Cantú-Paz & Goldberg, 1997a; Cantú-Paz & Goldberg, 1997b). Those studies considered two configurations that bound the migration rate and the connectivity between the populations, and resulted in useful design guidelines for parallel GAs. However, the

bounding configurations are not commonly used by practitioners because neither one of them is scalable. On one extreme, where the demes are isolated, the improvements in quality are marginal and the total computational cost grows very fast. At the other extreme, where each deme communicates with all the others, the communication cost quickly becomes impractical as more demes are used.

This paper addresses the problem of scalable communications by analyzing the effects of the topology of communications, the deme size, and the migration rate on the algorithmic cost and on the solution quality. The goal is to find the configuration of multiple demes that reaches the desired solution in the shortest time possible.

The models assume that the demes evolve in isolation until each deme converges to a unique solution. At this point, the demes exchange an arbitrary number of individuals and restart their execution. The interval of time between migrations is called an *epoch*. The same strategy of migration has been considered by others (Grosso, 1985; Braun, 1990; Munetomo, Takai, & Sato, 1993; Cantú-Paz & Goldberg, 1997a). The calculations in this paper also assume that all the demes have the same number of neighbors, and that all of them use the same migration rate, but there are no other restrictions on the topologies or on the magnitude of the migration rates.

The paper is organized as follows. The next section summarizes a model that predicts the quality of the solutions that a simple GA reaches based on the size of its population. Section 3 describes how the deme size, the migration rate, and the degree of the connectivity graph affect the probability that the desired solution is reached after the *second* epoch. The analysis shows how to find a configuration that minimizes the execution time. Section 4 generalizes the results to multiple epochs. It presents experiments that sug-

gest that the solutions reached by different topologies with the same degree are almost identical, and then it focuses on one family of topologies to determine how the quality improves after arbitrary epochs. Then, the results of the modeling are used to find a configuration that minimizes the execution time. Finally, section 5 summarizes the results of this paper and presents the conclusions of this study.

2 BACKGROUND

This section reviews a model that predicts the quality of solutions found by a simple GA based on the size of its population and the number of correct building blocks (BBs) present initially. Subsequent sections extend this model to consider multiple communicating populations. In this paper, a correct BB is the lowest-order schema that consistently leads to the global optimum.

To obtain a model of the quality of the solution of a GA, Harik, Cantú-Paz, Goldberg, and Miller (1997), modeled the selection process in a GA as a biased one-dimensional random walk. Their model considers one partition of length k , and it assumes that decisions are independent across partitions. The number of copies of the correct BB in the partition is represented by the position, x , of a particle on a one-dimensional space. The space is bounded with absorbing barriers at $x = 0$ and $x = n$, which represent ultimate convergence to the wrong and to the right solutions, respectively. The initial position of the particle, x_0 , is the expected number of copies of the best BB in a randomly initialized population, and is equal to $x_0 = n/2^k$, where k is the order of the BB and n is the population size.

At each step of the random walk there is a probability, p , of obtaining one additional copy of the correct BB. This probability depends on the particular problem faced by the GA, and it represents the probability of deciding correctly in a one-to-one competition between the best and the second best schemata of the partition. For functions composed by adding several uniformly-scaled subfunctions, p was computed by Goldberg, Deb, and Clark (1992) in their study of population sizing as

$$p = \Phi\left(\frac{d}{\sigma_{bb}\sqrt{2m'}}\right), \quad (1)$$

where Φ denotes the cumulative distribution function (CDF) of a normal distribution with a mean of zero and a standard deviation of one, d is the difference

of the fitness contribution between the best and the second best schemata in the partition, $m' = m - 1$, m is the number of subfunctions, and σ_{bb}^2 is the RMS average variance of k -th order partitions.

