Simple Analytical Models of Genetic Algorithms for Multimodal Function Optimization

Samir W. Mahfoud
Department of Computer Science
University of Illinois at Urbana-Champaign
1304 West Springfield Avenue
Urbana, IL 61801

IlliGAL Report No. 93001
February 1993

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

Simple Analytical Models of Genetic Algorithms for Multimodal Function Optimization

Samir W. Mahfoud
Department of Computer Science
University of Illinois at Urbana-Champaign
1304 West Springfield Avenue
Urbana, IL 61801
mahfoud@gal4.ge.uiuc.edu

Abstract

This paper presents simple analytical models of genetic algorithms which are commonly used in multimodal function optimization. The methodology for constructing the models is similar throughout the study. The predictive value of each model is verified by running the corresponding genetic algorithm on various multimodal functions of varying complexity.

1 Introduction

Genetic algorithm researchers have for a long time been aware that given a problem with multiple solutions, the traditional genetic algorithm (GA) will at best, ultimately converge to a population containing only one of those solutions. This observation prompted the development of GAs capable of forming "niches", often for the purpose of multimodal function optimization. While many such GA variations have been attempted, most can be classified either as *crowding* schemes (Cavicchio, 1970; Culberson, 1992; De Jong, 1975; Mahfoud, 1992), or as *sharing* schemes (Deb, 1989; Goldberg & Richardson, 1987; Oei, Goldberg, & Chang, in press; Yin & Germay, 1992). This study will construct simple analytical models of appropriately chosen crowding and sharing schemes. In addition, the general method of construction will be demonstrated on a simple GA. Since variation about local points in the search space is not of much interest in multimodal function optimization, all models and GAs will incorporate only the operators of selection and crossover, not mutation.

Previous GA models have for the most part been highly complex, and are typically based on Markov chains. Several studies have attempted string-level models of proportionate selection, crossover, and mutation, using Markov chains. While such models are theoretically sound, they either assume an infinite population size (Vose, in press), or the combinatorics become overwhelming as population size and string length increase. Even at the equivalence-class level, if modelling is via Markov chain, the number of possible states quickly becomes prohibitive as the number of classes increases beyond two, and as the population size grows (Mahfoud, 1991). This study models multimodal GAs at the equivalence-class level, and demonstrates simple, analytical alternatives to Markov chain models. It represents a preliminary step towards the development of a framework for the analysis and design of multimodal genetic algorithms, and ultimately, towards a comprehensive theory of diversity in genetic algorithms.

2 Crowding Methods

Crowding, originally proposed by De Jong (1975) in his "crowding factor model", is motivated by analogy with competition for limited resources among similar members of a natural population. Dissimilar population members, often of differing species, occupy different environmental niches, and therefore do not typically compete for resources. Similar individuals, on the other hand, tend to occupy the same environmental niches, and therefore must compete for the same limited resources. When a niche has reached its carrying capacity, weaker members of that niche will be crowded out of the population by stronger members. Older members of the niche will eventually be replaced by the fittest of the younger members.

In the artificial world, any GA which attempts to maintain stable subpopulations by replacing population elements with like individuals, can be called a *crowding method*. As an example, GAs which replace parents only with their offspring, fall under this definition of crowding method.

De Jong's crowding works by reproducing and killing off a fixed percentage of the population each generation. Each newly generated population member must replace an existing member, preferably the most similar one. To accomplish an approximation of this, a small sample is taken from the existing population, and the element closest in Hamming distance to the new one is replaced.

Mahfoud (1992) points out that stochastic "replacement errors" prevent the basic crowding algorithm from maintaining more than two peaks of a multimodal fitness landscape. In addition, measures other than Hamming distance, such as phenotypic distance (Goldberg & Richardson, 1987), are often effective in reducing stochastic errors. Mahfoud outlines an improved crowding algorithm called *deterministic crowding*, which nearly eliminates replacement error, and which is capable of locating and maintaining multiple peaks, given a multimodal function.

Deterministic crowding works by randomly pairing all population elements each generation, to yield $n/2$ pairs, where n is the population size. Each such pair of parents undergoes crossover, possibly in combination with mutation or other genetic operators, to yield two children. The two children then compete against the parents for inclusion in the population.

