Using Markov Chains to Analyze a Bounding Case of Parallel Genetic Algorithms

Erick Cantú-Paz
Department of Computer Science and
Illinois Genetic Algorithms Laboratory
University of Illinois at Urbana-Champaign
104 South Matthews Avenue, Urbana, IL 61801
cantupaz@illigal.ge.uiuc.edu

ABSTRACT

This paper uses Markov chains to analyze the search quality of a bounding case of parallel genetic algorithms with multiple populations. In the bounding case considered here, each population exchanges individuals with all the others. First, the migration rate is set to the maximum value possible, and later the analysis is refined to consider lower migration rates. In the algorithm examined by this paper, migration occurs only after each population converges. Then, incoming individuals are incorporated into the populations and the algorithm restarts. The analysis shows how to calculate the probability that each population will eventually converge to the correct solution, and the expected number of migration-restart events until all the populations converge to the same solution.

1 Introduction

Implementations of parallel genetic algorithms with multiple populations are common. However, this method of parallelization introduces many additional parameters, and their effect is not very well understood. Parameters such as the number of populations, their size, and the amount of communications have to be set carefully, because an incorrect configuration may result in a slow search or in an inadequate solution. The objective of this paper is to predict the quality of the search after multiple migration events and with varying migration rates.

To make some progress in the analysis the paper cannot consider all the parameters of the multi-population GA. Instead, it focuses on a case that is an upper bound on the topology: each population (or deme) communicates individuals to all the others (i.e., the topology is a fully-connected graph). The opposite bound is when the demes evolve in complete isolation, and it has been shown that in isolation the gains in quality are only marginal (Cantú-Paz & Goldberg, 1997). In the algorithm examined here, migration occurs only after the populations converge; then, the newly received individuals are incorporated into the populations, and the algorithm restarts. This evolve-migrate sequence is called an epoch, and the algorithm ends when all the populations converge to the same solution. Similar algorithms were investigated empirically by Grosso (1985), Braun (1990), and by Munetomo, Takai, and Sato (1993). This bounding case has been studied before (Cantú-Paz & Goldberg, 1997; Cantú-Paz & Goldberg, 1997), but the previous analysis only considered one migration event. This paper extends that study by examining the quality of the search after multiple epochs.

The analysis is based on the gambler's ruin (GR) model that predicts the convergence quality of simple GAs based on the population size and on how many copies of the correct building block are present initially (Harik, Cantú-Paz, Goldberg, & Miller, 1997). The number of building blocks (BBs) at the beginning of an epoch depends solely on how many demes converged correctly in the previous epoch. Therefore, Markov chains can be used to describe the distribution of demes that converge correctly. The analysis describes how to calculate the quality of the search after each epoch and the probability that the parallel GA converges to the correct solution in the long run. This long-run success probability of fully connected demes is the same as the probability that a simple GA with an aggregate population

Figure 1 The bounded one-dimensional space of the gambler's ruin problem.

size will converge correctly. Also, the paper shows how to calculate the expected number of epochs until all the populations converge to the same solution.

The remainder of the paper is organized as follows. The next section gives a brief description of the GR model for simple GAs. Section 3 examines the case where the migration rate is set to the maximum value possible, and section 4 refines the analysis by considering smaller migration rates. Finally, section 5 presents the conclusions of this study and discusses avenues of future research.

2 The gambler's ruin model

The basic mechanism in GAs is Darwinian evolution: bad traits are eliminated from the population because they appear in individuals which do not survive the selection process. Good traits survive and are mixed by recombination (mating) to form better individuals. In GAs, the notion of good traits is formalized with the concept of building blocks (BBs), which are string templates (schemata) that match a short portion of the individuals and act as a unit to influence their fitness. The prevailing theory suggests that GAs work by propagating BBs in the population using selection and crossover. We follow Goldberg, Deb, and Thierens (1993), and restrict the definition of a building block to the shortest schemata that contribute to the global optimum. In this view, the juxtaposition of two BBs of order k at a particular string does not lead to a BB of order $2k$, but instead to two separate BBs.

To obtain a model of the quality of the solution of a GA, Harik, Cantú-Paz, Goldberg, and Miller (1997) modeled selection in a GA as a random walk. The model concentrates on only one partition of order k, and it assumes that decisions are independent across partitions. In the random walk model the number of copies of the correct BB in the population is represented by the position, x, of a particle on a one-dimensional space, as depicted in figure 1. Absorbing barriers at $x = 0$ and $x = n$ bound the space, and represent ultimate convergence to the wrong and to the right solutions, respectively. Once the particle reaches the barriers it cannot escape, and this implies that the GA does not use mutation. The initial position of the particle, x_0, is the expected number of copies of the best BB in a randomly initialized population, that 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 is different for every problem and represents the chance of deciding well between two competing BBs. 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

$$p = N\left(\frac{d}{\sqrt{2m'\sigma_{bb}^2}}\right),$$

where N denotes the cumulative distribution function of a normal distribution with a mean of zero and a standard distribution 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} is the average variance of the k-th order partitions.

