Adaptive Niche Radii and Niche Shapes Approaches for Niching with the CMA-ES

Ofer M. Shir oshir@liacs.nl
Natural Computing Group, Leiden University, Leiden, The Netherlands

Michael Emmerich emmerich@liacs.nl
Natural Computing Group, Leiden University, Leiden, The Netherlands

Thomas Bäck baeck@liacs.nl
Natural Computing Group, Leiden University, Leiden, The Netherlands
NuTech Solutions, 44227 Dortmund, Germany
http://natcomp.liacs.nl

Abstract

While the motivation and usefulness of niching methods is beyond doubt, the relaxation of assumptions and limitations concerning the hypothetical search landscape is much needed if niching is to be valid in a broader range of applications. Upon the introduction of radii-based niching methods with derandomized evolution strategies (ES), the purpose of this study is to address the so-called niche radius problem. A new concept of an adaptive individual niche radius is applied to niching with the covariance matrix adaptation evolution strategy (CMA-ES). Two approaches are considered. The first approach couples the radius to the step size mechanism, while the second approach employs the Mahalanobis distance metric with the covariance matrix mechanism for the distance calculation, for obtaining niches with more complex geometrical shapes. The proposed approaches are described in detail, and then tested on high-dimensional artificial landscapes at several levels of difficulty. They are shown to be robust and to achieve satisfying results.

Keywords

Multi-global optimization, niching, evolution strategies, covariance matrix adaptation, niche radius problem, self-adaptation, Mahalanobis metric.

1 Introduction

Evolutionary algorithms (EAs) have the tendency to converge quickly into a single solution (Mahfoud, 1995; Bäck, 1996), that is, all the individuals of the artificial population evolve to become nearly identical. This can be attributed to fast takeover (Bäck, 1994), and to the effects of genetic drift and disruptive recombination (Preuss et al., 2005). Given a problem with multiple solutions, the traditional EAs will usually locate a single solution. This is the desired result for many complex tasks, but a problem arises when multimodal objective functions are considered, and the discovery of all local optima is required.

Niching methods are used to extend EAs to multimodal optimization. They address the loss of diversity and the takeover effects of traditional EAs by maintaining certain properties within the population of feasible solutions. The study of niching is

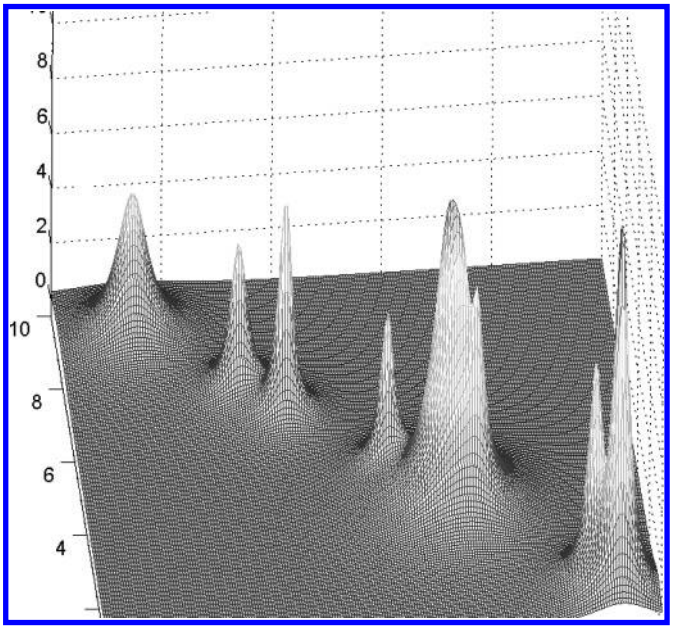

Figure 1: The Shekel function (see, e.g., Törn and Zilinskas, 1987) in a 2D decision space: introducing a dramatically uneven spread of optima; for more details see Table 1.

challenging both from the theoretical point of view and from the practical point of view. The theoretical challenge is two-fold, maintaining the diversity within a population-based stochastic algorithm from the computational perspective, but also having an insight into speciation theory from the biological perspective. The practical aspect provides a real-world motivation for this problem as there is an increasing interest from the applications' community in providing the decision maker with multiple solutions with different conceptual designs, for single- or multi-criterion search spaces (see, e.g., Avigad et al., 2004).

1.1 Motivation: The Niche Radius Problem

Niching techniques are often subject to criticism due to the so-called niche radius problem. The majority of the niching methods make an assumption concerning the fitness landscape, stating that the optima are far enough from one another with respect to some threshold distance, called the niche radius, which is estimated for the given problem and remains fixed during the course of evolution. Most prominently, the niche radius is used in the so-called sharing function, which penalizes fitness values of individuals whose distance to other individuals is below that threshold (Goldberg and Richardson, 1987). Obviously, there are landscapes for which this assumption is not applicable, and where this approach is most likely to fail (see Figures 1 and 2 for illustrations). Generally speaking, the task of defining a generic basin of attraction seems to be one of the most difficult problems in the field of global optimization, and there have been few attempts to treat it theoretically (Törn and Zilinskas, 1987).

In the literature, several GA niching sharing-based techniques can be found (Goldberg and Richardson, 1987; Yin and Germany, 1993; Mahfoud, 1995; Jelasity, 1998;

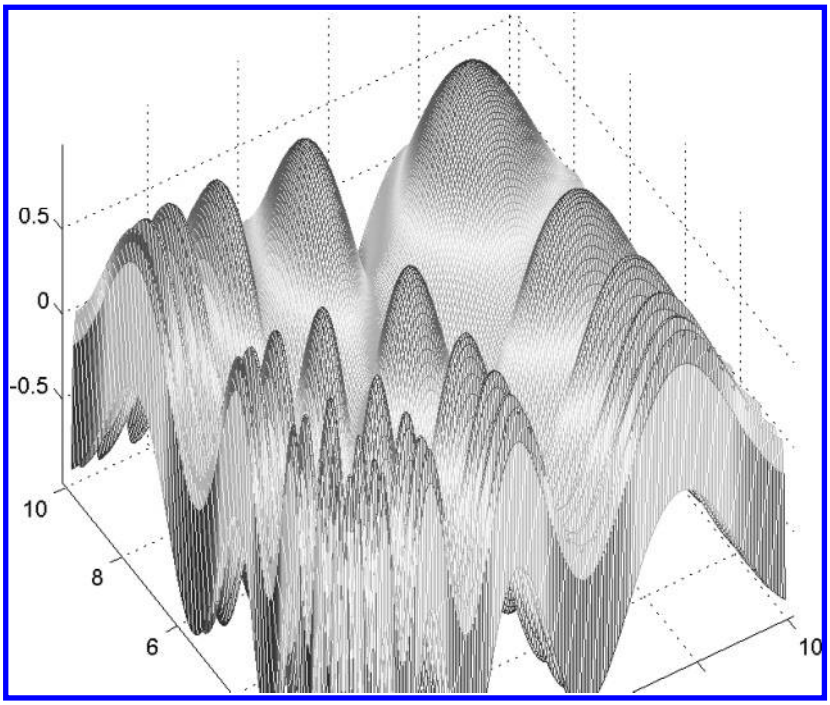

Figure 2: The Vincent function in a 2D decision space: a sine function with a decreasing frequency.

Miller and Shaw, 1996; Cioppa et al., 2004). As alternatives to sharing methods, clustering algorithms for niche formation were also proposed (Hanagandi and Nikolaou, 1998; Ursem, 1999; Branke, 2001; Gan and Warwick, 2001; Streichert et al., 2003; Ando et al., 2005; Schönemann et al., 2004; Stoean et al., 2005). Other approaches, such as crowding (Deb and Tiwari, 2005), or using tag-bits for species formation (Deb and Spears, 1997) aim for diversity maintenance in multimodal optimization, too. Within evolution strategies (ES; Bäck, 1996; Beyer and Schwefel, 2002), a canonical EA for continuous function optimization, several, relatively new, niching methods have been proposed (Aichholzer et al., 2000; Streichert et al., 2003; Schönemann et al., 2004; van der Goes et al., 2008). The introduction of quasi-sharing methods into self-adaptive ES, including the CMA-ES, has been put forward by Shir and Bäck (2005, 2006, 2009). Upon their successful application to high-dimensional artificial landscapes, they were also successfully applied to a challenging problem in quantum control (Shir et al., 2006). In that application, the niching technique was shown to be clearly qualitatively superior in comparison to multiple restart runs with a single population, in locating highly-fit unique optima that had not been obtained otherwise, and represented different conceptual designs. Here, a radius based approach was used, and the choice of the radius value was based on theoretical considerations about the application problem.

1.2 The Niche Radius Problem: Related Work

There were several GA-oriented studies that addressed this so-called niche radius problem, aiming to relax the assumption specified earlier, or even to drop it completely.