The method of competition between children and parents is a crucial step, whereby a small algorithmic change typically yields an enormous change in the algorithm's capabilities and behavior. For example, making a random choice at each opportunity regarding which of the two children competes against which of the two parents, yields an algorithm much like Cavicchio's (1970) "preselection". Due to replacement errors, this algorithm is incapable of maintaining multiple peaks. As a further example, holding a "partner's tournament", also referred to as double acceptance/rejection (Mahfoud & Goldberg, 1992), can yield highly interesting results, depending upon the acceptance criterion employed. In this variation, either both children win the tournament and are accepted into the population of the next generation, or both parents win it. One possible acceptance criterion, choosing the pair (either parents or children) containing the maximally fit element (of the four), results in a GA similar to the "gene invariant genetic algorithm" (Culberson, 1992), which, in the absence of operators other than crossover, maintains a constant frequency of each allele in the population. The behavior of this variation on multimodal functions requires further research.

Deterministic crowding requires that the closest individuals compete. To accomplish this, the two possible parent-child tournaments are considered: parent 1 against child 1 and parent 2 against child 2, or parent 1 against child 2 and parent 2 against child 1. The pair of tournaments which forces competition between the closest (on the average) parents and children is held. Closeness is computed according to some appropriate distance measure (phenotypic distance

was employed in the original study (Mahfoud, 1992)). Because of its previous success and high level of stability, deterministic crowding is used as the representative crowding method in the remainder of this paper.

3 Sharing methods

Sharing methods require that fitness be shared as a single resource among similar individuals. Fitness sharing, or more simply, *sharing*, was introduced by Goldberg and Richardson (1987) as the "method of sharing functions". Sharing works by derating the fitness of each population element by an amount related to the number of similar individuals in the population. Specifically, a member's new fitness is equal to its old fitness, divided by its *niche count*. An individual's niche count is a sum of sharing function values between it and every individual in the population (including itself). A *sharing function* is a function of two population elements, which returns a '1' if the elements are identical, a '0' if they cross some threshold of dissimilarity, and an intermediate value for intermediate levels of similarity. The threshold of dissimilarity is specified by a constant, σ_{share}, which indicates the maximum allowable distance for non-zero sharing to occur. To date, only sharing functions of the form,

$$sh(d) = 1 - \left(\frac{d}{\sigma_{share}}\right)^{\alpha} \quad (d < \sigma_{share}) \quad , \tag{1}$$

have been used, where d is the difference between population elements, and α is a constant (typically 1.0) used to regulate the shape of the sharing function. Both genotypic and phenotypic difference measures can be employed, the appropriate choice depending on the problem.

Sharing can be implemented using any selection method, though stochastic remainder selection has been the most popular to date. Stability problems in implementing sharing using tournament selection are addressed by Oei, Goldberg, and Chang (in press), who propose a new sharing method which incorporates binary tournament selection, and which calculates shared fitnesses with respect to the new population as it is being filled. This method, called, "tournament selection with continuously updated sharing", is used by Goldberg, Deb, and Horn (1992), in conjunction with fitness scaling prior to sharing, to overcome the inherent deceptiveness of a multimodal, deceptive function. The authors also implement a previous suggestion to estimate shared fitnesses through population sampling, rather than cycle through the entire population. Yin and Germay (1992) propose that a clustering algorithm be implemented prior to sharing to divide the population into niches so that each individual will share only with the individuals within its niche. Since shared fitness for each individual is not computed with respect to the entire population, the algorithm can be expected to be faster than traditional sharing. One further possibility for sharing is to use stochastic universal selection, because of its known stability, rather than other selection schemes. Because of its simplicity, sharing with roulette-wheel selection is used as the representative sharing method in the remainder of this study.

4 Two-Class Models

We commence with a discussion of the most basic type of model, the two-class model, upon which multi-class models will be based. The term, *class*, refers to an equivalence class, induced through partitioning the search space based on the local maximum to which each point is attracted. In other words, since the primary emphasis of this study is multimodal function

maximization, equivalence classes correspond to local maxima (peaks) in the fitness landscape, and each population element is considered a member of the peak to which it is attracted.

To fully understand the concepts of class membership and attraction, some basic definitions are necessary. Assume a search space, S. Assume also an objective function, f, which assigns a real function value to each element of S ($f : S \rightarrow R$). Assume without loss of generality that we wish to perform maximization with respect to f. Define a neighborhood generation function, $N : S \rightarrow 2^S$, which assigns each element i of S a set of elements in S which are close to i in some sense (i need not be in its own neighborhood).

