J Heuristics (2008) 14: 243–270
DOI 10.1007/s10732-007-9035-1

Crowding with nearest neighbors replacement for multiple species niching and building blocks preservation in binary multimodal functions optimization

Minqiang Li · Jisong Kou

Received: 30 June 2006 / Revised: 22 November 2006 / Accepted: 11 December 2006 /
Published online: 11 September 2007

Abstract This paper introduces a novel niching scheme called the q-nearest neighbors replacement (q-NNR) method in the framework of the steady-state GAs (SSGAs) for solving binary multimodal optimization problems. A detailed comparison of the main niching approaches are presented first. The niching paradigm and difference of the selection-recombination genetic algorithms (GAs) and the recombination-replacement SSGAs are discussed. Then the q-NNR is developed by adopting special replacement policies based on the SSGAs; a Boltzmann scheme for dynamically sizing the nearest neighbors set is designed to achieve a speed-up and control the proportion of individuals adapted to different niches. Finally, experiments are carried out on a set of test functions characterized by deception, epistasis, symmetry and multimodality. The results are satisfactory and illustrate the effectivity and efficiency of the proposed niching method.

Keywords Genetic algorithms · Crowding · Nearest neighbors replacement · Niching · Steady-state genetic algorithms · Binary multimodal functions optimization

1 Introduction

There is a large number of optimization problems that consist of binary multimodal functions which are characterized by epistasis, deception, multimodality, symmetry, scaling, and noise that pose difficulties for the standard genetic algorithms (GAs) (Deb and Goldberg 1989; Goldberg 1989; Davidor 1991; Goldberg et al. 1992; Reeves and Wright 1995; Pelikan and Goldberg 2000; Van Hoyweghen et al. 2002a; Sastry et al. 2005). The problems with these features, usually requiring prohibitive

M. Li (✉) · J. Kou
School of Management, Tianjin University, Tianjin 300072, People's Republic of China
e-mail: mqli@tju.edu.cn

computation, constitute challenging tasks for the researchers in the community of GAs or evolutionary computation (EC).

Various approaches in the GAs/EC, usually called niching methods or techniques (Goldberg and Richardson 1987; Mahfoud 1995; Goldberg and Wang 1997; Bäck et al. 2000), have been proposed to solve multimodal function optimization problems. These approaches allow the GAs to identify and locate multiple optima simultaneously or sequentially through introducing implicit or explicit niching mechanisms, and most of them require the proper parameterization to achieve good behavior and performance, which is quite an undertaking even with good apriori information of a problem. The binary multimodal function optimization problems remain difficult tasks for existing niching schemes to reliably and efficiently find the multiple niches regardless of parameters setting.

When the GAs are used to deal with multimodal function optimization problems, they should be equipped with a new mechanism to promote the identification, location, and coevolution of the multiple niches without employing particular niching parameters and elaborate but problem-dependent procedures. What we propose in this paper is a special niching method, called crowding with nearest neighbors replacement (NNR) in the recombination-replacement paradigm of the steady-state GAs (SSGAs).

In Sect. 2, this paper gives a brief review of the various niching methods for multimodal functions optimization, which is followed by a detailed comparison of the characteristics of the existing approaches. In Sect. 3, we discuss the need and inspiration of the crowding based niching schemes, and describe the procedure of the niching scheme with the nearest neighbors replacement embedded in the steady-state GAs. In Sect. 4, we introduce the main binary multimodal functions which are used to test the proposed niching method. Experiments are conducted and results are compared between the crowding with NNR scheme and the deterministic crowding. Finally, the paper concludes by giving a summary of the work done and recommending the possible extensions in the future work.

2 Review of niching methods

A niche represents a local optimum and its attraction basin (or region) in the feasible solutions space of a multimodal function (Goldberg 1989; Mahfoud 1995; Horn 1997; Eiben and Smith 2003). The niching GAs try to implement simultaneous exploration and exploitation of a number of different environmental niches in multimodal optimization problems.

Goldberg defined the niche and speciation techniques as having the following functions required (Goldberg 1989, pp. 185–197): (1) stable and noncompetitive subpopulations (species), formed around all peaks, serve different domains of a function, and the uncontrolled growth of a particular species within a population is limited; (2) it is expected to allocate subpopulations to peaks in proportion to their magnitude (fitness).

Whatever the niching mechanism is, the final population or multiple subpopulations should be distributed among all or most of the niches through niching methods, where only a subset of individuals (called a species) is permitted to adapt to the local environment or a niche (called local adaptation). All of the subpopulations reach gradually the equilibria in proportion to the fitness or the attraction basin sizes of niches, which is preserved afterward.

The GAs with niching techniques capable of solving the multimodal optimization problems are also called the niching GAs or the multimodal GAs. These methods may be classified into seven categories by the naming of different niching schemes: (1) crowding and preselection (implicit niching) (De Jong 1975; Mahfoud, 1992, 1995; Cedeño and Vemuri 1999; Thomsen 2004), (2) fitness sharing (implicit niching) (Goldberg and Richardson 1987; Goldberg 1989; Miller and Shaw 1996; Sareni and Krähenbühl 1998; Cioppa et al. 2004), (3) restricted competition (explicit niching) (Goldberg 1989; Goldberg et al. 1990; Mahfoud 1995), (4) sequential niching (explicit niching) (Beasley et al. 1993), (5) island model GAs and parallel GAs (explicit niching) (Siarry et al. 2000, 2002; Gudla and Ganguli 2005), (6) immune systems (implicit niching) (Forrest et al. 1993), (7) hybrid and miscellaneous explicit niching methods as the clustering based and niche preserving based fitness sharing (Yin and Germany 1993; Lin and Yang 1998; Pelikan and Goldberg 2000; Jelasity et al. 2001; Gan and Warwick 2000; Li et al. 2002), the over-specification (Diploid GAs) (Goldberg 1989; Mahfoud 1995), the clearing procedure (Pétrowski 1996), the estimation of distribution algorithm (EDA) (Peña et al. 2005), the parallel hill-climbing and the multi-start strategy of hill-climbing in conventional mathematical programming domain.

What we call the explicit niching or the implicit niching of a niching method is defined by how the niches are identified. If a niching method identifies niches in the evolutionary process by using heuristic procedures with distance-based threshold parameters, we will call it an explicit niching method; otherwise, it is called an implicit one.

A set of indexes is adopted to describe different niching methods based on their characteristics, see Table 1.

The sequential niching (Beasley et al. 1993) is actually a technique that searches in the feasible solution space for one niche at a time iteratively by using any unimodal optimization algorithms such as the simple GAs, hilling climbing, simulated annealing, etc. It has nothing to do with the co-adaption of individuals to niches. The resource sharing (Horn 1997) and immune systems methods bear some similarity with the fitness sharing.

Since the standard GAs rely on fitness to determine the survival of individuals and the production of new offspring in the next population, the genetic drift is unavoidable in the evolution process of population. Positively, the GAs is able to do exploration and doesn't depend on the initial population while the selection operation is implemented on the total population. But the replacement operation (generational or non-generational) also gives rise to genetic drift when it works on the total population. In other words, if the replacement is implemented on the total population, called the global replacement in population evolution, the individual competition in populations will be global, which will probably cause genetic drift.

The fitness sharing method adopts special sharing scheme to carry out global selection and local replacement, which makes it possible for individuals to only compete with the ones in the same niche to attain survival. The genetic drift can only happen temporarily in populations, and then it will be resolved by the counter force

Table 1 Comparison of the characteristics of main niching methods

Indexes	Niching methods			
	Fitness sharing and dynamic schemes	Crowding and various crowding schemes	Restricted competition	Island model and parallel GAs
(a) Genetic operations:				
Selection	Global	Global/Local	Local	Local
Replacement	Local	Global/Local	Global-Local	Global
Competition	Local	Global/ Local	Global-Local	Global-Local
(b) Performances:				
Exploration	High	High	Low	Intermediate
Exploitation	Intermediate	Intermediate/Low	High	High
Genetic drift	No	Yes	Yes	Yes
Locating niches	High	High	Low	Intermediate
Maintaining niches	High	High/Intermediate	High/Intermediate	Intermediate
(c) Control parameters	Niching radius/ Niches dissimilarity threshold	CF/No	Dissimilarity threshold	Migration policy
(d) Dependence:				
Problems	Yes	No	Yes	No
Initial population	No	No	Yes	Yes

of fitness sharing, which can locate and maintain all or the most of niches within an enough large population. Since the selection is global in fitness sharing approaches, the selection-recombination scheme could ensure the exploration power needed for locating new niches. This makes it possible that the distribution of individuals among niches is not dependent on the initial sampling and distribution.

The major crowding methods for multimodal optimization include the De Jong's crowding factor model and modified versions (De Jong 1975; Mahfoud 1995), the deterministic crowding model (DC) (Mahfoud 1995), the restricted tournament selection (RTS) (Harik 1995), the probabilistic crowding (PC) (Mengshoel and Goldberg 1999), crowding-based different evolution (CDE) (Thomsen 2004), and the multiniche crowding (MNC) (Cedeño and Vemuri 1999). The common feature of the former five crowding schemes for niching purpose is to implement special replacements. These crowding approaches hold a global selection through random selection and pairing of individuals. In De Jong's crowding scheme, CF individuals are selected randomly from the entire population for the offspring to do replacement, thus the individuals will compete globally. The DC and the PC use two rounds of tournaments to decide the replacement, and the offspring may belong to niches different from those of their parents, so that the individual competition could be either local or global. The CDE resembles what we propose in this paper and was designed only for the real-valued multimodal optimization problems. The MNC incorporates the crowding selection with crowding replacement, actually it is a hybrid of the crowding scheme and the restricted competition scheme. There still exist the possibility of genetic drift to some extent in the major crowding schemes. It is possible that all of the niches could be found in the interim populations, but some may be lost along with the progress of population evolution.

With a mating restriction policy, the restricted competition scheme makes use of local selection for recombination. By setting up a dissimilarity threshold for individuals, two individuals are allowed to produce offspring only when they are thought to be from the same niche by distance-based similarity. But there is still the possibility that two parents are selected from different niches and produce aliens that belong to other niches. The competition domain is between the local and the global, and the incurred genetic drift may prevent the population from maintaining multiple species. Besides, the restriction policy also restricts the exploration capability of recombination for finding new niches, and thus the sampling error in initialization may affect the niching results.

