Fitness sharing and niching methods revisited

Bruno Sareni, Laurent Krähenbühl

HAL Id: hal-00359799
https://hal.science/hal-00359799

Submitted on 9 Feb 2009

Fitness Sharing and Niching Methods Revisited

Bruno Sareni and Laurent Krähenbühl

Abstract—**Interest in multimodal optimization function is expanding rapidly since real-world optimization problems often require the location of multiple optima in the search space. In this context, fitness sharing has been used widely to maintain population diversity and permit the investigation of many peaks in the feasible domain. This paper reviews various strategies of sharing and proposes new recombination schemes to improve its efficiency. Some empirical results are presented for high and a limited number of fitness function evaluations. Finally, the study compares the sharing method with other niching techniques.**

Index Terms— **Evolutionary computation, fitness sharing, genetic algorithms, multimodal optimization, niching methods.**

I. INTRODUCTION

TRADITIONAL genetic algorithms (GA's) with elitist selection are suitable for locating the optimum of unimodal functions as they converge to a single solution of the search space. Real optimization problems, however, often lead to multimodal domains and so require the identification of multiple optima, either global or local. For this purpose, niching methods extend simple GA's by promoting the formation of stable subpopulations in the neighborhood of optimal solutions.

Niching methods have been developed to reduce the effect of genetic drift resulting from the selection operator in the standard GA. They maintain population diversity and permit the GA to investigate many peaks in parallel. On the other hand, they prevent the GA from being trapped in local optima of the search space. Niching GA's are based on the mechanics of natural ecosystems. In nature, animals compete to survive by hunting, feeding, grazing, breeding, etc., and different species evolve to fill each role. A niche can be viewed as a subspace in the environment that can support different types of life. A species is defined as a group of individuals with similar biological features capable of interbreeding among themselves but that are unable to breed with individuals outside their group. For each niche, the physical resources are finite and must be shared among the population of that niche. By analogy, niching methods tend to achieve a natural emergence of niches and species in the environment (search space). A niche is commonly referred to as an optimum of the domain, the fitness representing the resources of that niche. Species can be defined as similar individuals in terms of similarity metrics.

The sharing method is probably the best known and also used among niching techniques. It was originally introduced by Holland [1, p. 164] and improved by Goldberg and Richardson

[2]. Section II presents the main principles of fitness sharing and reviews the recent development of this technique. Section III is devoted to other niching schemes and especially crowding methods. Section IV investigates various forms of sharing on test problems defined in Section III and compares their efficiency with the other niching GA's. Empirical results are presented for high and a limited number of fitness function evaluations.

II. FITNESS SHARING

A. Principle

Fitness sharing modifies the search landscape by reducing the payoff in densely populated regions. It lowers each population element's fitness by an amount nearly equal to the number of similar individuals in the population. Typically, the shared fitness f_i' of an individual i with fitness f_i is simply

$$f_i' = \frac{f_i}{m_i} \qquad (1)$$

where m_i is the niche count which measures the approximate number of individuals with whom the fitness f_i is shared. The niche count is calculated by summing a sharing function over all members of the population

$$m_i = \sum_{j=1}^{N} sh(d_{ij}) \qquad (2)$$

where N denotes the population size and d_{ij} represents the distance between the individual i and the individual j. Thence, the sharing function (sh) measures the similarity level between two population elements. It returns one if the elements are identical, zero if their distance d_{ij} is higher than a threshold of dissimilarity, and an intermediate value at intermediate level of dissimilarity. The most widely used sharing function is given as follows:

$$sh(d_{ij}) = \begin{cases} 1 - (d_{ij}/\sigma_s)^{\alpha}, & \text{if } d < \sigma_s \\ 0, & \text{otherwise} \end{cases} \qquad (3)$$

where σ_s denotes the threshold of dissimilarity (also the distance cutoff or the niche radius) and α is a constant parameter which regulates the shape of the sharing function. α is commonly set to one with the resulting sharing function referred to as the triangular sharing function [3].

The distance d_{ij} between two individuals i and j is characterized by a similarity metric based on either genotypic or phenotypic similarity. Genotypic similarity is related to bitstring representation and is generally the Hamming distance. Phenotypic similarity is directly linked to real parameters of the search space. It can be the Euclidian distance for instance.

Sharing based on phenotypic similarity may give slightly better results than sharing with genotypic similarity [4].

Sharing must be implemented with the less biased selection methods. Stochastic remainder selection (SRS) and stochastic universal selection (SUS) have been widely used to reduce bias in the selection algorithm [5]. Tournament selection (TS) with continuously updated sharing is another possibility [6]. In the same way, sharing must use low recombination operators to promote stability of subpopulations. In effect, crossovers between individuals of different niches often lead to poor individuals (lethals). Mating restriction schemes have been successfully applied to reduce the formation of lethals [4], [7], [8].

B Limitations

Sharing tends to encourage search in unexplored regions of the space and favors the formation of stable subpopulations. Nonetheless, sharing is not without limitations.

- Setting the dissimilarity threshold σ_s requires *a priori* knowledge of how far apart the optima are. For real optimization problems, however, no information about the search space and the distance between the optima is generally available. On the other hand, σ_s is the same for all individuals. This supposes that all peaks must be nearly equidistant in the domain. For these reasons, sharing can fail to maintain all desired peaks if they are not equidistant or if the estimated distance between two peaks is incorrect. Various empirical formulas have been proposed to set the dissimilarity threshold but this problem remains the major flaw of the method [4], [9].
- The sharing scheme is very expensive as a result of the computation of niche counts of complexity $O(N^2)$ per generation. Clustering analysis and dynamic niching have been developed to reduce computational complexity and increase sharing effectiveness [7], [8]. In many domains, however, the computational time to obtain the fitness of individuals dominates the computational cost of comparisons. In that case, standard sharing can be implemented with only a small increase in the computational requirements.

C. Fitness Scaling

One way to improve sharing efficiency is to use fitness scaling [3]. A scaled shared function increases differentiation between optima and reduces deception[1] [16], [19]. It makes the optima more attractive than the surrounding regions of the space. A common technique to scale the fitness function is to use a power scaling. In that case (1) can be modified as follows:

$$f_i' = \frac{f_i^\beta}{m_i}. \tag{4}$$

The remaining problem is the choice of an appropriate parameter β for a given objective function. If the power of

[1] We talk about deception when the combination of good building blocks leads to reduced fitness rather than increased fitness. Deceptive problems are generally multimodal functions with attractive local optima (see [16]).

the scaling function is too high, the predominance of fitness scaling can prevent the reduction of genetic drift by the sharing method. The domination of "super-individuals" in the population can cause the niching GA to converge prematurely. On the other hand, if the power of the scaling function is too low, differentiation between optima can be insufficient. This can hinder a perfect detection of the optima by the sharing method. The compromise in the choice of the scaling power is directly related to the accurate balancing between exploration and exploitation necessary to all global stochastic optimization methods. To prevent premature convergence and increase the efficiency of the sharing method, annealing the scaling power during the search is recommended [19].