For an arbitrary $i \in S$, let $S_i = N(i) \subseteq S$ be called the *neighborhood* of i. i is a *local maximum* if $f(i) \geq f(j)$ for all $j \in S_i$. i is a *strict local maximum* if $f(i) > f(j)$ for all $j \in S_i$ ($j \neq i$). Define all local maxima (either all maxima or only the strict ones) to be *attractors*. Any point in the search space will be *primarily attracted* (we will also refer to this concept as simply, *attracted*) to one or more local optima. To find the local optimum/optima to which any point, i, in the search space is attracted, execute the following hillclimbing algorithm:

$CurSet = \{i\}$
$OldSet = \emptyset$
Repeat
    j = Find an element in $CurSet$ with highest value for f
    $Curset = Curset - (Curset \bigcap OldSet)$
    Eliminate all points from $CurSet$ with fitness value less than $f(j)$
    If any points in $CurSet$ are local optima, stop and return $CurSet$
    $OldSet = OldSet \bigcup Curset$
    $CurSet$ = Apply N to all elements of $CurSet$

The above algorithm is deterministic in that it will return the set of local optima to which a point is closest, even in the case of large plateaus. The algorithm terminates whenever it finds one or more local optima. The definition of local optimum can be restricted (to strict optima only), if one wishes to not terminate on plateaus. If we do not know all local optima beforehand, and are not restricting the definition of local optima, we can iterate until no further improvement occurs.

A local optimum's *region of attraction* is the set of all points in S which are primarily attracted to it. Regions of attraction may overlap. A local optimum's *strict region of attraction* is the set of all points in S which are primarily attracted to it, and are not primarily attracted to any other local optimum. Strict regions of attraction do not overlap. There will be a subset of S which contains points that are not strictly attracted to any local optimum (boundary points that are probably of low fitness). As an example, if N is defined as a neighborhood of $\pm\epsilon$ in the phenotype, and if the phenotype contains only one variable, then points which lie under a peak will be in the region of attraction of that peak. In fact, such an N is assumed throughout this study. Regions of attraction can be made non-overlapping, getting rid of the need for a "none of the above" class, by enforcing some preference on the peak to which a point will be considered attracted in the case of a tie.

In the GA, globally, we are interested in the peaks under which selection and crossover deposit points. The selection/crossover algorithm has done its job if it deposits in a desirable distribution. We assume selection/mutation or some other type of genetic hillclimbing can move a point to the top of a peak once it has been successfully deposited.

The two-class models assume two equivalence classes, A and B, containing respectively I_A and I_B elements ($I_A + I_B = n$). This corresponds to a bimodal objective function, where either

one or both local optima are global. The models also assume that for each class, all elements of that class will have identical fitness. This assumption becomes increasingly valid as runs approach equilibrium. Denote the fitnesses of the two classes by f_A and f_B. In these models, we assume all points in the search space are attracted to only one peak, and therefore there is no "none of the above" class. As specified earlier, the combination of selection and crossover is modelled without mutation. The final assumption has to do with the class membership of offspring produced by crossover. We assume that the cross of two elements from the same class will yield two members of that class, and that the cross of two elements from differing classes will yield one member of each class. This latter assumption will be relaxed in the multi-class models.

4.1 The simple genetic algorithm

For simplicity, we model roulette-wheel selection (RWS), although stochastic remainder selection and stochastic universal selection are more stable. Under RWS, in any given trial, the probability of selecting some element of class A is proportional to fitness, and is given by the expression,

$$P_s(A) = \frac{I_A f_A}{I_A f_A + I_B f_B} \quad . \tag{2}$$

Employing the formulas for the mean and variance of the binomial distribution for I_A, the expected number of population elements in A after one generation (n trials) is

$$\mu_A = n \; P_s(A) \quad . \tag{3}$$

The variance in the expected number of elements in class A is given by

$$\sigma_A^2 = n \; P_s(A) \; P_s(B) \quad , \tag{4}$$

where $P_s(B) = 1 - P_s(A)$. Crossover does not affect the mean or variance.

