

A Comparison of Localised and Global Niching Methods

Grant Dick

Spatial Information Research Centre
University of Otago, Dunedin, New Zealand
Phone: +64 3 479-5282 Fax: +64 3 479-8311
Email: gdick@infoscience.otago.ac.nz

Presented at SIRC 2005 - The 17th Annual Colloquium of the Spatial Information Research Centre
University of Otago, Dunedin, New Zealand
November 24th-25th 2005

ABSTRACT

Niching methods are a useful extension of evolutionary computation that allow evolutionary algorithms to be applied in multimodal problem domains. Current niching methods use either one of two methods to promote the formation of species within a population. Genetics-based methods, such as fitness sharing or clearing, work directly on the search space of the problem. Alternatively, spatially-structured evolutionary algorithms are used to place individuals onto a landscape and restrict mating to within isolated demes of population members. This isolation promotes the driving of geographically distant individuals towards separate parts of the search space. This paper introduces the concept of localised niching (LC). LC takes the traditionally global operations used in genetics-based niching and applies them locally in a spatially-structured population. Testing on two well known and difficult benchmark problems indicates that LC not only has the potential to significantly outperform the traditional global niching methods, but is also more resistant to some of the known limitations of genetics-based species formation.

Keywords and phrases: evolutionary algorithms, multimodal optimisation, spatially-structured populations, niching methods

1 Introduction

Many real-world problems have more than one optimal solution. While traditional evolutionary algorithms (EAs) can be successfully applied to unimodal problem domains, they often fail to locate and maintain multiple optima in multimodal environments. To enable an EA to properly search multimodal fitness landscapes, researchers in evolutionary computation have turned to techniques that drive subsets of a population towards different areas in the search space. These mechanisms, commonly referred to as niching methods or speciation, are designed to create and maintain diverse subpopulations of individuals within the global population.

Speciation within EAs has traditionally been implemented using two different techniques. One method, such as the ECOlogical GA, imposes a geography on the population and restricts mating to “close” individuals. Other methods, such as fitness sharing and crowding, work directly on the genetic information present in individuals and attempt to restrict inter-individual operations to genetically similar population members. While these methods of speciation work in completely different ways, the ultimate goal in their use is the same; to create and maintain diverse populations that are able to explore multiple areas of a search space simultaneously.

Although the mechanism by which the two methods of speciation work is completely different, there is no reason why they can not be combined to form a hybridised niching technique. The goal of this paper is to take an existing commonly used spatial structure and implement the traditionally global technique of clearing within the local confines of space. Experimentation with the localised clearing method indicates that it has the potential to outperform the established niching techniques. Also, localised speciation appears to be more robust with regards to some of the known limitations of existing niching methods.

The remainder of this paper is structured as follows: §2 introduces some of the more frequently encountered niching methods used in evolutionary computation. The technique of localised niching and a specific implementation, local clearing, are presented in §3 and §4 respectively. §5 presents the framework under which the various

algorithms are compared and the results are discussed in §6. Finally, implications of the findings of this paper and suggested directions of future work are presented in §7.

2 Niching Methods in Evolutionary Algorithms

There are numerous ways that an evolutionary algorithm can be extended to locate and maintain multiple optima within a single global population. This section gives an overview of some of the more popular methods of speciation used in evolutionary computation. A more comprehensive study into niching methods was conducted by Mahfoud as part of his doctoral dissertation (Mahfoud 1995b). Broadly speaking, niching methods can be divided into two categories – genetics-based techniques and spatially-structured populations. Genetics-based methods of niching use the genotypes/phenotypes of individuals as guides to drive speciation¹. Spatially-structured techniques impose geographic restrictions on individuals and produce isolated, genetically distinct solutions within the global population through restricted mating.

2.1 Sharing Methods

The most well-known method for creating subpopulations of like individuals is fitness sharing (Goldberg & Richardson 1987). It is based on the concept that a point in a search space has limited resources which must be shared by any individuals that occupy similar search space behaviours or genetic representations. Sharing in an EA is implemented by scaling the fitness of an individual based on the number of “similar” individuals present in the population. Derating an individual’s fitness is controlled by two operations, a similarity function, which measures the distance between two individuals in either the genotypic or phenotypic space, and a sharing function. The purpose of the sharing function is to take the distance between two individuals and return the degree to which they can be considered the same species.

The degree to which individuals are considered to be of the same species is controlled by the sharing radius. If the distance between two individuals is greater than the sharing radius, they are not occupying the same niche and hence their fitnesses do not interact. The value of the sharing radius is therefore important in determining the number of species in the system and their separation in the search space. When setting the sharing radius, one needs to take the following into consideration:

- The sharing radius needs to be set to a value that efficiently discriminates between the desired optima in the search space. This requires *a priori* knowledge about the distance between the peaks in the search space. As the search space has not been explored this information is rarely available. Instead, the assumption is usually made that perfect discrimination between different peaks is possible. (Mahfoud 1995b)
- The sharing radius is fixed, which implies that all the required optima in the search space should be equidistant (Smith, Forrest & Perelson 1993).

