A Clustering Based Niching EA for Multimodal Search Spaces

Felix Streichert[1], Gunnar Stein[2], Holger Ulmer[1], and Andreas Zell[1]

[1] Center for Bioinformatics Tübingen (ZBIT), University of Tübingen, Sand 1, 72074 Tübingen, Germany,
streiche@informatik.uni-tuebingen.de,
http://www-ra.informatik.uni-tuebingen.de/welcome.html

[2] Institute of Formal Methods in Computer Science (FMI), University of Stuttgart, Breitwiesenstr. 20/22, D-70565 Stuttgart, Germany,
http://www.informatik.uni-stuttgart.de/ifi/fk/index_e.html

Abstract. We propose a new niching method for Evolutionary Algorithms which is able to identify and track global and local optima in a multimodal search space. To prevent the loss of diversity we replace the global selection pressure within a single population by local selection of a multi-population strategy. The sub-populations representing species specialized on niches are dynamically identified using standard clustering algorithms on a primordial population. With this multi-population strategy we are able to preserve diversity within the population and to identify global/local optima directly without further post-processing.

1 Introduction

In this paper we describe the Clustering Based Niching method (CBN) for Evolutionary Algorithms (EA) to identify multiple global and local optima in a multimodal search space. The basic idea was to transfer the biological concept of non-interbreeding species living in separated ecological niches into EA to preserve diversity in the EA population. One of our goals was to make the CBN as independent of the underlying EA method as possible in such a way that it can be applied to multiple EA methods and that the impact of the CBN on the EA mechanism and the fitness function is as small as possible. Also, we aimed for the CBN to have as few critical parameters as possible to allow black box optimization.

In biology species have in common that they don't interbreed anymore. Taking into account that species in different ecological niches don't compete for the same resources, but evolve independently of each other. Using this property should be the most natural way to create and maintain diversity in an EA population.

Our method to artificially create species in a primordial EA population is to search for groups or clusters of EA individuals in the search space, which will naturally occur due to the general convergence behavior of EA by the means of clustering algorithms. These clusters can then be separated into isolated sub-populations. Such a sub-population would represent a single species. Therefore,

individuals of different sub-populations do not compete with each other and are not be allowed to interbreed. The individuals of a single sub-population on the other hand do compete and breed like in any traditional EA and each sub-population behaves like an EA converging to a global/local optimum. With additional mechanisms to split sub-populations if necessary, dynamic specialization can occur. Merging species that become too similar will allow only one sub-population per niche.
In Sec. 2 we give an overview over some niching EAs and multi-start algorithms for multimodal search spaces. The algorithm for the Cluster Based Niching method is given in Sec. 3, and in Sec. 4 results are presented comparing the CBN with an Evolution Strategy (ES) to Multi-Start Hill-Climbing and an Evolution Strategy with Fitness Sharing on several two-dimensional benchmark functions and one n-dimensional benchmark function.

2 Related work

2.1 Niching Evolutionary Algorithms

Niching Evolutionary Algorithms (NEAs) try to identify as many optima as possible, by preserving diversity within the EA population and by using this diversity as resource for exploratory crossover. Most NEAs are based on the idea to preserve the diversity within the population by altering the EA operators to prevent premature convergence to one optimum, like Fitness Sharing [7], Crowding [4] and Deterministic Crowding (DC) [10] and Tagging [1, Sec. 6.2.3]. Other NEAs use a multi-population approach like the Multinational (MN) GA [19, 20] and the Forking GA [18], which divide a primordial global EA population into sub-populations with reduced interaction to preserve diversity above the level of sub-populations.
The most common and best studied niching method, based on prevention of premature convergence, is Fitness Sharing by Goldberg and Richardson [7]. Sharing adds a penalty to the fitness $\Phi(x_i)$ of all individuals x_i in the population P of size μ that are too similar to another individual in respect of a given metric $\|x_i, x_j\|$ between two individuals and the sharing distance σ_{share}:

$$\Phi'(x_i) = \frac{\Phi(x_i)}{\sum_{j=0}^{\mu} sh(x_i, x_j)} \tag{1}$$

$$sh(x_i, x_j) = \begin{cases} 1 - \left(\frac{\|x_i, x_j\|}{\sigma_{share}}\right) & : \text{ if } \|x_i, x_j\| \leq \sigma_{share} \\ 0 & : \text{ else} \end{cases} \tag{2}$$