Using a well-known result from the random walk literature, the probability that the particle will be absorbed at $x = n$, or equivalently the probability that the GA will converge to the right solution is

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

where $q = 1 - p$ is the probability of making the wrong decision between two competing BBs. This has been shown to be an accurate predictor of the search quality of simple GAs. Note that the success probability depends greatly on the initial position of the particle, and a good portion of the analysis in the next section is dedicated to calculate the number of copies of the BB at the beginning of each epoch.

3 Analysis with maximum migration rates

Cantú-Paz and Goldberg (1997) predicted the expected quality of the solution at the end of the second epoch when the algorithm uses a maximal migration rate. The maximal migration rate is $\rho = 1/r$, where r is the number of demes. Their method is to calculate the number of BBs after the first migration as $x_1 = nP_{bb}$, and to substitute x_0 with x_1 in equation 1. This assumes that many demes are used, but the approximation works well even for cases with 4 demes.

Unfortunately, the gambler's ruin solution cannot be used accurately after the second epoch, because as the model iterates the starting point for the random walk increases steadily. After a few iterations the probability of converging to the correct BB would be 1, regardless of

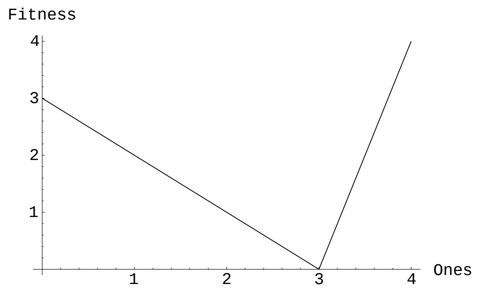


Figure 2 A fully deceptive 4-bit trap function. The horizontal axis is the number of bits set to 1 in and the vertical axis is the fitness value. Since all four bits need to be identified to reach the global optimum, the BBs in this function are of order four. The difficulty of this function can be varied easily by using more bits in the basin of the deceptive optimum, or by reducing the fitness difference between the global and deceptive maxima.

	Epoch	x_0	$P_{bb}(x_0)$
$n_d = 16$	1	1	0.2144
	2	3.4	0.5668
	3	9	0.9024
	4	14.4	0.9894
	5	15.8	0.9990
	6	15.98	0.9999
	7	15.99	0.9999
	8	16	1
$n_d = 32$	1	2	0.3753
	2	12	0.9410
	3	30.11	0.9996
	4	31.99	0.9999
	5	32	1

Table 1 The iterated gambler's ruin model always converges to 1, regardless of the deme size (n_d) or the number of demes.

the population size and the number of demes. For example, consider a fitness function formed by concatenating 20 copies of a 4-bit fully deceptive trap function such as the one pictured in figure 2. Table 1 shows that after only eight (five) epochs, the iterated GR model predicts that a parallel GA with a deme size of 16 (32) individuals will converge to the correct solution. In reality, the correct value of a partition may not appear in any of the demes, or it may be lost after some epochs.

The number correct BBs at the start of an epoch depends directly (and solely) on how many demes converged correctly in the previous iteration. In particular, if i demes converged correctly, each deme would start the current epoch with $\frac{in}{r}$ copies of the BB, and the probability that they converge correctly is $P_{bb}(\frac{in}{r})$. The problem consists on computing the number of demes i that have the right BB after each epoch.

Recall that migration only occurs after the demes have converged, so they execute independently, and the chance that each converges correctly is $P_{bb}(x_0)$. Therefore, after the first epoch the distribution of the number of demes with the right BB has a binomial distribution:

$$V_1(i) = \binom{r}{i} P_{bb}^i(x_0)\,(1 - P_{bb}(x_0))^{r-i}\,, \qquad (2)$$

where $V_t(i)$ denotes the probability that exactly i demes converge correctly after the t-th epoch, and $x_0 = \frac{n}{2^k}$ is the number of BBs in the initial random populations. The probability of converging correctly after the second epoch is given by the sum of the conditional probabilities

$$P_{bb_2} = \sum_{i=0}^{r} V_1(i) P_{bb}(\frac{in}{r}),$$

which can be generalized and expressed as a vector product

$$P_{bb_t} = V_t U, \qquad (3)$$

where $U(i) = P_{bb}(\frac{in}{r})$.