A well-known result about random walks is the probability that a particle will eventually be captured by the absorbing barrier at $x = n$ (Feller, 1966):

$$P_{bb} = \frac{1 - \left(\frac{q}{p}\right)^{x_0}}{1 - \left(\frac{q}{p}\right)^n} \approx 1 - \left(\frac{q}{p}\right)^{x_0}, \quad (2)$$

where $q = 1 - p$. For increasing values of n , the denominator of the equation above approaches 1 very quickly, and therefore it may be ignored in the calculations.

We measure the quality of the solutions as the number of partitions with the correct BB at the end of a run, and we denote the desired target quality as \hat{Q} . Using the assumption that partitions are independent, we can solve $P_{bb} = \frac{\hat{Q}}{m}$ for n to obtain the following population sizing equation:

$$n = \frac{2^k \ln(\alpha)}{\ln \frac{q}{p}}, \quad (3)$$

where $\alpha = 1 - \frac{\hat{Q}}{m}$.

When there are multiple populations, success is defined when at least one of the populations reaches the desired target quality \hat{Q} . An equivalent success criterion is to require that the highest quality found by any of the r demes equals the target quality. The effect of this is that the quality required in each deme can be relaxed in the following way (Cantú-Paz & Goldberg, 1999):

$$\hat{P} = \frac{\hat{Q}}{m} - \frac{\mu_{r:r}}{2\sqrt{m}} \approx \frac{\hat{Q}}{m} - \frac{\sqrt{\ln r}}{\sqrt{2m}}, \quad (4)$$

where $\mu_{r:r}$ is the expected value of the highest-order statistic of a standard normal distribution. The values of $\mu_{r:r}$ have been tabulated extensively (Harter, 1970), and the approximation used above for $\mu_{r:r}$ was suggested by Beyer (1993). Note that \hat{P} varies very slowly with respect to r , and that when $r = 1$, $\hat{P} = \hat{Q}$.

3 DEGREE OF CONNECTIVITY

An important property of the connectivity graph between the demes is its *degree*, which is the number of neighbors of each deme. This paper assumes that all the demes have the same degree, and we denote it as δ . The degree completely determines the cost of communications, and as we shall see, it also influences the size of the demes and consequently the time of computations.

This section analyzes how the deme size, the migration rate, and the degree of the topology affect the probability that the parallel GA reaches the desired solution. The analysis of this section considers only the first two epochs of the algorithm, because closed-form expressions may be derived easily. The next section extends the analysis to multiple epochs.

The analysis has several steps. First, we compute how many copies of the correct BB are necessary to reach the target quality per deme (\hat{P} , given by equation 4). Next, the probability that a given configuration brings together the critical number of BBs is calculated. The success probability is then used to derive a deme sizing equation, which in turn is used to minimize the execution time.

The first step of the analysis is straightforward. To determine how many BBs \hat{x}_1 are needed at the beginning of the second epoch to reach \hat{P} , we may use the solution of the gambler's ruin problem. Making

$$\hat{P} = 1 - \left(\frac{q}{p}\right)^{\hat{x}_1},$$

and solving for \hat{x}_1 results in

$$\hat{x}_1 = \frac{\ln(1 - \hat{P})}{\ln\left(\frac{q}{p}\right)}. \quad (5)$$

The next step is to determine the probability that a deme receives at least \hat{x}_1 BBs from its δ neighbors. The probability that one neighbor sends the right BB is the same probability that it converged correctly in the first epoch, and is given by P_{bb} . Assuming that all the neighbors of a deme use the same migration rate, ρ , then at least $\hat{\delta} = \frac{\hat{x}_1}{\rho n_d}$ neighbors must contribute the correct BB. Since the demes have evolved in isolation until this moment, the probability of receiving at least \hat{x}_1 BBs has a binomial probability:

$$P_{x_1} = 1 - \sum_{i=0}^{\hat{\delta}-1} \binom{\delta}{i} P_{bb}^i (1 - P_{bb})^{\delta-i}, \quad (6)$$