This penalty is intended to lead to an even distribution of the EA population around promising areas in the search space and to prevent the overall convergence of the whole population on one optimum. Unfortunately, there are several drawbacks:

- There is a fixed σ_{share} for all individuals. For multimodal optimization all optima in the search space must be nearly equidistant.

- To set σ_{share} *a priori* knowledge about the distribution of the optima and their fitness is required, see for example [11].
- The population size is also dependent on *a priori* knowledge of the number of optima, an example is given in [12].
- Using no or a bad scaling function on the fitness Φ can prevent the individuals from finding the actual peak of the niche [3]. Alternatively, an additional hill-climbing post-processing algorithm is needed to improve the results of sharing, see [13].

Yin and Germay [21] introduced an Adaptive Clustering Algorithm (ACA) to avoid the *a priori* estimation of σ_{share}. Instead of the original sharing function of equations (1) and (2) they used equations (3) and (4).

$$\Phi'(x_i) = \frac{\Phi(x_i)}{S(x_i)} \tag{3}$$

$$S(x_i) = n_{c_j} - n_{c_j} \cdot \left(\frac{\|x_i, c_j\|}{2 \cdot d_{max}} \right)^{\alpha} \quad \text{, with } x_i \in C_j \tag{4}$$

where α is a constant, $\|x_i, c_j\|$ is the distance between the individual x_i and the centroid c_j of the cluster C_j which x_i belongs to, and n_{c_j} is the number of individuals associated with cluster C_j. The ACA is able to get rid of σ_{share} by applying the sharing function only within identified clusters, but introduces two additional variables d_{min} and d_{max}, which define the minimum and the maximum radii of a cluster, which need to be set to reasonable values. Although Gan and Warwick [6] extended the ACA by retaining the niches found by the ACA, they did not implement a multi-population approach.
One multi-population approach is the Forking GA [18], which monitors a global GA population for the occurrence of dominating schemes in the population. If such a scheme occurs, a sub-population is created which includes all individuals that meet the scheme and this sub-population continues the search in a subspace of the original space reduced by the fixed elements of the scheme. The global population continues the search in the original space but the already identified schemes are forbidden.
The Multinational GA groups the individuals together into sub-populations called nations [19, 20]. Each nation consists of individuals which are not separated by valleys of lower fitness in the search space. This is tested with the 'hill-valley' function by evaluating several interpolating individuals between each individual of the GA population. A valley would obviously separate different local optima and therefore different nations. The Multinational GA is distinguished from other NEAs by the fact that it is the only NEA known to us which actually identifies the optima on the basis of the nations residing on them, without any further post-processing.

2.2 Multi-Start Algorithms

Although the Multi-Start Hill-Climber (MS-HC) represents a rather primitive technique it can become useful in simple and low dimensional multimodal search

spaces. The MS-HC performs several local HC searches in parallel, each one initialized randomly. A single HC will most likely converge to different global/local optima depending on the initialization.
There is a high probability that several HC will converge toward the same local optimum. To prevent this, Törn introduced the LC (multiple local searches with clustering) algorithm [16, 17]. After a number of parallel local search HC steps, he applies a clustering algorithm and continues the local search with only one sample from each cluster. Hanagandi and Nikolaou applied the principles of the LC algorithm on GA [8]. But instead of retaining the niches found by the clustering algorithm, they randomized all individuals after each clustering step except the best cluster representatives.
Beasley et. al. proposed a Multi-Start GA, the Sequential Niching algorithm, for the multimodal optimization problem [2]. The optimization result of each GA run is stored as identified optimum and a penalty function is added permanently to the fitness function to prevent the next run of the GA from finding the very same optimum again. The shape and the radii of the penalty function are crucial for this algorithm since it can add several artificial deceiving local optima to the search space. However, the Sequential Niching yields the advantage that the basic EA mechanisms remain unaltered.