The island model GAs or multideme GAs are easily taken as the tailored method for multimodal function optimization. Actually, there are many factors that may make them fail to identify and maintain niches as expected. The most salient fact is that the migration policy in the island model GAs is good for exploration, but may also result in the dominance and take over of a subpopulation by intruders. Therefore, we think that it has an intermediate capability for locating and maintaining niches. Besides, the punctuated equilibria are also dependent to some extent on the initial population.

While we may classify the equilibria of population distribution for single-population GAs or multi-populations GAs into two types: fitness dependent distribution and niche-size (attraction basin) dependent. In the former case, the number of individuals adapting to a niche is proportional to the fitness of the niche, such as the fitness sharing method and its modifications. The restricted competition belongs to the latter case. The crowding methods and island model GAs fall between the two extremes.

3 Niching: selection-recombination vs. recombination-replacement

The discussion in Sect. 2 shows that the localization of individual competition is the basis for the coevolution of multiple species in limited resources (Horn 1997). There are two paradigms to approach this problem: the selection-recombination GAs and the recombination-replacement SSGAs.

3.1 Selection-recombination niching

The primary idea of the selection-recombination GAs is that the GAs work through a mechanism of quasi-decomposition and reassembly (Goldberg 2002), where the identification and exchange of building blocks (BBs) must be promoted by keeping the bitwise diversity of the population, which is one of the most important traits in the selection-recombination GAs. In order to make an effective niching for multimodal functions optimization, the selection-recombination GAs usually change the fitness landscape to make sure that the selection pressure is moderate between individuals adapting to different niches. The perfect sharing combined with proportionate selection yields an ideal restorative force to overcome the genetic drifts once happened in former populations.

The multimodal GAs require not only bitwise diversity but also the BBs-wise diversity and niches-wise diversity, which can be attained by fitness sharing techniques in one population. For instances, Van Hoyweghen et al. (2001) employed fitness sharing to identify and preserve the non-inferior BBs to deal with the synchronization and symmetry (concurrently forming and preserving of the multiple non-inferior BBs which are symmetrical) in the Ising model. Sastry et al. (2005) used the sub-structural niching in the estimation of distribution algorithms to explicitly identify and preserve the building blocks, which successfully generated and maintained multiple global optima.

3.2 Recombination-replacement niching

The SSGAs adopt a different population evolution policy which only produces one or two offspring to replace the individuals in the population in each iteration (Syswerda 1991; Bäck et al. 2000, p. 207 in vol. 1). The SSGAs does better in maintaining population diversity than the generational GAs because of the latter's generational errors.

In the recombination-replacement paradigm of the SSGAs, the replacement of the worst or random individual policy often leads to the replacement of an individual in one niche by an offspring belonging to another niche, which gives rise to the disappearance of niches with lower fitness.

Compared with the fitness sharing, the crowding based niching methods do not have the rehabilitation mechanism that can restore the population from genetic drift caused by selection and generation errors. What the crowding approaches resort to is to avoid the happening of the genetic drift and dominance of a niche in the population. The major crowding techniques can not guarantee fully localization of the competition of individuals.

There is another perspective for the above analyses. What the selection-recombination paradigm implements is the top-down pulling mechanism, we call it the fitness-similarity based niching. Whereas the recombination-replacement paradigm carries out the down-top pushing mechanism to realize co-adaptation of individuals to multiple niches, which we call the similarity-fitness based niching. We will illustrate empirically that the latter is able to preserve BBs-wise diversity when proper replacement policy is adopted.

4 A niching scheme based on crowding with nearest neighbors replacement

Considering the situation of niching in a single population, we adopt the recombination-replacement paradigm of the SSGAs to design and test niching methods based on the crowding mechanism.

Suppose that $\mathbf{P}$ denotes a population, $\mathbf{P} = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N\}$, where the population size is N and is kept constant during the total evolution process. An individual or a bit string chromosome, $\mathbf{a}_i = \langle a_{i,1}, a_{i,2}, \ldots, a_{i,L} \rangle$ $(i = 1, 2, \ldots, N)$, $a_k \in \{0, 1\}$, is an ordered set of L-dimension binary variables in a binary optimization function. Two individuals $\mathbf{a}_i = \langle a_{i,1}, a_{i,2}, \ldots, a_{i,L} \rangle$, $\mathbf{a}_j = \langle a_{j,1}, a_{j,2}, \ldots, a_{j,L} \rangle$ are selected randomly from the population to produce two children $\mathbf{a}'_1 = \langle a'_{1,1}, a'_{1,2}, \ldots, a'_{1,L} \rangle$, $\mathbf{a}'_2 = \langle a'_{2,1}, a'_{2,2}, \ldots, a'_{2,L} \rangle$ by using recombination operators.

The nearest neighbor technique, which can be traced back to the work of Parzen (1962), is commonly used in the fields of statistics, machine learning, and pattern recognition. Here, it is employed to design a novel replacement policy in the crowding scheme, called the q-nearest neighbors replacement (q-NNR) which is presented as below.

4.1 Crowding with q-NNR

In order to implement totally local competition among individuals, we should consider how to insert the two offspring $\{\mathbf{a}'_1, \mathbf{a}'_2\}$ into the population $\mathbf{P}$, or which individuals in $\mathbf{P}$ are selected to be replaced. In contrast to selecting *CF* individuals randomly from population as in the De Jong's crowding scheme or using offspring to replace their parents as in the DC, we tend to select the most similar ones in the neighborhood of $\mathbf{a}'_1$ or $\mathbf{a}'_2$ in the genotypic space separately. The Hamming distance is employed to measure the similarity between $\{\mathbf{a}'_1, \mathbf{a}'_2\}$ and individuals in $\mathbf{P}$:

$$d(\mathbf{a}', \mathbf{a}_i) = \|\mathbf{a}' - \mathbf{a}_i\| = \sum_{k=1}^{L} |a'_k - a_{i,k}|, \quad \mathbf{a}_i \in \mathbf{P}, \tag{4.1}$$

where $\mathbf{a}'$ denotes $\mathbf{a}'_1$ or $\mathbf{a}'_2$. The Euclidean distance is used for the real-valued optimization problems as in our previous work (Li and Kou 2005).

The individuals in population $\mathbf{P}$ is sorted by ascending order based on distance $\{d(\mathbf{a}', \mathbf{a}_i) \mid i = 1, 2, \ldots, N\}$ to get the reordered individuals set as $\mathbf{P}' = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N\}$. The formerly ranked q individuals in $\mathbf{P}'$ are chosen to form the q-nearest neighbors set: $\mathbf{P}'' = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_q\}$ $(\mathbf{P}'' \subset \mathbf{P}', q \ll N)$, and the one with the lowest fitness in $\mathbf{P}''$ is to be replaced. Suppose that $\mathbf{a}^* = \{\mathbf{a}_j \mid f(\mathbf{a}_j) \le f(\mathbf{a}_i), \mathbf{a}_i, \mathbf{a}_j \in \mathbf{P}''; i, j = 1, 2, \ldots, q; i \ne j\}$, and if $f(\mathbf{a}') > f(\mathbf{a}^*)$, then $\mathbf{a}^*$ will be replaced by $\mathbf{a}'$. This procedure is implemented for $\mathbf{a}'_1$ and $\mathbf{a}'_2$ respectively.

Figure 1 shows a basic algorithm for the SSGAs with q-NNR crowding scheme, where the mutation operation is non-mandatory in the algorithm. The locally or globally dominant individuals will not take over the population in the q-NNR crowding SSGAs when q takes a small number.

It should be pointed out that the mutation operation is not used in the q-NNR crowding SSGAs on binary multimodal optimization problems. When mutation operation is used in the q-NNR crowding SSGAs, the building blocks or the optima would be destroyed with a very high probability even if they are constructed by crossover operations. The final population is not able to converge to the multiple global optima, and the algorithm can only find solutions in the neighborhoods of the global optima. The algorithm does not get premature without mutation operation because the q-NNR crowding SSGAs is able to maintain the diversity of the population with the special replacement policy. The mutation operation is not optional when the

```
procedure Crowding_$q$-NNR_SSGAs
    set: $T_{\max}$—the maximum iterations for the SSGAs
        $N$—the population size
        $p_c$ [, $p_m$]—the rates for crossover [and mutation] operations
        $q$—the size of the nearest neighbors set
    initialize: $\mathbf{P}(0)$
    for $t = 0$ to $T_{\max}$ do
        selection: $\mathbf{a}_i, \mathbf{a}_j \in \mathbf{P}(t), \mathbf{a}_i \neq \mathbf{a}_j$, random selection
        crossover: $\{\mathbf{a}'_1, \mathbf{a}'_2\} \leftarrow$ crossover$(\mathbf{a}_i, \mathbf{a}_j)$
        [mutation: $\{\mathbf{a}'_1, \mathbf{a}'_2\} \leftarrow$ mutate$(\{\mathbf{a}'_1, \mathbf{a}'_2\})$], optional in implementation
        for $\mathbf{a}'$ in $\{\mathbf{a}'_1, \mathbf{a}'_2\}$ do
            calculate: $\{d(\mathbf{a}', \mathbf{a}_i) \mid i = 1, 2, \ldots, N\}, \mathbf{a}' \in \{\mathbf{a}'_1, \mathbf{a}'_2\}$
            sort: $\mathbf{P}' = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N\}$
            get: $\mathbf{P}'' = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_q\}$, $q$ individuals in the nearest
              neighborhood
            find: $\mathbf{a}^* = \{\mathbf{a}_j | f(\mathbf{a}_j) \leq f(\mathbf{a}_i), \mathbf{a}_i, \mathbf{a}_j \in \mathbf{P}''; i, j = 1, 2, \ldots, q; i \neq j\}$
            if $f(\mathbf{a}') > f(\mathbf{a}^*)$ then
                replace: $\mathbf{a}^* \leftarrow \mathbf{a}''$
            end if
        end for
        $\mathbf{P}(t+1) \leftarrow \mathbf{P}(t)$
    end for
    output $\{\mathbf{P}(T_{\max})\}$
end (procedure)
```

Fig. 1 The basic algorithm for crowding scheme with q-NNR in the SSGAs

q-NNR crowding SSGAs deals with real-valued multimodal optimization problems as in the work of Li and Kou (2005) which illustrated the effectiveness of the proposed approach on complex continuous function landscapes consisting of non-linear and non-separable niches.