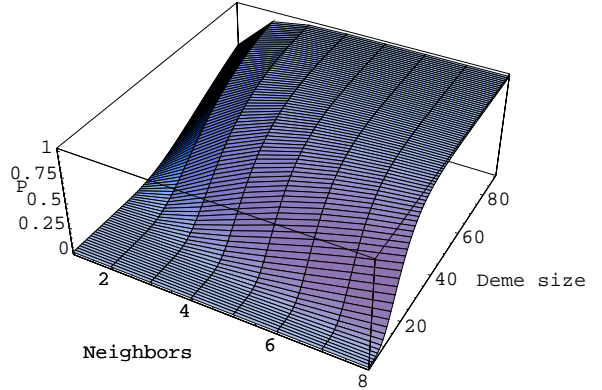


Figure 1: Plot of the probability of success with different configurations of deme sizes and number of neighbors (eq. 7). The migration rate in this example is 10%.

which can be approximated as

$$P_{x_1} = 1 - \Phi\left(\frac{\hat{\delta} - \delta P_{bb}}{\sqrt{\delta P_{bb}(1 - P_{bb})}}\right). \quad (7)$$

With higher migration rates, the number of neighbors that must contribute the right BB decreases, and therefore it is more likely that the deme receives the critical number of BBs and succeeds to find the solution. This observation is consistent with other theoretical results that show that the solution's quality increases with higher migration rates (Cantú-Paz, 1998).

Note that even if a deme receives less than \hat{x}_1 BBs, it may still reach the right solution, because the deme itself could have converged correctly in the first epoch, and it may contain enough BBs to converge correctly again. Also, a deme may start the second epoch with less than \hat{x}_1 BBs and converge correctly sometimes. However, we ignore these two possibilities and conservatively assume that a deme does not converge to the right answer if it does not receive at least \hat{x}_1 BBs from its neighbors. Under this assumption, P_{x_1} is the probability that at the end of the second epoch the deme will converge correctly.

There are different configurations that can bring together the critical number of BBs with the same probability (see figure 1). Configurations with large demes and few neighbors have the same chance of success than some configurations with smaller demes but with more neighbors. This is the usual tradeoff between computation and communications: smaller demes require more neighbors to succeed. We would like to use the configuration that achieves the desired objective with

the minimum cost.

The execution time of the parallel program is the sum of communication and computation times:

$$T_p = gn_d T_f + \delta T_c, \quad (8)$$

where g is the domain-dependent number of generations until convergence, n_d is the deme size, T_f is the time of one fitness evaluation, and T_c is the time required to communicate with one neighbor. T_c, T_f , and g can be easily determined empirically, but the required deme size depends on the degree of the topology, the migration rate, and the desired quality.

3.1 FINDING THE DEME SIZE

To find the deme size we need to make $P_{x_1} = \hat{P}$ and solve for n_d . First, the normal distribution of P_{x_1} has to be approximated as $\Phi(z) = (1 + \exp(-1.6z))^{-1}$ (Valenzuela-Rendón, 1989). With this approximation, P_{x_1} becomes

$$P_{x_1} = 1 - (1 + \exp(-1.6z))^{-1}, \quad (9)$$

where $z = \frac{\hat{\delta} - \delta P_{bb}}{\sqrt{\delta P_{bb}(1-P_{bb})}}$ is the normalized number of successes. We may bound z by considering that the variance is maximal when $P_{bb} = 0.5$, and thus it becomes $z \geq \frac{2}{\sqrt{\delta}}(\hat{\delta} - \delta P_{bb})$ (In the remainder we conservatively ignore the inequality.). Additionally, P_{bb} may be roughly approximated as $P_{bb} \approx \frac{cn_d}{2^k}$, where $c = 1 - q/p$. Substituting this form of P_{bb} and $\hat{\delta} = \frac{\hat{x}_1}{\rho n_d}$ into the bound of z gives

$$z = \frac{2}{\sqrt{\delta}} \left(\frac{\hat{x}_1}{\rho n_d} - \delta \frac{cn_d}{2^k} \right).$$