While the above results are neither new nor surprising, they do tell us a great deal about the behavior of a GA which runs RWS and crossover on a two-class problem. The distribution at time, $t + 1$, of population elements among classes depends both upon the distribution at time, t, and the relative fitnesses of the two classes. In addition, given two classes of equal fitness, where one class has become more frequent than the other, there is no restorative pressure — the expected number of each class in a given generation is the same as the actual number of each in the previous generation. The lack of restorative pressure, in combination with significant variance and finite population size, leads to genetic drift and the eventual disappearance of one class (Goldberg & Segrest, 1987). Given two classes of unequal fitness, the higher-fit class is expected to take over the entire population. A rough estimate of takeover time can be obtained by iterating the proportional form of Equation 3.

The above model of a traditional GA demonstrates a simple construction technique on a familiar GA. We now consider the application of such a technique to the modelling of multimodal GAs.

4.2 Sharing

Let us now add fitness sharing to the simple GA of the previous section. We will make the assumption that the two classes are fully distinguishable via such techniques as the proper setting of σ_{share} or clustering. Therefore if $a \in A$ and $b \in B$, then $sh(d(a,b)) = sh(d(b,a)) = 0$,

where d is the distance measure, and sh is the sharing function. Since we previously made the assumption that all elements of a class have identical fitness, the elements of a class contribute 100% to each other's fitness: $sh(d(a,a)) = sh(d(b,b)) = 1$. Shared fitnesses, denoted by f', are given by, $f'_A = f_A/I_A$ and $f'_B = f_B/I_B \quad (I_A, I_B \neq 0)$. Substitution into Equation 2 yields:

$$P_s(A) = \frac{I_A f'_A}{I_A f'_A + I_B f'_B} = \frac{f_A}{f_A + f_B} = \frac{r}{r+1} \quad , \tag{5}$$

where $r = f_A/f_B$. The mean and variance for the number of population elements in A after one generation (n trials) are given in the following two equations:

$$\mu_a = n\ P_s(A) = \frac{nr}{r+1} \tag{6}$$

$$\sigma_A^2 = n\ P_s(A)\ P_s(B) = \frac{nr}{(r+1)^2} \tag{7}$$

This time, the expected distribution and its variance are independent of the starting distribution. This explains the restorative pressure inherent in sharing methods, where classes which contain other than their share of population elements, can expect to be restored to the proper number in one generation. This restorative pressure keeps drift from becoming a major factor, since stochastic fluctuations are not given the time to accumulate over multiple generations. Given two classes of unequal fitness, the fitter class will not take over, unless the expected number of individuals in the less fit class is small enough to be overcome by noise (Deb, 1989). Finally, the expected distribution and its variance depend only upon the relative fitnesses of the two classes.

4.3 Crowding

In deterministic crowding, sampling occurs without replacement. Each individual gets exactly one chance for inclusion in the next generation. We will assume that an element in a given class, regardless of the difference measure employed, is closer to elements of its own class than to elements of other classes. Recall our prior assumptions that a cross between two elements of the same class yields two elements of that class, and a cross between two elements of differing classes yields one element from both classes. Therefore, if two elements of class A get randomly paired, the offspring will also be of class A, and the resulting tournament will advance two class A elements to the next generation. The random pairing of two class B elements will similarly result in no net change to the distribution of the next generation. If an element of class A gets paired with an element of class B, one offspring will be from class A, and the other from class B. The class A offspring will compete against the class A parent, the class B offspring against the class B parent. The end result will be that one element of both classes advances to the next generation — no net change. Since each element receives exactly one trial, the mean and variance for the number of population elements in A after one generation (n trials) are $\mu_A = I_A$ and $\sigma_A^2 = 0$.

Crowding's expected behavior is quite different from that of sharing. The expected distribution is identical to the prior distribution, with no variance. No restorative pressure is present, but no drift is either. Also, the expected distribution is independent of fitness. A highly fit class will not gain elements from a lowly fit class. Stability is guaranteed, regardless of class-fitness differential.

One might ask what a crowding algorithm can accomplish if there can never be shift between classes. The answer is that improvement occurs within classes. In addition, multi-class models

do not display such rigidity; as demonstrated later, highly interesting class interactions may occur.

5 Verification of Two-Class Models

The test problems used to verify the two-class models are bimodal functions, where the most significant (leftmost) bit of each population element determines the peak to which it is attracted, and therefore the class of which it is a member. Strings are eight bits in length, and decode from binary to a single integer variable from 0 to 255. Strings with a '0' in the leftmost position (ranging from 0 to 127) are considered members of class A, while strings with '1' in the leftmost position (ranging from 128 to 255) are considered members of class B. Figures 1a and 1c show the actual functions. Function F1 has two peaks of equal height. In F2, the peak corresponding to class A is four times the height of that corresponding to class B.