3 Clustering Based Niching EA

To introduce the biological concept of species into NEA we combined two methods: First a multi-population strategy with localized selection and limited interbreeding between sub-populations ($\widehat{=}$species) and second the use of clustering algorithms to identify species in an undifferentiated population. Both methods have been successfully applied independently, but to our knowledge they have not been merged into a single NEA approach so far.
The multi-population strategy has the advantage that it can preserve diversity through localized competition, for example the selection from localized clusters as performed by Hanagandi and Nikolaou, and localized interbreeding, compare mating restriction [1, Sec. 6.2.4]. However, each sub-population still behaves like a standard EA optimizer. This allows us to apply the full range of possible EA extensions or specialized EA operators on the level of sub-populations, like CMA [9] or MVA [14] mutation or even other NEA techniques like Fitness Sharing within each sub-population.
The goal of clustering algorithms is to group data units into cluster in such a way that units within a cluster are most similar while the clusters are relatively distinct from each other. In our application the EA individuals x_i are the data units and the resulting clusters will represent the species/niches. Using clustering algorithms to identify species allows us to move all relevant parameters for the niching behavior down to the clustering algorithm. There are numerous clustering algorithms available for multiple data types and applications with varying parameters requirements. We just need to choose the algorithm at yields the best ratio between clustering behavior and number of necessary parameters.

3.1 The Multi-Population Strategy

We implemented a multi-population strategy that starts with a randomized single primordial undifferentiated population D_0 and allows differentiated sub-populations(species) $D_{i\geq 1}$ to be generated dynamically. D_0 plays the special role of containing all individuals that do not belong to any identified species. While the undifferentiated population D_0 explores the general search space, species $D_{i\geq 1}$ exploit already identified niches.

After initialization of D_0 the CBN-EA generational cycle is entered until a EA termination criterion is met, like maximal number of generations reached, compare Fig. 1.

```
D0 = createInitialPop();
repeat {
  // species evolution phase
  for all i do simulateEAGeneration(Di);
  // species differentiation phase
  if (numOfClusters(D0) ≥ 1) split(D0);
  for all (i>0) do {
    if (numOfClusters(Di) > 1) split(Di);
    D0.add(Di.getLoners());
  }
  // species convergence phase
  TLP = createEmptyPop();
  for all (i>0) do TLP.addCentroidOf(Di);
  if (numOfCluster(TLP) ≥ 1) mergeDemes();
} until (maxGenerationReached());
```

Fig. 1. Pseudocode of CBN

First the **species evolution phase** is performed by simulating evaluation, selection and reproduction independently for each population D_i.

Then the **species differentiation phase** is entered: D_0 is tested whether differentiation occurs (`numOfClusters(D0)` $\geq$ `1`). If clusters can be identified by the clustering algorithm new sub-populations are created from the members of the clusters. Then the clustering algorithm is called on all sub-populations $D_{i\geq 1}$ to test if a species continues to differentiate further into new species which are to be separated. On the other hand, if the clustering algorithm finds individuals in $D_{i\geq 1}$ that do not belong to any species (`Di.getLoners()`), those individuals are moved to D_0 as straying loners.

Finally, the **species convergence phase** is performed: All species $D_{i\geq 1}$ add a representative (e.g. a centroid) to a temporary population of representatives (`TLP`). Clusters found in `TLP` indicate species that converge to the same niche which are to be merged to join their effort.

