

# **Genetic Algorithms with Dynamic Niche Sharing for Multimodal Function Optimization**

Brad L. Miller  
Dept. of Computer Science  
University of Illinois  
at Urbana-Champaign

Michael J. Shaw  
Beckman Institute  
University of Illinois  
at Urbana-Champaign

IlliGAL Report No. 95010  
December 1995

Department of General Engineering  
University of Illinois at Urbana-Champaign  
117 Transportation Building  
104 South Mathews Avenue  
Urbana, IL 61801

# Genetic Algorithms with Dynamic Niche Sharing for Multimodal Function Optimization

Brad L. Miller  
Dept. of Computer Science  
University of Illinois  
at Urbana-Champaign  
bmiller@uiuc.edu

Michael J. Shaw  
Beckman Institute  
University of Illinois  
at Urbana-Champaign  
mshaw@ux1.cso.uiuc.edu

December 1, 1995

## Abstract

Genetic algorithms utilize populations of individual hypotheses that converge over time to a single optimum, even within a multimodal domain. This paper examines methods that enable genetic algorithms to identify multiple optima within multimodal domains by maintaining population members within the niches defined by the multiple optima. A new mechanism, Dynamic Niche Sharing, is developed that is able to efficiently identify and search multiple niches (peaks) in a multimodal domain. Dynamic niche sharing is shown to perform better than two other methods for multiple optima identification, Standard Sharing and Deterministic Crowding.

To further improve performance, mating restrictions are used to increase the likelihood of producing highly fit offspring. Two new mating restriction mechanisms, Dynamic Line Breeding and Dynamic Inbreeding, are introduced that utilize dynamic niche sharing to proportionately populate local optima in a multimodal domain. Experiments presented in this paper demonstrate that dynamic niche sharing, dynamic line breeding, and dynamic inbreeding are fast and efficient mechanisms for identifying multiple optima within a multimodal domain.

## 1 Introduction

Genetic algorithms (GAs) have proven useful in a variety of unimodal domains, as they have been effective at identifying and converging populations of candidate hypotheses to a single global optimum. However, some domains require the identification of multiple optima in the domain. Examples of such domains include the learning of useful financial decision knowledge from examples, where identifying relevant knowledge and useful intermediate concepts is important (Sikora & Shaw, 1994). Another example is in identifying a set of diverse rules that together can be used as the basis for a classifier system (Horn, Goldberg, & Deb, 1994). Traditional GAs can successfully identify the best rule (optimum) in the domain, but are unable to maintain rules of secondary importance due to population convergence to the best rule. This illustrates why multimodal domains that require a genetic algorithm to identify multiple local optima in parallel have been difficult for GAs.

A standard GA will eventually converge to a single peak, even in domains characterized by multiple peaks of equivalent fitness. This is a result of *genetic drift* (De Jong, 1975), which is the tendency of a GA to converge over time to one hypothesis within the domain. Various population diversity mechanisms have been proposed that force a genetic algorithm to maintain a diverse population of members throughout its search, so as to avoid convergence of the population to a single peak. These mechanisms allow a GA to identify multiple optima in a multimodal domain, and not just a single global optimum. This study is interested in both the number and quality of optima obtained by genetic algorithms with different population diversity mechanisms.

This paper will explore several different population mechanisms that enable a GA to identify local as well as global optimum.

## 2 Niche Formation Methods

In a multimodal domain, each peak can be thought of as a niche. The analogy from nature is that within the environment there are different subspaces (niches) that can support different types of life (species, or organisms). The number of organisms contained within a niche is determined both by the fertility of the niche (or *carrying capacity*), and the efficiency of each organism at exploiting niche fertility. If there are too many organisms for a given niche, there will not be enough food to go around, and the least efficient organisms will die. Conversely, if there are few organisms in a very fertile niche, those organisms will quickly reproduce so as to fully exploit the niche’s ability to support life.

In a similar manner, genetic algorithms can maintain the population diversity of its members in a multimodal domain. Each peak of a multimodal domain can be thought of as a niche that can support a certain number of concepts. The number of individuals supported by a niche should be in direct proportion to the niche’s carrying capacity, as measured by the niche peak’s fitness relative to other niche peaks’ fitness values present in the domain. As the number of individuals contained within a niche indirectly indicates the amount of computational effort the GA will spend to improve the niche, the niches should be populated according to their fitness relative to the other peaks. This is known as niche proportionate population.

The difficulty of implementing population diversity mechanisms for GAs based on the niching metaphor is that the niche peak locations of a multimodal domain are not known *a priori*. Of course, if they were, we would not need a genetic algorithm to identify the local optima of the multimodal domain! As the niche boundaries are not known *a priori*, it is of course difficult to correctly populate each niche in proportion to its fitness due to the niche location uncertainty. Ideally, genetic algorithms should be able to, in parallel, both identify the niche locations and proportionally populate the niches according to their fitness. This section will explore several mechanisms that attempt to do just this.