In order to enhance the efficiency of calculating the q-nearest neighbors set, we design the KD-tree index data structure of the population to help search the nearest neighbors in the population. Actually, a dynamic KD-tree index is created periodically to handle the changing population.

The nearest neighbors sets for both offspring calculated by the q-NNR have no mandatory relation with their parents, which makes the replacement policy totally local even though the selection operation is global. Hence, the number of individuals adapted to a niche is proportional to the size of its attraction basin in the fitness landscape. However, it is easy to change the final population into fitness proportional distribution by using fitness sharing technique.

Since the replacement, when offspring are inserted into the population, only happens in small neighborhoods of offspring consisting of q individuals, it is possible that even less competitive niches with smaller attraction basins and lower fitness can be preserved. In the binary multimodal function optimization (for instance, the bipolar trap problem, the Ising model, etc.), there are a large number of medium-order building blocks (that are also small niches) which constitute the stepping stones to form the global optima. The crowding scheme with q-NNR does not eliminate these building blocks before they are recombined to form larger order ones. Meanwhile, even after the final global optima are found, their replicas in the population are confined within the bounds of the nearest neighbor size q. This scheme presents an effective way to control the genetic drift and dominance of high fitness individuals, and also afford an immune mechanism to encounter replacement errors that may be caused by non-uniform initialization and further amplified by crossover operations.

The parameter q ($q \ll N$) is problem independent, and is decided only by considering available resource as population size. In fact, we might as well take $q = 1$ for default. When q is set as a bigger number, the replacement policy will cover a larger region of the space, and there will be a higher probability for the competition to happen between individuals of different niches. If $q = N$ (the worst deletion scheme in the SSGAs), the global competition will happen certainly, and all of the individuals in the final population are going to adapt to only one niche.

There seems to have a difficulty for the proper choice of the nearest neighbor set size q, the default $q = 1$ may fit in with many cases whatever the multimodal optimization problems are. It would be beneficial to parameterize the niching procedure if we get specific information of a binary multimodal optimization problem. Besides, it can be seen that q is only related to the population size or the resources of the SSGAs. When we want to locate and maintain more niches, a smaller value is preferred; otherwise, a larger one will adapt individuals to fewer niches with much higher computation efficiency.

We use a set of index to describe the characteristics of the proposed niching schemes with the existing crowding methods.

Actually, the crowding-based differential evolution (CDE) (Thomsen 2004) was very similar to the crowding scheme with the proposed nearest neighbors replacement, but it used the differential procedure to produce offspring, which was more exploitive than explorative. It is robust and is able to maintain stable subpopulations adapting to different niches on the real-valued multimodal optimization problems.

For the multi-niche crowding (MNC) (Cedeño and Vemuri 1999), it still allows some competition for the slots among the niches while achieving the mating within members of the same niche. The WAMS procedure for replacement is identical to the proposed nearest neighbors replacement when the crowding factor is one. But if the crowding factor is equal to population size, it becomes the globally worst replacement scheme in the SSGAs. Thus, there still exist errors that an individual in a niche replaces the inferior individuals in another niche. The crowding group size g, and crowding factor f, that affect the performance of the MNC GA, are not easy to decide with a particular multimodal optimization problem.

The A4 algorithm, proposed by Mahfoud (1995) as an extended crowding method by using full sampling ($CF = N$), was also very close to the proposed q-NNR policy, but it did not use the elitist replacement. Experiments showed that the A4 algorithm was unable to niche individuals, and it could only yield random distribution of individuals.

Table 2 Characteristics of the crowding with q-NNR and other crowding methods

Strategies	De Jong's crowding	DC	PC	RTS	MNC	CDE	q-NNR
Crowding replacement:							
– Elitist;		X		X		X	X
– Non-elitist;	X		X		X		
Parental crowding:							
– Deterministic;		X		X	X		
– Probabilistic;			X		X		
Non-parental crowding:							
– Crowding factor $< N$;	X			X	X		
– Crowding factor $= N$;	X					X	X
Competition restriction:							
– Restricted replacement;	X	X	X	X	X	X	X
– Restricted mating;					X		
Control parameters:	CF			w: window size;	s: crowding selection group size; g: crowding group size		q: nearest neighbors set size;
Individuals competition	Global	Global-local				Local	Local
Performances:							
Exploration	High		High	Low	Intermediate	Intermediate	High
Exploitation	Low		Low	High	Intermediate	High	Intermediate
Genetic drift	Yes		Yes	Yes	Yes	No	No
Locating niches	High		High	Low	Intermediate	Intermediate	High
Maintaining niches	Low		Intermediate	High	High	High	High

4.2 A dynamic sizing for q-NNR

In order to increase the efficiency of the q-NNR crowding method and control the proportion of individuals adapted to those niches denoted by local or global optima, we design a Boltzmann scheme for sizing the nearest neighbor set dynamically:

$$q_t = q_{\min} + \left[\frac{e^{\left(\frac{t}{T_{\max}}\right)} - 1}{e - 1}\right]^{\alpha} \times (q_{\max} - q_{\min}), \tag{4.2}$$

where q_t is the size of the nearest neighbor set in t-th iteration. $q_{\min}$, $q_{\max}$ are the minimum size (also the initial value) and the maximum size for the nearest neighbors set in the steady-state evolution process. Usually we take $q_{\min} = 1$ for default. $T_{\max}$ is the maximum iterations of the algorithm. α is the scaling factor, and $\alpha = 1$ for default.

For the binary multimodal optimization problems, the $q_{\max}$ means the maximum number of copies of local or global solutions. Suppose that we want the algorithm to find at least M multimodal solutions with population of size N $(M < N)$, then we have $q_{\max} < N/M$.

With the Boltzmann scheme for sizing the nearest neighbors set, we can use different sizes for the nearest neighbors set to do niching dynamically in the evolution process. A smaller size in the initial phase of iterations is good for the preservation, mixing of lower order building blocks and also the creation of higher order ones. Meanwhile, the number of the duplicates of dominant individuals in the population is controlled within the limit of q, so that the co-adapting process to multiple niches can be improved. This will effectively hold the population free from the replacement errors, and further make the population to stay away from genetic drifts. In the latter phase of iterations, a bigger value for q could accelerate the distributed convergence of individuals among niches in the population. The Boltzmann scheme is a trade-off between the niching quality and efficiency of the q-NNR crowding method.

5 Experiments

This section first describes four binary multimodal functions which have been commonly used in conventional researches. Then, the performance of the proposed crowding method and the DC are tested on these functions, and the experimental results are reported.

5.1 Test functions

(1) TwoMax function

The TwoMax function is defined as (Van Hoyweghen et al. 2002a):

$$f(u) = \left|\frac{L}{2} - u\right| + \frac{L}{2}, \tag{5.1}$$

where u is the unitation of the bits in a chromosome, L is the bit string length. The global optima string are ‘$0 \cdots 0$’ and ‘$1 \cdots 1$’ with the maximum function value of L.

(2) Two-peak fully deceptive function

The commonly used 5-deceptive trap function is defined as (Deb and Goldberg 1992, 1993):

$$f(u) = \begin{cases} \frac{a}{4}(4-u) & \text{if } u \leq 4, \\ b & \text{otherwise,} \end{cases} \tag{5.2}$$

where u is the unitation of a bit string, $a < b$ and $r = a/b$ is the fully-deceptive rate. The global optimum is ‘$1 \cdots 1$’ with function value b, the deceptive attractor is ‘$0 \cdots 0$’ with function value a.

A class of multimodal functions is built as the m-k-deceptive trap function which consists of additively m number of k-deceptive trap functions. Usually, we take $k = 5$, $m = 6$, and get the 6-5-deceptive trap function with a length of 30 bit string.

(3) Bipolar deceptive function

A special 6-bipolar deceptive function is defined as (Goldberg et al. 1992; Deb et al. 1993):

$$f(u)=\begin{cases} b(1-u) & \text{if } u \le 1,\\ \frac{a}{2}(u-1) & \text{otherwise,}\end{cases} \quad \text{while } u \le 3, \tag{5.3}$$

$$f(u)=\begin{cases} \frac{a}{2}(5-u) & \text{if } u \le 5,\\ b(u-5) & \text{otherwise,}\end{cases} \quad \text{while } 6 \ge u > 3. \tag{5.4}$$

There are two global optimum '$1\cdots1$', '$0\cdots0$' with function value b, and 60 deceptive local optima with function value a, and $a < b$.

A class of massively multimodal functions of bounded deception are created by concatenating a number of bipolar deceptive functions together. For instance, the m-k-bipolar deceptive function consists of additively m number of k-bipolar deceptive function. When $k = 6$, $m = 5$, the 5-6-bipolar deceptive function with string length of 30 bit will have 5,153,632 optima, of which there are $2^5 = 32$ global optima with function value b (Goldberg et al. 1992).

(4) The scaled Ising model

The Ising model with one-dimensional nearest-neighbor interaction function takes the form:

$$f : \{-1, 1\}^L \to \Re : x \mapsto \sum_{i=1}^{L} x_i x_{i+1}, \tag{5.5}$$

where L is the string length and $x_{L+1} = x_1$. The two optima are '$1\cdots1$', '$-1\cdots-1$' with the fitness value of L.

The scaled Ising models with pinning effects are defined by Naudts and Naudts (1998), Van Hoyweghen et al. (2001):

$$\begin{aligned} &f : \{0, 1\}^L \to \Re : x \mapsto \sum_{i=1}^{L} \delta(x_i, x_{i+1}),\\ &\delta(x_i, x_{i+1}) = \begin{cases} a & \text{if } i \bmod b = 0 \text{ and } x_i = x_{i+1},\\ 1/2 & \text{if } i \bmod b \ne 0 \text{ and } x_i = x_{i+1},\\ 0 & \text{otherwise,}\end{cases} \end{aligned} \tag{5.6}$$

where L is the string length and $x_{L+1} = x_1$. Actually, the scaled Ising model introduce various degrees of epistasis.