Jelasity (1998) suggested a cooling-based mechanism for the niche radius, which adapts the global radius as a function of time during the course of evolution. Gan and Warwick (2001) introduced the so-called dynamic niche clustering, to overcome the radius problem by using a clustering mechanism. A complex subpopulation differentiation model, the so-called multinational evolutionary algorithms, was presented by Ursem (1999). It introduces a topological-based auxiliary mechanism of sampling, which detects whether feasible solutions share the same basin of attraction. A recent study by Stoean et al. (2007) considered the hybridization of the latter with a radii-based niching method, previously proposed by Stoean et al. (2005). Finally, an iterative statistical-based approach has lately been introduced (Cioppa et al., 2004) for learning the optimal niche radius without a priori knowledge of the landscape. It considers the *fitness sharing* strategy, and optimizes it as a function of the population size and the niche radius, without relaxing the landscape assumption specified earlier, that is, the niches are eventually obtained using a single fixed niche radius.

1.3 This Study

Our study introduces a new concept into the niche radius problem, inspired by the ES self-adaptation concept, an adaptive individual niche radius. The idea is that each individual, that is, feasible solution in the artificial population, updates every generation a niche radius along with its adaptive strategy parameters. This study is an adaptive extension to niching with the covariance matrix adaptation evolution strategy (Shir and Bäck, 2009). Two new approaches are presented here. The first exploits the self-adaptation of the step size in the CMA-ES mechanism, the cumulative step size adaptation (CSA) mechanism, and couples the individual niche radius to it. Since the step size does not hold any further spatial information concerning the landscape, the classification into niches uses hyperspheres, based on the Euclidean distance metric. The second approach introduces the Mahalanobis distance metric into the niching mechanism. The aim is to allow more accurate spatial classification of niches by working with rotatable ellipsoid shapes as classification regions. We propose to obtain the shapes of these ellipsoids from the covariance matrix of the evolving multivariate normal distribution adapted by the CMA mutation scheme. This will replace the default classification by means of Euclidean hyperspheres. This idea can be implemented in a straightforward manner using the Mahalanobis distance metric as a replacement for the Euclidean distance metric. This is due to the fact that the Mahalanobis distance metric, the hyperspheres of which are ellipsoids when viewed in the Euclidean space, is parameterized by a covariance matrix which is obtained without additional cost from the CMA. The two new approaches are tested with the CMA-($\dagger$,) routines, and evaluated on a suite of artificial landscapes, including problems with an uneven spread of optima as well as with non-isotropic attractor basins.

The remainder of the paper is organized as follows. Section 2 reviews the covariance matrix adaptation evolution strategy (CMA-ES), the evolutionary search engine of the algorithms in this study. Niching techniques that are relevant to this study are discussed in Section 3, reviewing their components which are used to construct the new approaches presented here. Section 4 presents our two new approaches of niche radius adaptation for niching with the CMA-ES. The experimental procedure, including the setup and the numerical results, are all presented in Section 5. Section 6 draws conclusions, summarizes this study, and offers possible directions of future research.

2 The Covariance Matrix Adaptation Evolution Strategy

In standard evolution strategies, mutative step size control tends to work well for the adaptation of a global step size, but tends to fail when it comes to the individual step sizes. This is due to several disruptive effects (Hansen and Ostermeier, 2001) as well as to the fact that the selection of the strategy parameters setting is indirect, that is, the vector of a successful mutation is not used to adapt the step size parameters, but rather the parameters of the distribution that led to this mutation vector. The so-called derandomized mutative step-size control aims to tackle those disruptive effects. The covariance matrix adaptation evolution strategy is considered to be the state of the art derandomized-ES variant, being the fourth proposed variant, in a succession of three derandomized variants (Ostermeier et al., 1993, 1994b; Hansen et al., 1995b).

It is important to note, in particular in the context of this study, that the CMA-ES is composed of two adaptation phases: the cumulative step-size adaptation (CSA) mechanism, which is based upon the path length control, as well as the actual covariance matrix adaptation (CMA) mechanism, which is based upon the evolution path. The CMA learns the covariance matrix based on the successful mutations, whereas the CSA learns the optimal global step size, using the projection of the successful mutation on the rotated axis of the decision space. See Beyer and Arnold (2003) for an analysis of the two components.

2.1 (1, λ)-CMA-ES

We consider the rank-one update with cumulation covariance matrix adaptation evolution strategy (Hansen and Ostermeier, 2001). This advanced method applies principal component analysis (PCA) to the selected mutations during the evolution, also referred to as the evolution path, for the adaptation of the covariance matrix of the distribution.

$\mathbf{C}^{(g)} \in \mathbb{R}^{n\times n}$ is the covariance matrix, which is eigenvalue-decomposed here to $\mathbf{C}^{(g)} = \mathbf{B}^{(g)}\mathbf{D}^{(g)}(\mathbf{B}^{(g)}\mathbf{D}^{(g)})^T$; Let $k = 1\ldots\lambda$ indicate the index of the kth offspring, and let the subscript "sel" refer to the selected offspring. Then the generation update rule reads:

$$\vec{x}_k^{(g+1)} = \vec{x}^{(g)} + \sigma^{(g)}\mathbf{B}^{(g)}\mathbf{D}^{(g)}\vec{z}_k^{(g+1)} \tag{1}$$

$$H_\sigma^{(g+1)} = \begin{cases} 1 & \text{if } \dfrac{\left\|\vec{p}_\sigma^{(g)}\right\|}{\sqrt{1-(1-c_\sigma)^2}} < H_{\text{thresh}} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$\vec{p}_c^{(g+1)} = (1-c_c)\cdot\vec{p}_c^{(g)} + H_\sigma^{(g+1)}\sqrt{c_c(2-c_c)}\cdot\mathbf{B}^{(g)}\mathbf{D}^{(g)}\vec{z}_{\text{sel}}^{(g+1)} \tag{3}$$

$$\mathbf{C}^{(g+1)} = (1-c_{\text{cov}})\cdot\mathbf{C}^{(g)} + c_{\text{cov}}\cdot\vec{p}_c^{(g+1)}\left(\vec{p}_c^{(g+1)}\right)^T \tag{4}$$

$$\vec{p}_\sigma^{(g+1)} = (1-c_\sigma)\cdot\vec{p}_\sigma^{(g)} + \sqrt{c_\sigma(2-c_\sigma)}\cdot\mathbf{B}^{(g)}\vec{z}_{\text{sel}}^{(g+1)} \tag{5}$$

$$\sigma^{(g+1)} = \sigma^{(g)}\cdot\exp\left(\frac{c_\sigma}{d_\sigma}\cdot\left(\frac{\left\|\vec{p}_\sigma^{(g+1)}\right\|}{\mathbf{E}\left(\|\mathcal{N}(0,\mathbf{I})\|\right)} - 1\right)\right) \tag{6}$$

where $\vec{p}_c^{(g)} \in \mathbb{R}^n$ is the so-called evolution path, the crucial component for the adaptation of the covariance matrix, and $\vec{p}_\sigma^{(g)} \in \mathbb{R}^n$ is the conjugate evolution path, which is responsible for the control of the step size $\sigma^{(g)}$. c_c, c_{cov}, c_σ, and d_σ are learning/adaptation rates, and $H_{\text{thresh}} = (1.5 + \frac{1}{n-0.5})\mathbf{E}(\|\mathcal{N}(0, \mathbf{I})\|)$.

2.2 (1 + λ)-CMA-ES

This elitist version (Igel et al., 2006, 2007) of the original CMA-ES algorithm combines the classical $(1 + \lambda)$ ES strategy (Rechenberg, 1973; Bäck, 1996; Beyer and Schwefel, 2002) with the covariance matrix adaptation concept. The so-called success rule based step size control replaces the path length control of the $(1, \lambda)$-CMA strategy. The same notation as in Section 2.1 is used here:

$$\vec{x}_k^{(g+1)} = \vec{x}^{(g)} + \sigma^{(g)}\mathbf{B}^{(g)}\mathbf{D}^{(g)}\vec{z}_k^{(g+1)} \tag{7}$$

After the evaluation of the new generation, the success rate is updated $p_{\text{succ}}^{(g+1)} = \lambda_{\text{succ}}^{(g+1)}/\lambda$, where:

$$\bar{p}_{\text{succ}}^{(g+1)} = (1 - c_p) \cdot \bar{p}_{\text{succ}}^{(g)} + c_p \cdot p_{\text{succ}}^{(g+1)} \tag{8}$$

$$\sigma^{(g+1)} = \sigma^{(g)} \cdot \exp\left(\frac{1}{d} \cdot \left(\bar{p}_{\text{succ}}^{(g+1)} - \frac{p_{\text{succ}}^{\text{target}}}{1 - p_{\text{succ}}^{\text{target}}}\left(1 - \bar{p}_{\text{succ}}^{(g+1)}\right)\right)\right) \tag{9}$$