While sharing techniques have been successfully applied to difficult problems, they are known to be particularly unsuited to the test problems used in this paper unless scaling functions are incorporated into the fitness measure. This type of scaling requires prior knowledge of the problem domain (Goldberg, Deb & Horn 1992). Other researchers have also noticed this limitation when sharing is applied to other problems (Darwen & Yao 1995). Alternative niching methods, such as the ones described below, do not exhibit these problems. For this reason, fitness sharing is not considered as a benchmark case for this paper.

2.2 Crowding Methods

An alternative to fitness sharing is deterministic crowding (DC) (Mahfoud 1992). It is an extension of a technique first used by DeJong to help promote diverse populations (DeJong 1975). The method for performing DC is shown in Algorithm 1. The basic premise of DC is that pairs of individuals are recombined to create offspring. These offspring then replace their closest parent if they are of greater fitness.

Like fitness sharing, deterministic crowding uses a distance measure (either in genotype or phenotype space) to determine similarity between individuals. Unlike fitness sharing however, DC does not require the use of a similarity radius. This relaxes the requirement of *a priori* domain knowledge and makes DC more suitable for difficult problems than fitness sharing. It is used in this paper as a comparative benchmark. Additionally, deterministic crowding is an elitist niching method. This means that once a peak is discovered, it is never lost from the population. This behaviour differs from fitness sharing, which cannot use elitism as it conflicts with what fitness derating is trying to achieve.

¹Mahfoud describes the techniques that work directly on the search space as spatial niching methods. To avoid confusion with spatially-structured populations, the term genetics-based niching will be used in this paper.

```

input : A population of individuals of size  $N$ 
output: The same population with reproduced individuals

1 Shuffle population;
2 for  $i \leftarrow 1$  to  $(N - 1)$  step 2 do
3    $p_1 \leftarrow \text{population}[i]$ ;
4    $p_2 \leftarrow \text{population}[i + 1]$ ;
5    $(c_1, c_2) \leftarrow \text{reproduce}(p_1, p_2)$ ;
6   if  $[\text{distance}(p_1, c_1) + \text{distance}(p_2, c_2)] \leq [\text{distance}(p_1, c_2) + \text{distance}(p_2, c_1)]$ 
7     then
8       if  $\text{fitness}(c_1) > \text{fitness}(p_1)$  then  $p_1 \leftarrow c_1$ ;
9       if  $\text{fitness}(c_2) > \text{fitness}(p_2)$  then  $p_2 \leftarrow c_2$ ;
10      else
11        if  $\text{fitness}(c_2) > \text{fitness}(p_1)$  then  $p_1 \leftarrow c_2$ ;
12        if  $\text{fitness}(c_1) > \text{fitness}(p_2)$  then  $p_2 \leftarrow c_1$ ;
13      end
14 end
15 return population;

```

Algorithm 1: The Deterministic Crowding Algorithm

2.3 Clearing

The final genetics-based niching method described in this paper is clearing. Clearing is best described as a variant of the sharing technique described in §2.1 (Pérowski 1996). The major difference between sharing and clearing is that rather than degrading the fitness of like individuals, clearing determines the dominant individuals of the subpopulations and removes the remaining population members from the mating pool. The overall algorithm for clearing is shown in Algorithms 2 and 3.

```

input : A set of individuals (the population) and the clearing radius  $\sigma_{clear}$ 
output: A set of the dominant individuals in the population

1 matingPool  $\leftarrow$  population;
2 Sort matingPool in descending order of fitness;
3 for  $i \leftarrow 1$  to  $|matingPool|$  do
4    $m_i \leftarrow matingPool[i]$ ;
5   for  $j \leftarrow i + 1$  to  $|matingPool|$  do
6      $m_j \leftarrow matingPool[j]$ ;
7     if  $\text{distance}(m_i, m_j) < \sigma_{clear}$  then remove  $m_j$  from matingPool;
8   end
9 end
10 return matingPool;

```

Algorithm 2: The Clearing Algorithm

This paper uses the modified version of clearing in which no actual selection operator is used (Petrowski 1997). Instead, selection pressure emerges from two factors:

- The mating pool is sorted before any comparison between individuals takes place. Imposing an order on the mating pool ensures that the dominant individuals will not be removed by the clearing process.
- Like DC, clearing incorporates elitism into the niching process. Elitism in the clearing method is performed by copying the dominant individuals of above average fitness into the next generation. This has the added bonus of ensuring that any optima discovered during the course of evolution are not lost.

The clearing method used in this paper has been shown to perform at least as well as clearing with an explicit selection operator with the added benefit of reduced complexity. (Petrowski 1997)

Like fitness sharing, clearing relies on a distance radius to determine similarity between individuals. However, despite this parameter, clearing has been shown to be markedly superior to fitness sharing on the two difficult test problems used in this paper. Clearing is therefore presented in this paper as a benchmark case.

```

input : The population of individuals
output: The next generation of individuals

1 matingPool ← clearing(population,  $\sigma_{clear}$ );
2 generation ← {};
3 foreach i in matingPool do
4   | if fitness(i) > meanFitness(population) then generation ← generation ∪ {i};
5 end
6 while generation size is less than population size do
7   |  $p_1$  ← random element of matingPool;
8   |  $p_2$  ← random element of matingPool;
9   |  $c$  ← reproduce( $p_1, p_2$ );
10  | generation ← generation ∪ {c};
11 end
12 return generation;