The Ising model contains the spin-flip symmetry, which means that the exchange of '1' and '-1' and vice versa does not improve the fitness. The synchronization leads a simple GA to get stuck in a local optimum far away from the global optima. Van Hoyweghen et al. (2002a, 2002b) proved empirically that the scaled Ising model for some sets of $\{a, b\}$ can hardly be solved accurately by the GAs in 30,000 generations without specially designed crossover and mutation operators. But the GAs with fitness sharing can solve badly scaled Ising models using reasonable small populations (Van Hoyweghen et al. 2001). It was also shown that the non-inferior building blocks is necessary to find an optimum, which can be obtained by using a niching technique as fitness sharing (Van Hoyweghen et al. 2002a).

5.2 The experiments

The experiments and discussions are done individually for each test function. Since the bias in the random generation of the initial population may give rise to difficulties in measuring the behavior of niching methods, we use a preprocessing program to check whether the bias in the initial population is over a threshold. If so, a new initial population is generated once more, which is very important especially in the case where a relatively small population size is used. Here, we employ the χ^2 (confidence 95%) distribution test to check the statistical significance of the initial populations with the unitations of bit-string individuals obeying the Gaussian distribution or not.

The algorithms are run 30 times on each function in the experiments. The statistical data (individual distribution, function evaluations, etc.) are the averaged ones of 30 independent executions with randomly generated initial populations. In order to test the statistical difference of the DC and the q-NNR, the student t-tests are done with confidence of 95% with regard to function evaluations, multimodal solution quality, and distribution.

There is no significant difference between the uniform crossover and the two-point crossover with the q-NNR crowding SSGAs on simple functions as the TwoMax function. For the trap and bipolar deceptive functions, the two point crossover yields better results than the uniform crossover. So we only report the experiments with the two-point crossover.

(1) TwoMax function

A set of bit string length is used as $L = \{60, 80, 100, 120, 140, 160, 180, 200\}$ for the TwoMax function. The size of population is taken as equal to the bit string length $(P = L)$. The q takes default value as $q = 1$, which is the mostly unbiased for niching process. The maximum iterations of the SSGA is set as $T_{\max} = 100P$.

An execution is taken as failing to the niching task for the TwoMax function when the number of individuals adapted to one niche is less than 20% of the total population or the best individuals in each niche are more than 2 bits away from peaks in the final population.

Table 3 shows that the q-NNR crowding and the DC algorithm perform equally well on the TwoMax function. The final populations are distributed among the two niches evenly, and the function evaluations needed to locate the two peaks are very close. The t-test shows no significant difference between two algorithms on either individuals distribution of the final population among niches or function evaluations for achieving convergence. The two niches can be maintained till the end of the evolution process once they are found.

Comparing with the hybrid method of k-means clustering and multi-subpopulation GAs (Pelikan and Goldberg 2000), there isn't a prerequisite for the number of clusters apriori, and also the computation for k-means clustering is exempted.

The dynamics of the q-NNR crowding on the TwoMax function is illustrated in Fig. 2 through six histograms, where the horizontal axis denotes the unitation of bit

Table 3 Results of different niching methods on the TwoMax function

Parameters		Deterministic crowding	Crowding with q-NNR
	(a)	30.2, 29.8, 11.1*;	31.5, 28.5, 9.3*;
$L = 60$	(b)	4112, 5132, 2522*;	4496, 4294, 1888*;
	(c)	100%	100%
	(a)	40.1, 39.0, 12.2*;	37.7, 42.3, 12.8*;
$L = 80$	(b)	7828, 8372, 3218*;	7876, 7144, 3148*;
	(c)	100%	100%
	(a)	47.4, 52.6, 16.5*;	52.2, 47.8, 12.6*;
$L = 100$	(b)	13058, 12164, 3562*;	10112, 11648, 3980*;
	(c)	100%	100%
	(a)	57.3, 62.7, 12.7*;	61.6, 58.4, 15.0*;
$L = 120$	(b)	16974, 15716, 4896*;	14816, 17308, 4816*;
	(c)	100%	100%
	(a)	70.2, 69.8, 10.8*;	72.6, 67.4, 16.3*;
$L = 140$	(b)	23698, 22482, 5488*;	18984, 21580, 5746*;
	(c)	100%	100%
	(a)	79.9, 80.1, 16.6*;	81.1, 78.9, 21.4*;
$L = 160$	(b)	26042, 24732, 4988*;	22774, 26800, 6558*;
	(c)	100%	100%
	(a)	90.4, 89.6, 25.6*;	88.9, 91.08, 17.7*;
$L = 180$	(b)	32270, 32560, 6780*	29498, 28276, 6566*;
	(c)	100%	100%
	(a)	99.0, 101, 23.0*;	104, 96, 19.5*;
$L = 200$	(b)	30896, 38268, 7372*	38316, 30318, 7926*;
	(c)	100%	100%

(a) Individuals distribution: the first figure and the second figure indicate the number of individuals adapted to two niches (‘$0 \cdots 0$’, ‘$1 \cdots 1$’) separately; the asterisked denotes the average difference of the number of individuals adapted to two niches

(b) Function evaluations: the first figure and the second figure indicate the evaluations for finding peaks in two niches separately; the asterisked denotes the average difference of the evaluations for finding peaks in two niches

(c) Success rate (%): the percentage of runs where two niches are found in the final population in total runs; an execution fails on the TwoMax function when either of the two optima {‘$0 \cdots 0$’, ‘$1 \cdots 1$’} is not in the final population

strings, the vertical axis represents the number of individuals adapted to two niches. These histograms show the states of individuals distribution after different number of function evaluations. The two optima are on the leftist end (‘$0 \cdots 0$’) and the rightist end (‘$1 \cdots 1$’) of the horizontal axis in histograms. In order to get a clear picture of the dynamics of the q-NNR in terms of distributed convergence, we set $q_{\max} = N$ after $t > 0.30T_{\max}$, where t is the iteration.

Figure 2 shows that the initial population is distributed around the center of bit unitation $(L/2)$. As the q-NNR crowding SSGAs proceeds, the implicit niching mechanism splits the population into two separate parts which are pushed gradually to the

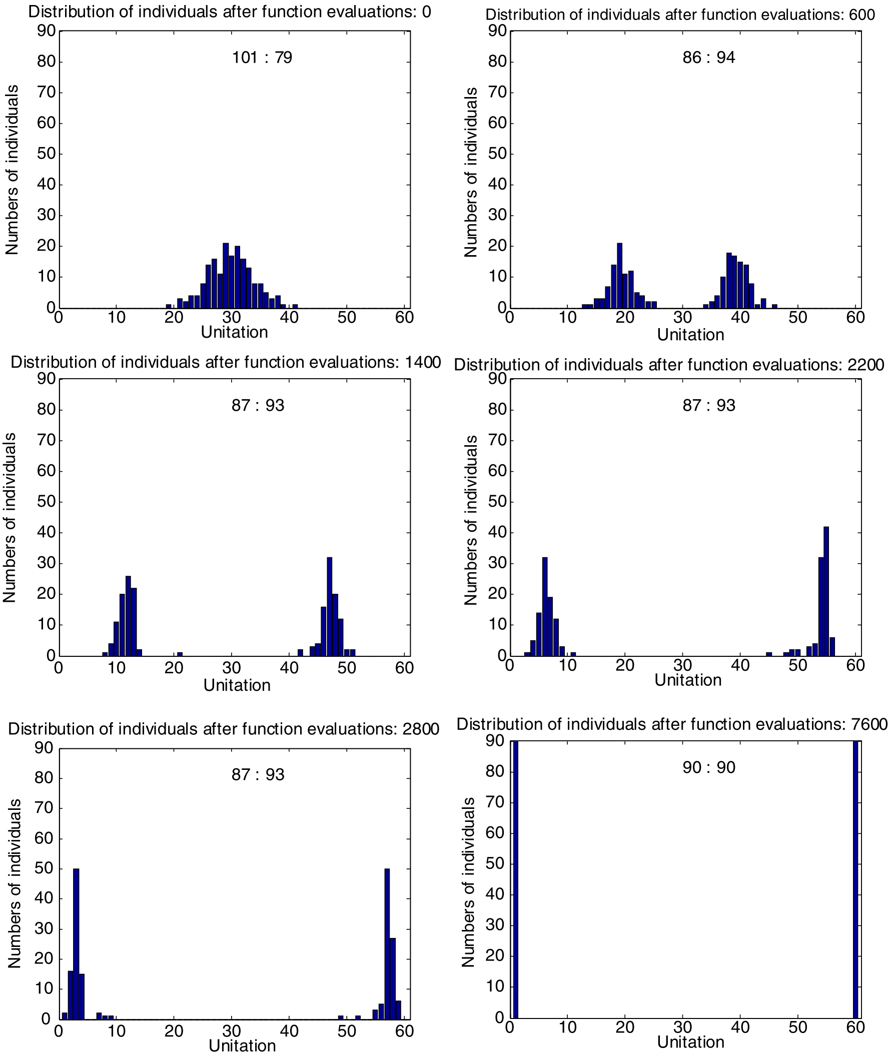


Fig. 2 The dynamics of the q-NNR crowding on the TwoMax function (string length: 60, population size: 180, maximum function evaluations: 36,000)

extreme ends of the horizontal axis. Since the TwoMax has two niches of equal fitness and attraction basins, the average numbers of individuals adapted to each niche are very close in the evolution process and in the final population as well.

In order to test the effects of the nearest neighbors size on the performance of the q-NNR crowding scheme, we take various values for q as $q \in \{1, 3, 5, 7, \ldots, 0.5L\}$. The parameters for the SSGAs are set as $L = 60$, $P = 120$, $T_{\max} = 50P$.