Depending on the convergence speed of the EA method, multiple EA generational steps can be performed before applying the species differentiation and convergence mechanisms. We found that early differentiation enables CBN to identify more optima, as the standard EA convergence behavior would progressively remove individuals located on unincisive local optima.

Merging of species that converge on the same niche will have the effect that niches with a bigger basin of attraction will be populated by species with more members. But this mechanism also guarantees, that if the CBN is converged there exists at most one species per niche. This enables the CBN to identify global/local optima directly, by returning just the best individual of each species. It is necessary to note, that currently there is no crossover between species and that there is no competition between species. We want to introduce these mechanisms to the CBN in the near future.

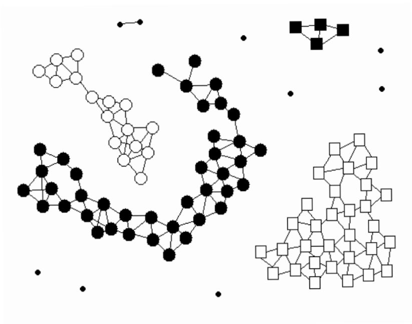


Fig. 2. Density-Based Cluster-Analysis

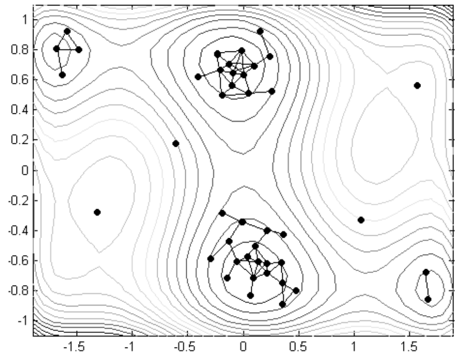


Fig. 3. Clustered population on M2

3.2 The Clustering Algorithm

We decided to use the 'density-based' clustering algorithm by Sanders, Ester, Kriegel and Xiaowei to identify species [5, 15]. This clustering algorithm identifies clusters by connecting individuals if the distance $\|x_i, x_j\|$ between them is lower than a given threshold value σ_{dist}. All interconnected groups of individuals, whose group size exceeds a minimum value *MinPts* are identified as cluster. Fig. 2 gives an example what kind of clusters can be identified by the 'density-based' clustering algorithm using $MinPts = 4$.
The 'density-based' clustering algorithm offers several advantages:

- it allows clusters of varying size and shape,
- it can identify clusters of *a priori* unknown number,
- the algorithm allows for loners which do not belong to any species, indicated as small dots in Fig. 2,
- it requires only two parameters that are easy to interpret.

Since *MinPts* gives the minimum size of a cluster it also gives the minimum size of a sub-population. And the minimum size of a sub-population can be chosen *a priori* not depending on the problem but on the EA method used. Small values for *MinPts* will be sufficient for mutation oriented EA methods like ES but larger values will be necessary for crossover based EA methods like GAs.
The parameter σ_{dist} is not as easy to select. It gives a lower bound, below which two optima can not be distinguished, because the clustering algorithm is not able to separate the clusters.
Fig. 3 gives an example for a clustered ES population with $MinPts = 2$ using an euclidean distance metric. Note that although an individual is located on the local optimum in the lower right corner, no species can be established. To increase the stability of identified species the mutation step size must be of the same order of magnitude as σ_{dist}. Otherwise individuals leave and enter species randomly simply because of mutation. Therefore, we currently limit the ES mutation step size to σ_{dist}, if an individual belongs to a species. This constraint can be removed, if the individual returns to D_0 as a loner.