### 2.1 Crowding

In De Jong’s crowding mechanism (De Jong, 1975), premature convergence is reduced in the GA by minimizing changes in the overall population distribution between generations. The crowding mechanism selects a proportion of the population  $G$  (generation gap) to reproduce each generation. They reproduce, and the resulting offspring then replace current members of the population in the following manner. For each offspring, a certain number  $CF$  (crowding factor) of individuals are selected at random from the population, and the most similar individual, according to a similarity metric, is then replaced by the offspring. The similarity metric can either be domain independent, as in the Hamming distance of genotype bit-string representations, or domain dependent, where the phenotypes are compared in a domain-specific manner. By choosing the most similar member for replacement, the crowding mechanism minimizes changes in the population distribution from one generation to the next. This slows down the overall tendency of a GA for the population to converge to a single point in subsequent generations. The drawback is that it retards the exploration of domain areas that are not near (similar) to population members in the initial distribution. Subsequent studies have shown crowding to not be very effective at identifying multiple optima (Deb & Goldberg, 1989; Mahfoud, 1992).

### 2.2 Deterministic Crowding

In Mahfoud’s paper exploring different crowding schemes (Mahfoud, 1992), a new mechanism, *deterministic crowding*, was developed to improve De Jong’s basic crowding scheme. First, it was demonstrated that similarity metrics based upon phenotypes were preferred to genotype similarity metrics for the multimodal domains examined. This is because domain-specific knowledge was available for the phenotype similarity metrics, giving them an edge over naive genotype similarity metrics. Second, De Jong’s use of a crowding factor  $CF$  to determine the replacement pool could be improved. It was demonstrated that there was a high probability that the most similar members in a population to an offspring were the offspring’s parents. The replacement scheme was thus modified from looking at a replacement pool of  $CF$  random members to only looking at the parents. This allowed the population diversity to be better maintained, as the replacement strategy could

find closer similarity matches on average. Lastly, the selection pressure in crowding was originally provided by proportionate selection methods, which select members to reproduce in proportion to their fitness. Mahfoud found that by choosing members randomly for reproduction, and then providing the selection pressure by only replacing a parent if the child had higher fitness, resulted in better performance. To determine which of the possible parents-children pairing should be used in comparing the parents to their children ((parent1-child1 and parent2-child2) OR (parent1-child2 and parent2-child1)), the total of the parent-child similarities for each of the two possible combinations was determined. The parents-children pairing that had the highest total similarity was used to determine if the child should replace the parent. Mahfoud called the scheme that utilized these three improvements over De Jong’s crowding algorithm deterministic crowding.

## 2.3 Sharing

Another mechanism for maintaining population diversity is sharing, first introduced by Holland (1975), and expanded by Goldberg and Richardson (1987). The premise behind sharing is to reduce the fitness of individuals that have highly similar members within the population. This rewards individuals that uniquely exploit areas of the domain, while discouraging redundant (highly similar) individuals in a domain. This causes population diversity pressure, which helps maintain population members at local optima.

The shared fitness of an individual  $i$  is given by  $f_{sh,i} = \frac{f_i}{m_i}$ , where  $f_i$  is the raw fitness of the individual, and  $m_i$  is the niche count, which defines the amount of overlap (sharing) of the individual  $i$  with the rest of the population. The niche count is calculated by summing a sharing function over all members of the population:  $m_i = \sum_{j=1}^N sh(d_{i,j})$ . The distance  $d_{i,j}$  represents the distance between individual  $i$  and individual  $j$  in the population, determined by a similarity metric (discussed below). If the sharing function determines that the distance is within a fixed radius  $\sigma_{sh}$ , it returns a value between 0 and 1 that increases with greater similarity (decreasing  $d_{i,j}$ ). Otherwise it returns 0. In essence, each member of the population is considered to be the center of a niche, and its shared fitness value is reduced for every other member of a population whose distance is less than a niche radius  $\sigma_{sh}$  from that individual.

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

Deb and Goldberg (1989) developed formulas for determining the appropriate value for  $\sigma_{sh}$  given the expected number of peaks in the multimodal domain  $q$  and the hyper-volume of the entire domain space. The ramifications of different values for the parameter  $\alpha$  are also explored in the paper. The parameter  $\alpha_{sh}$  is commonly set to 1, with the resulting sharing function denoted as the *triangular sharing function*.

### 2.3.1 Similarity Metrics

The similarity metric can be based on either phenotype or genotype similarity. A genotype similarity metric is often a domain independent metric, such as the Hamming distance between the genotype bit-string representations. A similarity metric for phenotypes can utilize domain knowledge to determine a more meaningful distance measurement. In either case, the similarity metric must return a distance  $d_{i,j}$  that decreases as similarity increases.

### 2.3.2 Sharing with Tournament Selection

Although tournament selection has been commonly used in GAs, it does not work well with sharing (Oei, Goldberg, & Chang, 1991), due to shared fitness values being calculated based on the last generation’s population, and not the current population being created. This results in chaotic fluctuations in the number of individuals populating each niche from one generation to the next. This problem can be remedied by calculating the shared fitness value of each tournament competitor based upon the current contents of the new population being generated. This approach is used throughout this study for experiments that utilize both tournament selection and sharing functions.

### 2.3.3 Sharing Issues

There are several difficulties with the sharing approach outlined above. First, it may be difficult to accurately estimate the number of expected peaks  $q$  in the domain. Second, determining  $\sigma_{sh}$  is often complicated by irregular peaks, where the peaks can vary in height, width, and shape. Values for  $q$  and  $\sigma_{sh}$  are often conservatively estimated, respectively, as the maximum number of peaks that could be in the domain, and the minimum niche radius of any optima within the domain. Lastly, calculating the shared fitness of the individuals incurs an additional complexity overhead of the order  $O(n^2)$  similarity metric calls per generation, where  $n$  is the population size. The additional complexity is a result of comparing each individual against all other members of the population to determine the niche count  $m$ , where the niche count is used to determine the shared fitness value. For very large populations, or for domains with expensive similarity metrics, this additional overhead can be prohibitive. However, in many domains, the overhead of obtaining raw fitness scores dominates the other computational costs, and thus sharing can be implemented with only a small increase in the overall computational requirements.