To calculate the distribution of correct demes after the second epoch, we use Markov chains. The states of the Markov chain represent the number of demes that converge to the right value. The transition matrix is defined with the probabilities of going from a state with i demes correct to a state with j demes correct:

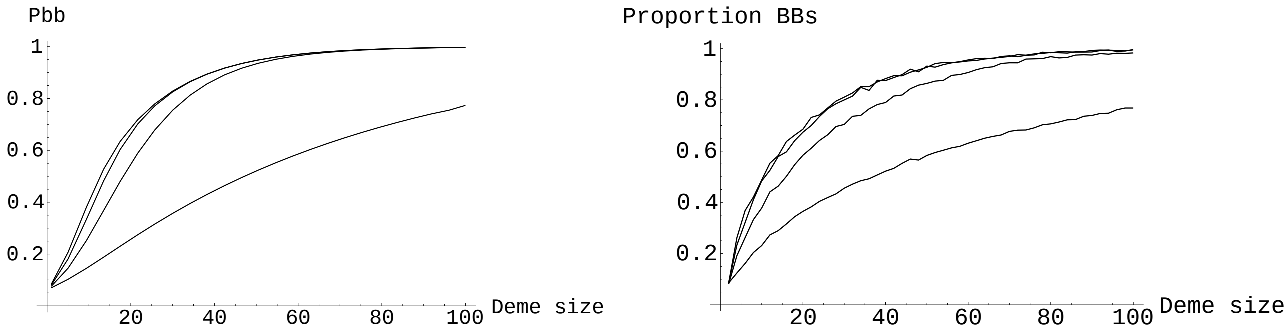
$$M_{ij} = \binom{r}{j} P_{bb}^j\left(\frac{in}{r}\right)\left(1 - P_{bb}\left(\frac{in}{r}\right)\right)^{r-j}. \qquad (4)$$

The distribution of the number of demes that converge correctly after an arbitrary number of epochs (t) is given by

$$V_t = V_1 M^t, \qquad (5)$$

and the probability of converging to the correct BB may be calculated using equation 3.

Figure 3 shows an example of the predictions of equation 3 after several epochs on four fully connected demes. The function used in the example is a 20-BB 4-bit trap problem. The GA uses binary tournament selection, two-point crossover with probability 1, and no mutation. Contrast the predictions with the iterated gambler's ruin example in table 1. Note that the major improvement



(a) **Theoretical predictions** (b) **Experimental results**

Figure 3 Probability of converging to the correct BB after 1,2,3,and 4 epochs (from right to left). This example considers a 20-BB 4-bit trap function on four fully connected demes with maximal migration.

in quality comes at the second epoch, and therefore the previous calculations of Cantú-Paz and Goldberg (1997) are very significant for the design of parallel GAs.

Another application of Markov chains is to calculate the long run distribution of the number of demes that find the BB, that is $\lim_{t\to\infty} V_t$. Substituting this distribution in equation 3 would give the probability that in the long run the parallel GA finds the BB. The remainder of this section will treat this problem, and find the expected absorption time.

First, note that when the none of the demes has the correct BB, it cannot be found in the next epoch. Once the correct BB disappears, there is no way of recovering it, because there is no mutation. Likewise, when all the demes converge to the correct BB, there is no chance of losing it. These facts are reflected in the transition matrix: the first row (corresponding to state 0) is 1,0,0,...0, and the last row (state r) is 0,0,...,1. The states 0 and r are called absorbing (or persistent) states, and because there is no possible transitions between them, the chain is said to have two closed absorbing sets. All the other states in the chain are transient states, because in the long run, the chain is expected to converge to one of the absorbing sets.

The fundamental matrix method (Isaacson & Madsen, 1976) is used to calculate the expected absorption time and the long run distribution of correct demes. To use this method the states need to be reordered, and the transition matrix rewritten as:

$$M = \begin{pmatrix} P_1 & 0 & 0 \\ 0 & P_2 & 0 \\ R_1 & R_2 & Q \end{pmatrix},$$

where P_1 and P_2 are the submatrices with the transition probabilities within the two closed persistent sets, which in our case consist of a single state each (therefore, $P_1 = P_2 = 1$); Q is a submatrix with the transition probabilities within the transient states; and R_1 and R_2 contain the probabilities of going from each transient state to the each of the persistent states.