III. FURTHER NICHING METHODS

An important variety of other niching methods have been reported in the literature including sequential niching [10], immune systems [11], speciation with implicit fitness sharing and co-evolution [20], ecological GA's [9], [12], and crowding schemes. This paper focuses on crowding techniques and explores a recent promising niching method called *clearing*.

A. Crowding Methods

Crowding methods insert new elements in the population by replacing similar elements.

- *Standard Crowding*: In DeJong's crowding [13], only a fraction of the global population specified by a percentage G (generation gap) reproduces and dies each generation. In this crowding scheme, an offspring replaces the most similar individual (in terms of genotypic comparison) taken from a randomly drawn subpopulation of size CF (crowding factor) from the global population. Because of a great number of replacement errors, the initial crowding of DeJong has been shown to be limited in multimodal function optimization [4], [9].
- *Deterministic Crowding*: Mahfoud improved standard crowding by introducing competition between children and parents of identical niches [9]. After crossover and eventually mutation, each child replaces the nearest parent if it has a higher fitness. Thus deterministic crowding (DC) results in two sets of tournaments: (parent 1 against child 1, and parent 2 against child 2) or (parent 1 against child 2, and parent 2 against child 1). The set of tournament that yields the closest competitions is held. Similarity is computed using preferably phenotypic distance. With two distance comparisons per set of tournaments and $N/2$ sets of tournaments per generation, the resulting order of complexity of deterministic crowding is $O(N)$.
- *Restricted Tournament Selection*: Restricted tournament selection (RTS) adapts standard tournament selection (TS) for multimodal optimization [14]. RTS initially selects two elements from the population to undergo crossover and mutation. After recombination, a random sample of CF individuals is taken from the population as in standard crowding. Each offspring competes with the closest sample element. The winners are inserted in the

population. This procedure is repeated $N/2$ times. The order of complexity of RTS is $O(\mathrm{CF} \cdot N)$. It can vary from $O(N)$ to $O(N^2)$ according to the crowding factor value CF.

B. Clearing

The clearing method is very similar to fitness sharing but is based on the concept of limited resources of the environment [15]. Instead of sharing the resources between all individuals of a single subpopulation as in fitness sharing, clearing attributes them only to the best members of the subpopulation. In practice, the capacity k of a niche specifies the maximum number of elements that this niche can accept. Thus, clearing preserves the fitness of the k best individuals (dominant individuals) of the niche and resets the fitness of the others that belong to the same subpopulation (dominated individuals). As in the sharing method, individuals belong to the same niche (or subpopulation) if their distance in the search space is less than a dissimilarity threshold σ_s (clearing radius). Clearing can be coupled with elitism strategies to preserve the best elements of the niches during the generations. The order of complexity of the basic clearing procedure is $O(qN)$ where q is the number of niches maintained during the search.

IV. TEST PROBLEMS

A. Test Functions

We consider three multimodal functions of different difficulty with nomenclature maintained from [9]

$$F3(x) = \sin^6\left(5\pi[x^{3/4} - 0.05]\right). \qquad (5)$$

This function defined on [0, 1] consists of five unequally spaced peaks of uniform height. Maxima are located at approximate x values of 0.080, 0.247, 0.451, 0.681, and 0.934. All peaks are of height 1.0

$$F4(x) = e^{-2(\ln 2)(\frac{x-0.08}{0.854})^2} \sin^6\left(5\pi[x^{3/4} - 0.05]\right). \qquad (6)$$

$F4$ is also defined on [0, 1] and consists of five unequally spaced peaks of nonuniform height. Maxima are located at approximate x values of 0.080, 0.247, 0.451, 0.681, and 0.934. Maxima are of approximate height 1.000, 0.948, 0.770, 0.503 and 0.250 respectively.

$F7$ is the massively multimodal deceptive function [15], [16]. $F7$ is defined by the sum of the fitness of five subfunctions

$$F7(x_0, x_1, \ldots, x_{29}) = \sum_{i=0}^{4} u\left(\sum_{j=0}^{5} x_{6i+j}\right) \qquad (7)$$

where $\forall k \in [0, 29]$, $x_k \in \{0, 1\}$. Each subfunction is a bimodal deceptive function of unitation as displayed in Fig. 3. $F7$ has 32 global optima of height 5 and several million local maxima lying between 3.203 and 4.641.
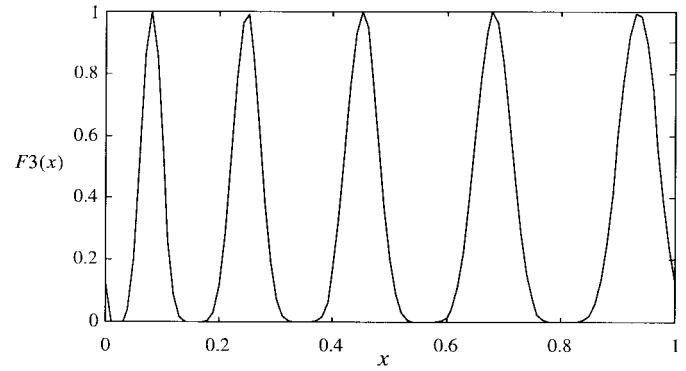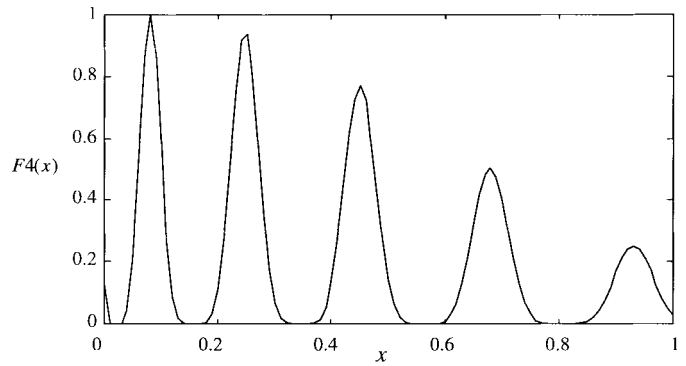

Fig. 1.   Function $F3$.
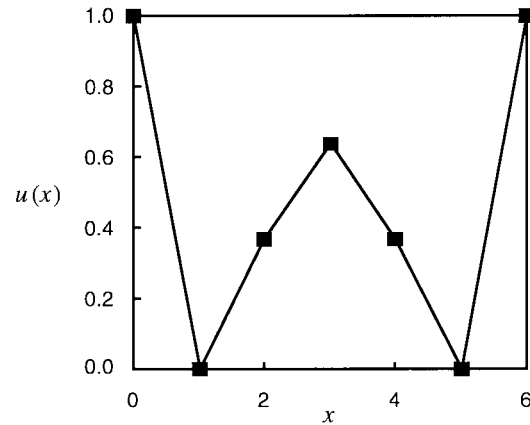

Fig. 2.   Function $F4$.