The covariance matrix is updated only if the selected offspring is better than the parent. Then,

$$\vec{p}_c^{(g+1)} = \begin{cases} (1 - c_c)\vec{p}_c^{(g)} + \sqrt{c_c(2 - c_c)} \cdot \dfrac{\vec{x}_{\text{sel}}^{(g+1)} - \vec{x}^{(g)}}{\sigma^{(g)}} & \text{if } \bar{p}_{\text{succ}}^{(g+1)} < p_\Theta \\ (1 - c_c)\,\vec{p}_c^{(g)} & \text{otherwise} \end{cases} \tag{10}$$

$$\mathbf{C}^{(g+1)} = \begin{cases} (1 - c_{\text{cov}}) \cdot \mathbf{C}^{(g)} + c_{\text{cov}} \cdot \vec{p}_c^{(g+1)}\left(\vec{p}_c^{(g+1)}\right)^T \\ \qquad \text{if } \bar{p}_{\text{succ}}^{(g+1)} < p_\Theta \\ (1 - c_{\text{cov}}) \cdot \mathbf{C}^{(g)} + c_{\text{cov}} \cdot \left(\vec{p}_c^{(g+1)}\left(\vec{p}_c^{(g+1)}\right)^T + c_c\,(2 - c_c)\,\mathbf{C}^{(g)}\right) \\ \qquad \text{otherwise} \end{cases} \tag{11}$$

All weighting variables and learning rates were applied as suggested by Hansen and Ostermeier (2001) and by Igel et al. (2006).

3 From Fitness Sharing to CMA-ES Niching

In this section, we describe the foundations for our study: previous niching techniques with certain components that are integrated in our approaches in Section 4. Sections 3.1 and 3.2 review the classical methods of fitness sharing and dynamic fitness sharing, respectively. Section 3.3 presents the niching with the CMA-ES algorithm, upon which our new approaches are mainly based.

3.1 Fitness Sharing

The fitness sharing approach (Goldberg and Richardson, 1987) was the pioneering GA niching method. Its idea is to consider the fitness as a shared resource and by that to aim at decreasing redundancy in the population. Given the similarity metric of the population, expressed by means of the distance d_{ij} between individuals i and j, which can be genotypic- or phenotypic-based, the sharing function is given by:

$$sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\rho}\right)^{\alpha_{sh}} & \text{if } d_{ij} < \rho \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where ρ (traditionally noted as σ_{sh}) is the fixed radius of every niche, and $\alpha_{sh} \geq 1$ is a control parameter, usually set to 1. Using the sharing function, the niche count is then defined as follows:

$$m_i = \sum_{j=1}^{N} sh(d_{ij}) \tag{13}$$

Given an individual raw fitness f_i, the shared fitness is then defined by:

$$f_i^{sh} = \frac{f_i}{m_i} \tag{14}$$

assuming that the fitness is strictly positive and subject to maximization.

3.2 Dynamic Niche Sharing

The dynamic niche sharing method (Miller and Shaw, 1996), which succeeded the fitness sharing method, aims to dynamically recognize the q peaks of the forming niches, and with this information to classify the individuals as either members of one of the niches, or as members of the "non-peaks domain." Explicitly, let us introduce the dynamic niche count:

$$m_i^{\text{dyn}} = \begin{cases} n_j & \text{if individual } i \text{ is within dynamic niche } j \\ m_i & \text{otherwise (non-peak individual)} \end{cases} \tag{15}$$

where n_j is the size of the jth dynamic niche (i.e., the number of individuals that were classified to niche j), and m_i is the standard niche count, as defined in Equation (13). The shared fitness is then defined as:

$$f_i^{\text{dyn}} = \frac{f_i}{m_i^{\text{dyn}}} \tag{16}$$

The identification of the dynamic niches can be done in the greedy approach, as proposed by Miller and Shaw (1996) as the dynamic peak identification (DPI) algorithm (see Algorithm 1).

Algorithm 1 Dynamic peak identification.

input: population Pop, (maximal) number of niches q, niche radius ρ

```
Sort Pop in decreasing fitness order
i := 1
NumPeaks := 0
DPS := ∅ {Set of peak elements in population}
while NumPeaks ≠ q and i ≤ popSize do
   if Pop[i] is not within sphere of radius ρ around peak in DPS then
      DPS := DPS ∪ {Pop[i]}
      NumPeaks := NumPeaks + 1
   end if
   i := i + 1
end while
```

output: DPS

3.3 CMA-ES Niching with Fixed Niche Radius

The advent of modern evolution strategies allows successful global optimization for a large range of practically relevant problems with minimal settings, mostly without recombination, and with a low number of function evaluations. In particular, consider the $(1 \dagger \lambda)$ CMA-ES presented in Sections 2.1 and 2.2. In the context of niching, the CMA-ES allows the construction of a fairly simple and elegant niching algorithm. Here, we provide the reader with details concerning the general niching framework and the fixed radius approach.

The niching technique is based upon interacting search processes, which simultaneously perform a derandomized $(1, \lambda)$ or $(1 + \lambda)$ search in different locations of the space. In case of multimodal landscapes, these search processes are meant to explore different attractor basins of local optima. The speciation interaction occurs every generation when all the offspring are considered together to become niches' representatives for the next iteration, or simply the next search points, based on the rank of their fitness and their location with respect to higher-ranked individuals.

Explicitly, given q, the estimated/expected number of peaks, $q + p$ D-sets are initialized, where a D-set is defined as the collection of all the dynamically adapted strategy as well as decision parameters of the derandomized algorithm, which uniquely define the search at a given point of time. These parameters are the current search point, the covariance matrix, the step size, as well as other auxiliary parameters. At every point in time, the algorithm stores exactly $q + p$ D-sets, which are associated with $q + p$ search points: q for the peaks and p for the non-peaks domain. The $(q + 1)$th$\ldots(q + p)$th D-sets are individuals that are randomly regenerated every epoch, that is, a cycle of κ generations, as potential candidates for niche formation. This is basically a semi-restart mechanism, which allows new niches to form dynamically. It should be noted that the total number of function evaluations allocated for a run is proportional to q, so setting the value of p reflects a trade-off between applying a wide restart approach for exploring further the search space to exploiting computational resources for the existing niches. In any case, due to the curse of dimensionality, higher values of p lose their significance as the dimension n of the problem gets higher, since random sampling in high dimensions becomes ineffective very quickly.

Until the stopping criterion is met, the following procedure takes place. Each search point samples λ offspring, based on its evolving D-set. After the fitness evaluation of the new $\lambda \cdot (q + p)$ individuals, the classification into niches of the entire population is done using the DPI routine (Miller and Shaw, 1996; see Algorithm 1)—based on the fixed niche radius ρ—and the peaks then become the new search points. Their D-sets are inherited from their parents and updated.

A pseudo code for the niching routine is presented as Algorithm 2.

Algorithm 2 CMA-ES niching with a fixed niche radius.

for $i = 1 \ldots (q + p)$ search points **do**
  Generate λ samples based on the D-set of i
end for
Evaluate fitness of the population
Compute the dynamic peak set (DPS) with the DPI algorithm (Algorithm 1)
for all elements of DPS **do**
  Set peak as a search point
  Inherit the D-set and update it
end for
if $N_{\text{DPS}} =$ size of DPS $< q$ **then**
  Generate $q - N_{\text{DPS}}$ new search points, reset D-sets
end if
if $gen \bmod \kappa \equiv 0$ **then**
  Reset the $(q + 1)$th $\ldots (q + p)$th search points
end if

3.4 Fixed Niche Radius Calculation

Whenever one cannot make a reasonable choice for the niche radius from a priori knowledge of the application problem, a default value would be desirable. The original formula for ρ for phenotypic sharing in GAs was derived by Deb and Goldberg (1989). Analogously, by considering the ES decision parameters as the decoded parameter space of the GA (phenotype), the same formula can be applied, using the same distance metric. Obviously, this would not have been the case, had it been derived for genotypic sharing, using a discrete metric such as the Hamming metric.

This approach chooses a minimal niche radius, such that the entire search space can be covered by a specific number of spheres of that radius. Explicitly, given q, every niche is thus considered to be surrounded by an n-dimensional hypersphere with radius ρ that occupies $\frac{1}{q}$ of the entire volume of the space. The volume of the hypersphere that contains the entire space is

$$V = cr^n \tag{17}$$

where c is a constant, given explicitly by:

$$c = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2} + 1)}, \qquad \Gamma(n) = \int_0^\infty x^{n\text{-}1} \exp(-x) dx \tag{18}$$

Given lower and upper bound values $x_{k,\min}$, $x_{k,\max}$ of each coordinate in the decision parameters space, r is defined as follows:

$$r = \frac{1}{2}\sqrt{\sum_{k=1}^{n}(x_{k,\max} - x_{k,\min})^2} \tag{19}$$

If we divide the volume into q parts, we may write

$$c\rho^n = \frac{1}{q}cr^n \tag{20}$$

which yields

$$\rho = \frac{r}{\sqrt[n]{q}} \tag{21}$$

Hence, by applying this niche radius approach, two assumptions are made:

1. The expected/desired number of peaks, q, is given or can be estimated.
2. All peaks are at least a distance 2ρ from each other, where ρ is the fixed radius of every niche.