A simple GA with RWS, sharing with RWS, and deterministic crowding are not altered to accommodate these class definitions, but are run as usual, using population size, $n = 30$, probability of crossover, $p_c = 1.0$, probability of mutation, $p_m = 0.0$, number of generations, $G = 50$, single-point crossover, random initialization, and a phenotypic difference measure (for crowding and sharing). For sharing, the sharing function of Equation 1 is employed, with $\sigma_{share} = 64$, and $\alpha = 1.0$.

Figure 1b tracks the number of individuals in class B (I_B) from generation to generation ($I_A = n - I_B$), for each of the three algorithms, running on F1. Figure 1d does the same for F2. Figures 1a and 1c show respectively, the final distributions for sharing on F1, and crowding on F2.

The test runs illustrate the predictive value of the two-class models. As expected, RWS's high variance and lack of restorative pressure cause the simple GA, on F1, to fluctuate about the current point in both directions, never regaining ground, until class B has lost all its elements (Figure 1b). On F2, class B, the weaker of the two, is rapidly and steadily driven to extinction (Figure 1d).

Sharing, since it incorporates RWS, also exhibits much the same fluctuation. However, since the expected distribution under sharing does not depend upon the previous distribution, drift does not occur. Instead, the sharing algorithm fluctuates about the mean point of Equation 6, an amount indicated by the variance calculation in Equation 7. On F1, sharing, as expected, maintains two classes on the average in nearly equal proportions (Figure 1b). On F2, sharing maintains on the average the expected number of six elements in class B (Figure 1d). Note that at times, sharing fluctuates dangerously close to zero. This will be a greater problem for higher fitness ratios. In fact, when the fitness ratio was increased from 4:1 to 8:1, sharing was unable to maintain to generation 50, any of the expected 3.33 elements in class B.

The sharing model sets the groundwork for computing a good estimate for the expected time to extinction of the lesser class. First, the probability of extinction in a single generation from equilibrium must be calculated (call this quantity p_e. Then, assuming independent trials (note that the sharing model is not a Markov chain), the expected extinction time is simply $1/p_e$. Slightly more sophisticated estimates are also possible. Once the above quantity is known, population sizing calculations can proceed, based on maintaining the weakest class a sufficient number of generations.

Crowding, as expected, maintains the status quo distribution, exhibiting the predicted stability of zero variance on both F1 and F2 (Figures 1b and 1d). Although random initialization has given class B 16 elements to class A's 14 elements, crowding exhibits no pressure to move

towards any other ratio, even on F2, where A has four times the fitness of B.

6 Multi-Class Models

We now consider the extension of the crowding model to multiple classes. While in the two-class model, the products of crossover do not affect the results produced by selection, in the multi-class model, crossover yields interesting interactions among classes. Although in sharing, selection will ultimately compensate for such interactions, in crowding, selection may be steered toward trajectories other than the perfectly horizontal (of Figures 1b and 1d). This is where crowding starts to gain power over a simple parallel hillclimber — the power to migrate to higher peaks.

Recall our previous assumption that all class elements have identical fitness. This assumption is valid when a multimodal GA runs at or near equilibrium, when population individuals rapidly congregate at the top of peaks. Let the class fitnesses, f_A, f_B, ..., correspond to peak fitnesses (ϵg to the locally maximal values of the objective function). We must account for the possible offspring of each crossover. For crowding, let us also make the assumption that an offspring from a class of higher fitness, when competing against a parent from a class of lower fitness, will always win.

Consider the four-peaked functions, F3 and F4, of Figures 1e and 1g. Class membership, in both cases, is determined solely by the leftmost two bits of the eight-bit chromosome. Class A individuals have '00' in the most significant bits, class B, '01', class C, '10', and class D, '11'. Obviously, crosses between two elements from the same class will yield two offspring from that class, with no net change after the tournament. Let us now consider hybrid crosses. Crossing A with B or C always yields one element from each class involved in the cross. After like competes against like, we have no net change in distribution. Likewise, crossing B with D yields no net change. The two remaining possibilities are the ones of interest. Most of the time, the cross point will fall away from the most significant bits, resulting in no net change. However, with probability, $1/(l-1)$, the crossover point will fall between the most significant bits, resulting in two offspring of different classes. Such a cross between elements of A and D will yield two offspring, one from B and one from C; such a cross between B and C will yield one A and one D. Note that the special cases above are symmetric, so that given a uniform starting distribution, no one class is expected to be produced by crossover more frequently than any other.