Fig. 3.   The bimodal deceptive subfunction used in $F7$.

B. Performance Criteria

Maximum Peak Ratio: The maximum peak ratio is the sum of the fitness of the local optima identified by the niching technique divided by the sum of the fitness of the actual optima in the search space [8]. An optimum is considered to be detected if it is within a niche radius of the real optimum and if its fitness value is at least 80% of the real optimum. When an optimum is not identified, the local optimum value is set to zero. Thence, the maximum value for the maximum peak ratio is one corresponding to a perfect detection of all optima.

Effective Number of Peaks Maintained: We also consider the effective number of optima maintained at the end of the search according to the previous assumptions.

TABLE I

SELECTION IN THE SHARING METHOD. THIS TABLE PRESENTS THE VALUES OF THE PERFORMANCE CRITERIA (SEE SECTION IV) ON FUNCTION $F3$ FOR EACH SELECTION SCHEME. TEST1 IS PERFORMED USING 100 INDIVIDUALS AND 200 GENERATIONS. TEST2 IS PERFORMED USING 30 INDIVIDUALS AND 30 GENERATIONS

Selection Scheme	Nb of Peaks maintained		Maximum Peaks Ratio		Mean Chi-Square		End Chi-Square	
	test1	test2	test1	test2	test1	test2	test1	test2
SRS	5	3.6	0.998	0.703	2.615	4.352	2.283	4.255
SUS	5	4.8	0.997	0.935	2.214	2.957	1.672	1.789
TS	5	3.2	0.998	0.678	4.575	5.043	5.076	5.042

TABLE II

SELECTION IN THE SHARING METHOD. THIS TABLE PRESENTS THE VALUES OF THE PERFORMANCE CRITERIA (SEE SECTION IV) ON FUNCTION $F4$ FOR EACH SELECTION SCHEME. TEST1 IS PERFORMED USING 100 INDIVIDUALS AND 200 GENERATIONS. TEST2 IS PERFORMED USING 30 INDIVIDUALS AND 30 GENERATIONS

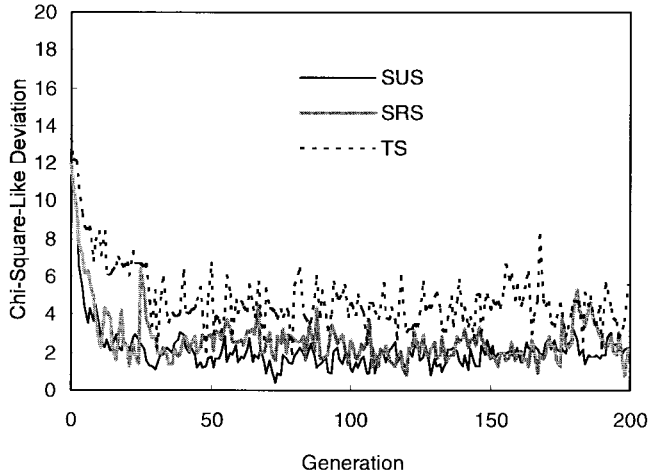| Selection Scheme	Nb of Peaks maintained		Maximum Peaks Ratio		Mean Chi-Square		End Chi-Square	
	test1	test2	test1	test2	test1	test2	test1	test2
SRS	5	3.2	0.996	0.628	2.552	4.957	2.514	4.617
SUS	5	4.8	0.999	0.941	2.395	3.461	2.109	2.281
TS	5	3.2	0.987	0.678	5.145	5.114	6.220	4.627

Fig. 4. Chi-square-like deviation of the selection schemes investigated on function $F3$. The population size is $N = 100$.
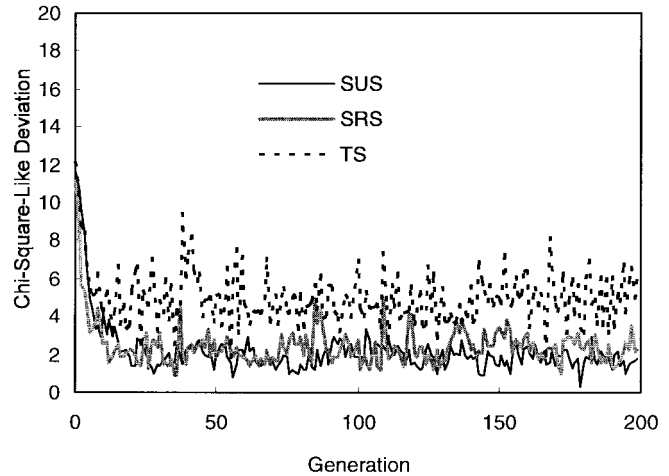

Fig. 5. Chi-square-like deviation of the selection schemes investigated on function $F4$. The population size is $N = 100$.

Chi-Square-Like Performance Criteria: The "chi-square-like" performance statistic measures the deviation between the population distribution and an ideal proportionally populated distribution [4], [7], [8]. This criterion is computed using the actual distribution of individuals X_i and an ideal distribution mean μ_i in all the i niches (q peak niches plus the nonpeak niche)

$$\text{chi-square-like deviation} = \sqrt{\sum_{i=1}^{q+1} \left(\frac{X_i - \mu_i}{\sigma_i^2} \right)^2} \qquad (8)$$

where

$$\mu_i = N \frac{f_i}{\sum_{k=1}^{q} f_k} \quad \text{and} \quad \sigma_i = \mu_i (1 - \mu_i/N) \qquad (9)$$

for the i peak niches and

$$\mu_{q+1} = 0 \quad \text{and} \quad \sigma_{q+1} = \sum_{i=1}^{q} \sigma_i^2 \qquad (10)$$

for the nonpeak niche. N denotes the population size, and f_i corresponds to the fitness value of the peak i. The variable X_i represents the observed number of individuals in a niche i, μ_i represents the expected ideal number, and σ_i represents the standard deviation of the number of individuals in the ideal distribution.

The chi-square-like performance statistic characterizes the ability of the niching technique to proportionally populate the niches of the search space. The smaller the measure, the better the method.