### 2.3.4 Dynamic Niching

To reduce the added computational expense of sharing, a new approach was developed. It utilizes two assumptions that are also present in sharing: first, it assumes that the number of niche peaks,  $q$ , can be estimated, and second, that the peaks are all a minimum distance  $2\sigma_{sh}$  from each other. The key observation is that for GAs using sharing, individuals increasingly populate the niches as time passes. Dynamic niching attempts to identify the  $q$  peaks of these forming niches, and uses these dynamically identified peaks to classify all individuals as either belonging to one of these dynamic niches (within  $\sigma_{sh}$  of a dynamic peak), or else belonging to the “non-peak” category.

In the dynamic niche sharing scheme, the shared fitness values for an individual within a dynamic niche is its raw fitness value divided by the dynamic niche population size. Otherwise, the individual belongs to the non-peak category, and its niche count  $m_i$  is calculated using the standard niche count equation given above for standard sharing. Thus, the dynamic niche sharing method calculates the shared fitness value  $f_{dsh,i}$  by  $f_{dsh,i} = f_i/m_{dsh,i}$ , where the dynamic niche count  $m_{dsh,i}$  is calculated by the following:

$$m_{dsh,i} = \begin{cases} n_j & \text{if ind. } i \text{ is within dyn. niche } j, \\ m_i & \text{otherwise (non-peak individual).} \end{cases}$$

The variable  $n_j$  is the niche population size of the  $j$ th dynamic niche, and  $m_i$  represents the standard niche count defined previously. An individual  $i$  is considered to be within a dynamic niche  $j$  if its distance  $d_{i,j}$  from peak  $j$  in the dynamic peak set is less than  $\sigma_{sh}$ .

A greedy approach is used to identify the dynamic peaks for each generation, as outlined in table 1. The peaks are “dynamic” because they are recalculated every generation, and thus can change from generation to generation. Note that more sophisticated methods, such as clustering (Yin & Gerny, 1993) or statistical analysis, could be used to identify dynamic peaks that more closely approximate the true niche peaks than the greedy approach outlined in table 1. However, the greedy approach proved to be accurate, computationally efficient, and easy to implement, and was thus used for this study.

Dynamic niche sharing is more efficient than standard sharing techniques. The computational requirements per generation for dynamic niching consists of identifying the dynamic peak set, plus the  $O(nq)$  similarity calls to determine dynamic niche membership for each individual in the population. The computational overhead of identifying the dynamic peak set is the time to sort the population by the raw fitness values, plus the  $nq$  similarity calls to determine dynamic niche membership. As the  $nq$  similarity calls dominate the relatively inexpensive sorting time, the computational complexity of the dynamic peak identification process can be characterized as  $O(nq)$  similarity calls per generation. Thus when all members of a population belong to dynamic niches (no individuals are categorized in the “non-peak” category), the total overhead for dynamic niche sharing is  $O(nq)$ . However, initially the population will probably have a majority of the members in the non-peak category, and for this case, dynamic niching is reduced to standard sharing with its overhead of  $O(n^2)$  similarity metric calls. As time progresses, the selection pressure forces the population to increasingly populate the dynamic niches. Thus a GA employing dynamic niching will have an overhead of  $O(n^2)$  similarity metric calls per generation initially, and gradually improve to  $O(nq)$  similarity metric calls in later generations. This complexity improvement in later generations implies that genetic algorithms employing dynamic niche sharing

```

input:  Pop   - array of population members
        N     - population size
        q     - number of peaks to identify
         $\sigma_{sh}$  - niche radius.

Sort Pop in decreasing raw fitness order
i = 1
NumPeaks = 0
DPS =  $\phi$  ; (Dynamic Peak Set)
loop until NumPeaks = q or i=N+1
    if Pop[i] is not within  $\sigma_{sh}$  of peak in DPS
        insert Pop[i] into DPS
        NumPeaks = NumPeaks + 1
    endif
    i = i + 1
endloop

output: Dynamic Peak Set

```

Table 1: Greedy Dynamic Peak Identification.

will have a run-time savings over selection processes utilizing the standard sharing mechanism. Furthermore, greater run-time savings can be expected with greater selection pressure, as greater selection pressures causes faster population convergence towards the dynamic peaks. The experiments presented later in this paper demonstrate the run-time savings of dynamic niching over standard sharing.

## 2.4 Niche Population Diversity

Both standard sharing and dynamic niche sharing are good at proportionately populating niches in multimodal domains. However, the two different sharing methods cause different population distributions within the niches. The standard sharing mechanism adjusts the fitness of an individual based on the relative distance (similarity) of its neighbors. Each individual in a niche will have a slightly different set of neighbors, which results in a different niche count for each member. This causes the final (steady-state) distribution of individuals to be dispersed through the niche, as each individual has pressure to maximize distance (minimize niche count) between itself and its neighbors. This diversity pressure within the niches retards the exploration of the niche peak areas, as fewer individuals are able to populate and explore the peak areas. This can adversely affect the quality of the peaks that are identified by a GA employing standard sharing techniques.