Making the approximate form of $P_{x_1} = \hat{P}$ and solving for z yields the ordinate where the probability of success reaches the required value:

$$\hat{z} = 0.625 \ln \left(\frac{\hat{P}}{1 - \hat{P}} \right).$$

Making $z = \hat{z}$, solving for n_d , and simplifying terms gives the deme size:

$$n_d = \frac{2^{k-2} \hat{z} + \sqrt{\hat{z}^2 + \frac{c \hat{x}_1}{\rho 2^{k-2}}}}{\sqrt{\delta} c}. \quad (10)$$

Observe that the deme size decreases with higher mi-

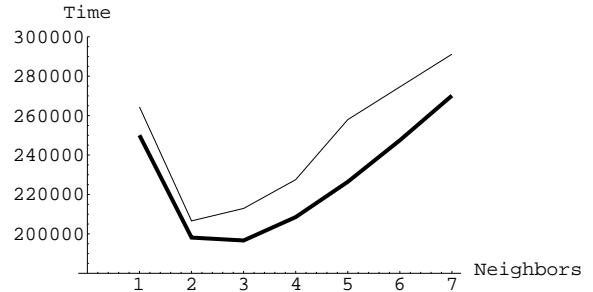


Figure 2: Comparison of theoretical (thick line) and experimental (thin line) execution times (in microseconds) of eight demes connected by topologies with different degrees.

gration rates and as the number of neighbors increases, which is what we expected. For clarity, this deme-sizing equation may be rewritten in a more compact form by grouping all the domain-dependent constants into one (n_0) as follows:

$$n_d = \frac{n_0}{\sqrt{\delta}}. \quad (11)$$

Now, the total execution time as given by equation 8 may be easily optimized with respect to δ by making $\frac{\partial T_p}{\partial \delta} = 0$ and solving for δ :

$$\delta^* = \left(\frac{gn_0 T_f}{2T_c} \right)^{2/3}, \quad (12)$$

and the optimal deme size can be found by substituting δ^* in equation 10.

Figure 2 compares the theoretical predictions of the execution time with experimental results on a network of eight IBM workstations. The fitness function is $\sum_{i=1}^{20} f_4(u_{4i})$, where u_{4i} is the number of bits set to 1 in the substring that starts at position $4i$, and f_4 is a fully deceptive trap function of order $k = 4$ defined as: $(u_{4i}, f_4(u_{4i})) = ((0, 3), (1, 2), (2, 1), (3, 0), (4, 4))$. In this example there are $m = 20$ copies of the trap function, and the difference between the best and second best BB is $d = 1$. The fitness variance in the partition is $\sigma_{bb}^2 = 1.215$, and therefore $p = 0.5585$. The objective is to find a solution with at least 16 partitions correct ($\hat{Q} = 0.8$). The time to evaluate a single individual is $T_f = 51$ microseconds, the communications time is $T_c = 29$ ms, and the number of generations until convergence is $g = 50$. The figure shows the average of 100 runs using pairwise tournament selection, two-point crossover with probability 1.0, and no mutation.

The migration rate was $\rho = 0.1$.

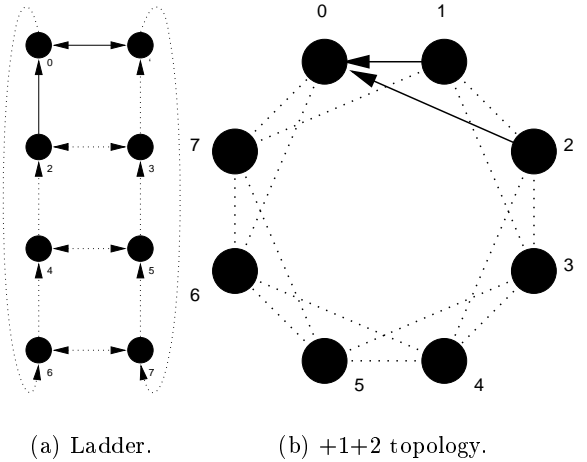
4 CONSIDERING MULTIPLE EPOCHS

The previous section showed how to find the optimal degree of connectivity that minimizes the execution time for a particular domain. However, even with a fixed degree there are $\binom{r-1}{\delta}$ ways to connect the demes, and we still face the question of how to choose a particular topology. Certainly, if the algorithm is only executed for two epochs, it does not matter how the demes are connected, because only the immediate neighbors affect the search. But if more than two epochs are used, a deme would receive indirect contributions from other demes. The purpose of this section is to quantify the effect of those contributions on the quality of the search, and to determine how to minimize the execution time after multiple epochs.