In the two special cases above, phenotypic closeness will result in tournaments between A and B, and between C and D, with the fitter class winning. Consider F3 (Figure 1e), where all four peaks are of identical height. The crowding algorithm, as specified in an earlier section, prefers parents to their offspring in case of a tie. Therefore, no migration between peaks will occur under the model. In the actual simulation, using the same parameters as described earlier, after the population had reached equilibrium, no further migration did occur. Note in Figure 1f the symmetry of the curves corresponding to the number of individuals in A and B, and also the symmetry of those for C and D. Prior to equilibrium, A can only gain an individual if it takes it from B, and vice-versa. The combined number of individuals in C and D likewise remains constant.

As an example of analysis contributing to design, let us alter deterministic crowding so that in the case of a tie, it always advances the child over the parent. For functions such as F3, with multiple global optima, the migration discussed above will occur freely, even after equilibrium. The inevitable result of such migration for an algorithm with no restorative pressure, will be

drift, though crowding will drift much more slowly than did the simple GA. In an earlier paper (Mahfoud, 1992), deterministic crowding was actually implemented in this way. This explains why, given a sine function with five peaks of equal height, one peak contained only a few elements, while the other peaks divided the rest. Similar instability in preliminary runs on the functions of the current study led to the requirement that offspring be strictly fitter than their parents in order to advance.

On F4 (Figure 1g), class A is twice as fit as the other three. Our model predicts that A will win all tournaments against B when a cross between B and C yields elements of A and D. Therefore, B will eventually contribute all its elements to A. C and D, however, will not transfer elements between themselves. In Figure 1h, we see that C and D reach equilibrium around generation 40, at which point they no longer exchange elements. A and B, on the other hand, reach a quasi-equilibrium around generation 35, where A has taken most, but not all of B's elements. This quasi-equilibrium lasts to generation 60, after which, A gains the rest of B's elements. We will refer to peak A as a *dominating* peak (or class), and to peak B as a *dominated* peak.

This brings to mind two phenomena. First, if one is interested in identifying dominated classes, one can probably do so by stopping the run upon preliminary convergence, rather than waiting for full convergence. The second phenomenon is that crowding demonstrates some properties of sharing, albeit in different ways. Based on preliminary results, we can define a dominated class as one which, with the assistance of another class, can cross to form a fitter class, and which, due to close proximity in the search space to the fitter class, must compete against it and be sacrificed. In sharing, peaks in close proximity to higher peaks may donate all their elements to the higher peak if σ_{share} is too large to distinguish the two. Furthermore, crowding may allocate elements to classes somewhat proportionally (to fitness), if fitter classes can gain elements by dominating less fit classes in close proximity.

One further phenomenon deserves mention: the *assist*. Recall that on F4, migration from class B to class A occurs with the assistance of C, since in order for a single migration to occur, B and C must cross to produce A and D. Suppose that class D is also made a dominating class, so that peaks A and D are each twice the height of both B and C. We might expect that B will eventually transfer all its elements to A, and that C will eventually transfer all its elements to D. However, our model tells us otherwise. Since B and C must assist each other in migration, whenever one of those classes has been depleted, the other becomes stable and no longer transfers any of its elements. Without the assist, the weaker class is no longer dominated. One run was performed to verify this. After class C had contributed all its elements to class D, class B still had three elements remaining. The three elements were maintained to generation 1000, at which point the run was stopped.

7 Conclusion

This study has been a departure from the full Markov chain models of previous studies, towards a simpler, analytical model of the multimodal genetic algorithm. Intrinsic properties of sharing and crowding algorithms were both verified and uncovered using the simple models. Previously, little was known about the expected behavior of deterministic crowding. Sharing and crowding were chosen for examination because they are representative of many potential multimodal genetic optimizers. Geographical, parallel genetic algorithms are a third classification that is rapidly gaining favor, and that deserves future study (for example, see Spiessen's and Manderick's (1991) paper).