Dynamic niche sharing takes a different approach, by adjusting fitness based upon the dynamic niche population size. Every individual within a dynamic niche has their raw fitness adjusted by the same scaling factor (dynamic niche population size) when calculating their shared fitness values. Thus there is no incentive to maintain distance between the individuals within a dynamic niche. The final (steady-state) distribution consists of every member of the niche population converged at the dynamic niche peak. This allows dynamic niche sharing to more thoroughly explore the peak areas of the niches than standard sharing. The experiments presented later in this paper will demonstrate that dynamic niching is more accurate than standard sharing at identifying the peaks of the niches, which can be attributed to the capability of dynamic niching to more thorough explore the peak areas.

## 3 Mating Restriction Schemes

In multimodal domains, crossover between two individuals from different niches often produces low performance offspring (lethals). To remedy this, different mating restriction schemes have been developed to restrict mating between individuals from different niches. In a normal mating restriction scheme, the first parent is chosen randomly, and then a mating restriction mechanism is used to select a suitable mate for the first parent.

This study adapts mating restriction schemes for multimodal domains in two ways. First, to adapt mating restriction techniques to multimodal domains, this study selects the first parent using binary tournament

selection based upon either the standard sharing or dynamic niche sharing fitness values. The sharing functions provides the diversity pressure that prevents the population from converging to a single point. The second adaptation involves limiting the size of the mating pool considered for the second parent. Standard techniques can use the entire population as the mating pool for the second parent. This can be computationally expensive, as it has an overhead of  $O(n^2)$  comparisons per generation. This study instead uses a mating pool whose size is defined by a mating factor,  $MF$ . A mating pool is created for each first parent by randomly selecting without replacement  $MF$  members from the population. The second parent is then selected from the mating pool according to the mating restriction criteria. This reduces the overhead to  $O(n * MF)$  comparisons per generations, where  $MF \ll n$ . The mating factor mechanism is essentially a sampling technique, where the effective sampling rate is  $MF/PopSize$ .

The sections below describe different mating restriction methods used to determine a suitable mate for a given first parent, where the first parent was selected using techniques that help maintain population diversity.

### 3.1 Mating Radius

A simple mechanism for mating restriction was employed by Deb and Goldberg (1989), in which mating is restricted to parents within a mating radius  $\sigma_{mating}$  of each other. If a second parent can not be found within the given  $\sigma_{mating}$  radius of the first parent, the second parent is chosen randomly. One potential problem with this method is that individuals on the edge of a niche have a 50% probability of being mated with individuals that are not within their true niche. This can increase the generation of poorly performing individuals (lethals), thereby decreasing genetic algorithm performance.

### 3.2 Line breeding

A mating restriction practice common in livestock breeding is *line breeding*, in which a champion individual is repeatedly bred with others. Line breeding was used in genetic algorithms by Hollstien (1971), who noted that it performed well on unimodal domains, but not well on multimodal domains due to premature convergence.

This study explores a variation of line-breeding, called *dynamic line-breeding*, that takes advantage of the ability to calculate dynamic peaks. The first parent is selected through tournament selection based on shared fitness values. The second parent is simply the closest (most similar) dynamic peak. In this way, each dynamic peak acts as a “stud” that continually mates with other individuals within or close to its defined dynamic niche.

### 3.3 Inbreeding

To overcome the problem of line breeding converging to a single optimum in a multimodal domain, Hollstien (1971) proposed *inbreeding with intermittent cross-breeding*. This approach causes close members to repeatedly mate (intra-niche mating) as long as average niche fitness continues to rise. If it did not continue to rise, inter-niche mating was tried. This scheme showed improvement over standard line breeding for a multimodal domain. This approach will be referred to as niche inbreeding in this paper.

This study explores a variation of inbreeding, called *dynamic inbreeding*, in which the first parent is selected through tournament selection based upon shared fitness values. The second parent is then determined by examining  $MF$  members selected randomly without replacement from the population, and selecting the highest fit individual that belongs to the same niche as the first parent. If there are no individuals belonging to the same dynamic niche, the closest member (most similar) of the mating pool is selected as the second parent to decrease the possibility of lethal offspring. Note that the size of the mating factor  $MF$  indirectly controls the intermittent cross-breeding rate. The smaller the  $MF$ , the larger the chance that the mating pool will not contain a member from the same niche as the first parent, and thus the effective intermittent cross-breeding rate is higher.

## 4 Experimental Design

### 4.1 Test Function

The experiments in this paper were performed using the multimodal domain, F2, developed in Deb and Goldberg (1989) and also later used by Mahfoud (1992). The F2 domain has several characteristics which make it ideal for testing the ability of a GA to identify optima in a multimodal domain. First, the location of each optimum in the domain is known. This made it easy to compare the ending population distributions with the ideal population distributions, which was necessary to assess if the GAs were proportionately populating the niches. Second, the peaks are evenly spaced throughout the domain. This simplified the choice, and calculation, of what value should be used for the niche radius  $\sigma_{sh}$ . Lastly, the peak heights are varied, which is used to test the ability of the GA to proportionately populate the niches.

However, one weakness of the F2 domain was that given a population of 100 members, there is a high probability of initially having at least one member within each niche. While this is true of many domains, there are also many domains where the optima are sparsely located throughout the domain. In sparse domains, dynamic niche sharing would be expected to not help as much, as initially not as many members would be contained within dynamic niches. This would prolong the phase where the dynamic sharing function performs essentially like the standard sharing function, and thus reduce the computational advantage of dynamic niche sharing over standard sharing functions.