Consider the topologies with degree $\delta = 2$ depicted in figure 3. These are only three of the $\binom{7}{2} = 21$ possible topologies of degree two. Figure 4 shows the results of experiments with a 20-BB 4-bit fully deceptive trap function on eight demes connected with the two topologies of figure 3 and a bi-directional ring. The results are averaged over 100 repetitions at each deme size; the demes used pairwise tournament selection, two-point crossover with probability 1.0, and no mutation. The migration rate was set to its maximal value of $\rho = 1/3$. The figure shows the proportion of correct BBs per deme after one, two, three, and four epochs. The quality of the solutions improves after successive epochs, and the largest increase occurs after the second epoch. As one would expect, the results for different topologies after the first two epochs are indistinguishable, and the difference after three and four epochs is not significant. This observation will be used to derive a model of solution quality that depends only on the degree of the connectivity graph and on the migration rate, but that ignores the specific topology. Before doing so, we first introduce the concept of the extended neighborhood of a deme.

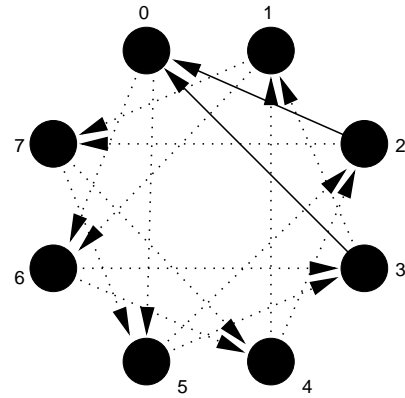
4.1 EXTENDED NEIGHBORHOODS

To visualize how the choice of topology affects the quality of the search, imagine a tree rooted on a particular deme. The descendants of a node in the tree are the immediate neighbors of the deme it represents, and the τ -th level in the tree contains the demes that are



(a) Ladder.

(b) +1+2 topology.



(c) +2+3 topology.

Figure 3: Different topologies with two neighbors.

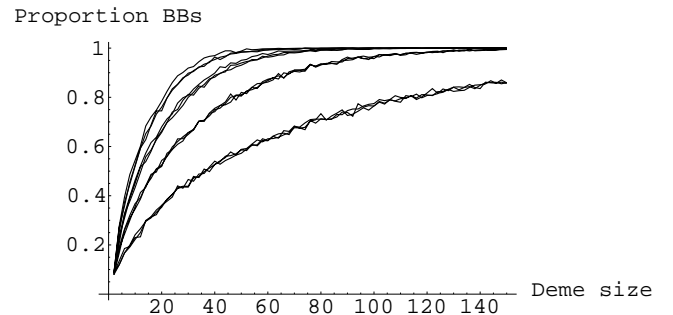


Figure 4: Average quality per deme after one, two, three, and four epochs (from bottom to top) using eight demes connected with different topologies of degree two.

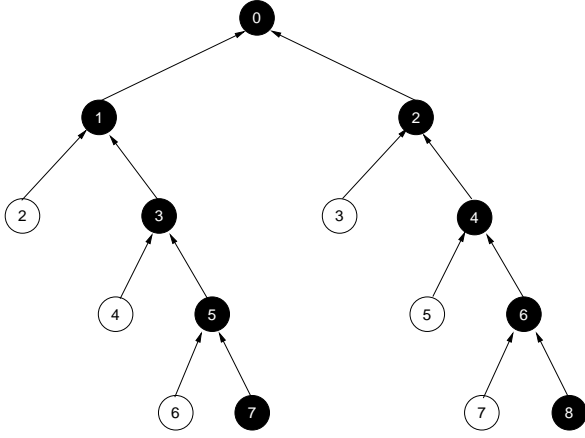


Figure 5: A tree representation of the extended neighborhood of deme 0 in the +1+2 topology with 16 demes. The black nodes represent the new members of the extended neighborhood after each epoch. The white nodes represent demes that already belong to the neighborhood, and they are not expanded to avoid clutter in the graph.