```

Algorithm 3: Algorithm for breeding a generation using clearing

2.4 Spatially-Structured Evolutionary Algorithms

An alternative to the genetically-based techniques shown above is to impose a spatial structure on the overall population in an EA. EAs which use such populations are commonly referred to as Spatially-Structured EAs (SSEAs). SSEAs are frequently implemented in one, two or three dimensional space. Each individual in the global population occupies a single location in space. Demes are subpopulations which are formed by collecting the individuals from closely related locations. Genetic operations, such as selection and crossover, are limited to within the deme. A location, and hence an individual, takes part in more than one deme. This enables communication of genetic information throughout the global population.

The imposition of space onto a population creates an “isolation by distance” phenomenon which naturally allows speciation to take place. Perhaps the first person to notice this was Davidor with his work on the ECOlogical GA (Davidor 1991). This method placed population members onto a two-dimensional grid in which the edges met (a torus). Davidor noticed that the inherent isolation produced by the torus promoted different areas of the grid to converge to genetically different solutions of a given problem. However, these subpopulations were not persistent. Eventually, one genotype would dominate and take over the entire population.

A population structure similar in nature to Davidor’s ECOlogical GA was used by Collins and Jefferson on a bimodal graph partitioning problem (Collins & Jefferson 1991). For this problem, the population was structured as either a two dimensional torus or one dimensional ring. The authors noted that the two spatial populations outperformed panmictic (unstructured) populations in terms of time required to find a solution and overall diversity. Unlike the ECO GA, the structured populations were able to maintain both optima in nearly equal proportions for at least 1000 generations, although large populations (> 8000 individuals) were required to do so. The two optima were separated in space by regions of low-fitness hybrids. The authors considered the ring topology to be the more stable of the two spatially-structured populations investigated.

While the system used by Collins and Jefferson was able to locate and maintain two distinct solutions, the issue of scaling to problems with many global optima was not addressed. Sarma tested spatially-structured populations on two well known multimodal problems as part of her doctoral dissertation (Sarma 1998). Each problem had five known peaks in the defined region of the search space. Sarma noted that the two dimensional population structure she was investigating was unable to support all five optima, with at best three peaks being maintained for more than 200 generations. This, coupled with the work done by Davidor and Collins and Jefferson, implied that SSEAs may not be entirely useful for optimisation of problems with numerous global optima. Perhaps as a consequence of this work, research into using spatial populations for speciation stagnated. This is unfortunate, as even the simplest spatial topologies have interesting niching properties, as will be shown later in this paper.

3 Localised Niching

The previous section introduced several techniques for promoting speciation within a population. While each technique is different, they can all be grouped into two major categories; techniques which promote speciation through genetic information and those that use geographic isolation to induce niches. This section introduces a further possibility for niching – the use of genetic-based speciation techniques within the confines of a spatially-structured population.

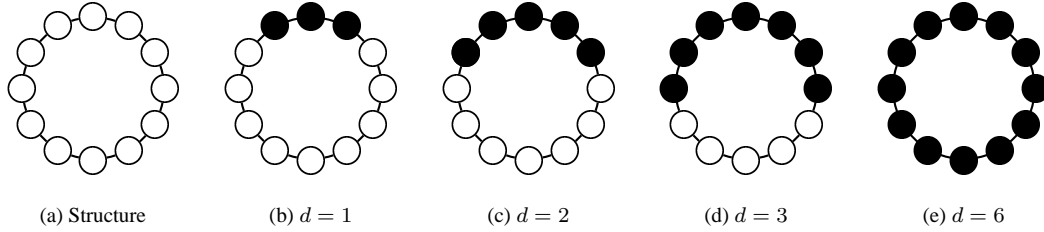


Figure 1: The ring topology for SSEAs. The shaded cells indicate the demes members for a given value of d . All demes shown are centred around the individual in the “12 o’clock” position.

The combination of genetics-based niching with SSEAs does not appear to have been extensively explored. In a previous study, Spears implemented a system that incorporated both a sharing-like niching method with a ring topology and noticed an improvement in performance on some problems (Spears 1994). However, he also noted that the ring topology operated independently of the sharing method. The use of the spatial structure of the ring within the sharing operation was left as a topic for future research. Other work has focused on using island models with modified migration operations to promote species formation (Bessaou, Pérowski & Siarry 2000). However, migrants are placed into a global pool in order to decide which would be the best destination in terms of genetic similarity. Hence, as with the work of Spears, the niching operator acts globally, and does not take into consideration any effects of locality and restricted mating.