4 Results

We compared the CBN on five multimodal benchmark functions, given in the Appendix, to Fitness Sharing (FS) and a Multi-Start Hill-Climber (MS-HS). Since we used real-valued benchmark functions we decided to apply Evolution Strategies (ES). We used a $(60+120)$-ES with best selection scheme simulated for $T = 100$ generations. For the CBN we used $MinPts = 2$ as minimum cluster size. To increase the performance of the FS-ES and to avoid the problem of finding a suitable scaling function, we used a hill-climbing post-processing step start at $T = 70$ but only for the FS-ES. The MS-HC was simulated by 120 independent (1+1)+ES trials with a fixed mutation rate σ_{mut}.
The performance of the algorithms is measured by the number of optima each algorithm found, averaged over 25 runs. An optimum o_j was considered 'found' if $\exists\ x_i \in P_{t=T} \mid \|x_i, o_j\| \leq \epsilon = 0.005$, where $P_{t=T}$ is the complete population at the end of each run and x_i an individual in $P_{t=T}$.
We applied all three algorithms on a normalized search space and varied a 'resolution' parameter to find the best parameter settings for each algorithm. In case of the MS-HC 'resolution' represents the fixed mutation rate σ_{mut}, for the FS-ES the parameter gives the critical value of σ_{share} and for the CBN the 'resolution' gives the σ_{dist} of the 'density-based' clustering algorithm.

4.1 Two Dimensional Benchmark Functions

On all four benchmark problems the MS-HC achieved the best results due to the limited search space and the high number of multi-starts and the available number of fitness calls. Only in case of M2 and M3 the MS-HS failed to identify all local optima if σ_{mut} becomes too big, since the high mutation rates enable a single HC to escape local optima, see Fig. 6 and Fig. 7.
With the additional HC post-processing step the FS-ES became less prone to bad values for σ_{share}, but performed not as well as the MS-HC. But although extremely high values for σ_{share} rendered the FS ineffective, they caused an evenly distributed start for the HC post-processing step, compare Fig. 7.

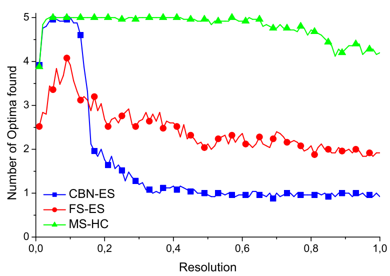


Fig. 4. M0, avg. number of optima found

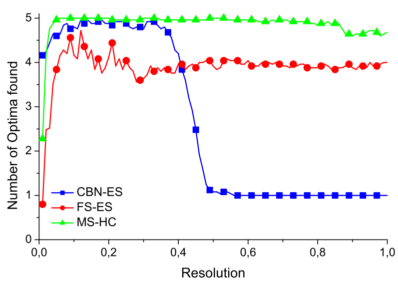
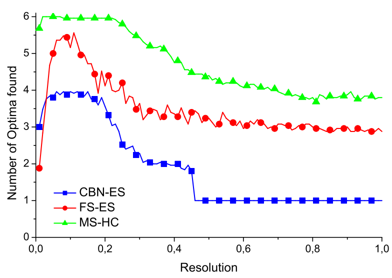


Fig. 5. M1, avg. number of optima found



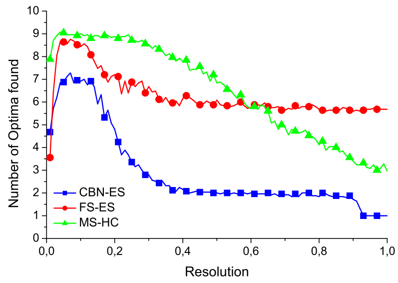


Fig. 6. M2, avg. number of optima found **Fig. 7.** M3, avg. number of optima found