Number of Fitness Function Evaluations: In many applications such as electromagnetic design, the computational cost of fitness functions can be very expensive. Therefore, we are interested in evaluating the efficiency of niching methods at limited numbers of function evaluations. Experimental results were established for 900 fitness function evaluations (30 individuals, 30 generations denoted by test2 in the following). Simulations were also carried out with a higher number of fitness function evaluations 200 000 (100 individuals, 200
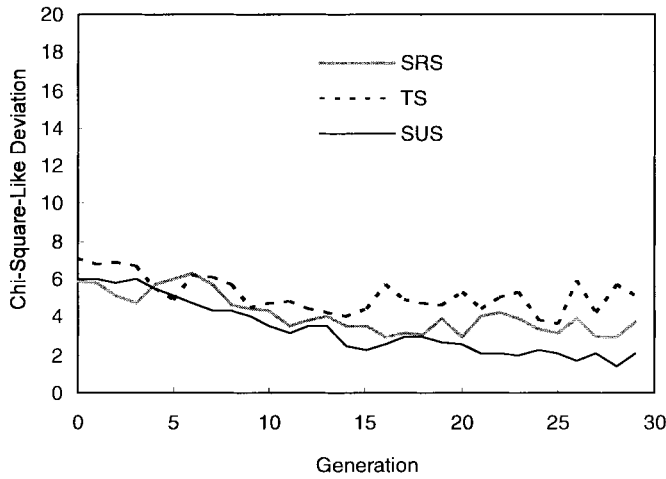
Fig. 6. Chi-square-like deviation of the selection schemes investigated on function $F3$. The population size is $N = 30$.
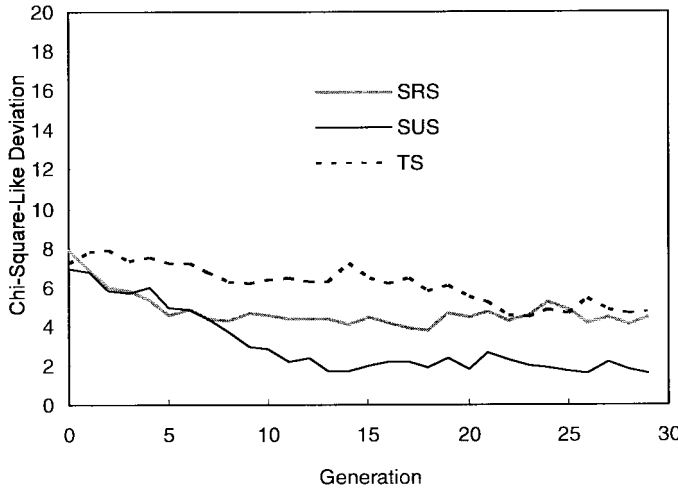

Fig. 8. Chi-square-like deviation of the crossover schemes investigated on function $F3$. The population size is $N = 100$.

Fig. 7. Chi-square-like deviation of the selection schemes investigated on function $F4$. The population size is $N = 30$.

Fig. 9. Chi-square-like deviation of the crossover schemes investigated on function $F4$. The population size is $N = 100$.

generations denoted by test1 in the following) for comparison with other experimental studies. These values were considered reasonable, and no claim is made to their optimality.

V. EXPERIMENTAL STUDY

All experiments were performed with a genotype coded in a 30-bit number using Gray parameter encoding. Mutation rates and crossover probabilities were chosen according to earlier recommendations [8], [9].

The mutation was removed for functions $F3$ and $F4$ to prevent the restoration of lost diversity. Recall that the main role of mutation is to protect individuals from the loss of genetic material by always maintaining diversity in the population. If we want to assess the efficiency of the niching schemes, it is necessary to isolate the different population diversity mechanisms by resetting the mutation rate.

Since RTS and DC use implicitly a full crossover probability, $p_c = 1$ was set in all other niching methods. Moreover, this allows to evaluate the niching GA's performance in the most disruptive case.
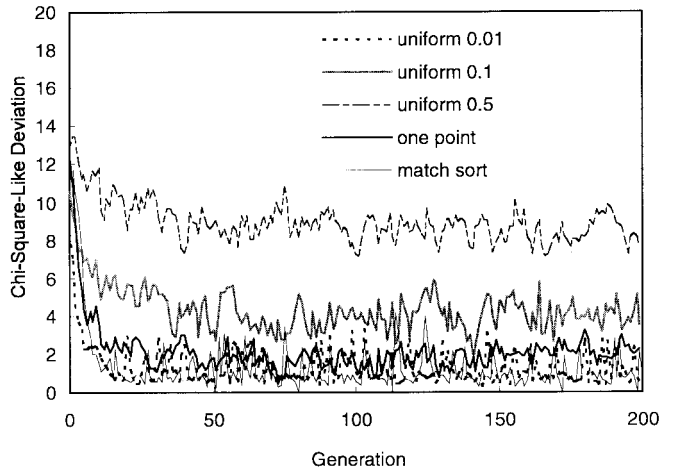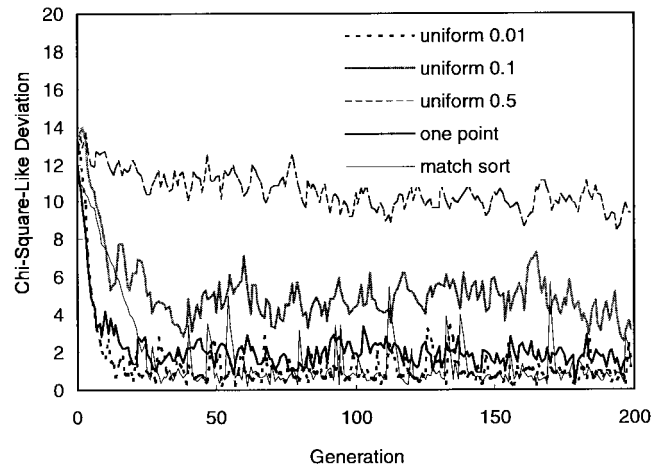
For illustration, ten runs were made with different populations generated at random for each scheme to take into account the stochastic nature of GA's. An average is calculated for the performance criteria.

A. Sharing Experiments

Selection in the Sharing Method: We investigate the effect of the selection scheme on the sharing efficiency. The different schemes reviewed in the first section, namely SUS, SRS and TS, with continuously updated sharing, are compared. The crossover operator is the standard one point crossover with probability $p_c = 1$, the mutation probability being set to zero. The parameter α is set to one and a value of 0.1 is taken for the niche radius. Tables I and II summarize statistics on performance criteria for the functions $F3$ and $F4$ respectively.

Typical chi-square-like deviations on functions $F3$ and $F4$ are displayed in Figs. 4–7 for the selection schemes investigated.

