Genetic Algorithms with Dynamic Niche Sharing
for Multimodal Function Optimization

Brad L. Miller
Dept. of Computer Science
bmiller@uiuc.edu

Michael J. Shaw
Beckman Institute
mshaw@ux1.cso.uiuc.edu

University of Illinois at Urbana-Champaign

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.**

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 multiple optima in multimodal domains.

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! This makes it 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 pop-

ulate 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 (e.g., the Hamming distance between genotypes), 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

Deterministic Crowding (Mahfoud, 1992), was developed to improve De Jong's basic crowding scheme. For replacement, instead of using De Jong's crowding factor, deterministic crowding compares the new offspring directly to their parents, and replaces the parents only if the children have higher fitness. This provides the selection pressure for the GA. The parents-children pairing of the two possible child-parent pairings that has the highest total similarity is used to determine the replacement comparisons.

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 σ_{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 σ_{sh} from that individual.

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

Deb and Goldberg (1989) developed formulas for determining the appropriate value for σ_{sh} given the expected number of peaks in the multimodal domain q and the hyper-volume of the entire domain space. The parameter α_{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 phenotype similarity metric utilizes domain knowledge to calculate a more meaningful distance measurement. In either case, the similarity metric must return a distance $d_{i,j}$ that decreases as similarity increases. Generally, phenotype similarity metrics are used if there is sufficient domain knowledge to construct them.

2.3.2 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 σ_{sh} is often complicated by irregular peaks, where the peaks can vary in height, width, and shape. Values for q and σ_{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 optimum 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 its niche count m, which is needed to calculate its 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.

2.4 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 σ_{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 jth 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 σ_{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 & Germay, 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

```
input:   Pop    - array of population members
         N      - population size
         q      - number of peaks to identify
         σ_sh   - niche radius.

Sort Pop in decreasing raw fitness order
i = 1
NumPeaks = 0
DPS = φ ; (Dynamic Peak Set)
loop until NumPeaks = q or i=N+1
  if Pop[i] is not within σ_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.

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 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.

3 Experimental Design

3.1 Test Function

The experiments in this paper were performed using the multimodal domain, F2, which was also used in the studies by Deb and Goldberg (1989) and 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 makes 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 σ_{sh}. Lastly, the peak heights are varied, which is used to test the ability of the GA to proportionately populate the niches.

$F2$ is a periodic function, having five equally spaced peaks of varying height in the interval [0,1]. The peaks, in descend-

ing 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 in figure 1.
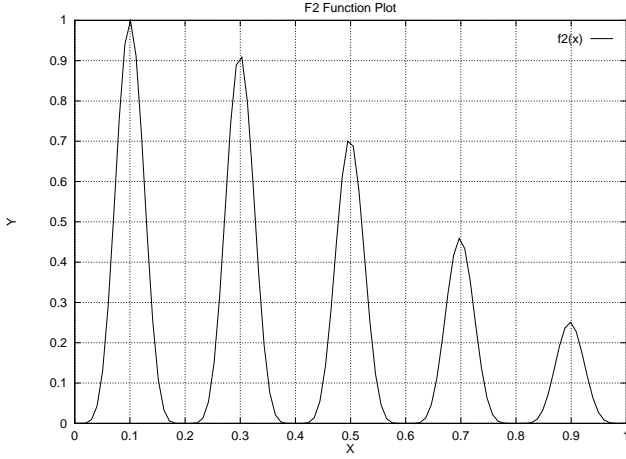

Figure 1: $F2(x) = e^{-2(\ln(2))(\frac{x-0.1}{.8})^2} \sin^6(5\pi x)$.

3.2 Performance Criteria

This study is primarily interested in the ability of a GA to identify local as well as the 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.

3.2.1 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.

3.2.2 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 proportionately 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 μ_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, μ_i represents the ideal number of individuals in subspace i, and σ_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 μ_i and standard deviations σ_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 ϵ value of 0.8 in Deb and Goldberg (1989)).

subspace i	x	$F2(x)$	μ_i	σ_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.

4 Experimental Results

This section describes experiments that evaluate different GA mechanisms for identifying multiple optima within a multimodal domain. The first subsection gives GA configuration information, and the second subsection presents the experimental results.

4.1 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 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 σ_{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.

4.2 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 optimum 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 deterministic crowding, 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 shared fitness values, which are calculated using dynamic niching.

Table 3 summarizes the experimental results for the four different GA variations, averaged over ten trials. 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 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 niche 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

	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.

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 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   Conclusions

In this paper we have studied GA mechanisms for identification of multiple optima within multimodal domains. This paper focuses on niche formation, including deterministic crowding, sharing, and dynamic niching. 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 requirements. 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-á-vis the standard sharing mechanism which tends to result in groups of individuals dispersed throughout the niches. Further testing is needed to see if the performance advantages of dynamic niching over sharing are significant. This study demonstrates that dynamic niching achieves at least comparable performance to that of sharing, at a much lower computational cost.

Further research will test dynamic niching on more complex multimodal domains, and investigate the use of mating restrictions utilizing dynamic niching to further improve GA performance. A more detailed report on dynamic niching, including the investigation of mating restrictions used in conjunction with dynamic niching, can be found in Miller and Shaw (1995). The experimental results, the comparative performances, and the analysis we were able to do in this study 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.

6   Acknowledgments

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., & 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.

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.

Miller, B. L., & Shaw, M. J. (1995). *Genetic algorithms with dynamic niche sharing for multimodal function optimization* (IlliGAL Report No. 95010). Urbana: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.

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 class ification: Integrating genetic algorithms with similarity-based learning. *ORSA Journal on Computing, 6*(2), 174–187.

Yin, X., & Germay, 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.