The CBN-ES performed better than FS-ES on M0 and M1. But the importance of σ_{dist} becomes evident. As discussed before, σ_{dist} gives the minimal distance between two separable clusters, see 3.2. In case of the M0 benchmark function the optima are about 1/5 units apart, compare Fig. 10, and about 1/2 units in case of M1, compare Fig. 11. These are exactly the σ_{dist} values where the performance of the CBN-ES drops to the behavior of a standard ES without niching, thus converging to a single optima. The same effect can be observed for M2 and M3, compare Fig. 12 and Fig. 13. But here the optima are not as evenly spaced and the CBN-ES fails step-by-step as σ_{dist} is increasing.
On M2 and M3 CBN performed not as well as the FS-ES with the additional post-processing step. Regarding M2 CBN failed to identify the two most unincisive local optima, compare Fig. 3. This is caused by the greedy best selection strategy used. Most individuals of D_0 converged to the more attractive optima before species can be identified. On both problems the explorative character of D_0 seemed to fail.

4.2 The n-Dimensional Benchmark Function

To examine how the algorithms react to increasing dimensionality we compared them on the n-dimensional M5 benchmark function with 5^n optima, see Fig. 8 and 9.
Although the MS-HC has the advantage that it could track one optima per deployed individual, it finds considerably less optima. Also the performance of the HS-HC suffers if the problem dimension is increased.
The FS-ES on the other hand fails completely with the increasing problem dimension, although the evenly spaced optima with equal fitness should provide perfect conditions for Fitness Sharing.
Regarding the CBN the number of optima identified with $\mu = 60$ never exceeds fifteen even for $n < 4$. This suggests that CBN is limited by the population size rather than the problem dimension. Although $MinPts = 2$ allows a minimum species size of two, merging of species causes much bigger sub-populations.

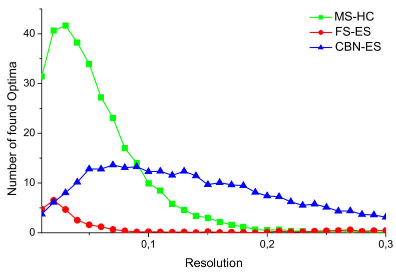


Fig. 8. M5 $n = 4$, avg. number of optima

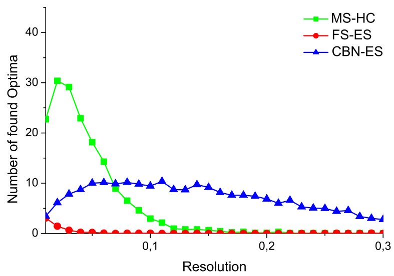


Fig. 9. M5 $n = 5$, avg. number of optima

These results indicate, that CBN scales better with increasing search space dimension than MS-HC and FS-ES but that CBN needs more individuals to retain a niche than *MinPts* would suggest.

5 Conclusions and Future Research

The CBN-EA we proposed, is a new niching EA method based on the formation of species. CBN joins a multi-population strategy with clustering analysis for species detection. This approach is virtually independent of the EA method it is applied to, because it does not put any restriction on the EA method used. Furthermore, CBN does not alter the search space and therefore does not disrupt the normal convergence behavior of the EA. Additionally CBN is able to actually identify optima using the concept of species without further post-processing.

Since we used benchmark functions that do not benefit from EA crossover as an exploratory operator, except for the M4 function, it is no surprise that the simple Multi-Start-(1+1)ES strategy performs best on all benchmark functions. And although Fitness Sharing was allowed to make extensive use of the HC post-processing step, it never squares the performance of the Multi-Start-(1+1)ES. CBN on the other hand is actually able to equal the Multi-Start-(1+1)ES on M0 and M1, if suitable values for σ_{dist} are used. The failure of CBN on M2 and M3 indicates, that currently the exploratory elements of CBN are too weak.

Regarding the n-dimensional M4 benchmark function Fitness Sharing fails even in spite of the available exploratory crossover. Most likely because of the increasing problem dimension. CBN on the other hand scales rather well with the increase of problem dimension, but it is limited by the fixed population size and the fact that currently no exploratory interbreeding between species is implemented.