reachable from the root deme after τ epochs. These demes form the *extended* neighborhood of the root and are taken into account only the first time they are reached¹. Figure 5 shows such a tree that corresponds to the +1+2 with 16 demes.

A simple way to bound the contribution from the extended neighborhood is to assume that the demes form panmictic groups as soon as they come in contact with others. In this view, after τ epochs the aggregate population size would be $r_\tau n_d$, where r_τ is the number of demes in the extended neighborhood after the τ -th epoch, and n_d is the size of each deme. Under this assumption, the solution quality would be given by $P_{bb}(r_\tau n_d)$. Of course, demes do not become panmictic as soon as they reach one another, and therefore the size of the extended neighborhood should be adjusted with a mixing coefficient $c_m < 1$, and the quality becomes $P_{bb}(c_m r_\tau n_d)$.

¹For the mathematically inclined, the definition of the extended neighborhood is as follows. Consider a directed graph $G = (V, E)$, where V is the set of vertices that represent the demes, and E is the set of edges that represent connections between demes. The extended neighborhood of a deme v is the set $R_\tau = \bigcup_{i=0}^{\tau} a : a \xrightarrow{i} v$, where $a \xrightarrow{i} v$ denotes a path of length i from a to v . $r_\tau = |R_\tau|$.

4.2 DESIGNING FOR MULTIPLE EPOCHS

The observation that topologies of the same degree reach almost identical solutions has an important implication: if an accurate quality predictor can be derived for one topology, it would be accurate for any topology of the same degree.

Some topologies are easier to study than others because the size of their extended neighborhoods increases in a regular form. In particular, there are topologies where the size of the extended neighborhood is simply $r_\tau = \delta(\tau - 1) + 1 = \delta\tau' + 1$. Examples of such topologies are the +1+2 binary topology depicted in figure 3 and a +1+2+3 ternary topology. We will use this family of topologies to study the effect of the degree of the network on the solution quality after several epochs.

Using the simple model introduced in the previous subsection, the quality after τ epochs is $P_{bb\tau} = P_{bb}(c_m r_\tau n_d)$. For simplicity, we use $n_\tau = c_m r_\tau n_d$ to represent the number of individuals in the extended neighborhood. The key to obtain an accurate quality prediction is to adjust n_τ with an appropriate c_m . We can deduce the value of the mixing coefficient by considering some of the properties it should have. First, n_τ should grow linearly as τ increases, and when $\tau = 1$ the value of n_τ should be equal to n_d , because the demes are isolated during the first epoch. In addition, the previous section showed that the deme size $n_d \propto \frac{1}{\sqrt{\delta}}$. Putting everything together, we may write n_τ as:

$$n_\tau = (\sqrt{\delta}\tau' + 1)n_d, \quad (13)$$

which means that $c_m = \frac{\sqrt{\delta}\tau'+1}{\delta\tau'+1} \approx \frac{1}{\sqrt{\delta}}$. Experimental tests were performed to assess the accuracy of this model. The experiments use eight demes connected by a +1+2 topology (and the same experimental conditions as previous experiments in this paper: 100 repetitions at each deme size, pairwise tournament selection, two-point crossover with probability 1.0, and no mutation). The quality was measured at the end of the first four epochs. Figure 6 shows that the predictions of $P_{bb}(n_\tau)$ match very well the experimental results.