Table 4 shows that we can accelerate the niching process by using a bigger q, but the replacement error may lead to a higher probability of losing one niche in prescribed function evaluations. And more, the distribution of individuals among the

Table 4 Niching effects of the q value on the TwoMax function ($L = 60$)

Parameters	Crowding with q-NNR		
	(a)	(b)	(c)
$q = 1$	59.4, 60.6, 11.2*	6556, 6716, 1260*	100
$q = 3$	61.3, 58.7, 14.5*	4934, 5024, 1764*	100
$q = 5$	59.1, 60.9, 17.7*	4344, 4806, 2032*	90.0
$q = 7$	57.0, 63.0, 19.5*	5056, 4390, 2236*	83.3
$q = 9$	60.1, 59.9, 25.9*	4806, 4638, 2562*	56.7
$q = 15$	53.8, 66.2, 30.4*	4674, 4060, 1588*	16.7
$q = 20$	65.3, 54.7, 38.3*	3348, 5258, 1910*	6.7
$q = 25$	58.8, 61.2, 34.3*		0
$q = 30$	60.6, 59.4, 34.8*		0

(a) Individuals distribution: the first figure and the second figure indicate the number of individuals adapted to two niches (‘$0 \cdots 0$’, ‘$1 \cdots 1$’) separately; the asterisked denotes the average difference of the number of individuals adapted to two niches

(b) Function evaluations: the first figure and the second figure indicate the evaluations for finding peaks in two niches separately; the asterisked denotes the average difference of the evaluations for finding peaks in two niches

(c) Success rate (%): the percentage of runs where two niches are found in the final population in total runs; an execution fails on the TwoMax function when either of the two optima {‘$0 \cdots 0$’, ‘$1 \cdots 1$’} is not in the final population

two niches becomes much more unevenly. It is evident that the size of the nearest neighbors set has an important affection on the performance of the q-NNR crowding scheme.

In order to test the Boltzmann scheme sizing of the nearest neighbors set on the performance of the q-NNR crowding, we conduct experiments with dynamical nearest neighbors sizes (see Table 5), where $L = 60$, $P = 120$, $T_{\max} = 50P$, $q_{\min} = 1$.

The Boltzmann scheme sizing of the nearest neighbors set is beneficial to increasing greatly the success rate of the q-NNR crowding on locating both niche peaks, and the function evaluations required are decreased by one third. So the Boltzmann sizing of dynamic q values provides a nice way to raise the efficiency while maintaining a good niching capability.

(2) 6-5-deceptive trap function

A length of 30 bit string ($a = 0.8$, $b = 1$) is used in experiments on the 6-5-deceptive trap function. The critical parameters are set as $L = 30$, $P = \{2L, 3L, 4L, 5L\}$, $T_{\max} = 100P$, $q = 1$, and $\{q_{\min} = 1, q_{\max} = 5\}$ for dynamic sizing of the nearest neighbors set. The experimental results of both the q-NNR crowding and the DC are displayed in Table 6.

The performance of the tested crowding methods rely on the population size on the trap function, which shows that the global optimum can hardly be recombined from individuals at bitwise diversity when a smaller population size is used. In contrast, the diversity of BBs-wise is crucial to the successful creation of the global optimum ‘$1 \cdots 1$’ which can only be achieved with a lager population size. To make

Table 5 Niching with dynamical nearest neighbors sizes on the TwoMax function ($L = 60$)

Parameters	Crowding with q-NNR		
	(a)	(b)	(c)
$q_{\max} = 1$	59.4, 60.6, 11.2*	6556, 6716, 1260*	100
$q_{\max} = 3$	59.5, 60.5, 12.2*	6359, 6667, 1459*	100
$q_{\max} = 5$	61.9, 58.1, 11.7*	5260, 5794, 1492*	100
$q_{\max} = 7$	61.7, 58.3, 12.7*	4734, 5262, 1236*	100
$q_{\max} = 9$	60.9, 59.1, 12.5*	4740, 5096, 1418*	100
$q_{\max} = 15$	61.4, 58.6, 13.8*	4424, 4482, 1380*	100
$q_{\max} = 20$	57.9, 62.1, 9.7*	4694, 4170, 1038*	96.7
$q_{\max} = 25$	57.7, 62.3, 13.2*	4698, 4396, 1648*	90.0
$q_{\max} = 30$	59.2, 60.8, 3.5*	4282, 4592, 1498*	83.3

(a) Individuals distribution: the first figure and the second figure indicate the number of individuals adapted to two niches (‘$0\cdots0$’, ‘$1\cdots1$’) separately; the asterisked denotes the average difference of the number of individuals adapted to two niches

(b) Function evaluations: the first figure and the second figure indicate the evaluations for finding peaks in two niches separately; the asterisked denotes the average difference of the evaluations for finding peaks in two niches

(c) Success rate (%): the percentage of runs where two niches are found in the final population in total runs; an execution fails on the TwoMax function when either of the two optima {‘$0\cdots0$’, ‘$1\cdots1$’} is not in the final population

Table 6 Niching with crowding methods on the 6-5-deceptive trap function

Population sizes	Success rate* (%)		
	Deterministic crowding	q-NNR crowding ($q = 1$)	Dynamic q-NNR crowding ($q_{\max} = 5$)
$P = 30$	23.3	23.3	16.7
$P = 60$	66.7	63.3	60.0
$P = 90$	93.3	90.0	86.7
$P = 120$	96.7	100	96.7
$P = 150$	100	100	100

Success rate (%): an execution fails on the trap functions when the optima (‘$1\cdots1$’) is not in the final population

this clearer, let’s watch the changing process of the located build blocks of different orders.

For the 6-5-deceptive trap function, there are $2^6 = 64$ optima, of which the global optimum is ‘$11111\cdots11111$’ with six building blocks of ‘11111’ juxtaposed together. The local optima include solutions with building blocks of ‘11111’ fewer than six. For simplicity, we use the number of the elementary building blocks ‘11111’ to denote different orders of building blocks as $\{5, 10, 15, 20, 25, 30\}$. Figure 3 shows the changes of the mean numbers of different orders of building blocks with the number of function evaluations.

Figure 3 indicates the process of BBs-wise formation and preservation, and the creation of high order BBs. The evolution process starts with the initial population, where there is a few of building block '11111'. Along with the iterations of the GAs, the basic building block of '11111' are created more and more, and takes an increasing proportion in the population. Then the order-10 building blocks are created from these basic ones. The process continues by mixing lower order building blocks to create higher order ones until the global optimum '$11111 \cdots 11111$' is formed.

With the deterministic crowding method, the proportion of lower building blocks decreases along with the increment of higher order ones, and the final population consists of the global optimum and the order-25 building blocks, which illustrates that it is not able to maintain all local optima or niches because of the global competition.

On the contray, the q-NNR crowding with $q = 1$ is good at confining the competition between individuals as totally local, and limiting the replacement of less fitted individuals by dominant ones in the population through the nearest neighbors size. It can locate the multiple optima or niches and maintain them forever in the population. Meanwhile, the BBs-wise diversity makes it possible for the SSGAs to create higher building blocks from lower ones steadily. Actually, there is only a single copy of the global optimum in the final population.

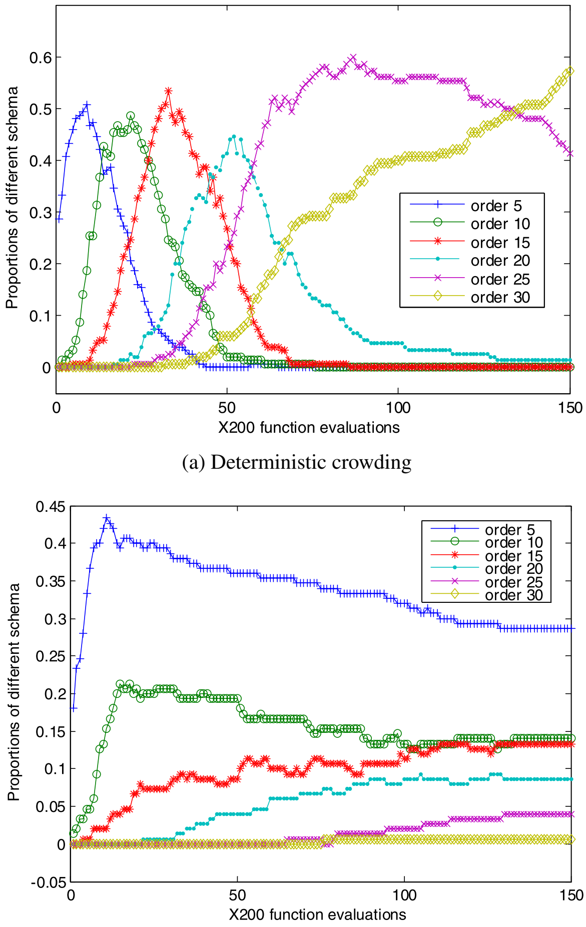
The q-NNR crowding with $q_{\max} > 1$ maintains fewer niches in the final populations while $q_{\max}$ gets bigger, but fewer function evaluations are required. Since the q-NNR crowding allows the competition between neighbors of building blocks when $q_{\max} \geq 5$, the order-5 and order-10 building blocks or niches disappear in the final population. The individuals belonging to order-5 building blocks are replaced by order-10 building blocks, and the later will be replaced by order-15 building blocks. There are 15-order, 20-order, 25-order and order-30 building blocks in the final population when the population is evolved further. The order-30 building blocks, $q_{\max}$ replicas at maximum, can not take over the population because of the neighbors size limit.

The q-NNR is able to preserve the BBs of different orders even when higher order BBs or the global optima are created in the population, while the DC fails to preserve the BBs of lower order and to prevent the global optima from taking over the population eventually.

(3) 5-6-bipolar deceptive function

We use the 30 bit string ($a = 0.8$, $b = 1$) for the 5-6-bipolar deceptive function, where the $2^5 = 32$ global optima have function values as 5. We choose the main parameters as $P = \{5L, 10L, 15L\}$, $T_{\max} = 100P, q = 1$, and $\{q_{\min} = 1, q_{\max} = 3\}$ for dynamic sizing of the nearest neighbor set. Table 7 shows the results of the q-NNR crowding and the DC.