Therefore our future work will concentrate on introducing exploratory interbreeding between species to the CBN-EA, to take the full advantage of the positive properties of niching EAs. And we will focus on problems where exploratory crossover yields a greater benefit, otherwise Multi-Start Local-Search strategies will most likely prevail over an niching EA.

To free resources for interspecies crossover within the CBN-EA, we will introduce mechanisms that use redundant individuals from extremely large species or already converged species for reinitialization. If the reinitialized individuals are descendants from parents from different competing species, we are also able to introduce a additional selection pressure on species.

Acknowledgements

This research has been funded by the German Federal Ministry of Research and Education (BMBF) under contract No. 03C0309E.

References

1. T. Baeck, D. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation*. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.
2. D. Beasley, D. Bull, and R. Martin. A sequential niche technique for multimodal function optimization. *Evolutionary Computation*, 1(2):101–125, 1993.
3. P. Darwen and X. Yao. How good is fitness sharing with a scaling function. Technical Report CS 8/95, University College, The University of New South Wales, Canberra, April 1995.
4. K. De Jong. An analysis of the behavior of a class of genetic algorithms. (Doctoral dissertation, University of Michigan) Dissertation Abstracts International, 36(10), 5140B. (University Microfilms No. 76-9381), 1975.
5. M. Ester, H.-P. Kriegel, J. Sander, and X. Xiaowei. A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, and U. Fayyad, editors, *2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, Oregon, 1996. AAAI Press.
6. J. Gan and K. Warwick. A genetic algorithm with dynamic niche clustering for multimodal function optimisation. (Internal Report No. 98-001) Cybernetics Dept, Reading University, 1998.
7. D. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In Grefenstette, editor, *Proceedings of the 2nd International Conference on Genetic Algorithms*, pages 41–49, 1987.
8. V. Hanagandi and M. Nikolaou. A hybrid approach to global optimization using a clustering algorithm in a genetic search framework. *Computers and Chemical Engineering*, 22(12):1913–1925, 1998.
9. N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaption. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pages 312–317, 1996.
10. S. W. Mahfoud. Crowding and preselection revisited. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature II*, pages 27–36, Amsterdam, 1992. North-Holland.
11. S. W. Mahfoud. Genetic drift in sharing methods. In *Proceedings of the 1st IEEE Conference on Evolutionary Computation*, volume 1, pages 67–72, Piscataway, NJ, 1994. IEEE Service Center.

12. S. W. Mahfoud. Population sizing for sharing methods. Technical Report IlliGAL Report No 94005*, 1994.
13. S. W. Mahfoud. A comparison of parallel and sequential niching methods. In L. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 136–143, San Francisco, CA, 1995. Morgan Kaufmann.
14. J. Poland and A. Zell. Main vector adaptation: A cma variant with linear time and space complexity. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1050–1055. Morgan Kaufman, 2001.
15. J. Sander, M. Ester, H.-P. Kriegel, and X. Xiaowei. Density-based clustering in spatial databases, the algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery*, 2(2):169–194, June 1998.
16. A. Törn. Cluster analysis using seed points and density-determined hyperspheres as an aid to global optimization. *IEEE Transactions on Systems, Man and Cybernetics*, 7:610–616, 1977.
17. A. Törn. A search-clustering approach to global optimization. *Towards Global Optimization II*, 1978.
18. S. Tsutsui and Y. Fujimoto. Forking genetic algorithm with blocking and shrinking modes (FGA). In S. Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 206–215, San Mateo, CA, 1993. Morgan Kaufman.
19. R. K. Ursem. Multinational evolutionary algorithms. In P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 3, pages 1633–1640, Mayflower Hotel, Washington D.C., USA, 6-9 1999. IEEE Press.
20. R. K. Ursem. Multinational GAs: Multimodal optimization techniques in dynamic environments. In D. Whitley, D. E. Goldberg, E. Cantu-Paz, L. Spector, I. Parmee, and H.-G. Beyer, editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 19–26, Las Vegas, Nevada, USA, 10-12 2000. Morgan Kaufmann.
21. X. Yin and N. Germany. A fast genetic algorithm with sharing using cluster analysis methods in multimodal function optimization. In *Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms*, pages 450–457, Innsbruck, Austria, 1993.