The expected absorption time from each transient state i is given by the i-th element of

$$\tau = N\mathbf{1},$$

where the matrix $N = (I-Q)^{-1}$ is called the fundamental matrix, I is the identity matrix, and $\mathbf{1}$ is a column vector of ones. Augment τ with a zero at the beginning and a zero at the end to account for the expected absorption times from state 0 and state r, respectively. Now the mean time until absorption may be calculated by multiplying the initial distribution of demes with the right BB (given by equation 2) by the extended τ (τ'):

$$\overline{\tau} = V_1 \tau'. \tag{6}$$

The absorption probabilities from the transient state i to the persistent state l is given by the (i,l)-th entry of NR, where R is the matrix formed with the elements of $R1$ and $R2$. If P_2 is the submatrix for the state that represents finding the correct BB in all the demes, the distribution of probabilities, α, of being absorbed into that state (r) is given by the second column of NR. Augment α with a zero at the beginning and a one the end, to account for the chances of being absorbed from state 0 and state r, respectively. To find the mean probability of being absorbed at state r, multiply the initial distribution of demes with the correct BB by the extended vector (α'):

$$\overline{P_{bb}} = V_1 \alpha'. \tag{7}$$

Figure 4 shows that the probability that in the long

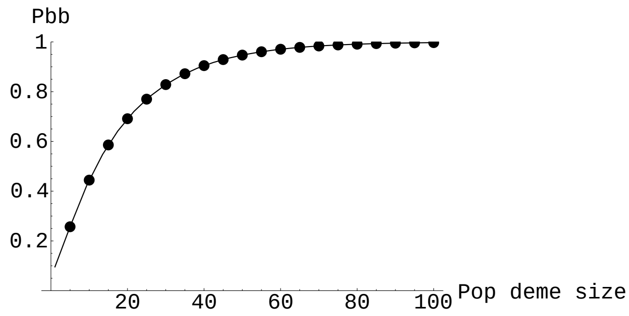


Figure 4 In the limit, a parallel GA with r fully connected populations using a maximal migration rate (dots) has the same chance of finding the solution than a simple GA with an aggregate population (continuous line).

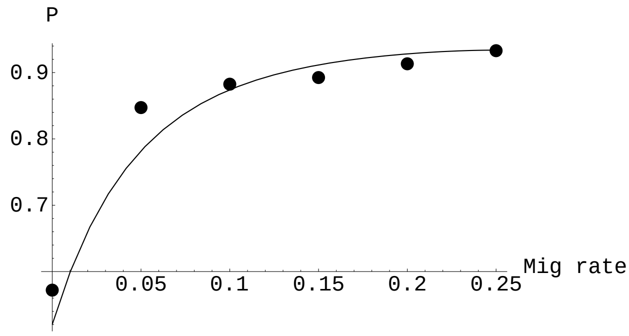


Figure 5 The probability of converging to the BB increases with higher migration rates. The example uses four demes with 50 individuals each; the test function is a 20-BB 4-bit trap problem; and only one migration epoch is considered. The theoretical predictions (continuous line) are compared against experimental results.

run r fully connected demes of size n_d will converge to the correct value in a partition is the same as the probability that a GA with a single population with rn_d individuals succeeds.

4 Lower migration rates

Using the maximal migration rate simplifies the calculation of the number of BB after each epoch, because the contribution from all the demes is uniform. This section extends the analysis to cases with lower migration rates. The form of the analysis is very similar, but with small migration rates the initial number of BBs in a particular deme depends greatly on whether the local deme converged correctly in the previous epoch.

For example, consider that in a given epoch two out of three demes of a parallel GA converge correctly, and suppose that each population sends 5% of its individuals to the two others. At the start of the next epoch there are two possibilities for a particular deme: if it converged correctly in the previous epoch, then 95% of its population has the correct BB (90% was already there, and it obtained 5% from the other correct deme). If it did not converge correctly, only 10% of the population would have the correct BB, contributed by the other two demes.

To reflect this situation, the Markov chain has additional states. As before, there is one absorbing state for when all the demes converge to the correct BB, and there is another for when they all converge incorrectly. For all the intermediate cases there are two states: one to denote when the major fraction of the population contains the BB (because the deme converged correctly in the previous iteration), and another state to represent when the major fraction is incorrect. Therefore, there are $2r$ total states, and they are numbered so that states 0 to $r-1$ represent the cases where the major fraction of the deme is incorrect, and states r to $2r$ represent the cases where major fraction is correct.