Criteria for choosing the representative crowding and sharing algorithms were stability and ease of analysis. Analysis of other sharing methods, such as those which incorporate increasingly stable selection methods, is in progress. Analysis of additional crowding methods is also in progress. Other current research deals with extension and enhancement of multi-class models for various multimodal genetic algorithms, time-to-extinction calculations, population sizing, and further exploration of crossover-induced interactions, including dominating, dominated, and assisting classes.

Ultimately, one goal of such modelling is to be confident that the predictions of a model will be characteristic of an algorithm's behavior on both difficult problems and application problems. Currently, deterministic crowding is being applied to the massively multimodal, deceptive function of Goldberg, Deb, and Horn (1992), with good results in locating all 32 global optima. Crowding's behavior corresponds well to that predicted by the model, with local, deceptive optima being dominated by globals. Another goal is that analysis of an algorithm will lead to its improvement; we have witnessed one such case in this paper. In fact, the deterministic crowding algorithm was developed upon analysis of De Jong's crowding and Cavicchio's preselection (Mahfoud, 1992), as an improved hybrid of both. Such analysis should also lead to a better understanding of prior results, such as why De Jong's crowding maintains exactly two classes (Deb, 1989), and why deterministic crowding was losing elements from one of five equal peaks of a sine function (Mahfoud, 1992).

In addition, one may be driven to ask whether certain enhancements to an algorithm are desirable. For example, mating restrictions, when added to sharing, were shown to yield a cleaner version of sharing, without lethal hybrid crosses between elements of different niches (Deb, 1989). However, mating restrictions also prevent beneficial hybrids from forming. In the case of this paper's crowding examples, migration to higher peaks would have been prevented had mating restrictions been additionally enforced.

In summary, the small steps taken in this paper have started to yield large benefits in the understanding of multimodal genetic function optimizers. Follow-up to and extension of the models of this study, should yield a general framework for the analysis and design of multimodal genetic algorithms, and more generally, a theory of diversity in genetic algorithms.

Acknowledgments

This work is supported by the National Science Foundation under Grant ECS-9022007.

References

Cavicchio, D. J. (1970). *Adaptive search using simulated evolution.* Unpublished doctoral dissertation, University of Michigan, Ann Arbor.

Culberson, J. C. (1992). *Genetic invariance: a new paradigm for genetic algorithm design.* Unpublished manuscript, Department of Computing Science, University of Alberta, Edmonton, Alberta (ftp thorhild.cs.ualberta.ca in pub/GIGA).

De Jong, K. A. (1975). An analysis of the behavior of a class of genetic adaptive systems. (Doctoral dissertation, University of Michigan). *Dissertation Abstracts International, 36*(10), 5140B. (University Microfilms No. 76-9381).

Deb, K. (1989). *Genetic algorithms in multimodal function optimization* (Masters Thesis and TCGA Report 89002). Tuscaloosa: University of Alabama, The Clearinghouse for Genetic Algorithms.

Goldberg, D. E., Deb, K., & Horn, J. (1992). Massive multimodality, deception, and genetic algorithms. In R. Manner and B. Manderick (Eds.), *Parallel Problem Solving From Nature, 2* (pp. 37–46). Elsevier: Amsterdam.

Goldberg, D. E., & Richardson, J. J. (1987). Genetic algorithms with sharing for multimodal function optimization. *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, 41–49.

Goldberg, D. E., & Segrest, P. (1987). Finite Markov chain analysis of genetic algorithms. *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, 1–8.

Mahfoud, S. W. (1991). *An analysis of Boltzmann tournament selection* (IlliGAL Report 91007). Urbana: University of Illinois, Illinois Genetic Algorithms Lab. Submitted for Publication.

Mahfoud, S. W. (1992). Crowding and preselection revisited. In R. Manner and B. Manderick (Eds.), *Parallel Problem Solving From Nature, 2* (pp. 27–36). Elsevier: Amsterdam.

Mahfoud, S. W., & Goldberg, D. E. (1992). A genetic algorithm for parallel simulated annealing. In R. Manner and B. Manderick (Eds.), *Parallel Problem Solving From Nature, 2* (pp. 301–310). Elsevier: Amsterdam (also IlliGAL Report 92002).

Oei, C. K., Goldberg, D. E., & Chang, S. J. (in press). Tournament selection, niching, and the preservation of diversity. *Proceedings of the Second Workshop on Foundations of Genetic Algorithms.*

Spiessens, P. & Manderick, B. (1991). A massively parallel genetic algorithm. *Proceedings of the Fourth International Conference on Genetic Algorithms*, 279–286.