In Table 7, there are significant differences in terms of the global optima found between the q-NNR crowding and the DC by the t-test. The DC needs a much bigger population to find all of the global optima because of the replacement errors and genetic drift. Meanwhile there are almost only replicas of the global optima in the final population, and each of the global optima has different number of replicas. Although the building blocks of different orders constitute the stepping stones to creating the global ones, they are also replaced in the evolution process in the context of the deterministic crowding. This gives rise to the rare chances to create some of the global



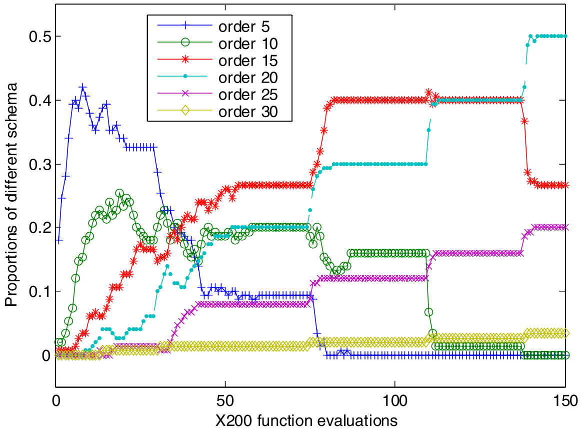
(a) Deterministic crowding

(b) q-NNR crowding ($q = 1$)

Fig. 3 The proportion of different order building blocks in population with function evaluations with three crowding methods on the 6-5-deceptive trap function (bit string length: 30, population size: 150, maximum evaluations: 30,000)

optima. However, the q-NNR crowding and dynamic q-NNR crowding techniques are both more efficient (see Table 7, Fig. 4), and they are able to maintain the building blocks of medium orders that are also niches with lower fitness (see Fig. 5). The limited sizes of nearest neighbors lead to the impossibility for the global optima or any other higher fitness niches to take over the population, which results in the equilibria of all niches and the much more even distribution of individuals adapted to niches of either global or local optima.

Figure 5a indicates the number of replicates of the global optima. The horizontal axis represents the 32 global optimum denoted by the unitation of binary strings of the optima. The curves show that the number of the replicates of all of the global optima



(c) Dynamic q-NNR crowding ($q_{\max} = 5$)

Fig. 3 (*Continued*)

Table 7 Niching with different crowding methods on the 5-6-bipolar deceptive function

Niching methods	Population sizes	Global optima found (%)	Average number and deviation of replicas for each of the global optima	
			Average number	Deviation
Deterministic crowding	$P = 150$	46.7	4.69	10.37
	$P = 300$	96.7	9.34	6.64
	$P = 450$	96.7	14.06	11.36
q-NNR crowding ($q = 1$)	$P = 150$	100	1	0
	$P = 300$	100	1	0
Dynamic q-NNR crowding ($q_{\max} = 3$)	$P = 150$	93.3	2.32	0.26
	$P = 300$	100	2.99	0.02

found by q-NNR are the same and equal to $q_{\max}$. But in the converged population of the DC, the number of the replicates of the global optima are different.

Figure 5b shows that the converged population by the q-NNR has a good diversity, the local optima (with different number of BBs) are still preserved in the population. In contrast, the global optima almost take over the population with the DC.

In the experiments with fitness sharing and deterministic crowding in the context of generational GAs, much bigger population sizes were used. The fitness sharing based niching exploited a population of 5,000 individuals with power sharing ($\beta = 15$), and all peaks were located in 200 generations and maintained afterwards (Goldberg et al. 1992). The adaptive niching by using coevolutionary sharing employed two sets of 2,000 and 35 individuals for the customer population and businessman population, the total number of global optima were found in 400 generations through the imprint technique (Goldberg and Wang 1997). The deterministic crowd-

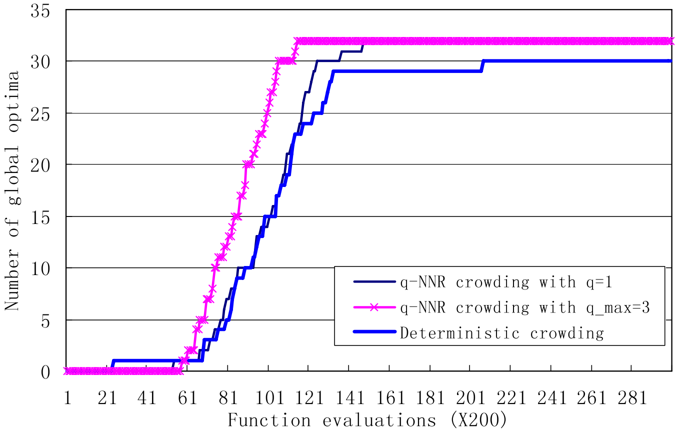


Fig. 4 The numbers of global optima found with function evaluations on the 5-6-bipolar deceptive function (bit string length: 30, population size: 300, maximum evaluations: 60,000)

ing adopted a population of 2,000 individuals in locating the 32 global optima in 160 generations, and the final population converged to the global optima unevenly, which could be maintained forever (Mahfoud 1995).

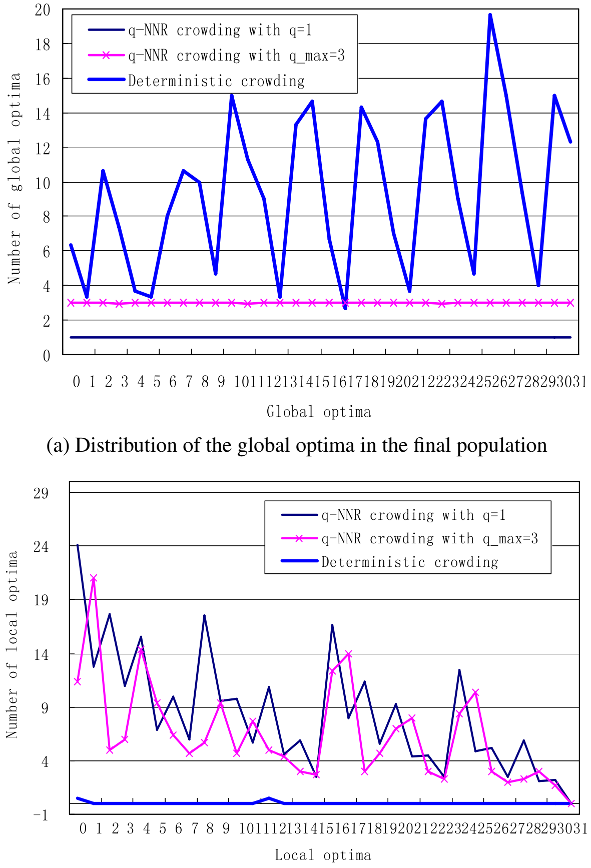
Compared with these attempts on the 5-6-bipolar deceptive function, the DC and q-NNR crowding methods in our experiments are much more efficient. This may be partly owed to the steady-state executions of GAs, which could be able to avoid the generational replacement errors. Further experiments are needed to verified the difference of performances of the DC in the paradigm of the generational GAs and the SSGAs separately.

(4) The scaled Ising model

In contrast to the aim of finding a global optimum of the Ising model (Naudts and Naudts 1998; Van Hoyweghen et al. 2002a), we try to take it as a binary multimodal function and find the two peaks in one population, which constitutes a tough task for niching methods. We consider the unscaled and the scaled Ising model with a bit string length of 80 in the SSGAs. The scaled parameters are taken as four groups $\{a = 3, b = 4\}$, $\{a = 3, b = 6\}$, $\{a = 4, b = 7\}$, $\{a = 4, b = 8\}$, which are identified empirically as hard problems in conventional researches (Naudts and Naudts 1998).

The main experiment parameters are set as $P = \{3L, 4L, 5L\}$, $T_{\max} = 100P$, $q = 1$, and $\{q_{\min} = 1, q_{\max} = 5\}$ for dynamic sizing of the nearest neighbors set. The mean results of both the q-NNR crowding and the DC are displayed in Tables 8 and 9. The success rate is defined as the percentage of runs where both of the two global optima are located.

Tables 8 and 9 show clearly that the q-NNR crowding and its dynamic scheme achieve good success rates when regular population sizes are used. In contrast, the DC does not always locate both of the global optima in the populations; otherwise a much bigger population size is required. The t-test illustrates that there are significant differences between the q-NNR crowding and the DC with regard to the success rate on the scaled Ising models.



(a) Distribution of the global optima in the final population

(b) Distribution of the local optima in the final population

Fig. 5 The distribution of the global optima and local optima in the final population on the 5-6-bipolar deceptive function (bit string length: 30, population size: 300, maximum evaluations: 60,000)

Figures 6 and 7 provide more evidence about the dynamics of two crowding methods. The histograms illustrate that the individual distributions with number of function evaluations evolve in parallel regarding the two global niches with the q-NNR crowding. However, the DC often leads to genetic drift that causes the population to move to only one global niche (see Fig. 7).

These experiments reveal that the preservation of building blocks of medium orders is crucial for the creation and mixing of higher building blocks and the parallel evaluation of niches of either local or global optima for the binary multimodal fitness landscape. The q-NNR crowding is able to effectively localize the competition of individuals by its special replacement policy. Meanwhile, the scaled Ising model is an interesting problem of binary multimodal optimization for testing niching methods, and it is hoped that more researchers would use it as a benchmark problem to evaluate various niching methods hereafter.

Table 8 Niching with crowding methods on the Ising models

Niching methods	Population sizes	Success rate (%)	Average number of individuals in niches of global optima	
			Optimum: '$0\cdots0$'	Optimum: '$1\cdots1$'
Deterministic crowding	$P = 160$	0	65.0	95.0
	$P = 240$	20.0	95.5	144.5
q-NNR crowding ($q = 1$)	$P = 160$	40.0	81.5	78.5
	$P = 240$	100	116.0	124.0
Dynamic q-NNR	$P = 160$	36.7	77.2	82.8
crowding ($q_{\max} = 5$)	$P = 240$	100.0	105.5	134.5

Table 9 Niching with crowding methods on the scaled Ising models

Niching methods	Population sizes	Success rate (%)			
		$\{a = 3, b = 4\}$	$\{a = 3, b = 6\}$	$\{a = 4, b = 7\}$	$\{a = 4, b = 8\}$
Deterministic crowding	$P = 240$	16.7	20.0	16.7	16.7
	$P = 320$	26.7	36.7	33.3	26.7
	$P = 400$	73.3	60.0	73.3	66.7
q-NNR crowding ($q = 1$)	$P = 240$	73.3	63.3	76.7	63.3
	$P = 320$	100	100	100	100
Dynamic q-NNR	$P = 240$	76.7	83.3	86.7	80.0
crowding ($q_{\max} = 5$)	$P = 320$	100	100	100	100