The hybridisation of two different niching techniques at first appears to be somewhat of a contradiction. The global, genetic-based niching techniques need suitably large samples of individuals to form reasonable stable subpopulations. Conversely, SSEAs typically require small, tightly clustered demes in order to establish species within the population. However, as will be shown, a compromise can be found that allows the two niching methods to cooperate. Once the potential shortcomings have been resolved, hybridising individual-based niching with SSEAs introduces the following potential benefits:

- *Reduced Complexity* Traditionally, genetics-based niching requires large numbers of comparisons between individuals. At worst, fitness sharing requires each individual to be compared with every other population member, which results in an $O(N^2)$ runtime complexity. Given a small enough deme size d , this complexity should reduce to $O(Nd)$.
- *Parallel Implementation* SSEAs frequently implement space as a set of discrete locations. Each location is a semi-independent unit of computation, requiring only limited communication with its neighbours. The fine-grained nature of space in SSEAs makes them natural candidates for implementation on parallel machines, which can result in significant reductions in physical (wall-clock) execution time.
- *Behavioural Changes* Changing the size and shape of subpopulations presented to the genetic-based niching methods will obviously alter the system’s behaviour, with the hope of a reduction of computation required (in terms of fitness evaluations).

4 Local Clearing

The specific implementation of localised niching in this paper is *local clearing* (LC), which takes the traditionally global operation of clearing and applies it to each location in space. LC makes use of one particular spatial structure, the ring topology. In this spatial structure, N individuals are placed at equidistant locations in one dimensional space. The boundaries of this space, that is the ends of the line, are joined to form one continuous dimension, as is shown in Figure 1. Each dot in the ring represents a single individual. Demes are constructed via a parameter d which is the neighbourhood step size. The demes overlap, allowing for the slow exchange of genetic information throughout the population. Figure 1 shows demes around the individual at “12 o’clock” for several settings of d . The members of the deme are shown in black. The deme size for a given value of d is $2d + 1$. This gives a fine-grained tuning of deme sizes as necessary; a setting of $d = 1$ provides very small demes while a value of d approaching $\frac{N}{2}$ results in the ring converging to a panmictic population. Rings are known to be more resistant to genetic drift than panmictic populations so long as the condition $d < 0.1N$ holds (Dick & Whigham 2005). This paper uses $d = 0.05N$ for all experiments.

The algorithm for performing local clearing in a ring is shown in Algorithm 4. It introduces two major differences over the global clearing method. The most obvious change is in the application of clearing itself; rather than a single instance of clearing being applied to the whole population, LC applies clearing to every deme at

every location in space. This introduces a location-dependent component to clearing. An individual participates in multiple demes and must undergo the clearing process for each deme.

The second alteration to the global clearing process is in the way elitism is performed. LC as it appears in this paper uses a modified form of elitism as typically found in SSEAs (DeJong & Sarma 1995). If the current occupant of a given location is of below average fitness for the deme, it is automatically replaced. Otherwise, the incoming offspring must be of greater fitness than the current occupant in order to be introduced into the population.

input : The population of individuals and its containing spatial structure
output: The next generation of individuals

```

1 generation ← {};
2 foreach location in space do
3   | deme ← constructDeme(location);
4   | matingPool ← clearing(deme,  $\sigma_{clear}$ );
5   |  $p_1$  ← random element of matingPool;
6   |  $p_2$  ← random element of matingPool;
7   |  $c$  ← reproduce( $p_1, p_2$ );
8   |  $o$  ← population[location];
9   | if fitness( $c$ ) < fitness( $o$ ) and fitness( $o$ ) ≥ meanFitness(deme) then
10  | | generation[location] =  $o$ ;
11  | | else
12  | | | generation[location] =  $c$ ;
13  | | end
14 end
15 return generation;

```

Algorithm 4: Algorithm for breeding a generation using Localised Clearing

5 Methodology and Test Problems

The test functions used in this paper are taken from the the test suite defined by Mahfoud for his thesis on niching methods (Mahfoud 1995b). They are frequently used by other authors for evaluating niching EAs (Goldberg et al. 1992, Horn & Goldberg 1995, Mahfoud 1995a, Pérowski 1996, Petrowski 1997). Both are considered to be extremely hard problems to discover all global optima.

The $M7$ problem is a massively multimodal and deceptive problem proposed by Goldberg, Deb and Horn to evaluate the limitations of fitness sharing (Goldberg et al. 1992). It consists of five subfunctions. Each subfunction is a deceptive unimodal function $u(s)$, which is defined over strings of six bits

$$u(s) = \begin{cases} 1.000000, & \text{if } s = 6 \text{ or } s = 0 \\ 0.000000, & \text{if } s = 5 \text{ or } s = 1 \\ 0.360384, & \text{if } s = 4 \text{ or } s = 2 \\ 0.640576, & \text{otherwise} \end{cases}$$

where s is the number of 1 bits in the string. As shown in Figure 2(a), the function has two global optima at $u(x) = 0$ and $u(x) = 1$, which correspond to the bitstrings 000000 and 111111 respectively. There are also 20 local optima for all bitstrings where $u(x) = 3$. Hence, as defined over five units, $M7$ has $22^5 = 5,153,632$ total optima of which 32 are considered global and desirable.