4 Adaptive Niche Radii Approaches

While the motivation and usefulness of niching is beyond doubt, the relaxation of assumptions and limitations concerning the hypothetical search landscape is much needed if niching methods are to be valid over a broader range of applications. We choose to treat the particular limiting assumption of the fixed niche radius by introducing self-adapting niche-radii and niche-shapes mechanisms. In this section, we present two new approaches for the adaptation of the niche radius, in the framework of niching with the CMA-ES. Section 4.1 introduces the self-adaptive niche radius mechanism which is based upon the coupling to the step size, first introduced by Shir and Bäck (2006), and Section 4.2 introduces niching with the Mahalanobis distance metric, relying on the evolving covariance matrix.

4.1 Self-Adaptive Radius: Step Size Coupling

Aiming to follow the successful mechanism of the step size adaptation, the idea of this approach is to couple the niche radius to the global step size σ, whereas the indirect selection of the niche radius is governed by the objective that every niche should ideally consist of λ individuals. This is implemented by means of a quasi dynamic fitness sharing mechanism. A detailed description follows.

The CMA-ES niching method is used as outlined earlier (Section 3.3), with the following modifications. q is given as an input to the algorithm, but it is now merely a prediction or a demand for the maximal number of solutions the decision maker would like to obtain. Given the ith individual in the population, a niche radius in generation g, denoted by $\rho_i^{(g)}$, is initialized by means of a rule ($\rho_i^{(0)} = \sqrt{n} \cdot \sigma_{\text{init}}$) in the beginning of the search. Its update step in generation $(g+1)$ is based upon both the parent's radius

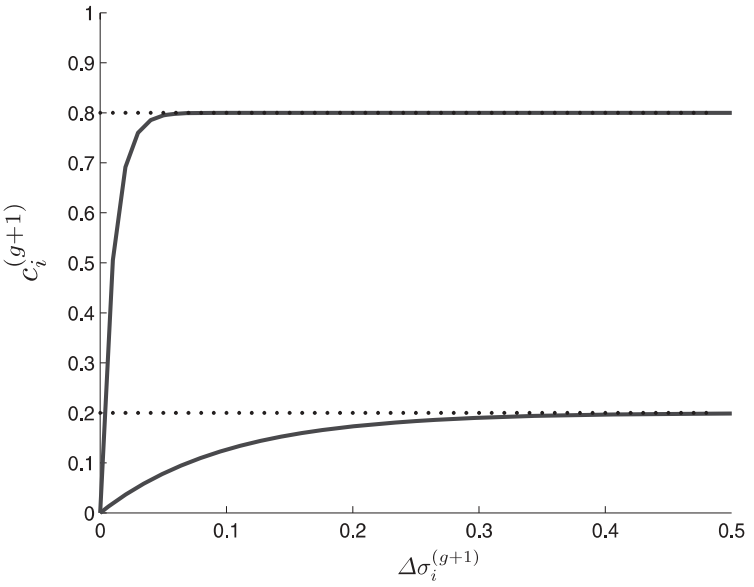

Figure 3: The learning coefficient $c_i^{(g+1)}$ [Equation (23)] is plotted as a function of the step size derivative, $\Delta\sigma_i^{(g+1)}$, for the two strategies, as derived from Equation (24): (γ, α) are substituted for the two strategies: $\{(\frac{1}{5}, 10), (\frac{4}{5}, 100)\}$.

and its step size:

$$\rho_i^{(g+1)} = \left(1 - c_i^{(g+1)}\right) \cdot \rho_{\text{parent}}^{(g)} + c_i^{(g+1)} \cdot \sqrt{n} \cdot \sigma_{\text{parent}}^{(g+1)} \tag{22}$$

where $c_i^{(g)} \in [0, 1)$ is the individual learning coefficient, which is updated according to the value $\Delta\sigma_i^{(g+1)} = |\sigma_{\text{parent}}^{(g+1)} - \sigma_{\text{parent}}^{(g)}|$ (see Figure 3 for an illustration):

$$c_i^{(g+1)} = \gamma \cdot \left(1 - \exp\left\{-\alpha \cdot \Delta\sigma_i^{(g+1)}\right\}\right) \tag{23}$$

γ and α are set differently for the two selection strategies:

$$\gamma = \begin{cases} \frac{1}{5} & \text{for } (1, \lambda)\text{-selection} \\ \frac{4}{5} & \text{for } (1 + \lambda)\text{-selection} \end{cases} \qquad \alpha = \begin{cases} 10 & \text{for } (1, \lambda)\text{-selection} \\ 100 & \text{for } (1 + \lambda)\text{-selection} \end{cases}. \tag{24}$$

γ determines the saturation value of the learning coefficient: strong coupling to the parent's step size for the plus strategy, versus a weak coupling for the comma strategy. α dictates the strength of the exponential convergence toward the saturation value: slow convergence for the plus strategy, versus a rapid convergence for the comma strategy. This rule for parametric setting works reliably on a wide range of problems, as we will show later. The rationale behind it stems from the different niching convergence behavior of the two strategies, as will be discussed in Section 5.1. Furthermore, we discuss the use of new parameters in Section 6.

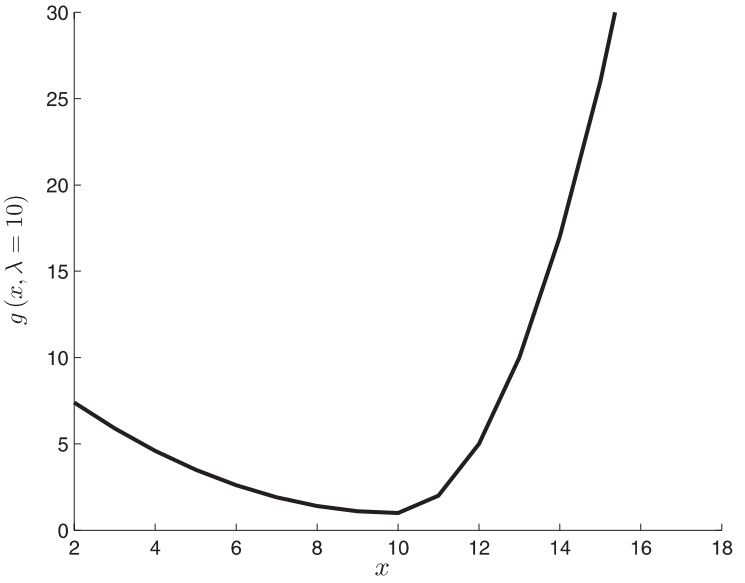

Figure 4: An illustration for $g(x, \lambda = 10)$ [Equation (25)]: a parabola with uneven branches; the niche fitness [Equation (26)] is penalized more for an overpopulated niche ($x > \lambda$) due to the steep branch, in comparison to an underpopulated niche.

The DPI algorithm is run using the individual niche radii, for the identification of the peaks and the classification of the population.

Furthermore, we introduce:

$$g(x, \lambda) = 1 + \Theta(\lambda - x) \cdot \frac{(\lambda - x)^2}{\lambda} + \Theta(x - \lambda) \cdot (\lambda - x)^2 \tag{25}$$

where $\Theta(y)$ is the Heaviside step function. Given the population size λ, $g(x, \lambda)$ is a parabola with unequal branches, centered at ($x = \lambda$, $g = 1$; see Figure 4 for illustration). The justification for its geometrical asymmetry will be described shortly.

Then, by applying the calculation of the dynamic niche count m_i^{dyn} [Equation (15)], based on the appropriate radii, we define the niche fitness of individual i by:

$$f_i^{\text{niche}} = \frac{f_i}{g\left(m_i^{\text{dyn}}, \lambda\right)} \tag{26}$$

We assume, again, that the raw fitness is strictly positive and subject to maximization. Finally, the selection of the next parent in each niche is based on this niche fitness.

Equation (26) enforces the requirement for having a fixed resource of λ individuals per niche, since $g(x, \lambda)$ yields values greater than 1 for any niche count different from λ. The asymmetry of $g(x, \lambda)$ is therefore meant to penalize more those niches that exceeded λ members, in comparison to those with less than λ members. This equation is a variant of the dynamic shared fitness [Equation (16)], and is used now in the context of niche radius adaptation.

The pseudo code for the self-adaptive niching routine is presented in Algorithm 3.