Results show the superiority of SUS regardless of the population size and the number of fitness function evaluations. TS and SRS fail to maintain all peaks at low number of

TABLE III
RECOMBINATION IN THE SHARING METHOD. THIS TABLE PRESENTS THE VALUES OF THE PERFORMANCE CRITERIA (SEE SECTION IV) ON FUNCTION $F3$ FOR EACH RECOMBINATION SCHEME. TEST1 IS PERFORMED USING 100 INDIVIDUALS AND 200 GENERATIONS. TEST2 IS PERFORMED USING 30 INDIVIDUALS AND 30 GENERATIONS

Crossover Scheme	Nb of Peaks maintained		Maximum Peaks Ratio		Mean Chi-Square		End Chi-Square	
	test1	test2	test1	test2	test1	test2	test1	test2
one-point	5	4.8	0.997	0.935	2.214	2.957	1.672	1.789
uniform 0.5	3.8	3.6	0.757	0.696	8.926	5.865	8.145	5.547
uniform 0.1	5	4.2	0.995	0.830	4.307	4.020	3.717	3.660
uniform 0.01	5	4.6	0.999	0.889	1.479	3.422	1.041	2.120
match. sort + uniform 0.5	5	4.8	0.999	0.947	1.287	2.839	0.740	1.972

TABLE IV
RECOMBINATION IN THE SHARING METHOD. THIS TABLE PRESENTS THE VALUES OF THE PERFORMANCE CRITERIA (SEE SECTION IV) ON FUNCTION $F4$ FOR EACH RECOMBINATION SCHEME. TEST1 IS PERFORMED USING 100 INDIVIDUALS AND 200 GENERATIONS. TEST2 IS PERFORMED USING 30 INDIVIDUALS AND 30 GENERATIONS

| Selection Scheme	Nb of Peaks maintained		Maximum Peaks Ratio		Mean Chi-Square		End Chi-Square	
	test1	test2	test1	test2	test1	test2	test1	test2
one-point	5	4.8	0.999	0.941	2.395	3.464	2.109	2.281
uniform 0.5	4.2	3.2	0.767	0.670	10.58	6.377	9.613	5.415
uniform 0.1	5	3.8	0.994	0.798	5.144	5.080	4.474	3.883
uniform 0.01	5	4.4	0.999	0.838	1.355	2.998	1.130	2.335
match. sort + uniform 0.5	4.8	4.4	0.984	0.920	2.041	2.712	1.380	1.826
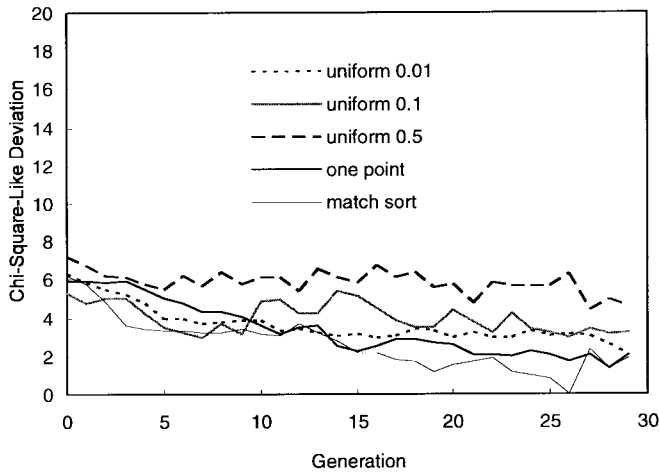
Fig. 10. Chi-square-like deviation of the crossover schemes investigated on function $F3$. The population size is $N = 30$.

Fig. 11. Chi-square-like deviation of the crossover schemes investigated on function $F4$. The population size is $N = 30$.

fitness function evaluations. In particular, TS was unable to form stable subpopulations yielding to a chi-square-like deviation with high fluctuations during the generations. These results are in accordance with Baker's predictions since the SUS is described as the less biased proportional selection technique with minimum spread [5]. As could be expected, it yields a minimum genetic drift and allows the population to proportionally populate the niches with more accuracy.

Recombination in the Sharing Method: As already mentioned in Section II, recombination in the sharing method should prevent the formation of lethals. The first solution to achieve this is to use restrictive mating techniques. In this paper, we propose to sort the population before applying the crossover and mutation operators. A pseudocode of our matching sort algorithm is described as follows.
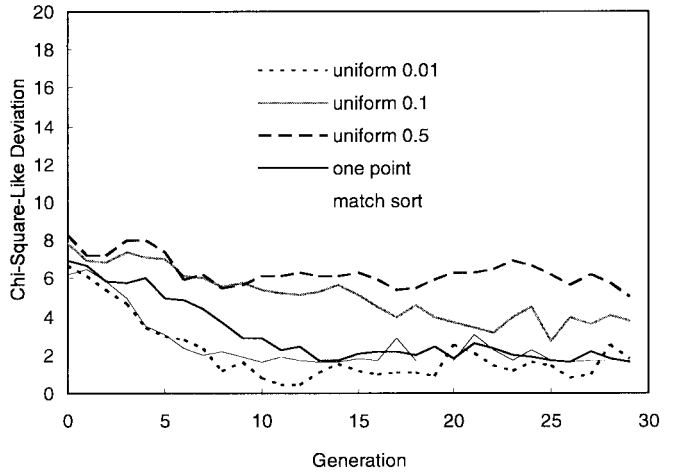
1) Sort the population in decreasing fitness order. Set $i = 1$.
2) Loop until $i = N - 2$
   find the kth element (among $k = i+1 \cdots N$ individuals) that is closest to the ith element of the population.

$$d_{ik} = \min_{j=i+1\cdots N} d_{ij}$$

   exchange the $i + 1$th element with the kth element.
   increase i $(i = i + 1)$.

After the selection of parents, the matching sort is applied and individuals are crossed pairwise following the order of the sort. Note that this scheme is rather costly since it realizes $N(N - 1)/2 - 1$ distance comparisons per generation.