F2 is a periodic function, having five equally spaced peaks of varying height in the interval [0,1]. The peaks, in descending order by peak height, are at approximately  $x$  equals 0.1, 0.3, 0.5, 0.7, and 0.9. The F2 definition and plot are given below:

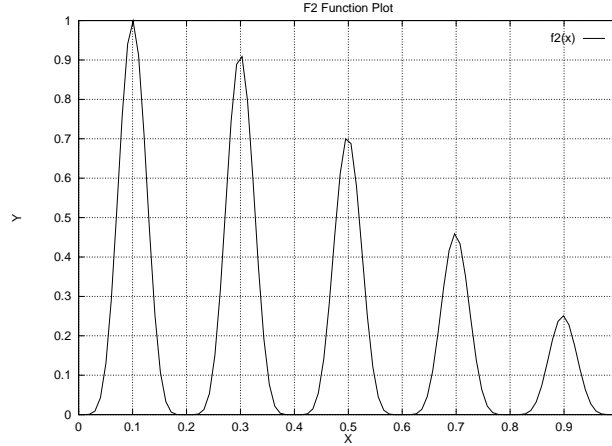


Figure 1:  $F2(x) = e^{-2(\ln(2))(\frac{x-0.1}{.8})^2} \sin^6(5\pi x)$ .

### 4.2 Performance Criteria

This study is primarily interested in the ability of a GA to identify local as well as global optimum. The maximum peak ratio statistic, described below, is used to indicate both the quality and number of the optima identified by the different GA variations. Also of interest is how well the different GA variations proportionately populate the niches, which is measured by the chi-square-like performance statistic, and is also described below. Lastly, the efficiency of the different mechanisms is also critical, so the mean average run time per trial for the different GA variations is also determined.

#### 4.2.1 Maximum Peak Ratio

The maximum peak ratio is simply the sum of the fitness of the local optima in the ending GA population divided by the sum of the fitness of the actual optima in the domain. To determine which members of the ending GA population constitute the local optima set, each member is compared against the real optima. A member is considered to be a local optimum corresponding to a real optimum if: 1) it is within a niche radius



of the real optimum, 2) it has a fitness value of at least 80% of the real optimum, and 3) it has the maximum fitness value of all members of the population that satisfy the first two criteria. There may be real optima for which no corresponding local optima may be found, in which case the local optima fitness value is taken to be zero (assuming positive fitness values only). The maximum value for the maximum peak ratio is thus 1.0, corresponding to the case where the GA has exactly identified every optimum in the domain. The maximum peak ratio therefore reflects both the quality and quantity of the optima in the ending GA population.

#### 4.2.2 Chi-square-like Performance Criteria

The chi-square-like performance statistic is used to determine how far the final population distribution differs from the ideal F2 proportionally populated distribution. This measures the ability of the GA to proportionately populate the niches in the multimodal F2 domain. The chi-square-like statistic (Deb & Goldberg, 1989), given below, returns a positive number that decreases as the final GA population distribution gets closer to the ideal F2 distribution; it returns 0 if the two distributions are identical. The chi-square-like performance metrics for both the starting and ending GA populations are calculated so that the change in population distribution due to the GA population diversity mechanism can be observed. The ideal F2 niche proportionally populated distribution is derived in Deb and Goldberg (1989), and is given below in table 2.

$$\text{Chi-square-like performance metric} = \sqrt{\sum_{i=1}^{q+1} \left( \frac{X_i - \mu_i}{\sigma_i} \right)^2}$$

The chi-square-like performance criterion measures the deviation of the actual distributions of individuals  $X_i$  from the ideal distribution mean  $\mu_i$  in all the  $i$  subspaces ( $q$  peak subspaces plus the non-peak subspace). The variable  $X_i$  represents the actual number of individuals in subspace  $i$ ,  $\mu_i$  represents the ideal number of individuals in subspace  $i$ , and  $\sigma_i$  represents the standard deviation of the number of individuals in the ideal distribution. The smaller the measured chi-square-like metric for a given population, the more ideal the population distribution.

Table 2 gives the ideal distribution for the  $F2$  function and a population of size 100. This distribution is considered ideal, as each niche population is proportionately populated according to the niche peak fitness. The expected standard niche count  $\mu_i$  and standard deviations  $\sigma_i$  are calculated as in Deb and Goldberg (1989). The ideal distribution is such that each niche peak's fitness value divided by its niche count is a constant:  $\frac{f_i}{m_i} = c$  for  $i = 1$  to  $q$  peaks. For this study, an individual is considered to be contained within a niche if it is within a niche radius  $\sigma_{sh} = 0.1$  of the ideal niche peak given in table 2, and its fitness is within 80% of the ideal niche peak fitness (corresponding to an  $\epsilon$  value of 0.8 in Deb and Goldberg (1989)).

subspace i	x	$F2(x)$	$\mu_i$	$\sigma_i^2$
1st peak	0.10	1.00	30.01	21.00
2nd peak	0.30	0.92	27.52	19.95
3rd peak	0.50	0.71	21.22	16.72
4th peak	0.70	0.46	13.86	11.86
5th peak	0.90	0.25	7.51	6.94
non-peak	NA	NA	0.00	76.47

Table 2: Ideal Proportionate Population Distribution of 100 Individuals for  $F2$  Function.