The initial distribution of the demes that converge correctly is

$$V_1(i) = \begin{cases} \binom{r-1}{i} P_{bb}^i(x_0)(1-P_{bb}(x_0))^{r-i} & \text{if } i < r, \\ \binom{r-1}{i-r} P_{bb}^{i-r+1}(x_0)(1-P_{bb}(x_0))^{2r-i-1)} & \text{if } i \geq r. \end{cases}$$

The transition matrix now becomes:

$$M_{ij} = \begin{cases} \binom{r-1}{j} P_{bb}(x)^j(1-P_{bb}(x))^{r-j} & \text{if } j < r, \\ \binom{r-1}{j-r} P_{bb}(x)^{j-r+1}(1-P_{bb}(x))^{2r-j-1} & \text{if } j \geq r, \end{cases}$$

where x is the starting point of the random walk for each case, and it depends on the migration rate ρ:

$$x = \begin{cases} ni\rho & \text{if } i < r, \\ n((i-r)\rho + 1 - \rho(r-1)) & \text{if } i \geq r. \end{cases}$$

As before, the probability of converging to the correct BB after t epochs is $P_{bb_t} = V_t U$, where $U(i) = P_{bb}(x)$, and x is defined as above.

Figure 5 shows that the probability of reaching the correct solution increases rapidly with higher migration rates. Since all the individuals are the same when migration occurs, there are no cost penalties associated with high rates, because only one individual needs to be sent, and it can be replicated any number of times at the receiving deme.

As before, the fundamental matrix method can be used to predict the long-term behavior of the parallel GA with low migration rates. The states have to be reordered as in the previous section. Figure 6 shows the probability that in the long-run the parallel GA will converge to the correct solution as a function of the migration rate. Note that in the long run only a small migration rate is sufficient to reach a solution of the same quality as a simple GA with a aggregate population.

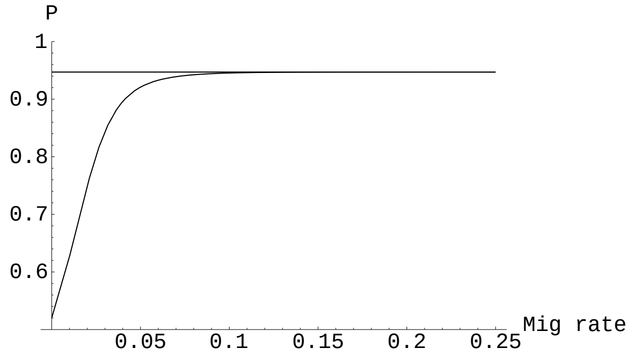


Figure 6 The long-term probability of converging to the correct BB increases rapidly with higher migration rates. The example considers four fully connected demes, with 50 individuals each, working on a 20-BB 4-bit trap function. The horizontal line is the probability that four fully connected demes with maximal migration (or a simple GA with 200 individuals) will eventually converge to the right BB.

5 Conclusions

This paper examined a bounding case of a parallel GA with multiple fully connected populations. The analysis focused on calculating the probability that a partition converges to the correct BB, and Markov chains were used to determine the convergence quality for an arbitrary number of epochs, and also in the long run. Initially, the case where migration is set at its maximum possible value was studied, and it was later refined to consider smaller migration rates.

The first conclusion of this study is that in the long run, a set of fully connected populations using the maximal migration rate is expected to find a solution of the same quality as a simple GA with an aggregate population. The second conclusion is that the greatest gain in quality comes after the second epoch. Running the populations in isolation (one epoch) does not produce very good results, and iterating the algorithm more than two times does not seem to yield significantly better solutions.

The second part of the analysis shows that the quality improves with higher migration rates. There is no reason to use a low migration rate in the algorithm examined here, specially since the communications cost is the same regardless of the number of migrants.

However, these results about the quality of the search have to be interpreted carefully, because we need to consider the cost of reaching the solution before committing to a particular configuration. Even though the analysis shows how to calculate the expected number of epochs until all the populations converge to the same solution, there is no mention about the duration of each epoch or about the cost of communications. That is a necessary extension of the work presented here. Another possible extension is to consider other less densely connected topologies that might be more attractive to users.

The conclusions of this study should aid in the design of multiple-deme parallel GAs. High migration rates and few (as low as two) epochs seem to be good design choices for the algorithm described in the paper.

Acknowledgments

I wish to thank Prof. David E. Goldberg for his useful comments on the material presented here.

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

This study was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number 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.

References

Braun, H. C. (1990). On solving traveling 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., & Goldberg, D. E. (1997). 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. (1997). 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.

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.

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.

Isaacson, D. L., & Madsen, R. W. (1976). *Markov chains theory and applications*. New York, NY: John Wiley and Sons, Inc.

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.