The second way to reduce the formation of lethals is to limit the disruption rate of schemata. This can be achieved by using

TABLE V
NICHING METHODS COMPARED. THIS TABLE PRESENTS THE VALUES OF THE PERFORMANCE CRITERIA (SEE SECTION IV) ON FUNCTION $F3$ FOR EACH NICHING GA.
TEST1 IS PERFORMED USING 100 INDIVIDUALS AND 200 GENERATIONS. TEST2 IS PERFORMED USING 30 INDIVIDUALS AND 30 GENERATIONS

Niching GA	Nb of Peaks maintained		Maximum Peaks Ratio		Mean Chi-Square		End Chi-Square	
	test1	test2	test1	test2	test1	test2	test1	test2
sharing+sort	5	4.8	0.999	0.947	1.287	2.839	0.740	1.972
clearing	5	5	1.000	0.990	0.293	1.045	0.212	0.129
Deterministic Crowding	5	5	1.000	0.999	5.683	3.537	5.820	3.155
RTS (CF=30%N)	5	4.8	1.000	0.958	2.839	2.743	2.520	3.763

TABLE VI
NICHING METHODS COMPARED. THIS TABLE PRESENTS THE VALUES OF THE PERFORMANCE CRITERIA (SEE SECTION IV) ON FUNCTION $F3$ FOR EACH NICHING GA.
TEST1 IS PERFORMED USING 100 INDIVIDUALS AND 200 GENERATIONS. TEST2 IS PERFORMED USING 30 INDIVIDUALS AND 30 GENERATIONS

Niching GA	Nb of Peaks maintained		Maximum Peaks Ratio		Mean Chi-Square		End Chi-Square	
	test1	test2	test1	test2	test1	test2	test1	test2
sharing+sort	4.8	4.4	0.984	0.920	2.041	2.712	1.380	1.826
clearing	5	4.8	1.000	0.933	1.448	2.397	1.354	1.477
Deterministic Crowding	4	4	0.778	0.768	9.582	5.263	10.83	5.264
RTS (CF=30%N)	5	5	1.000	0.998	6.516	3.698	6.279	2.924

low recombination operators such as uniform parameterized crossover [17], [18]. Standard uniform crossover swaps two parents' alleles with a probability of 0.5. Under uniform parameterized crossover, an additional parameter P_0 defines the probability of swapping. This operator combines a high recombination potential and a good exploration power with a low level of disruption [18].

We compare the efficiency of these recombination schemes with standard uniform crossover and one-point crossover. Experiments are made on functions $F3$ and $F4$ with SUS and no mutation. The crossover probabilities are set to one in each recombination schemes. The parameter α is set to one, and a value of 0.1 is taken for the niche radius. Tables III and IV summarize statistics on performance criteria for the functions $F3$ and $F4$ respectively.

Typical chi-square-like deviations on functions $F3$ and $F4$ are displayed in Figs. 8–11 for the recombination schemes investigated.

Results show a slight superiority of our matching sort algorithm regardless of the number of fitness function evaluations. Uniform parameterized crossover with a very low switching probability works well for the test1 but is less efficient when the number of fitness function evaluations is reduced (test2). This reveals its difficulty in exploring the entire search space by producing individuals in different niches when the number of fitness functions is limited. In that case, standard one-point crossover is better. Nonetheless, this recombination operator is obviously more disruptive than uniform parametrized crossover with low probability of swapping (see test1). Moreover, note that it gives poor results when the size of the chromosome is reduced; recall that the disruption rate of the schemata under one-point crossover is $1/l$ where l is the size of the chromosome. Standard uniform crossover was unable to form stable subpopulations because of a massive disruption rate of solutions detected.

Niching Methods Compared: We compare the efficiency of fitness sharing coupled with the matching sort algorithm with the other niching GA's reported in Section III. RTS and DC are implemented with standard uniform crossover. An optimal crowding factor for RTS has been determined empirically for the functions $F3$ and $F4$. We use CF $= 30\%N$, where N is the population size. This leads to CF $= 30$ for the test1 and CF $= 9$ for the test2. Clearing is combined with SUS and an elitist strategy as recommended by Petrowski [15]. In each generation, the dominant individual of each subpopulation competes with the corresponding one of the previous generation. The winners of the resulting tournaments are conserved in the current population. Following this procedure, the best individual of each niche is always preserved during the search. The capacity of the niches is set to ten for test1 and two for test2, respectively. All niching GA's are performed with full crossover probability ($p_c = 1$) and no mutation. The parameter α is set to one, and a value of 0.1 is taken for the niche radius. Tables V and VI summarize statistics on performance criteria for the functions $F3$ and $F4$ respectively.

Typical chi-square-like deviations on functions $F3$ and $F4$ are displayed in Figs. 12–15 for the niching GA's investigated.

The efficiency of the niching GA is related to its capacity to find new niches by producing new individuals without discarding the niches already identified. Clearing was the best niching GA that realizes this compromise. It produces a great quantity of new individuals by randomly recombining elements of different niches and controls this production (and obviously the genetic drift caused by selection) by resetting the fitness of poor individuals in each different niche. Furthermore, the elitist strategy prevents the rejection of the best individual of each niche from the population. For these reasons, clearing surpasses all other niching GA's and combines a very low chi-square-like deviation with a good detection of the peaks.
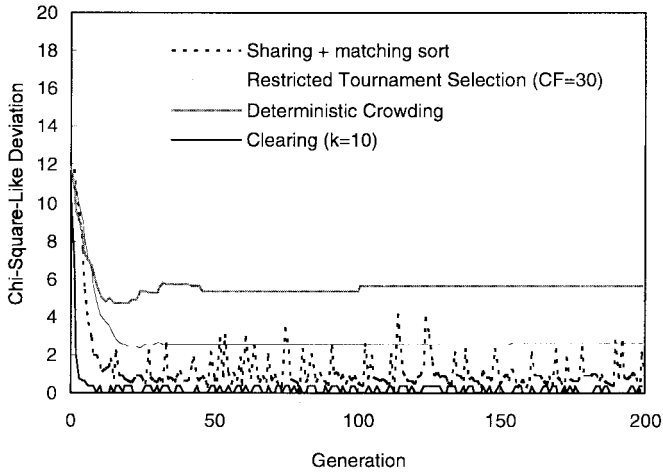
Fig. 12. Chi-square-like deviation of the niching GA's investigated on function $F3$. The population size is $N = 100$.
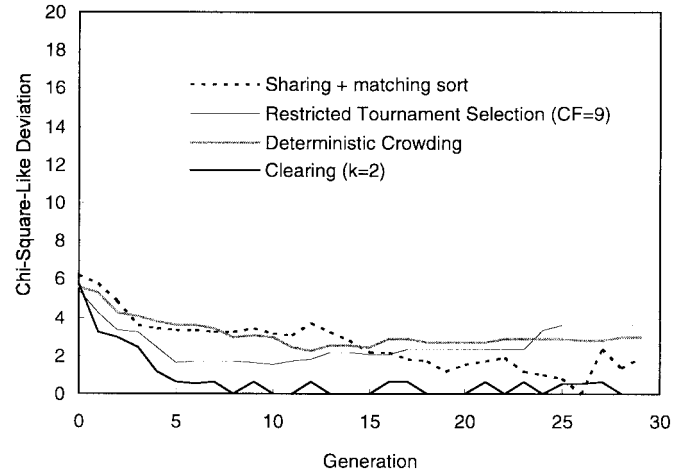