## 5 Experimental Results

This section describes experiments based upon different GA variations, and presents the results of each experiment. Common GA configuration information is given in the first section. The experiments are divided into two sets. The first set of experiments explores basic GA mechanisms for identifying multiple optima within a multimodal domain. The second set of experiments investigate the effectiveness of different mating restriction schemes for identifying multiple optima within a multimodal domain.

## 5.1 Genetic Algorithm Configuration

All GA variations use the same encoding for the  $F2$  function. Each individual’s phenotype is the variable  $x$  between 0 and 1, with the genotype being coded as a 30 bit number using standard binary encoding. The genotype is converted to a phenotype by dividing by  $2^{30} - 1$ . The raw fitness for an individual is  $F2(x)$ , where  $x$  is the phenotype of the individual. The similarity metric is based on the phenotype, and is simply the absolute value of the difference between the phenotypes of the individuals. The use of a genotype similarity metric was explored by Deb and Goldberg (1989), and was shown to be less efficient than the phenotype similarity metric for the  $F2$  domain.

The GA parameters that are standard across experiments are given below. The mutation rate was set to zero in order to better isolate the effects of the different population diversity mechanisms on GA performance.

Generations	100
Population Size	100
2-pt Crossover, with Crossover Rate	1.0
Mutation Rate	0.0
Generation Gap	None
Tournament Size	2

The experiments primarily differ on the selection mechanisms and their associated sharing functions. For those experiments that use tournament selection based upon shared fitness values, the share (or niche) radius  $\sigma_{sh}$  is 0.1, as calculated by Deb and Goldberg (1989) for phenotype sharing in the  $F2$  domain. The standard sharing function is the triangular sharing function ( $\alpha_{sh} = 1$ ), which was described earlier. The shared fitness values are calculated relative to the new population currently being generated, as suggested by Oei, Goldberg, and Chang (1991), which stabilizes the selection pressure for genetic algorithms utilizing tournament selection.

The experimental results were obtained by generating ten random starting populations, and running each experiment once with each of the ten random starting populations. The experiment results were then averaged over the ten trials. Both sets of experiments used the same ten random starting populations. The mean and standard deviation of the chi-square-like statistic over the ten initial random starting populations was 13.71 and 0.44 respectively.

## 5.2 Niche Formation Experiments

This section explores the results of four different GA variations that did not employ mating restriction schemes. The objective was to identify each of the five optima within the  $F2$  domain. The first variation is a standard GA, in which the selection method was tournament selection based on the raw fitness values of the competitors. This experiment is included as a baseline, as it shows the performance of a GA that is not modified for a multimodal environment. The second GA variation employs Mahfoud’s deterministic crowding scheme, in which children replace their parents only if the children have a higher raw fitness. The third variation uses tournament selection based upon shared fitness values, which are calculated using the triangular sharing function. The fourth variation uses tournament selection based upon the shared fitness values, which are calculated using the dynamic niching sharing function. Table 3 gives the experimental results for the four different GA variations.

Table 3 summarizes the experimental results for the four different GA variations, averaged over ten trials. The maximum peaks ratio indicates the quality and quantity of the identified optima in the ending populations of the different GA variations. The higher the maximum peaks ratio, the better the performance. The mean average and standard deviation of the chi-square-like metrics for the initial and ending populations are also given. The smaller the ending chi-square-like metric, the closer the ending population distribution is to the ideal distribution for the  $F2$  domain. The last two columns give the mean average and standard deviation of the running time, in CPU seconds, used by each GA variation.

The standard GA, as expected, had most of the population converge to the optimal peak. This caused a low max peaks ratio score, as all optima besides the global optimum were unlikely to be populated. The standard GA variation also had a high average ending chi-square-like score, reflecting that it showed substantial convergence to the global peak, and thus was not very similar to the ideal distribution.

The second variation, deterministic crowding, did not fare well. It had the ending population distribution evenly spread out across the domain, instead of proportionately populating the niches. The  $F2$  domain is

	Max Peaks		End Chi-Sq		CPU Time	
	Avg	Std	Avg	Std	Avg	Std
Standard GA	0.40	0.13	15.00	0.57	27.95	0.40
Deterministic Crowding	0.96	0.09	11.93	0.68	34.99	0.42
Standard Sharing	0.98	0.03	6.05	0.90	70.76	1.35
Dynamic Niche Sharing	1.00	0.00	5.09	0.62	32.02	0.57

Table 3: Population Statistics for Niche Formation Experiments.

characterized by niches of equal width, which results in the initial population be evenly distributed across the niches. Deterministic crowding uses a parent replacement strategy, which minimizes changes in the population distribution from generation to generation. These two factors cause deterministic crowding to maintain fairly even niche population sizes, resulting in lower chi-square-like results. Deterministic crowding can successfully identify niche peaks, but does so slowly, as exploration of the domain is sacrificed so as to minimize population distribution changes. This explains the lower max peaks ratio score compared to the experiments using sharing functions.

The third variation, using the standard sharing technique, performed well. It has a much smaller ending chi-square-like average than the first two, reflecting that it has a close to ideal ending distribution. Its main drawback is the computational expense of the sharing functions - it averaged 69 seconds per run, as opposed to the standard GA averaging 28 seconds. This additional computational expense is due to the overhead of calculating the standard shared fitness values each generation, which adds  $O(n^2)$  candidate comparisons over traditional GA selection methods based on raw fitness values. This disparity can obviously be expected to get worse as the population size grows.