Despite the accuracy of the predictions of $P_{bb}(n_\tau)$, a word of caution is necessary at this point. Note that $\lim_{\tau \rightarrow \infty} P_{bb}(n_\tau) = 1$. The derivation of n_τ is based on the concept of extended neighborhoods, but the size of the extended neighborhood n_τ is bounded by the sum of the sizes of all the demes (rn_d). Therefore, making $n_\tau \leq rn_d$, and solving for τ gives a bound

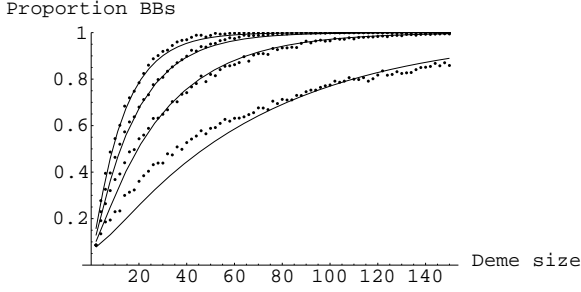


Figure 6: Theoretical predictions (line) and experimental results (dots) of the average quality per deme after 1, 2, 3, and 4 epochs (from right to left) using eight demes connected by a +1+2 topology.

on the number of epochs on which the predictions are valid:

$$\tau \leq \frac{r-1}{\sqrt{\delta}} + 1. \quad (14)$$

The next step of the analysis is to find a deme-sizing equation. The procedure is straightforward using the gambler's ruin model. Making $P_{bb}(n_\tau) = 1 - \left(\frac{q}{p}\right)^{n_\tau/2^k} = \hat{P}$ and solving for n_d results in

$$n_d = \frac{1}{\sqrt{\delta\tau'} + 1} \frac{2^k \ln(1 - \hat{P})}{\ln \frac{q}{p}}, \quad (15)$$

which can be rewritten in a much compact form by grouping all the problem-dependent constants (the second term above) into one constant n_0 , so $n_d = \frac{n_0}{\sqrt{\delta\tau'} + 1}$.

This form of the deme size with $\tau = 2$ is similar to the equation found in the previous section. With a closed-form expression for the deme size after multiple epochs, the execution time of the parallel GA may be easily minimized. In this case, the time is

$$T_p = \tau (gn_d T_f + \delta T_c), \quad (16)$$

and τ is restricted by equation 14.

To simplify the calculations, n_d may be approximated as $\frac{n_0}{\sqrt{\delta\tau'}}$. Making $\frac{\partial T_p}{\partial \delta} = 0$ and solving for δ gives the optimal degree of the topology as

$$\delta^* = \left(\frac{gn_0 T_f}{2\tau' T_c} \right)^{2/3}, \quad (17)$$

which is equivalent to the optimum found in the previous section when $\tau = 2$.

An important point to note is that δ^* depends on r , because n_0 depends on $\mu_{r:r}$. However, since $\mu_{r:r}$ varies very slowly with respect to r , δ^* also varies very slowly. This is significant because if δ^* does not change much as more demes (processors) are used, then the execution time (equation 16) would not change much either. This issue must be explored in more detail in the future, but we should be careful not to dismiss the algorithm's capability to reduce the execution time. After all, δ^* strongly depends on the number of epochs τ , and both δ^* and the execution time will decrease significantly as more epochs are used.

This raises another point: sometimes the choice of topology is restricted by hardware constraints. In this case, δ may be considered to be constant and the execution time may be optimized with respect to τ . Making $\frac{\partial T_p}{\partial \tau} = 0$ and solving for τ gives the optimal number of epochs:

$$\tau^* = 1 + \sqrt{\frac{gn_0 T_f}{\delta^{3/2} T_c}}. \quad (18)$$

The corresponding deme size may be found by substituting δ^* (or τ^*) in equation 15.

5 SUMMARY AND CONCLUSIONS

This paper presented accurate predictions of the quality of the solutions that multiple-population parallel GAs are expected to reach. The first part of the paper described the relation between the deme size, the migration rate, and the degree of the topology with the probability of success after two epochs. It showed how to find the configuration that optimizes the execution time while reaching a predetermined target quality. These results were generalized to multiple epochs in the second part of the paper.