Fig. 14. Chi-square-like deviation of the niching GA's investigated on function $F3$. The population size is $N = 30$.
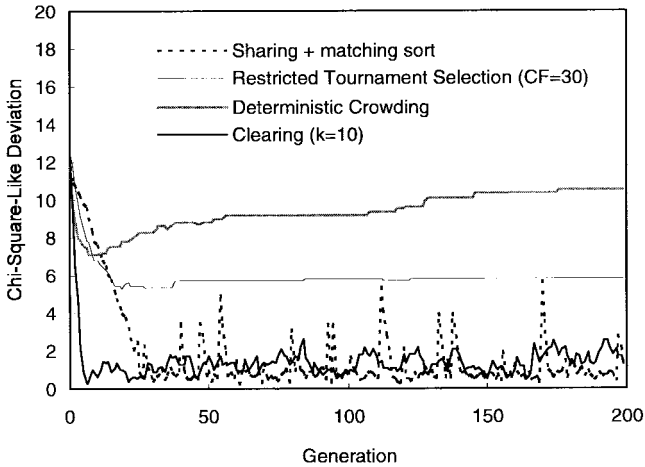

Fig. 13. Chi-square-like deviation of the niching GA's investigated on function $F4$. The population size is $N = 100$.
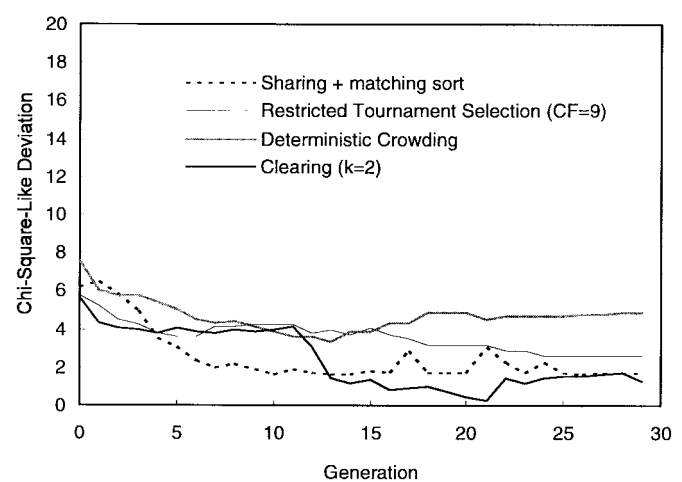

Fig. 15. Chi-square-like deviation of the niching GA's investigated on function $F4$. The population size is $N = 30$.

Crowding schemes were unable to maintain low chi-square-like deviations during generations. The first reason for this is mentioned in [8]. Crowding schemes use a replacement strategy which minimizes the changes in the population. The distribution of the population in the different niches strongly depends on the initial distribution. This explains the higher chi-square-like deviations noted for RTS and DC in comparison with those corresponding to the sharing and clearing methods which directly use a proportional selection. Second, replacement errors can occur for individuals located at the edge of the niches. This explains poor results noted for DC when it is applied to $F4$. DC detects the five peaks of this function in the first generations. Nevertheless, in the following generations, it appears that individuals located on the third peak (of coordinate $x = 0.451$) progressively migrate to the next peak (the fourth peak of coordinate $x = 0.681$) because of replacement errors. At the two-hundredth generation, all individuals are discarded from the third peak yielding a poor chi-square distribution. RTS is less sensitive to these errors with the size of the crowding factor CF used in the experiments. Therefore, it surpasses DC in all cases. Sharing works well on these easy problems.

Let us examine now the efficiency of these niching GA's on the massively multimodal function $F7$. Goldberg [6] solved this problem with the sharing method by raising the shared fitness to a power of 15 and using a huge population of 5000. Mahfoud [9] reported that more than 20 individuals per niche are necessary for DC to find the global optima. Such parameters are inconceivable for applications with high computational time of the objective function such as finite element applications. Recall that we are interested in assessing the efficiency of the niching GA's at a limited number of function evaluations. Therefore, we prefer solving this problem with the test1 (100 individuals, 200 generations). For each GA, we use the Hamming distance as a similarity metric and a crossover probability of 1.0. Considering the difficulty of this function, the mutation rate is not removed but set to the low value of 0.001 to increase exploration rate. The distance between two individuals is normalized by the biggest distance value in the search space according to [15] and [16]. The niche radius is set to 0.2 in the clearing and the sharing methods. Both of these methods are implemented with SUS. We compare their efficiency with and without scaling. Two

TABLE VII
NICHING METHODS COMPARED. THIS TABLE PRESENTS THE NUMBER OF PEAKS MAINTAINED DURING THE
SEARCH FOR THE FUNCTION $F7$. ALL TESTS ARE PERFORMED WITH 100 INDIVIDUALS AND 200 GENERATIONS

Niching GA	Nb of Peaks maintained	Confidence Margin (95%)
RTS (CF=5)	1.63	± 0.21
RTS (CF = 10)	2.28	± 0.23
RTS (CF = 20)	1.64	± 0.25
RTS (CF = 50)	0.56	± 0.14
DC	0.43	± 0.13
Standard sharing	0.00	± 0.00
Standard sharing + fixed scaling (β=15)	1.48	± 0.17
Standard sharing + *dynamic scaling*	3.13	± 0.32
clearing (k=2)	14.10	± 1.59
clearing (k=2) + fixed scaling (β=15)	15.58	± 1.66
clearing (k=2) + *dynamic scaling*	14.06	± 1.72

forms of scaling are investigated. The first one uses a fixed power of value $\beta = 15$ similarly to [6]. The second increases the scaling during the generations (*dynamic scaling*). A value of $\beta = 1$ is taken for the first fiftieth generations and is linearly increased to 15 in the following generations.

Table VII shows the effective number of peaks maintained after 200 generations for each niching GA. One hundred runs are made with different initial population generated at random and an average over these runs is taken for the number of peaks maintained at the two-hundredth generation.

For this problem, no niching GA was able to maintain the 32 global maxima. Nonetheless, clearing was obviously better than any other technique by finding and preserving between 14 and 15 optima. In agreement with [16], we find that sharing without a scaled function fails to detect any global solution. Raising the fitness to a power of 15 magnifies the differences between the global and local optima. This makes it easier for the sharing method to find the global optima. Nonetheless, using such a power for scaling the shared fitness discards many individuals from the population and reduces genotypic diversity. This explains the better results obtained for sharing with a dynamic scaled function, which allows more diversity in the population at the beginning of the search. Unlike the sharing method combined with a scaled function, the basic clearing procedure does not reject a great number of promising solutions. It only discards individuals located in the neighborhood of one dominant of a subpopulation and preserves good configurations. Moreover, clearing does not require a scaled function to find one global solution. On the contrary, a dynamic scaled function seems to be somewhat misleading for the clearing algorithm and in particular for dominant individuals of different generation who compete through our elitism scheme.