The second function used in this paper, $M9$, described as a *minimum distance* function, was first presented in (Horn & Goldberg 1995). The equation for $M9$ is the sum of three elementary subfunctions f_{mdG} defined over eight-bit strings:

$$M9(x_0, \dots, x_{23}) = \sum_{i=0}^2 f_{mdG}(x_{8i}, \dots, x_{8i+7})$$

with $x_i \in \{0, 1\}$. Given a set of optimal bitstrings \mathbf{G} , f_{mdG} is defined as:

$$f_{mdG}(s) = \begin{cases} 10 & \text{if } s \in \mathbf{G} \\ \min_{g \in \mathbf{G}} H(s, g) & \text{otherwise} \end{cases}$$

where $H(s, g)$ is the hamming distance between the bitstrings s and g . The set \mathbf{G} contains the vectors 00000000, 10001100 and 01001010. A graphical representation of $f_{mdG}(s)$ is shown in Figure 2(b).

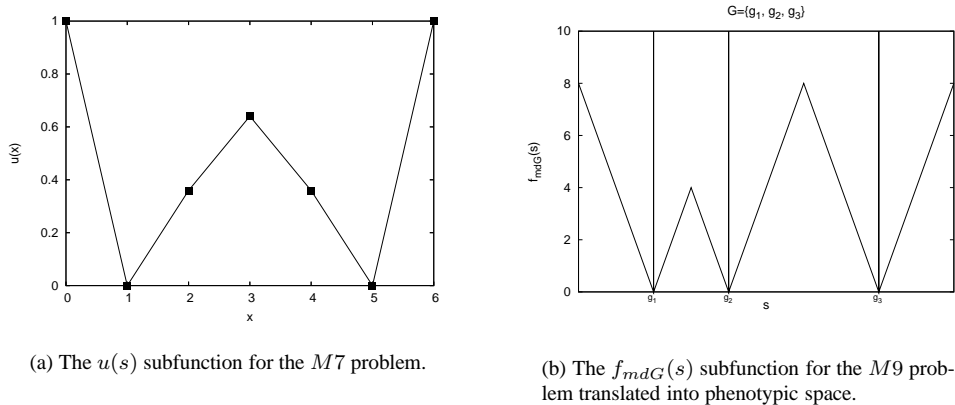


Figure 2: A graphical representation of the unitation functions used in the $M7$ and $M9$ test problems.

5.1 Methodology

The deterministic crowding, clearing and localised clearing algorithms were applied to the two problems $M7$ and $M9$. The sole performance criterion measure was the number of optima covered by a population for a given generation. The best algorithm would require the fewest generations to discover all optimal solutions, and would maintain all optima for the longest period of time. Each combination of problem and algorithm was run 100 times. Comparisons between algorithms were made on the averages for these runs.

Each problem employed one-point crossover which was applied with 100% probability. Mutation was applied on a bit-by-bit case with probability of 0.002.

The experiments using the $M7$ were run with a population size of 600. This is the same population size as used in (Pétrowski 1996). Deterministic crowding was also run using larger population sizes of 2400 and 4800. Each run lasted for 100 generations. The similarity between two individuals was determined by measuring the hamming distance of the bitstrings. The clearing radius σ_{clear} was set to 6, which is the value used in earlier experiments using this function (Mahfoud 1995a, Pétrowski 1996).

All experiments involving the $M9$ problem were run with a larger population size of 1000 to be consistent with earlier work (Petrowski 1997). Again, additional experiments using deterministic crowding were run with larger population sizes of 4000 and 8000. Each run lasted 100 generations. As with $M7$, hamming distance was applied to two individuals to determine a threshold of similarity. σ_{clear} was set to 3, which is the largest value that allows discrimination between global optima to take place. Again, the value for σ_{clear} was taken from previous work using this function (Mahfoud 1995a, Petrowski 1997).

A final set of experiments were run using a ring SSEA without localised clearing. These experiments formed a baseline of comparison and ensured that any performance benefits seen from localised clearing were not purely a result of the spatial structure.

6 Results

A comparison of the performance of the four niching methods on the $M7$ problem are shown in Figure 3. Deterministic crowding is able to locate and maintain all 32 optima in the search space, provided it is given a suitably large population size. Clearing is also able to find all 32 peaks within a single global population, with a significantly smaller population size. It also locates most of the optima earlier than DC. Both the ring SSEA and the local clearing algorithm start to find optima in the search space much faster than either DC or clearing. However, without clearing the ring SSEA stagnates in its search for all optima and halts on average at approximately 19 optima. In all 100 runs, LC required fewer than 40 generations to discover and maintain all 32 peaks in the search space.

Results for the $M9$ tests are shown in Figure 4. As before, deterministic crowding requires a much (4x) larger population size in order to discover all 27 optima within the required 100 generations. Compared to DC, clearing is able to locate and maintain all 27 peaks in fewer generations with a population size of 1000. The ring SSEA quickly discovers the majority of the optima in the search space, but is unable on average to discover all 27 peaks.