Algorithm 3 CMA-ES niching with an adaptive niche radius.

for $i = 1 \ldots (q + p)$ search points **do**
  Generate λ samples based on the D-set of i
  Update the niche radius $\rho_i^{(g+1)}$ [Equation (22)]
end for
Evaluate fitness of the population
Compute the dynamic peak set with the DPI algorithm, based on individual radii
Compute the dynamic niche count of every individual
for all elements of DPS **do**
  Compute the niche fitness [Equation (26)]
  Set individual with best niche fitness as a search point
  Inherit the D-set and update it respectively
end for
if $N_{\text{DPS}} =$ size of DPS $< q$ **then**
  Generate $q - N_{\text{DPS}}$ new search points, reset D-sets
end if
if $gen \bmod \kappa \equiv 0$ **then**
  Reset the $(q + 1)$th $\ldots (q + p)$th search points
end if

4.2 Mahalanobis Distance Metric: Covariance Exploitation

Existing niching techniques, and in particular those presented in Sections 3.3 and 4.1, use the Euclidean distance metric in the decision space for the classification of feasible solutions to the niches under formation. This approach is likely to encounter problems in high-dimensional landscapes with non-isotropic basins of attraction. Since the CMA-ES algorithm already learns the covariance matrix of the decision space distribution, it is worthwhile to use it for a better spatial classification mechanism within the niching framework. In essence, this can be considered as an upgrade of the niching mechanism, as it captures a more accurate spatial formation of the niches. Most importantly, this approach is also self-adaptive.

After giving this motivation, we proceed to discussing the details of this idea.

4.2.1 The Mahalanobis Distance Metric

Consider the Mahalanobis distance metric, for instance, in a probability distribution. Given a mean vector $\vec{m}$ and a covariance matrix Σ, the Mahalanobis distance of a vector $\vec{v}$ from the *mean vector* is defined as:

$$d(\vec{v}, \vec{m}) = \sqrt{(\vec{v} - \vec{m})^T \Sigma^{-1} (\vec{v} - \vec{m})} \tag{27}$$

It can be shown that the iso-distance surfaces of this metric are ellipsoids that are centered about the mean $\vec{m}$. In the special case where $\Sigma \sim \mathbf{I}$ (e.g., features are uncorrelated and all variances equal) the Mahalanobis distance metric is equivalent to the normalized Euclidean distance metric, and the iso-distance surfaces become Euclidean hyperspheres. Though the Mahalanobis distance metric is typically applied in statistics, it can also be applied in different contexts as a metric on vector spaces given a positive-semidefinite and symmetric matrix Σ determining the elliptic iso-distance surfaces.

4.2.2 Mahalanobis CMA-ES Niching

In the context of niching, given an individual $\vec{x}$, representing a niche with a covariance matrix $\mathbf{C}_x$, we choose to define, accordingly, the Mahalanobis distance metric of an individual $\vec{y}$ to the niche by

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \mathbf{C}_x^{-1} (\vec{x} - \vec{y})}.$$

Since different individuals have different covariance matrices, this operation is asymmetric. Hence, the actual classification into niches depends not only on the identity of the so-called peak individuals, which are selected according to their better fitness, but is also based upon their individual covariance matrices. Due to the fact that the classification itself is done differently with individual distance metrics, an equivalent classification with the Euclidean distance metric could possibly result in a different outcome when compared to this approach.

Notably, the proposed routine does not have a secondary selection mechanism, which was necessary for the self-adaptive niche radius approach, as introduced in Section 4.1. The reason why it is not required here is that the local shape of the attractor basins, as approximated by the CMA, is equivalent to the desired shape for the niches, and thus suffices for successful classification of individuals to the niche.

4.2.3 Numerical Implementation

As for the technical details, we discuss here the numerical implementation of the Mahalanobis distance metric, considering the matrix inversion which is required. We show here that the matrix inversion, in this context, can be replaced by matrix multiplication, which leads to a significant performance gain for the dimensions that are commonly under study.

In the CMA-ES mechanism, the eigenvalue-decomposition of the covariance matrix $\mathbf{C}$, which is calculated every generation, yields

$$\mathbf{C} = \mathbf{BD}\,(\mathbf{BD})^T \tag{28}$$

where $\mathbf{D} = \mathrm{diag}(\sqrt{\Lambda_1}, \sqrt{\Lambda_2}, \ldots, \sqrt{\Lambda_n})$, with the eigenvalues $\{\Lambda_i\}_{i=1}^n$. Upon inversion, one can derive,

$$\mathbf{C}^{-1} = [\mathbf{BD}(\mathbf{BD})^T]^{-1} = \mathbf{B}^{T-1}\mathbf{D}^{T-1}\mathbf{D}^{-1}\mathbf{B}^{-1} = \mathbf{B} \cdot \mathrm{diag}\left(\frac{1}{\Lambda_1}, \frac{1}{\Lambda_2}, \ldots, \frac{1}{\Lambda_n}\right) \cdot \mathbf{B}^T \tag{29}$$

and thus the matrix inversion calculation can be replaced, within the CMA-ES routine, with a matrix multiplication calculation. Despite the fact that these two operations are equivalent in terms of numerical complexity (see, e.g., Giorgi et al., 2003), we observe in practice a difference between the two procedures for obtaining $\mathbf{C}^{-1}$. For dimensions up to $n = 30$, it is observed that the multiplication procedure takes on average half the CPU calculation time in comparison to the inversion procedure.[1] Hence, it pays off to follow the derivation given here.

[1]The calculations were carried out in Matlab 7.0.

Due to numerical features of the eigenvalue-decomposition, which were also discussed by Hansen and Ostermeier (2001, p. 20), but are crucial here for the inversion operation of the covariance matrix, we choose to introduce a lower bound to the eigenvalues: $\Lambda_{\min} = 10^{-10}$.

4.2.4 Self-Adaptive Mahalanobis Approach

The self-adaptive niche radius mechanism presented in Section 4.1 can easily be adjusted to employ the Mahalanobis distance metric for the classification of the niches. In the context of this study, it would become a hybrid approach, so to speak, and it will be tested in the experimental phase.

4.3 Sizing the Population

We follow the recommended population size for $(1, \lambda)$ derandomized ES (see, e.g., Ostermeier et al., 1994a), and set $\lambda = 10$. On this note, we would like to mention a theoretical work on sizing the population in a derandomized $(1, \lambda)$ ES with respect to the local progress (Hansen et al., 1995a). The latter work obtained theoretical results showing that the local serial progress is maximized when the expected progress of the second best individual vanishes. These results allowed for the construction of a population size adaptation scheme, which sets the value of λ as a function of the fitness difference of the second fittest offspring and its parent. This adaptation scheme was shown to perform well on a set of simple theoretical landscapes (Hansen et al., 1995a). However, for the sake of simplicity, we choose to restrict our study to a fixed population size. We shall further comment on studying population sizes within the current niching frameworks in Section 6.

5 Experimental Procedure

We describe here the experimental phase of this study. We begin by introducing the niching performance criteria, and proceed with the numerical results.

5.1 MPR Analysis

Our research focuses on the ability to identify global as well as local optima and to converge in those directions through time, with no particular interest in the distribution of the population. Thus, as has been done in earlier studies of GA niching by Miller and Shaw (1996), we adopt the performance metric called the maximum peak ratio statistic. This metric measures the quality of optima given as a final result by the evolutionary algorithm. Explicitly, assuming a minimization problem, given the fitness values of the optima in the final population $\{\tilde{f}_i\}_{i=1}^{q}$, and the values of the real optima of the objective function $\{\hat{\mathcal{F}}_i\}_{i=1}^{q}$, the maximum peak ratio is defined as follows:

$$\mathrm{MPR} = \frac{\sum_{i=1}^{q} \hat{\mathcal{F}}_i}{\sum_{i=1}^{q} \tilde{f}_i} \tag{30}$$

where all values are assumed to be strictly positive. If this is not the case in the original parameterization of the landscape, the latter should be scaled accordingly with an additive constant for the sake of this calculation. Also, given a maximization problem, the MPR is defined as the obtained optima divided by the real optima. A drawback of

this performance metric is that the values of the real optima need to be known a priori. However, for many artificial test problems, these can be derived analytically, or tight numerical approximations to them are available. Also, values of optima in real world landscapes are often known.

Although this metric was originally introduced to be analyzed by means of the saturation MPR value for performance evaluation, a new perspective was introduced by Shir and Bäck (2005). That recent study investigated the MPR as a function of time, focusing on the early stages of the run, for investigating the behavior of the niching process. It was shown experimentally that the time-dependent MPR data fits a theoretical function, the logistic curve:

$$y(t) = \frac{a}{1 + \exp\{c(t - T)\}} \tag{31}$$

where a is the saturation value of the curve, T is its time shift, and c (in this context always negative) determines the strength of the exponential rise.