Appendix: Benchmark Functions

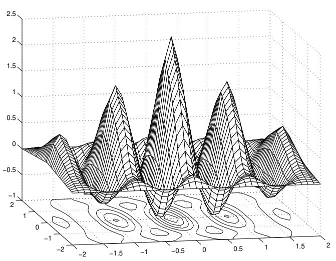


Fig. 10. Benchmark function M0

M0: Five hills and four valleys (5 peaks), as suggested in [19]:

$$\begin{aligned} M0(x,y) = {} & sin(2.2\pi x + 0.5\pi) \\ & \cdot \frac{2 - abs(y)}{2} \cdot \frac{3 - abs(x)}{2} \\ & + sin(0.5\pi y^2 + 0.5\pi) \\ & \cdot \frac{2 - abs(y)}{2} \cdot \frac{3 - abs(x)}{2} \end{aligned} \tag{5}$$

where $-2 \leq x, y \leq 2$

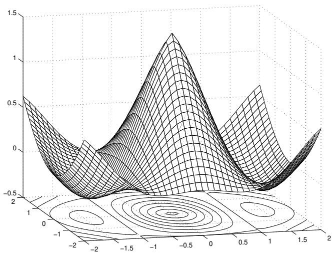


Fig. 11. Benchmark function M1

M1: One center peak and four neighbors (5 peaks), as suggested in [19]:

$$M1(x,y) = 3sin((0.5x\pi + 0.5\pi) \cdot \frac{2-\sqrt{x^2+y^2}}{4}) \quad (6)$$

where $-2 \leq x, y \leq 2$

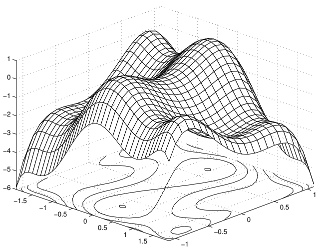


Fig. 12. Benchmark function M2

M2: Six hump camel back (6 peaks), as suggested in [19]:

$$M2(x,y) = -((4 - 2.1x^2 + \frac{x^4}{3})x^2 + xy + (-4 + 4y^2)y^2) \quad (7)$$

where $-1.9 \leq x \leq 1.9$ and $-1.1 \leq y \leq 1.1$

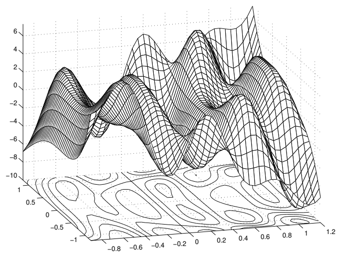


Fig. 13. Benchmark function M3

M3: Waves (10 peaks), as suggested in [19]:

$$M3(x,y) = -(y^2 - 4.5y^2)xy - 4.7cos(3x - y^2(2+x)) \cdot sin(2.5\pi x) + (0.3x)^2 \quad (8)$$

where $-0.9 \leq x \leq 1.2$ and $-1.2 \leq y \leq 1.2$

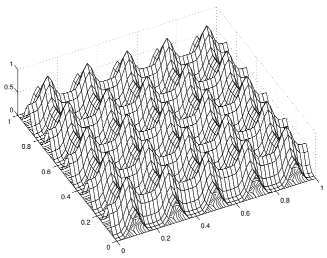


Fig. 14. Benchmark function M5

M5: n-dimensional sine (5^n peaks):

$$M5(\bar{x}) = 1 - \frac{1}{n}\sum_{i=1}^{n}(1 - sin^6(5\pi x_i)) \quad (9)$$

where $0 \leq x_i \leq 1$, $n = 2$