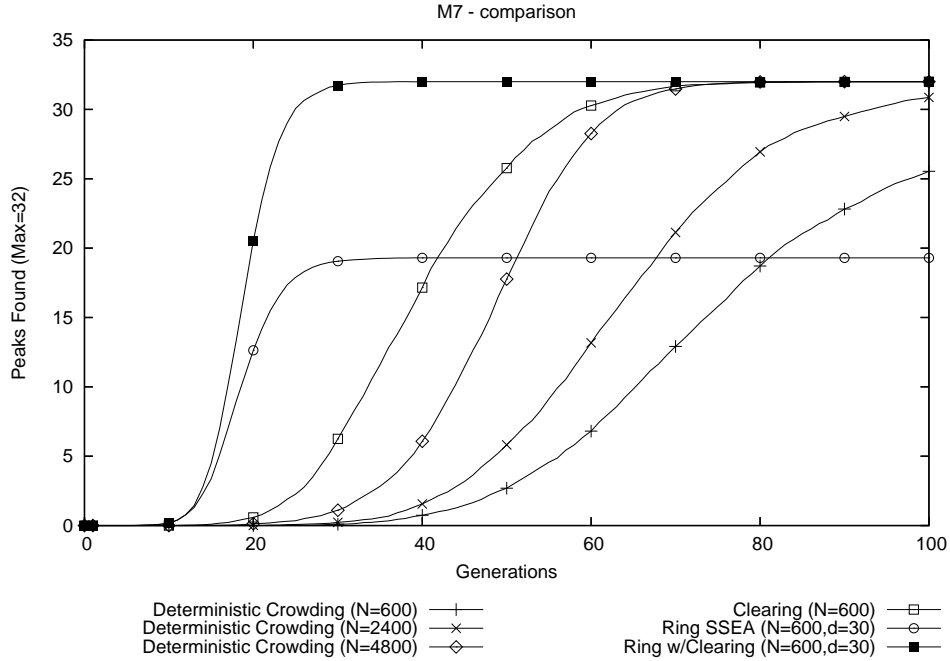


Figure 3: Performance of localised clearing against established niching techniques on the *M7* problem.

Local clearing is able to use the properties of an SSEA to quickly discover peaks in the search space. LC is also able to use the properties of clearing to help prevent search stagnation and allow it to locate all 27 optima in the search space.

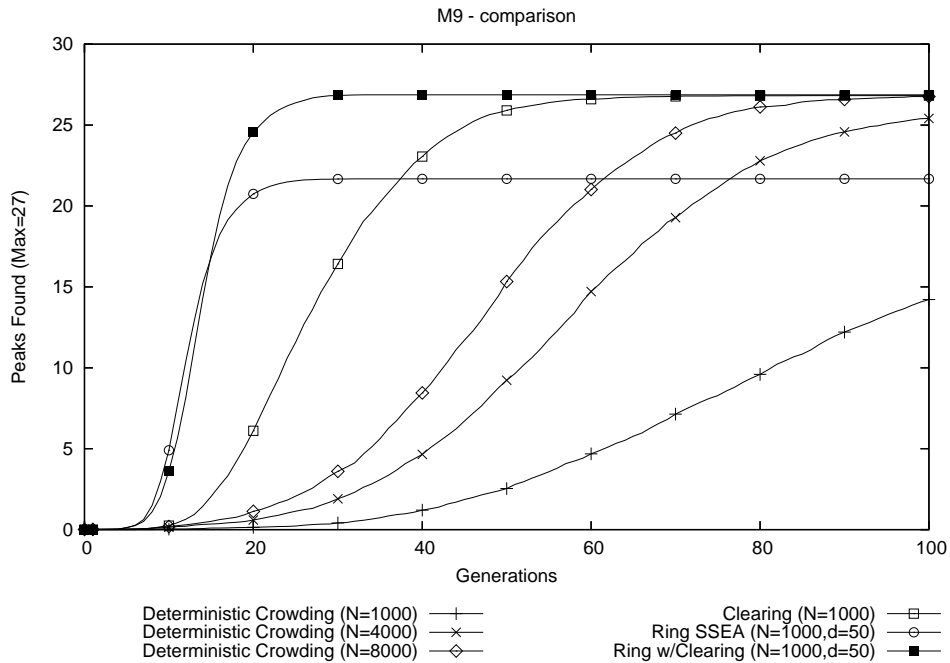


Figure 4: Performance of localised clearing against established niching techniques on the *M9* problem.

6.1 Localised Clearing and Sensitivity of σ_{clear}

As mentioned in §2, most genetics-based niching methods required a parameter that determines the degree to which two individuals are considered the same species. The notable exception here is deterministic crowding. Failure to set this parameter correctly will often result in suboptimal performance of the niching method, as it