In the context of evolutionary niching methods, it was argued in that study that the logistic parameters should be interpreted in the following way, with T as the learning period of the algorithm, and the absolute value of c as its niching formation acceleration. It focused on a comparison between the CMA-ES to the standard-ES in the context of niching. Here, some of the conclusions of that study are outlined:

1. The niching formation acceleration, expressed as the absolute value of c, had larger values for the CMA-ES mechanism for all the test cases. That implied stronger niching acceleration and faster convergence.

2. A trend concerning the absolute value of c as a function of the dimensionality was observed: the higher the dimensionality, the lower the absolute value of c.

3. The learning period, expressed as the value of T in the curve fitting, took negative as well as positive values. Negative values mean that the niche formation process, expressed as the exponential rise of the MPR, started immediately from generation zero.

4. The averaged saturation value a was larger in all of the test cases for the CMA-ES mechanism. This result also supported the claim that the CMA-ES had a faster convergence, as it got better fitness values earlier.

The study concluded with the claim that there was a clear trade-off: either a long learning period followed by a high niching acceleration (CMA-ES) or a short learning period followed by a low niching acceleration (standard-ES).

Here we focus on the location of the global optima and the desired local sub-optima, and also consider the MPR analysis of the different niching variants under investigation.

5.2 Test Functions: Artificial Landscapes

We consider a set of artificial landscapes with different characteristics. Some of the landscapes have a symmetric or equal distribution of optima (e.g., Ackley, $\mathcal{L}$, $\mathcal{M}$), and some do not (e.g., Shekel). Some of the functions are separable, that is, they can be optimized by solving n one-dimensional problems separately (see, e.g., Whitley et al., 1996), while some of them are nonseparable. Table 1 summarizes the unconstrained

Table 1: Test functions to be minimized and initialization domains. For some of the nonseparable functions, we apply translation and rotation: $\vec{y} = \mathcal{O}(\vec{x} - \vec{r})$, where $\mathcal{O}$ is an orthogonal rotation matrix, and $\vec{r}$ is a shifting vector (see also *a note on implementation* at the end of Section 5.2).

Name	Function	Init	Niches
Separable			
$\mathcal{M}$	$\mathcal{M}(\vec{x}) = -\frac{1}{n}\sum_{i=1}^{n} \sin^{\alpha}(5\pi x_i)$	$[0,1]^n$	100
$\mathcal{A}$ [Ackley]	$\mathcal{A}(\vec{x}) = -c_1 \cdot \exp\left(-c_2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)$ $- \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(c_3 x_i)\right) + c_1 + e$	$[-10,10]^n$	$2n+1$
$\mathcal{L}$	$\mathcal{L}(\vec{x}) = -\prod_{i=1}^{n} \sin^{k}(l_1\pi x_i + l_2) \cdot \exp\left(-l_3\left(\frac{x_i - l_4}{l_5}\right)^2\right)$	$[0,1]^n$	$n+1$
$\mathcal{R}$ [Rastrigin]	$\mathcal{R}(\vec{x}) = 10n + \sum_{i=1}^{n}\left(x_i^2 - 10\cos(2\pi x_i)\right)$	$[-1,5]^n$	$n+1$
$\mathcal{G}$ [Griewank]	$\mathcal{G}(\vec{x}) = 1 + \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)$	$[-10,10]^n$	5
$\mathcal{S}$ [Shekel]	$\mathcal{S}(\vec{x}) = -\sum_{i=1}^{10}\frac{1}{k_i(\vec{x}-a_i)(\vec{x}-a_i)^T + c_i}$	$[0,10]^n$	8
$\mathcal{V}$ [Vincent]	$\mathcal{V}(\vec{x}) = -\frac{1}{n}\sum_{i=1}^{n}\sin(10 \cdot \log(x_i))$	$[0.25,10]^n$	50
Nonseparable			
$\mathcal{F}$ [Fletcher-Powell]	$\mathcal{F}(\vec{x}) = \sum_{i=1}^{n}(A_i - B_i)^2$ $A_i = \sum_{j=1}^{n}\left(a_{ij} \cdot \sin(\alpha_j) + b_{ij} \cdot \cos(\alpha_j)\right)$ $B_i = \sum_{j=1}^{n}\left(a_{ij} \cdot \sin(x_j) + b_{ij} \cdot \cos(x_j)\right)$ $a_{ij},\ b_{ij} \in [-100,100]\,;\quad \vec{\alpha} \in [-\pi,\pi]^n$	$[-\pi,\pi]^n$	10
$\mathcal{R}_{SR}$ [S.R. Rastrigin]	$\mathcal{R}_{SR}(\vec{x}) = 10n + \sum_{i=1}^{n}\left(y_i^2 - 10\cos(2\pi y_i)\right)$	$[-5,5]^n$	$n+1$
$\mathcal{G}_{SR}$ [S.R. Griewank]	$\mathcal{G}_{SR}(\vec{x}) = 1 + \sum_{i=1}^{n}\frac{y_i^2}{4000} - \prod_{i=1}^{n}\cos\left(\frac{y_i}{\sqrt{i}}\right)$	$[0,600]^n$	5

multimodal test functions as well as their initialization intervals. Explicitly, we consider the following multimodal test functions:

- $\mathcal{M}$ is a basic hyper-grid multimodal function with uniformly distributed minima of equal function value of -1. It is meant to test the stability of a particularly large number of niches: in the interval $[0,1]^n$ it has 5^n minima.
- The well known Ackley function has one global minimum, regardless of its dimension n, which is surrounded isotropically by $2n$ local minima in the first hypersphere, followed by an exponentially increasing number of minima in successive hyperspheres. Ackley's function has been widely investigated in the context of evolutionary computation (see, e.g., Bäck, 1996). We used $c_1 = 20$, $c_2 = 0.2$, and $c_3 = 2\pi$.
- $\mathcal{L}$—also known as $F2$, as originally introduced by Goldberg and Richardson (1987)—is a sinusoid trapped in an exponential envelope. The parameter k determines the sharpness of the peaks in the function landscape; we set it to $k = 6$. $\mathcal{L}$ has one global minimum, regardless of n and k. It has been a popular test function for GA niching methods. We used $l_1 = 5.1$, $l_2 = 0.5$, $l_3 = 4 \cdot \ln(2)$, $l_4 = 0.0667$, and $l_5 = 0.64$.
- The Rastrigin function (Törn and Zilinskas, 1987) has one global minimum, surrounded by a large number of local minima arranged in a lattice configuration.

We also consider its shifted-rotated variant (Suganthan et al., 2005), with a linear transformation matrix of condition number 2 as the rotation operator (see a note on implementation below).

- The Griewank function (Törn and Zilinskas, 1987) has its global minimum ($f^* = 0$) at the origin, with several thousand local minima in the area of interest. There are four suboptimal minima $f \approx 0.0074$ with $\vec{x}^* \approx (\pm\pi, \pm\pi\sqrt{2}, 0, 0, 0, \ldots 0)$.
  We also consider its shifted-rotated variant (Suganthan et al., 2005), with a linear transformation matrix of condition number 3 as the rotation operator (see a note on implementation below).
- The Vincent function is a sine function with a decreasing frequency. It has 6^n global optima in the interval $[0.25, 10]^n$.
- The Shekel function, suggested by Törn and Zilinskas (1987), introduces a landscape with a dramatically uneven spread of optima. It has one global optimum, and seven ordered local optima. See a note on implementation below.
- The function after Fletcher and Powell (Bäck, 1996) is a nonseparable nonlinear parameter estimation problem, which has a nonuniform distribution of 2^n minima. It has non-isotropic attractor basins. See a note on implementation below.

5.2.1 A Note on Implementation

Most of the data for the functions, and in particular the translation and rotation operators, was retrieved from Suganthan et al. (2005).[2] The Shekel data was retrieved from Törn and Zilinskas (1987), and the Fletcher-Powell data was retrieved from Bäck (1996).

5.3 Modus Operandi

We consider *four* niching approaches, subject to *two* selection strategies, comma and plus: fixed-radius (CMA), self-adaptive niche radius (S-CMA), Mahalanobis (M-CMA), and Mahalanobis with self-adaptive niche radius (M-S-CMA).[3] The eight niching routines are tested on the specified functions for various dimensions. Each test case includes 100 runs per routine. All runs are performed with a search engine of a $(1 \dot{+} 10)$-strategy per niche and initial points are sampled uniformly within the initialization intervals. Initial step sizes σ_{init} are set to $\frac{1}{4}$ of the intervals, and initial niche radii for the self-adaptive routines are set to $\sqrt{n} \cdot \sigma_{\text{init}}$. Otherwise, niche radii are set based on some a priori knowledge of the landscape. The parameter q is set based on a priori knowledge when available, or arbitrarily otherwise. In order to keep the behavior as simple as possible, the parameter p was set here to $p = 0$ (no so-called restart mechanism). It should be noted that there are negligible CPU time differences among the routines.