After multiple epochs, the topology would seem to be an important factor in the solution quality because a deme receives indirect contributions from varying number of demes. However, section 4 showed that different topologies with the same degree reach almost identical solutions after any number of epochs. The small differences may be explained by the different sizes of the extended neighborhoods, but most importantly, the equivalence of topologies with the same degree facilitated the derivation of a general model of

solution quality. The quality model was transformed into an accurate deme sizing equation, which in turn was used to find the degree and the number of epochs that minimize the execution time.

The results of this paper facilitate the use of multi-population GAs, because they eliminate the need to guess or to do excessive experimentation to find appropriate values for the deme sizes or the topology of communications. Instead, the equations presented here enable users to optimize the algorithms to their particular hardware environment and problem domain.

Acknowledgments

I wish to thank Prof. David E. Goldberg for his guidance during the research reported here. I was supported by a Fulbright-García Robles Fellowship.

The work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-97-1-0050. Research funding for this project was also provided by a grant from the US Army Research Laboratory Program, Cooperative Agreement DAAL01-96-2-003. The US 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 and endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the US Government.

References

- Beyer, H.-G. (1993). Toward a theory of evolution strategies: Some asymptotical results from the $(1, + \lambda)$ -Theory. *Evolutionary computation*, 1(2), 165–188.
- Braun, H. C. (1990). On solving travelling salesman problems by genetic algorithms. In Schwefel, H.-P., & Männer, R. (Eds.), *Parallel Problem Solving from Nature* (pp. 129–133). Berlin: Springer-Verlag.
- Cantú-Paz, E. (1998). Using Markov chains to analyze a bounding case of parallel genetic algorithms. In Koza, J., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D., Garzon, M., Goldberg, D. E., Iba, H., & Riolo, R. (Eds.), *Genetic Programming 1998: Proceedings of the Third Annual Conference* (pp. 456–462). San Francisco, CA: Morgan Kaufmann Publishers.
- Cantú-Paz, E., & Goldberg, D. E. (1997a). Modeling idealized bounding cases of parallel genetic algorithms. In Koza, J., Deb, K., Dorigo, M., Fogel, D., Garzon, M., Iba, H., & Riolo, R. (Eds.), *Genetic Programming 1997: Proceedings of the Second Annual Conference* (pp. 353–361). San Francisco, CA: Morgan Kaufmann Publishers.
- Cantú-Paz, E., & Goldberg, D. E. (1997b). Predicting speedups of idealized bounding cases of parallel genetic algorithms. In Bäck, T. (Ed.), *Proceedings of the Seventh International Conference on Genetic Algorithms* (pp. 113–121). San Francisco: Morgan Kaufmann.
- Cantú-Paz, E., & Goldberg, D. E. (1999). Parallel genetic algorithms: theory and practice. *Computer Methods in Applied Mechanics and Engineering*. In press.
- Feller, W. (1966). *An introduction to probability theory and its applications* (2nd ed.), Volume 1. Wiley.
- Goldberg, D. E., Deb, K., & Clark, J. H. (1992). Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6, 333–362.
- 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. E., & Miller, B. L. (1997). The gambler's ruin problem, genetic algorithms, and the sizing of populations. In *Proceedings of 1997 IEEE International Conference on Evolutionary Computation* (pp. 7–12). Piscataway, NJ: IEEE.
- Harter, H. L. (1970). *Order statistics and their use in testing and estimation*. Washington, D.C.: U.S. Government Printing Office.
- Munetomo, M., Takai, Y., & Sato, Y. (1993). An efficient migration scheme for subpopulation-based asynchronously parallel genetic algorithms. In Forrest, S. (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms* (pp. 649). San Mateo, CA: Morgan Kaufmann.
- Valenzuela-Rendón, M. (1989). *Two analysis tools to describe the operation of classifier systems*. Doctoral dissertation, University of Alabama, Tuscaloosa. Also available as TCGA Report No. 89005.