Vose, M. (in press). Modeling simple genetic algorithms. *Proceedings of the Second Workshop on Foundations of Genetic Algorithms.*

Yin, X., & Germay, N. (1992). *A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization.* Submitted for Publication.

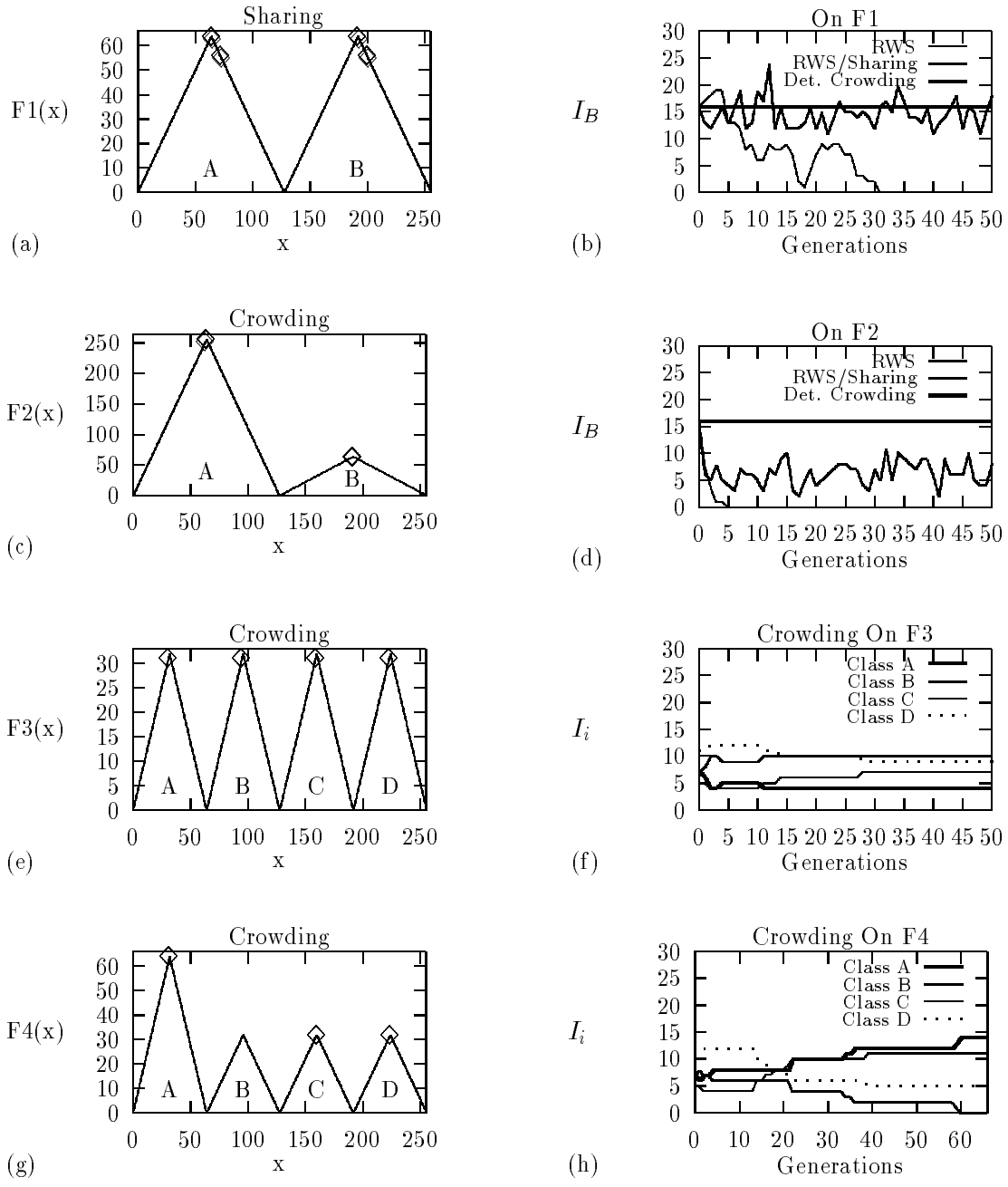

Figure 1: Graphs a, c, e and g show the distribution of population elements after either 50 (a, c, and e) or 100 (g) generations. Graphs b, d, f, and h track the number of elements in various classes.