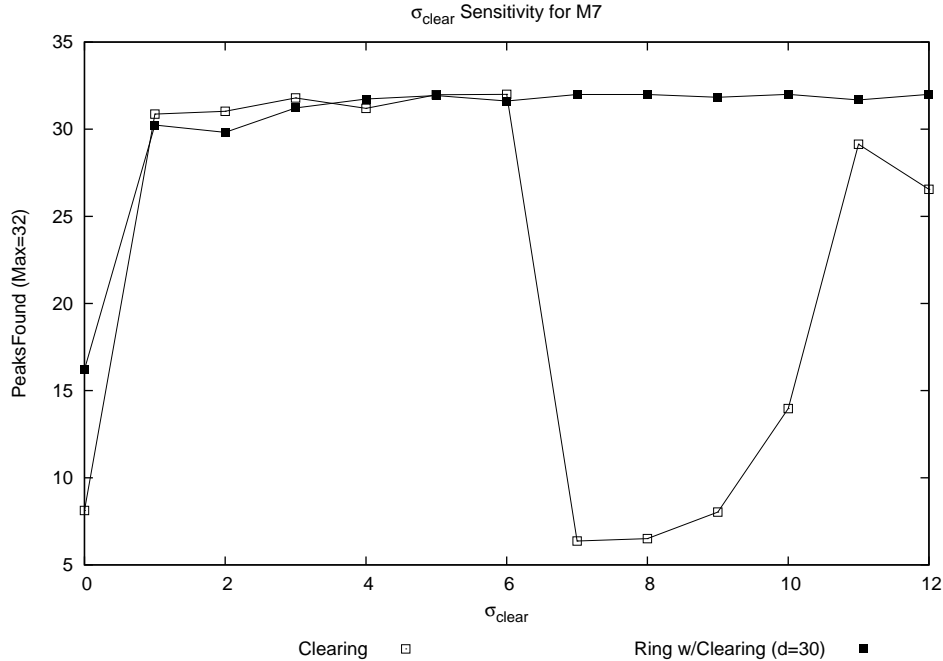


Figure 5: Performance of localised clearing and global clearing on the function $M7$ when using an incorrect setting for σ_{clear} .

is no longer able to accurately distinguish between optima. The niching methods of clearing and local clearing were able to discover all global optima in the test problems. This was in part due to the suitable calibration of the clearing radius. The $M7$ and $M9$ problems are well understood and optimal values for σ_{clear} known. Had σ_{clear} been set incorrectly, the performance of these two algorithms may have been quite different.

The sensitivity of both clearing and LC with regards to the clearing radius was investigated by repeating the $M7$ and $M9$ experiments with varying values of σ_{clear} . Each combination of problem and clearing radius was run 100 times, with the other parameters remaining constant. The averages number of peaks after 100 generations was taken for these 100 runs and are shown in Figures 5 for the $M7$ problem and in Figure 6 for $M9$. In both cases, clearing is significantly affected by an incorrect value for σ_{clear} , which was expected. Surprisingly, the value that σ_{clear} took did not appear to have a significant impact on the overall behaviour of LC. One possible explanation for this is that elitism in clearing is constrained to some degree by the results of clearing, and hence the value of σ_{clear} may have an impact there. Elitism in LC does not require any information from the clearing process and uses only local information, so the impact of an incorrect value for σ_{clear} is less significant. Elitism is known to be very important for determining the success of niching methods on these problems (Pétrowski 1996).

7 Conclusion and Future Work

Niching techniques are an important addition to evolutionary computation that allow EAs to be applied to multimodal problem domains. Traditionally, niching methods for EAs have worked by creating subpopulations of like individuals based of information extracted directly from the search space. Spatially-structured EAs have performed similar roles in the goal of obtaining diverse populations via geographic isolation of individuals. This paper introduces a new model of niching, local clearing, which combines the search space oriented operator of clearing with a ring-based SSEA. LC is able to outperform globally-applied clearing on two well-known and difficult multimodal problems. Additionally, LC is more stable with regards to the clearing radius. This second factor should be of particular interest as correct calibration of this parameter has traditionally required *a priori* knowledge of the problem at hand. As LC is less sensitive towards the value of clearing radius, it is more appealing as a tool for investigating problems where information about the size and shape of optima is unknown.

This paper is a preliminary investigation into localised niching methods. There are many possible directions that future work in this field might take. Obviously, LC should be applied to more problems so that its true behaviour can be properly ascertained. Another area that would be of interest is to investigate different local elitism strategies. Currently, LC uses no information about the niching process when deciding if an individual is to survive or be replaced. While this paper has suggested that this strategy may be to LC's advantage, a thorough

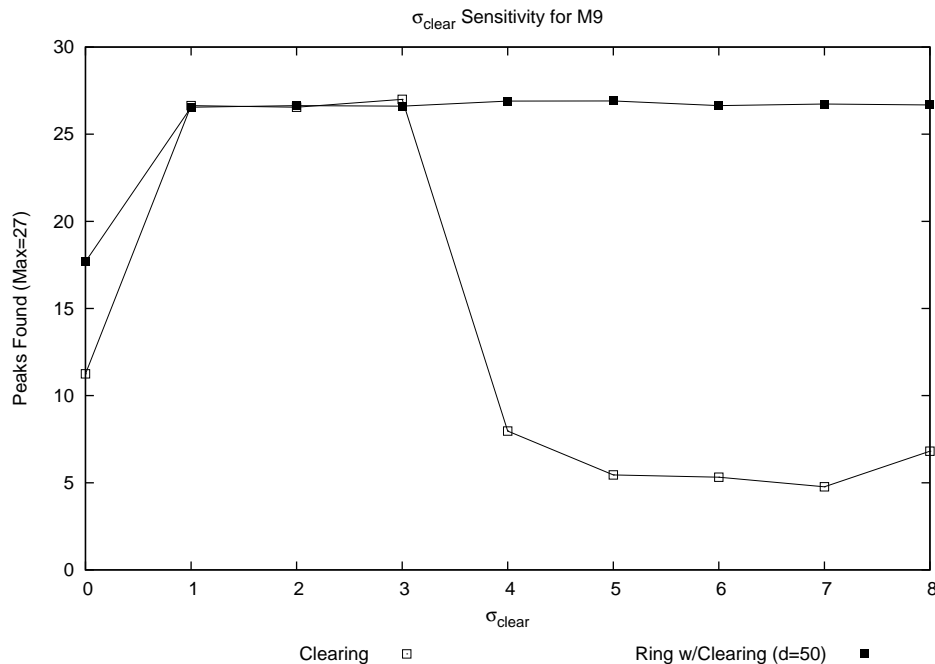


Figure 6: Performance of localised clearing and global clearing on the function $M9$ when using an incorrect setting for σ_{clear} .