The fourth variation, utilizing the dynamic niche sharing strategy, performed the best of the four variations. It had the highest max peak ratio, the lowest ending chi-square-like average, and a respectable average running time of 35 CPU seconds. Dynamic Sharing had a better max peaks ratio and ending chi-square-like statistics compared with standard sharing. This is a result of the niche populations being able to converge towards the peaks of the niches when using dynamic niche sharing. This allows for a more thorough search of the peak areas for better peak members. It also explains the better ending chi-square-like scores, as more members are able to fit within 80% of the peak values than with standard sharing. The improved efficiency over standard sharing technique, as measured by CPU time, is due to the fact that most individuals throughout the run were located within a niche radius of a dynamic peak. This allowed for the constant-time calculation of each niche count used for computing the shared fitness value, as opposed to the  $O(n)$  candidate comparisons required by standard sharing to calculate each niche count.

This first set of experiments demonstrates that dynamic niche sharing is an efficient mechanism for identifying local optima in a multimodal domain. Standard sharing also does well, but is slower and less accurate in identifying optima than dynamic niche sharing.

### 5.3 Mating Restriction Experiments

This set of experiments explored different mating restriction schemes to further improve performance in multimodal domains. In the mating restriction schemes examined, the first parent is selected through tournament selection, based upon fitness scores calculated using either standard sharing or dynamic niche sharing. These sharing functions provide population diversity pressure in the GAs, thus helping to prevent convergence of the population. The second parent is then chosen using a mating restriction scheme, which selects an appropriate individual to be the mate of the first parent. The second parent is selected to be a suitable mate for the first parent, which reduces the chance of lethal offspring of the parents. This results in an increase in the average fitness of the offspring. This set of experiments explored combinations of different sharing functions with different mating restrictions.

The first experiment selected the first parent using tournament selection based upon standard sharing fitness values, and the second parent was selected using niche inbreeding. Niche inbreeding examines  $MF$  random members without replacement, and selects the most highly fit member that belongs to the same niche as the first parent. If none belong to the first parent's niche, it selects the member that is most similar to the

first parent. A member was considered to be within the first parent’s niche if its phenotype was within a niche radius  $\sigma_{mating} = 0.1$  of the first parent’s phenotype. The mating factor  $MF$  was arbitrarily set to  $2q$ , where  $q = 5$  for the F2 domain. This configuration is labeled Standard Sharing with Niche Inbreeding in Table 4.

The second experiment was similar to the first experiment, but the first parent is selected using fitness values calculated using the dynamic niche sharing function. The second parent was selected through niche inbreeding, as described above. This configuration is labeled Dynamic Sharing with Niche Inbreeding in Table 4.

The third experiment uses tournament selection based upon shared fitness values calculated through dynamic niche sharing, and uses dynamic niche inbreeding to select the second parent. Dynamic niche inbreeding is identical to niche inbreeding, except that it determines niche membership differently. Two individuals are considered to be in the same niche if they are both within niche radius  $\sigma_{sh} = 0.1$  of the same dynamic peak, which denotes that they are both members of the same dynamic niche. The mating factor  $MF$  was again arbitrarily set to  $2q$ , where  $q = 5$  for the F2 domain. This configuration is labeled Dynamic Sharing with Dynamic Inbreeding in table 4.

The fourth and fifth experiments are identical line-breeding configurations, except that the fourth experiment uses the standard sharing function to select the first parent, while the fifth experiment uses the dynamic sharing function. For both experiments, the first parent is selected using tournament selection based upon shared fitness values calculated by their respective sharing functions. The first parent is then line-bred to the most similar niche champion. The most similar niche champion is determined to be the dynamic peak that has the closest phenotype to the first parent’s phenotype for both experiments.

The mating restriction experiments demonstrate that the use of mating restrictions can further improve the performance of a genetic algorithm within multimodal domains. The results of the mating restriction experiments are given below in table 4. The table is in the same format as table 3, described earlier. All of the experiments performed well at identifying the local optima, and proportionately populating the niches. The first three inbreeding experiments demonstrate that dynamic niche sharing is more effective than standard sharing when used in conjunction with inbreeding mating restrictions. Furthermore, dynamic niche inbreeding is shown to be effective at both determining the local optima, and proportionately populating the niches. The difference in the max peaks ratio between the first two experiments is a result of dynamic niche sharing allowing the niche populations to converge to the peaks of the local optima, whereas standard sharing distributes the niche population throughout the niche. This minimizes the possibility that a mate from another niche is within a niche radius, and thus less lethals are generated when using dynamic niche sharing. Furthermore, dynamic niche sharing concentrates the GA exploration effort at the peaks of the niches, allowing better local optima able to be identified. The small advantage in CPU time between the second and third experiments could be negated in future experiments by augmenting the Dynamic Peak Identification routine to label each member of the population with their respective dynamic niche membership as it calculates the dynamic niche peaks. This would eliminate the  $q$  comparisons currently needed to determine niche membership in the dynamic niche inbreeding scheme.