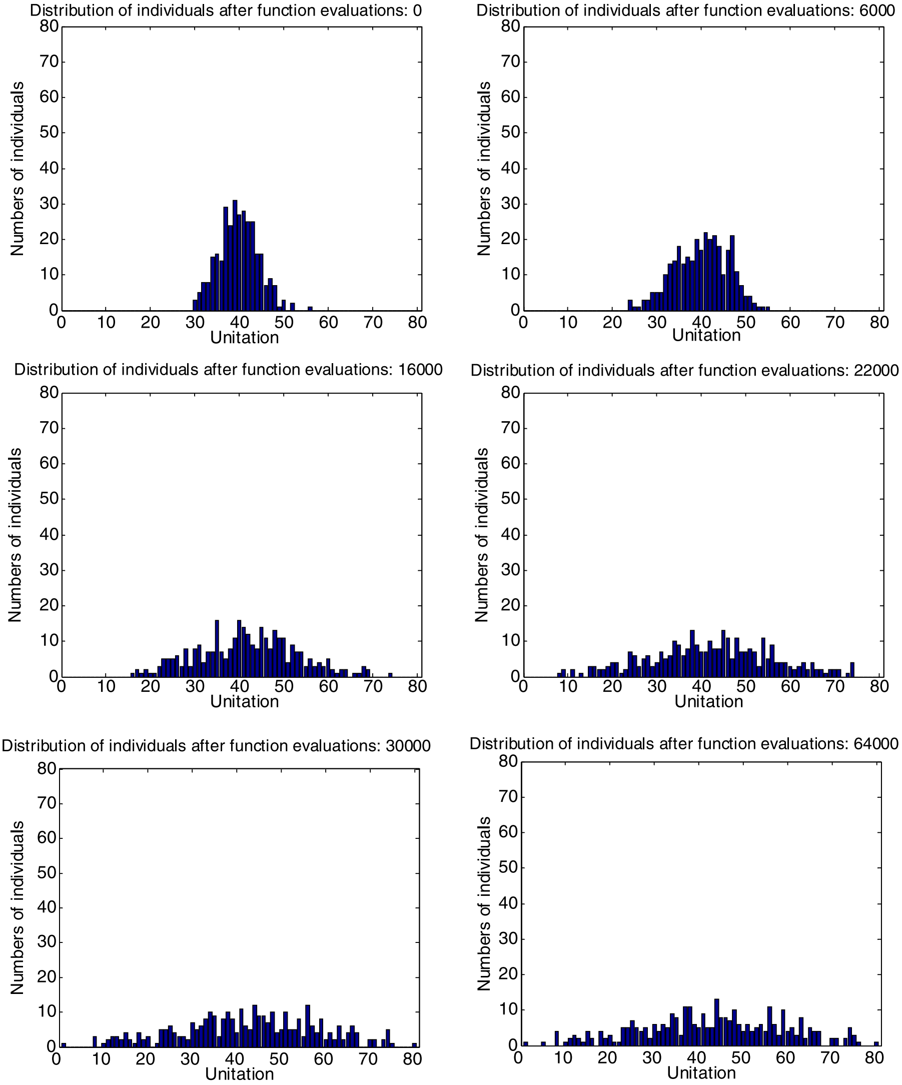
Since it is the first time that the Ising model is solved as a multimodal optimization problem, we do not make a comparison with the conventional researches.

5.3 Discussions

The experiments illustrate that the q-NNR crowding SSGAs is a simple, robust, and effective niching method on many hard problems as the trap function, the bipolar function, the scaled Ising model, etc. We just compared the performance of the q-NNR and the DC mainly because both of them do not have particular parameters to decide with the apriori information of the problems, e.g. the number of niches, the distance of niches, the radius of niches and so on. They are both problem independent.

We tested the probability crowding method on all functions, and found that it was only elitist in probability, and the evolution needed a quite long time to find the multimodal global optima, especially on the trap function, the bipolar deceptive function, the Ising model. So, we did not think it was fitted to the binary multimodal optimization problems.

The fitness sharing techniques rely on the niche radius to realize the coevolution of multiple species to different niches. A smaller niche radius is good for maintaining multiple species in the population, but the distributed convergence is quite slow.



Distribution of individuals after function evaluations: 0

Distribution of individuals after function evaluations: 6000

Distribution of individuals after function evaluations: 16000

Distribution of individuals after function evaluations: 22000

Distribution of individuals after function evaluations: 30000

Distribution of individuals after function evaluations: 64000

Fig. 6 Individuals distribution and the successful locating of both global optima in population on the scaled Ising model by niching with the q-NNR crowding ($q = 1$) (bit string length: 80, population size: 320, maximum evaluations: 64,000, $\{a = 3, b = 4\}$)

A bigger niche radius is much more efficient, but only fewer niches can be maintained in the final population. A big population size is usually used to overcome the randomness of generational-GA (Goldberg and Richardson 1987; Sareni and Krähenbühl 1998; Cioppa et al. 2004). When properly parameterized, the fitness sharing scheme works very well on the binary functions.

Since there are particular parameters to decide for other explicit niching methods with regard to specific problems, we did not do experiments with these niching methods. They are usually able to find binary multimodal optima when the parameters are

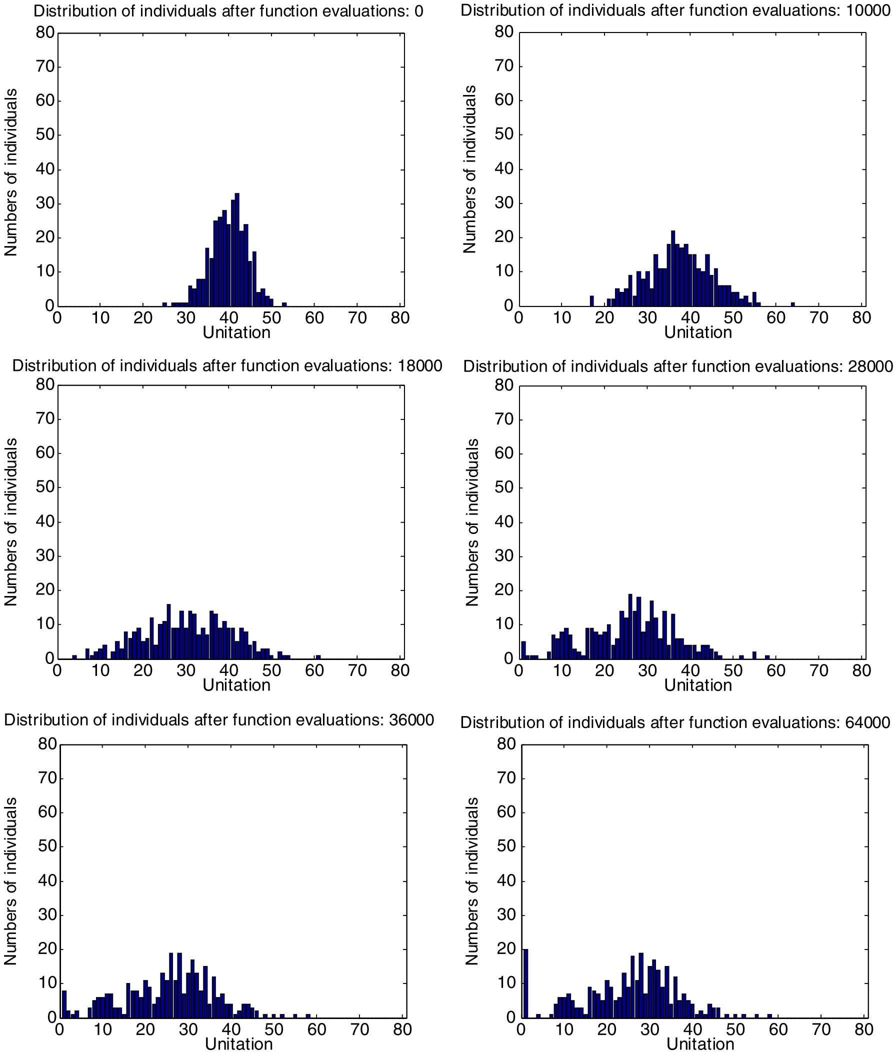


Fig. 7 Individuals distribution and the unsuccessful locating of both global optima in population on the scaled Ising model by niching with the DC (bit string length: 80, population size: 320, maximum evaluations: 64,000, $\{a = 3, b = 4\}$)

correctly set (Yin and Germany 1993; Lin and Yang 1998; Pétrowski 1996; Pelikan and Goldberg 2000; Siarry et al. 2000; Jelasity et al. 2001; Li et al. 2002; Sastry et al. 2005).

The q-NNR SSGA is a general-purpose technique for implicit niching. In case that we have a good knowledge of the multimodal optimization problems, and we only attempt to get find multimodal solutions with specific meaning, we might as well try various hybrid schemes of implicit niching and explicit niching.

6 Conclusions

The main contribution of this paper is the introduction and experimental evaluation of the nearest neighbors replacement crowding in the recombination-replacement paradigm of the steady-state GAs. The peculiarity of the proposed method in comparison with the existing niching techniques is that it is able to realize multiple species niching by effectively preserving the BBs-wise diversity, which is paramount for the GAs in solving binary multimodal functions. In terms of genetic drift usually happening in the evolving population, the q-NNR crowding scheme creates an effective mechanism to localize the individual competition, which makes it possible for the SSGAs to search multiple niches simultaneously. Regarding the speed-up of the efficiency of the proposed method, we design the dynamic sizing of the nearest neighbors which makes a good way to reduce function evaluations required to locate multiple optima. When the proposed method is applied to solve a set of difficulty binary multimodal functions with deceptiveness, symmetry, and multimodality, satisfactory experimental results are obtained. Another beneficial feature of the q-NNR crowding is that it is a parameterless niching method, easy to manipulate and efficient to implement. It has a good scalability to large binary multimodal optimization problems.

The integration of the proposed crowding algorithm with the fitness sharing is also an interesting research direction because the former can be used to find the proper niching parameters for the latter. In view of speeding up the proposed niching algorithm for complicated binary multimodal optimization problems when a large population is required, the parallel implementation scheme is a necessary approach to reduce the computation time of the proposed crowding method. Preliminary results look quite promising.

Acknowledgements The work was done when the first author was doing visiting research in the Illinois Genetic Algorithms Laboratory, Department of General Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA. The authors would like to thank Professor David E. Goldberg for valuable advises and evaluations that helped shape the paper; and thank Dr. Kumara Sastry, Dr. Xavier Llorà, and Dr. Tian-Li Yu for helpful discussions. The work was supported by the National Science Foundation of China (Grant Nos. 70171002, 70571057) and by the Program for New Century Excellent Talents in Universities of China (NCET-05-0253).