RTS surpasses DC for all studied cases. The efficiency of RTS, however, strongly depends on the value of the crowding factor. This can be a significant flaw because the optimal value for this parameter is generally unknown for a given objective function.

VI. CONCLUSIONS AND OUTLOOKS

This paper gives an overview of multiple niching GA's and points out some important issues of multimodal learning such as selection, recombination, restrictive mating, and fitness scaling. A simple analysis led us to class niching GA's in two different groups. The first one involves GA's which are characterized by an explicit neighborhood since they need an explicit niche radius (clearing and sharing). This can be an important drawback for problems for which distance between optima cannot be estimated. The second consists of techniques for which neighborhood is implicit (crowding schemes). In that case, the algorithm requires no information about the search space and can be easily applied to various problems without restrictions.

Among all niching GA's reviewed in this paper, clearing can be considered as the best method provided that the niche radius and the niche capacity are correctly estimated. Sharing works well on easy problems with some precautions. One should use stochastic universal selection and mating restriction schemes or low recombination operators to maintain stable subpopulations and avoid disruption of peaks detected. Nonetheless, sharing fails on hard problems reflecting its difficulty to differentiate the global from the local optima in multimodal deceptive landscapes through the constant modification of the fitness during generations. Therefore, sharing often requires a scaled fitness to increase peaks differentiation with a risk of premature convergence. This scaling is not necessary for crowding schemes since they are based on tournament rules. Restricted tournament selection gives slightly better results than deterministic crowding which has difficulties to preserve the niches in some cases as a result of replacement errors.

The application of new recombination operators in multimodal landscapes such as the matching sort algorithm or uniform parameterized crossover seems to be a promising way to ensure the stability of the niches. Finally, we also mention the necessity of investigating clustering techniques and adaptive niche radius methods to cleverly set the similarity threshold of niching GA's with explicit neighborhood.

REFERENCES

[1] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. of Michigan Press, 1975.

[2] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proc. 2nd Int. Conf. Genetic Algorithms*, J. J. Grefensette, Ed. Hillsdale, NJ: Lawrence Erlbaum, 1987, pp. 41–49.

[3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. New York: Addison-Wesley, 1989.

[4] K. Deb and D. E. Goldberg, "An investigation of niche and species formation in genetic function optimization," in *Proc. 3rd Int. Conf. Genetic Algorithms*, J. D. Schaffer, Ed. San Mateo, CA: Morgan Kaufmann, 1989, pp. 42–50.

[5] J. E. Baker, "Reducing bias and inefficiency in the selection algorithm," in *Proc. 2nd Int. Conf. Genetic Algorithms*, J. J. Grefensette, Ed. Hillsdale, NJ: Lawrence Erlbaum, 1987, pp. 14–21.

[6] C. K. Oei, D. E. Goldberg, and S. J. Chang, "Tournament selection, niching and the Preservation of diversity," Univ. of Illinois, Urbana, Illinois Genetic Algorithms Laboratory, Tech. Rep. 91011, 1991.

[7] X. Yin and N. Germay, "A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization," in *Artificial Neural Networks and Genetic Algorithms, Proceedings of the International Conference in Innsbruck*, R. F. Albrecht, C. Reeves, and N. C. Steele, Eds. Berlin, Germany: Springer-Verlag, 1993, pp. 450–457.

[8] B. L. Miller and M. J. Shaw, "Genetic algorithms with dynamic niche sharing for multimodal function optimization," in *Proc. 1996 IEEE Int. Conf. Evolutionary Computation*, 1996. Piscataway, NJ: IEEE Press, pp. 786–791.

[9] S. W. Mahfoud, "Niching methods for genetic algorithms," Ph.D. dissertation, Univ. of Illinois, Urbana-Champaign, 1995.

[10] D. Beasley, D. R. Bull, and R. R. Martin, "A sequential niche technique for multimodal function optimization," *Evol. Comput.*, vol. 1, no. 2, pp. 101–125, 1993.

[11] S. Forrest, B. Javornik, R. E. Smith, and A. S. Perelson, "Using genetic algorithms to explore pattern recognition in the immune system," *Evol. Comput.*, vol. 1, no. 3, pp. 191–211, 1993.

[12] Y. Davidor, "A naturally occurring niche & species phenomenon: The model and first results," in *Proc. 4th Int. Conf. Genetic Algorithms*, R. Belew and L. Booker, Eds. San Mateo, CA: Morgan Kaufmann, 1991, pp. 257–263.

[13] K. A. DeJong, "An analysis of the behavior of a class of genetic adaptative systems," Ph.D. dissertation, Univ. of Michigan, Ann Arbor, 1975.

[14] G. Harik, "Finding multimodal solutions using restricted tournament selection," in *Proc. 6th Int. Conf. Genetic Algorithms*, L. J. Eshelman, Ed. San Mateo, CA: Morgan Kaufmann, 1995, pp. 24–31.

[15] A. Pétrowski, "A clearing procedure as a niching method for genetic algorithms," in *Proc. 1996 IEEE Int. Conf. Evolutionary Computation*, Nagoya, Japan, 1996, pp. 798–803.

[16] D. E. Goldberg, K. Deb, and J. Horn, "Massive multimodality, deception and genetic algorithms," in *Parallel Problem Solving from Nature (PPSN-2)*, R. Männer and B. Manderick, Eds. North Holland: Amsterdam, pp. 37–46, 1992.

[17] G. Syswerda, "Uniform crossover in genetic algorithms," in *Proc. 3rd Int. Conf. Genetic Algorithms and Their Applications*, J. D. Schaffer, Ed. San Mateo, CA: Morgan Kaufmann, 1989, pp. 2–8.

[18] W. M. Spears and K. A. DeJong, "On the virtues of parametrized uniform crossover," in *Proc. 4th Int. Conf. Genetic Algorithms*, R. Belew and L. Booker, Eds. San Mateo, CA: Morgan Kaufmann, 1991, pp. 230–236.

[19] P. Darwen and X. Yao, "A dilemma for fitness sharing with a scaling function," in *Proc. 1995 IEEE Conf. Evolutionary Computation*, Piscataway, NJ: IEEE Press, 1995, pp. 166–171.

[20] _____, "Speciation as automatic categorical modularization," *IEEE Trans. Evol. Comput.*, vol. 1, pp. 101–108, July 1997.