investigation into alternative elitism schemes would provide a more complete understanding of the behaviour of local clearing. Finally, this paper has suggested that LC may have a computational advantage over existing niching methods under certain conditions. A full investigation into the complexity of localised niching, including upper and lower bounds of run-time would be of particular use for applying LC in real-world conditions.

References

- Bessaou, M., Pétrowski, A. & Siarry, P. (2000). "Island Model cooperating with Speciation for Multimodal Optimization" In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo & H.-P. Schwefel (eds), *Parallel Problem Solving from Nature – PPSN VI*. Springer Berlin pp. 437–446.
- Collins, R. J. & Jefferson, D. R. (1991). "Selection in Massively Parallel Genetic Algorithms" In R. K. Belew & L. B. Booker (eds), *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers San Mateo, CA.
- Darwen, P. J. & Yao, X. (1995). "A Dilemma for Fitness Sharing with a Scaling Function" *Proceedings of the Second IEEE International Conference on Evolutionary Computation*. IEEE Press Piscataway, New Jersey.
- Davidor, Y. (1991). "A Naturally Occuring Niche & Species Phenomenon: The Model and First Results" In R. K. Belew & L. B. Booker (eds), *Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA'91)*. Morgan Kaufmann Publishers San Mateo, California pp. 257–263.
- DeJong, K. A. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis University of Michigan Ann Arbor, MI. Dissertation Abstracts International 36(10), 5140B, University Microfilms Number 76-9381.
- DeJong, K. & Sarma, J. (1995). "On Decentralizing Selection Algorithms" In L. J. Eshelman (ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA'95)*. Morgan Kaufmann Publishers San Francisco, California pp. 17–23.
- Dick, G. & Whigham, P. (2005). "The Behaviour of Genetic Drift in a Spatially-Structured Evolutionary Algorithm" *2005 IEEE Congress on Evolutionary Computation*. IEEE Press. pp. 1855–1860.
- Goldberg, D. E., Deb, K. & Horn, J. (1992). "Massive Multimodality, Deception, and Genetic Algorithms" In R. Männer & B. Manderick (eds), *Parallel Problem Solving from Nature, 2*. Elsevier Science Publishers, B. V. Amsterdam pp. 37–46.

- Goldberg, D. E. & Richardson, J. (1987). "Genetic algorithms with sharing for multi-modal function optimisation" *Proc of the 2nd Int. Conf. on Genetic Algorithms and Their Applications..* pp. 41–49.
- Horn, J. & Goldberg, D. E. (1995). "Genetic Algorithm Difficulty and the Modality of Fitness Landscapes" In L. D. Whitley & M. D. Vose (eds), *Foundations of Genetic Algorithms 3*. Morgan Kaufmann. San Francisco, CA pp. 243–269.
- Mahfoud, S. W. (1992). "Crowding and preselection revisited" In R. Männer & B. Manderick (eds), *Parallel problem solving from nature 2*. North-Holland Amsterdam pp. 27–36.
- Mahfoud, S. W. (1995a). "A Comparison of Parallel and Sequential Niching Methods" In L. Eshelman (ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms*. Morgan Kaufmann San Francisco, CA pp. 136–143.
- Mahfoud, S. W. (1995b). *Niching methods for genetic algorithms*. PhD thesis University of Illinois at Urbana-Champaign Urbana, IL, USA. IlliGAL Report 95001.
URL: <ftp://ftp-illigal.ge.uiuc.edu/pub/papers/IlliGALs/95001.ps.Z>
- Pétrowski, A. (1996). "A Clearing Procedure as a Niching Method for Genetic Algorithms" *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation..* pp. 798–803.
- Petrowski, A. (1997). "A New Selection Operator Dedicated to Speciation" In T. Bäck (ed.), *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)*. Morgan Kaufmann San Francisco, CA pp. 144–151.
- Sarma, J. (1998). *An Analysis of Decentralized and Spatially Distributed Genetic Algorithms*. PhD thesis George Mason University Fairfax VA, USA. double sided.
- Smith, R. E., Forrest, S. & Perelson, A. S. (1993). "Searching for Diverse, Cooperative Populations with Genetic Algorithms" *Evolutionary Computation*. **1**(2): 127–149.
- Spears, W. M. (1994). "Simple Subpopulation Schemes" In A. V. Sebald & L. J. Fogel (eds), *Evolutionary Programming: Proc. of the Third Annual Conf.*. World Scientific Press Singapore pp. 296–307.