The authors are very grateful to the two anonymous reviewers whose invaluable comments and suggestions helped in improving the quality of the paper.

References

Bäck, T., Fogel, D.B., Michalewicz, Z.: Evolutionary Computation. Institute of Physics, Bristol (2000)

Beasley, D., Bull, D.R., Martin, R.R.: A sequential niche technique for multimodal function optimization. Evol. Comput. **1**(2), 10–125 (1993)

Cedeño, W., Vemuri, V.: Analysis of speciation and niching in multi-niche crowding genetic algorithms. Theor. Comput. Sci. **222**(1–2), 177–197 (1999)

Cioppa, A.D., Stefano, C., De, Marcelli, A.: On the role of population size and niche radius in fitness sharing. IEEE Trans. Evol. Comput. **8**(6), 580–592 (2004)

Davidor, Y.: Epistasis variance: a viewpoint on GA-hardness. In: Rawlins, G.J.E. (ed.) Foundations of Genetic Algorithms, pp. 23–35. Morgan Kaufmann, San Mateo (1991)

Deb, K., Goldberg, D.E.: 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. George Mason University/Morgan Kaufmann, San Mateo (1989)

Deb, K., Goldberg, D.E.: Analyzing deception in trap functions. In: Whitley, L.D. (ed.) Foundations of Genetic Algorithms, 2, pp. 93–108 (1992). Also IlliGAL Report No. 91009

Deb, K., Goldberg, D.E.: Sufficient conditions for deceptive and easy binary functions. Ann. Math. Artif. Intell. **10**(4), 385–408 (1993)

Deb, K., Horn, J., Goldberg, D.E.: Multimodal deceptive functions. Complex Syst. **7**, 131–153 (1993). Also IlliGAL Report No. 92003

De Jong, K.A.: An analysis of the behavior of a class of genetic adaptive systems. Doctoral Dissertation, University of Michigan, MI: Ann Arbor (1975). University Microfilms No. 76-9381

Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Berlin (2003)

Forrest, S., Smith, R.E., Javornik, B., Perelson, A.S.: Using genetic algorithms to explore pattern recognition in the immune system. Evol. Comput. **1**(3), 191–211 (1993)

Gan, J., Warwick, K.: A variable radius niching technique for speciation in genetic algorithms. In: Whitley, D., et al. (eds.) Proceedings of the Genetic and Evolutionary Computational Conference (GECCO2000), pp. 96–103. Morgan Kaufmann, San Mateo (2000)

Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison–Wesley, New York (1989)

Goldberg, D.E.: Design of Innovation: Lessons from and for Competent Genetic Algorithms. Kluwer Academic, Boston (2002)

Goldberg, D.E., Richardson, J.J.: Genetic algorithms with sharing for multimodal function optimization. In: Grefenstette, J. (ed.) Proceedings of the Second International Conference on Genetic Algorithms (ICGA 2nd), pp. 41–49. Lawrence Erlbaum Associates, Cambridge (1987)

Goldberg, D.E., Wang, L.: Adaptive niching via coevolutionary sharing. In: Quagliarella, D., Periaux, J., Poloni, C., Winter, G. (eds.) Genetic Algorithms in Engineering and Computer Science, pp. 21–38. Wiley, Chichester (1997). Also IlliGAL Report No. 97007

Goldberg, D.E., Deb, K., Korb, B.: Messy genetic algorithms revisited: studies in mixed size and scale. Complex Syst. **4**, 415–444 (1990)

Goldberg, D.E., Deb, K., Horn, J.: Massive multimodality, deception, and genetic algorithms. In: Manner, R., Manderick, B. (eds.) Parallel Problem Solving from Nature, 2, pp. 37–46. North-Holland, Amsterdam (1992). Also IlliGAL Report No. 92007

Gudla, P.K., Ganguli, R.: An automated hybrid genetic-conjugate gradient algorithm for multimodal optimization problems. Appl. Math. Comput. **167**(2), 1457–1474 (2005)

Harik, G.R.: Finding multimodal solutions using restricted tournament selection. In: Eshelman, L.J. (ed.) Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 24–31. Morgan Kaufmann, San Mateo (1995). Also IlliGAL Report No. 94002

Horn, J.: The nature of niching: genetic algorithms and the evolution of optimal, cooperative populations. Doctoral Dissertation, University of Illinois at Urbana-Champaign, Urbana, IL61801 (1997). Also IlliGAL Report No. 97008

Jelasity, M., Ortigosa, P.M., Garcia, I.: UEGO, an abstract clustering technique for multimodal global optimization. J. Heuristics **7**(3), 215–233 (2001)

Li, M., Kou, J.: A novel type of niching methods based on steady-state genetic algorithm. In: Wang, L., Chen, K., Ong, Y.S. (eds.) Advances in Natural Computation: First International Conference on Natural Computation (ICNC 2005). Lecture Notes in Computer Science, vol. 3612/2005, pp. 37–47. Springer, Berlin (2005)

Li, J.-P., Balazs, M.E., Parks, G.T., Clarkson, P.J.: A species conserving genetic algorithm for multimodal function optimization. Evol. Comput. **10**(3), 207–234 (2002)

Lin, C.-Y., Yang, Y.-J.: Cluster identification techniques in genetic algorithms for multimodal optimization. Comput. Aided Civ. Infrastruct. Eng. **13**(1), 53–62 (1998)

Mahfoud, S.W.: Crowding and preselection revisited. In: Manner, R., Manderick, B. (eds.) Parallel Problem Solving from Nature, 2, pp. 27–36. Elsevier, Amsterdam (1992). Also IlliGAL Report No. 92004

Mahfoud, S.W.: Niching methods for genetic algorithms. Doctoral Dissertation, University of Illinois at Urbana-Champaign, Urbana, IL61801 (1995). Also IlliGAL Report No. 95001

Mengshoel, O.J., Goldberg, D.E.: Probabilistic crowding: deterministic crowding with probabilistic replacement. IlliGAL Report No. 99004 (1999)

Miller, B.L., Shaw, M.J.: Genetic algorithms with dynamic niche sharing for multimodal function optimization. In: Grefenstette, J.J. (ed.) International Conference on Evolutionary Computation, pp. 786–791. IEEE Press, Piscataway (1996). Also IlliGAL Report No. 95010

Naudts, B., Naudts, J.: The effect of spin-flip symmetry on the performance of the simple GA. In: Eiben, A.E. et al. (eds.) Proceedings of the Fifth Conference on Parallel Problem Solving from Nature. Lecture Notes in Computer Science, vol. 1498, pp. 67–76 (1998)

Parzen, E.: On estimation of a probability density function and mode. Ann. Math. Stat. **33**, 1065–1076 (1962)

Pétrowski, A.: A clearing procedure as a niching method for genetic algorithms. In: Grefenstette, J.J. (ed.) Proceedings of 1996 IEEE International Conference on Evolutionary Computation, pp. 798–803. IEEE Press, Piscataway (1996)

Pelikan, M., Goldberg, D.E.: Genetic algorithms, clustering, and the breaking of symmetry. In: Schoenauer, M. et al. (eds.) Proceedings of Parallel Problem Solving from Nature VI, pp. 385–394. Springer, Berlin (2000). Also IlliGAL Report No. 2000013

Peña, J.M., Lozano, J.A., Larrañaga, P.: Globally multimodal problem optimization via an estimation of distribution algorithm based on unsupervised learning of Bayesian networks. Evol. Comput. **13**(1), 43–66 (2005)

Reeves, C., Wright, C.C.: Epistasis in genetic algorithms: an experimental design perspective. In: Eshelman, L. (ed.) Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 217–224. Morgan Kaufmann, San Francisco (1995)

Sareni, B., Krähenbühl, L.: Fitness sharing and niching methods revisited. IEEE Trans. Evol. Comput. **2**(3), 97–106 (1998)

Sastry, K., Abbass, H.A., Goldberg, D.E., Johnson, D.D.: Sub-structural niching in estimation of distribution algorithms. In: Beyer, H.-G., O'Reilly, U.-M. (eds.) Proceedings of Genetic and Evolutionary Computation Conference, 2005 (GECCO2005), June 25–29, pp. 671–678. ACM Press, Washington (2005). Also IlliGAL Report No. 2005003

Siarry, P., Pétrowski, A., Bessaou, M.: Island model cooperating with speciation for multimodal optimization. In: Schoenauer, M. et al. (eds.) Proceedings of 6th International Conference on Parallel Problem Solving from Nature (PPSN-VI), Parallel Problem Solving from Nature, pp. 437–446. Springer, Paris (2000)

Siarry, P., Pétrowski, A., Bessaou, M.: A multipopulation genetic algorithm aimed at multimodal optimization. Adv. Eng. Softw. **33**(4), 207–213 (2002)

Syswerda, G.: A study of reproduction in generational and steady state Genetic Algorithms. In: Rawlings, G.J.E. (eds.) Foundations of Genetic Algorithms (FOGA 1), pp. 94–101. Morgan Kaufmann, San Mateo (1991)

Thomsen, R.: Multimodal optimization using crowding-based differential evolution. In: Proceedings of Congress on Evolutionary Computation, 2004 (CEC2004), vol. 2, June 19–23, pp. 1382–1389. IEEE Press (2004)

Van Hoyweghen, C., Goldberg, D.E., Naudts, B.: Building block superiority, multimodality and synchronization problems. In: Spector, L., et al. (ed.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001), pp. 694–701. Morgan Kaufmann, San Francisco (2001). Also IlliGAL Report No. 2001020

Van Hoyweghen, C., Goldberg, D.E., Naudts, B.: From TwoMax to the Ising model: easy and hard symmetrical problems. In: Langdon, W., et al. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2002), pp. 626–633. Morgan Kaufmann, San Francisco (2002a). Also IlliGAL Report No. 2001030

Van Hoyweghen, C., Naudts, B., Goldberg, D.E.: Spin-flip symmetry and synchronization. Evol. Comput. **10**(4), 317–344 (2002b)

Yin, X., Germany, N.: A fast algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. In: Albrecht, R.F., Reeves, C.R., Steel, N.C. (eds.) Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms, pp. 450–457. Springer, Berlin (1993)