5.3.1 Function Evaluations

The idea of function allocations is to allocate a fixed number of evaluations per peak $(n \cdot 10^4)$, and thus each run is stopped after $q \cdot n \cdot 10^4$ function evaluations.

A curve-fitting routine is applied to each run in order to retrieve the characteristic parameters of its logistic curve (the routine uses the least-squares-error method).

---

[2] Data are available for download at http://www.ntu.edu.sg/home/epnsugan/index_files/.

[3] Code is available from the authors upon request.

Table 2: Global minimum reached in 100 runs (CMA denotes a $(1, \lambda)$-strategy, CMA+ denotes a $(1 + \lambda)$-strategy). The best result, per strategy, is shown in boldface type.

Test case	CMA	M-CMA	S-CMA	M-S-CMA	CMA+	M-CMA+	S-CMA+	M-S-CMA+
$\mathcal{A} : n = 3$	100%	100%	100%	100%	100%	100%	100%	100%
$\mathcal{A} : n = 10$	94%	100%	100%	100%	97%	100%	100%	100%
$\mathcal{L} : n = 3$	64%	**66%**	43%	54%	**94%**	89%	65%	70%
$\mathcal{L} : n = 10$	**16%**	8%	2%	13%	**9%**	5%	1%	0%
$\mathcal{R} : n = 3$	54%	**59%**	13%	40%	**67%**	62%	14%	30%
$\mathcal{R} : n = 10$	0%	0%	0%	0%	0%	0%	0%	0%
$\mathcal{G} : n = 3$	12%	19%	10%	**25%**	19%	19%	16%	**52%**
$\mathcal{G} : n = 10$	20%	**31%**	27%	27%	0%	0%	0%	0%
$\mathcal{S} : n = 5$	91%	97%	82%	**100%**	83%	62%	98%	**91%**
$\mathcal{S} : n = 10$	21%	48%	46%	90%	97%	92%	**100%**	75%
$\mathcal{F} : n = 4$	100%	100%	100%	100%	100%	100%	100%	100%
$\mathcal{F} : n = 10$	25%	40%	36%	**46%**	17%	22%	34%	**37%**
$\mathcal{R}_{SR} : n = 3$	46%	**54%**	14%	24%	50%	**66%**	10%	26%
$\mathcal{R}_{SR} : n = 10$	**8%**	2%	0%	0%	0%	0%	0%	0%
$\mathcal{G}_{SR} : n = 3$	**10%**	0%	0%	0%	**9%**	0%	0%	0%
$\mathcal{G}_{SR} : n = 10$	0%	0%	0%	0%	0%	0%	0%	0%

5.4 Performance Analysis

A recent study evaluated the performance of niching techniques, based on derandomized ES variants, on a similar suite of theoretical test functions (Shir and Bäck, 2009). Notice, however, that it applied a fixed radius approach, and set $p = 1$.

We discuss here the performance analysis at three levels: global minimum, MPR saturation, and niching acceleration.

5.4.1 Global Minimum

Table 2 contains the percentage of runs in which the global minimum was located. $\mathcal{M}$ and $\mathcal{V}$ are discarded from the table, as their global minimum was always found, by all algorithms, for every dimension n under investigation. For the comma strategy (the four left columns in Table 2), we observe that the Mahalanobis metric usually improves the global optimization, both for the fixed, as well as for the self-adaptive niche radius approaches. On the other hand, this does not seem to be the general trend for the plus strategy; on average, the usage of the Mahalanobis distance metric does not improve the global optimization. We may conclude that there is no clear winner, and that the routines using the Mahalanobis distance metric do not achieve a dramatic improvement in global optimization. This is an expected result, as the usage of this metric primarily assists in the formation of the niches.

5.4.2 MPR Saturation

This scalar value represents, to some degree, the quality of the obtained minima, as it captures the ratio between attained fitness and desired fitness. Thus, it is another indication for the final result of the niching process. Tables 3 and 4 present the mean and the standard deviation of the saturation MPR values for the different test cases.

We observe a trend of better performance for the routines using the Mahalanobis distance metric for both strategies. On average, the MPR values are higher, reflecting a better niching process.

Table 3: MPR saturation values for the $(1, \lambda)$-strategy: mean values and standard deviations over 100 runs. Emphasized in boldface type are winner algorithms with respect to the specified landscape, also in reference to the results of Table 4. Landscapes with several winners do not have boldface type.

Test case	CMA	M-CMA	S-CMA	M-S-CMA
$\mathcal{M}: n=3$	1 ± 0	1 ± 0	1 ± 0	1 ± 0
$\mathcal{M}: n=10$	0.994 ± 0.002	0.967 ± 0.003	1 ± 0	1 ± 0
$\mathcal{M}: n=40$	0.956 ± 0.006	0.953 ± 0.008	0.994 ± 0.001	$\mathbf{0.995 \pm 0.002}$
$\mathcal{A}: n=3$	0.938 ± 0.044	N.A.	0.860 ± 0.143	N.A.
$\mathcal{A}: n=10$	0.909 ± 0.033	N.A.	N.A.	N.A.
$\mathcal{L}: n=3$	0.864 ± 0.092	0.870 ± 0.106	0.713 ± 0.083	0.834 ± 0.099
$\mathcal{L}: n=10$	0.240 ± 0.086	0.389 ± 0.114	0.478 ± 0.080	$\mathbf{0.564 \pm 0.105}$
$\mathcal{R}: n=3$	0.301 ± 0.081	0.228 ± 0.063	0.159 ± 0.041	0.305 ± 0.103
$\mathcal{R}: n=10$	$\mathbf{0.103 \pm 0.045}$	0.062 ± 0.011	0.082 ± 0.019	0.094 ± 0.022
$\mathcal{G}: n=3$	0.249 ± 0.126	0.234 ± 0.045	0.283 ± 0.092	0.255 ± 0.064
$\mathcal{G}: n=10$	$\mathbf{0.252 \pm 0.169}$	0.195 ± 0.040	0.186 ± 0.092	0.190 ± 0.041
$\mathcal{S}: n=5$	0.840 ± 0.320	0.911 ± 0.307	0.819 ± 0.300	$\mathbf{0.979 \pm 0.067}$
$\mathcal{S}: n=10$	0.820 ± 0.722	0.931 ± 0.073	0.596 ± 0.136	$\mathbf{0.959 \pm 0.062}$
$\mathcal{V}: n=3$	0.972 ± 0.011	0.920 ± 0.005	0.613 ± 0.028	0.552 ± 0.078
$\mathcal{V}: n=10$	0.998 ± 0.007	0.998 ± 0.001	0.999 ± 0.001	$\mathbf{0.9998 \pm 0.0002}$
$\mathcal{F}: n=4$	0.0004 ± 0.0006	0.0049 ± 0.0052	0.0005 ± 0.0004	0.0173 ± 0.0915
$\mathcal{F}: n=10$	0.0001 ± 0.0002	0.0002 ± 0.0003	0.0003 ± 0.0003	0.0004 ± 0.0012
$\mathcal{R}_{SR}: n=3$	0.331 ± 0.103	0.231 ± 0.041	0.138 ± 0.051	0.268 ± 0.074
$\mathcal{R}_{SR}: n=10$	$\mathbf{0.130 \pm 0.039}$	0.087 ± 0.042	0.069 ± 0.019	0.093 ± 0.018
$\mathcal{G}_{SR}: n=3$	0.0009 ± 0.0009	0.0010 ± 0.0009	0.0007 ± 0.0008	0.0010 ± 0.0011
$\mathcal{G}_{SR}: n=10$	0.0001 ± 0	0.0001 ± 0	0.0001 ± 0	0.0001 ± 0

Note that the niching routines, except for the fixed niche radius case, fail on the Ackley function, that is, they locate only the global minimum (all niches are located in the global basin of attraction). This effect has been observed in the past (Shir and Bäck, 2006), and was explained by the strong basin of attraction of the global minimum, in comparison to the suboptimal minima. Moreover, most of the MPR values for the Fletcher-Powell and shifted-rotated Griewank test cases are much lower than unity, due to the extreme scaling of the landscape: it has false traps with very high function values. Thus, upon being trapped in these local minima, the MPR value is expected to be very low.

5.4.3 Niching Acceleration

The MPR analysis allows us to compare the niching acceleration of the different routines [the parameter c in Equation (31)]. Tables 5 and 6 present the mean values and the standard deviation of the niching acceleration values for the different test cases. The curve-fitting routine did not attain data with acceptable high quality for the Fletcher-Powell test case, and it suffered from extremely large standard deviations. We thus choose to discard it from these tables.