	Max Peaks		End Chi-Sq		CPU Time	
	Avg	Std	Avg	Std	Avg	Std
Std Share w/Niche Inbreeding	0.81	0.06	5.35	0.49	68.42	1.97
Dyn Share w/Niche Inbreeding	1.00	0.00	4.34	0.67	46.62	0.69
Dyn Share w/Dyn Inbreeding	1.00	0.00	4.38	0.60	52.29	0.38
Std Share w/Dyn Line-Breed	0.99	0.02	2.85	0.68	46.87	0.32
Dyn Share w/Dyn Line-Breed	0.99	0.02	2.81	0.44	28.39	0.25

Table 4: Population Statistics for Mating Restriction Experiments.

For the F2 domain, the line-breeding mating restrictions also worked very well. This is a stronger bias than inbreeding, as it rapidly promotes the convergence to the dynamic peaks by repeatedly mating the dynamic peaks. This is only appropriate in domains where the initial dynamic peaks can be considered to be suitable approximations to the local optima. Otherwise, line-breeding will quickly propagate the genes of misleading dynamic peaks, which degrades the ability of the GA to successfully identify the local optima, as well as degrading the proportionate population of the niches. Line-breeding is well suited for the F2 domain, as the

actual peaks are likely to be close to the initial dynamic peaks. This explains the high performance obtained, as indicated by the max peaks ratio and the ending chi-square-like statistics, for the line-breeding variations. Note that the dynamic line-breeding techniques are computationally more efficient than inbreeding techniques, as the shared fitness values are only calculated in determining the first parent.

## 6 Conclusions and General Discussion

In this paper we have studied GA mechanisms for identification of multiple optima within multimodal domains. Two groups of mechanisms are discussed and evaluated: the first group focuses on niche formation, including deterministic crowding, sharing, and dynamic niching; the second group focuses on mating restriction, including niche inbreeding, dynamic inbreeding, and dynamic line-breeding.

Through analysis and experimental testing with a multimodal function, we have demonstrated the efficacy of dynamic niching as a useful GA mechanism for identifying multiple optima (i.e., peaks), as measured by the maximum peak ratio, the chi-square-like test, and computational time needed. This can be explained by the ability of the dynamic niching mechanism to populate niches proportionately, thus maintaining population diversity. Moreover, dynamic niching can help focus more on the peaks of optima, vis-à-vis the standard sharing mechanism which tends to result in groups of individuals dispersed throughout the niches.

When mating restrictions are added to the GA procedure, the combination of dynamic sharing and dynamic line-breeding performed best in our multimodal experiments, again in terms of the maximum peak ratio, chi-square-like test, and the computational time needed. This can be explained by the ability of the line-breeding mechanism to quickly converge to the dynamic peaks. As demonstrated in our experiments using the  $F2$  test function, the dynamic peaks identified are likely to be close to the actual peaks. It should be noted that the identification of dynamic peaks for domains characterized by peaks sparsely located throughout the domain will be more difficult, and will be the study of future research.

This study focuses on understanding the characteristics, behaviors, and comparative performances of the niche formation and mating restriction mechanisms. To meet that objective, the experiments conducted for evaluation are based on an illustrative test function, the  $F2$  domain. The experimental results, the comparative performances, and the analysis we were able to do based on the findings provide valuable insights for us to understand the characteristics of niching mechanisms, and their effects on the performance of GAs, especially in identification of multiple optima in multimodal domains.

## 7 Acknowledgments

We would like to thank Jeff Horn, Sam Mahfoud, and Troy Strader for their valuable comments on early drafts of this paper. We would also like to thank David Goldberg for the insightful discussions that helped inspire this research. Finally, the support of both the Research Board of the University of Illinois and NASA (grant NG-9-4) is gratefully acknowledged.

## References

- De Jong, K. A. (1975). An analysis of the behavior of a class of genetic adaptive systems. *Dissertation Abstracts International*, 36(10), 5140B. (University Microfilms No. 76-9381).
- Deb, K., & Goldberg, D. E. (1989). An investigation of niche and species formation in genetic function optimization. In Schaffer, J. D. (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 42–50). San Mateo, CA: Morgan Kaufmann.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. New York: Addison-Wesley.
- Goldberg, D. E., & Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In Grefenstette, J. J. (Ed.), *Proceedings of the Second International Conference on Genetic Algorithms* (pp. 41–49). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press.

- Hollstien, R. B. (1971). Artificial genetic adaptation in computer control systems. *Dissertation Abstracts International*, 32(3), 1510B. (University Microfilms No. 71-23,773).
- Horn, J., Goldberg, D. E., & Deb, K. (1994). Implicit niching in a learning classifier system: nature's way. *Evolutionary Computation*, 2(1), 37-66.
- Mahfoud, S. W. (1992). Crowding and preselection revisited. In Männer, R., & Manderick, B. (Eds.), *Parallel Problem Solving from Nature*, 2 (pp. 27-36). Amsterdam, The Netherlands: Elsevier Science.
- Oei, C. K., Goldberg, D. E., & Chang, S. J. (1991). *Tournament selection, niching, and the preservation of diversity* (IlliGAL Report No. 91011). Urbana: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.
- Sikora, R., & Shaw, M. J. (1994). A double-layered learning approach to acquiring rules for classification: Integrating genetic algorithms with similarity-based learning. *ORSA Journal on Computing*, 6(2), 174-187.
- Yin, X., & Gernay, N. (1993). A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. In Albrecht, R. F., Reeves, C. R., & Steele, N. C. (Eds.), *Proceedings of the International Conference in Austria on Artificial Neural Nets and Genetic Algorithms* (pp. 450-457). New York: Springer-Verlag.