There are some general trends in the attained data. The comma strategy has typically higher niching acceleration values, in agreement with past observations (Shir and Bäck, 2009). Within each strategy, there is a trend of higher niching acceleration for the Mahalanobis-distance based routines. This result is pretty much intuitive, a more accurate spatial classification, as usually obtained by the Mahalanobis distance metric,

Table 4: MPR saturation values for the $(1+\lambda)$-strategy: mean values and standard deviations over 100 runs. Emphasized in boldface type are winner algorithms with respect to the specified landscape, also in reference to the results of Table 3. Landscapes with several winners do not have boldface type.

Test case	CMA+	M-CMA+	S-CMA+	M-S-CMA+
$\mathcal{M}: n=3$	1 ± 0	1 ± 0	1 ± 0	1 ± 0
$\mathcal{M}: n=10$	0.991 ± 0.003	0.986 ± 0.003	1 ± 0	1 ± 0
$\mathcal{M}: n=40$	0.975 ± 0.008	0.980 ± 0.007	1 ± 0	1 ± 0
$\mathcal{A}: n=3$	0.989 ± 0.026	$\mathbf{0.999 \pm 0.009}$	0.930 ± 0.030	0.937 ± 0.159
$\mathcal{A}: n=10$	0.946 ± 0.017	$\mathbf{0.987 \pm 0.019}$	N.A.	N.A.
$\mathcal{L}: n=3$	0.959 ± 0.033	$\mathbf{0.962 \pm 0.036}$	0.819 ± 0.079	0.919 ± 0.065
$\mathcal{L}: n=10$	0.454 ± 0.116	0.373 ± 0.115	0.423 ± 0.108	0.432 ± 0.090
$\mathcal{R}: n=3$	0.528 ± 0.118	$\mathbf{0.552 \pm 0.107}$	0.163 ± 0.072	0.250 ± 0.089
$\mathcal{R}: n=10$	0.102 ± 0.040	0.077 ± 0.027	0.049 ± 0.009	0.053 ± 0.011
$\mathcal{G}: n=3$	0.326 ± 0.094	0.334 ± 0.101	0.305 ± 0.114	$\mathbf{0.494 \pm 0.234}$
$\mathcal{G}: n=10$	0.037 ± 0.008	0.053 ± 0.015	0.062 ± 0.019	0.060 ± 0.015
$\mathcal{S}: n=5$	0.681 ± 0.114	0.897 ± 0.109	0.920 ± 0.073	0.882 ± 0.086
$\mathcal{S}: n=10$	0.658 ± 0.054	0.957 ± 0.104	0.916 ± 0.311	0.939 ± 0.085
$\mathcal{V}: n=3$	0.962 ± 0.012	$\mathbf{0.999 \pm 0.001}$	0.815 ± 0.072	0.689 ± 0.114
$\mathcal{V}: n=10$	0.953 ± 0.016	0.990 ± 0.004	0.996 ± 0.002	0.999 ± 0.001
$\mathcal{F}: n=4$	0.0007 ± 0.0003	0.862 ± 0.385	0.0044 ± 0.0024	$\mathbf{0.991 \pm 0.038}$
$\mathcal{F}: n=10$	0.0001 ± 0.0001	0.0001 ± 0.0001	$\mathbf{0.0005 \pm 0.0005}$	0.0001 ± 0.0001
$\mathcal{R}_{SR}: n=3$	0.486 ± 0.137	$\mathbf{0.563 \pm 0.140}$	0.135 ± 0.051	0.249 ± 0.129
$\mathcal{R}_{SR}: n=10$	0.081 ± 0.030	0.080 ± 0.018	0.044 ± 0.006	0.041 ± 0.006
$\mathcal{G}_{SR}: n=3$	0.0009 ± 0.0011	0.0007 ± 0.0006	0.008 ± 0.0008	$\mathbf{0.0012 \pm 0.0017}$
$\mathcal{G}_{SR}: n=10$	0.0002 ± 0	0.0002 ± 0	0.0002 ± 0	0.0002 ± 0

Table 5: Niching acceleration values for the $(1,\lambda)$-strategy: mean values and standard deviations over 100 runs.

Test case	CMA	M-CMA	S-CMA	M-S-CMA
$\mathcal{M}: n=3$	-0.068 ± 0.010	-0.069 ± 0.002	-0.049 ± 0.007	-0.060 ± 0.008
$\mathcal{M}: n=10$	-0.038 ± 0.001	-0.043 ± 0.002	-0.029 ± 0.002	-0.032 ± 0.001
$\mathcal{M}: n=40$	-0.014 ± 0.001	-0.014 ± 0.001	-0.010 ± 0.001	-0.010 ± 0.001
$\mathcal{A}: n=3$	-0.133 ± 0.015	N.A.	-0.035 ± 0.013	N.A.
$\mathcal{A}: n=10$	-0.063 ± 0.002	N.A.	N.A.	N.A.
$\mathcal{L}: n=3$	-0.179 ± 0.038	-0.184 ± 0.048	-0.128 ± 0.044	-0.167 ± 0.036
$\mathcal{L}: n=10$	-0.174 ± 0.024	-0.176 ± 0.025	-0.144 ± 0.016	-0.153 ± 0.019
$\mathcal{R}: n=3$	-0.043 ± 0.007	-0.131 ± 0.109	-0.045 ± 0.027	-0.125 ± 0.058
$\mathcal{R}: n=10$	-0.043 ± 0.013	-0.052 ± 0.012	-0.064 ± 0.016	-0.081 ± 0.015
$\mathcal{G}: n=3$	-0.079 ± 0.079	-0.112 ± 0.033	-0.097 ± 0.080	-0.152 ± 0.095
$\mathcal{G}: n=10$	-0.001 ± 0.002	-0.006 ± 0.002	-1.051 ± 6.983	-1.120 ± 5.418
$\mathcal{S}: n=5$	-0.004 ± 0.005	-0.019 ± 0.009	-0.080 ± 0.056	0.072 ± 0.020
$\mathcal{S}: n=10$	-0.004 ± 0.010	-0.003 ± 0.005	-0.012 ± 0.024	-0.005 ± 0.004
$\mathcal{V}: n=3$	-0.004 ± 0.004	-0.104 ± 0.010	-0.010 ± 0.027	-1.023 ± 2.018
$\mathcal{V}: n=10$	-0.004 ± 0.009	-0.037 ± 0.024	-0.055 ± 0.002	-0.061 ± 0.003
$\mathcal{R}_{SR}: n=3$	-0.079 ± 0.068	-0.153 ± 0.098	-0.031 ± 0.019	-0.113 ± 0.042
$\mathcal{R}_{SR}: n=10$	-0.077 ± 0.029	-0.087 ± 0.032	-0.051 ± 0.011	-0.069 ± 0.010
$\mathcal{G}_{SR}: n=3$	-0.147 ± 0.088	-0.150 ± 0.076	-0.274 ± 0.284	-0.129 ± 0.076
$\mathcal{G}_{SR}: n=10$	-0.101 ± 0.046	-0.107 ± 0.045	-0.204 ± 0.297	-0.196 ± 0.276

Table 6: Niching acceleration values for the $(1 + \lambda)$-strategy: mean values and standard deviations over 100 runs.

Test case	CMA+	M-CMA+	S-CMA+	M-S-CMA+
$\mathcal{M}: n=3$	-0.055 ± 0.007	-0.056 ± 0.007	-0.046 ± 0.004	-0.049 ± 0.005
$\mathcal{M}: n=10$	-0.015 ± 0.001	-0.016 ± 0.001	-0.015 ± 0.001	-0.015 ± 0.001
$\mathcal{M}: n=40$	-0.006 ± 0.001	-0.006 ± 0.001	-0.004 ± -0.001	-0.004 ± 0.001
$\mathcal{A}: n=3$	-0.044 ± 0.004	-0.048 ± 0.004	-0.016 ± 0.015	-0.043 ± 0.016
$\mathcal{A}: n=10$	-0.017 ± 0.001	-0.016 ± 0.001	N.A.	N.A.
$\mathcal{L}: n=3$	-0.066 ± 0.015	-0.066 ± 0.020	-0.053 ± 0.012	0.058 ± 0.012
$\mathcal{L}: n=10$	-0.029 ± 0.011	-0.034 ± 0.007	-0.040 ± 0.002	-0.040 ± 0.002
$\mathcal{R}: n=3$	-0.054 ± 0.005	-0.053 ± 0.005	-0.041 ± 0.007	-0.043 ± 0.014
$\mathcal{R}: n=10$	-0.015 ± 0.002	-0.007 ± 0.001	-0.019 ± 0.001	-0.020 ± 0.001
$\mathcal{G}: n=3$	-0.065 ± 0.009	-0.064 ± 0.013	-0.061 ± 0.014	0.050 ± 0.017
$\mathcal{G}: n=10$	-0.808 ± 5.670	-1.080 ± 10.380	-0.748 ± 6.995	-2.023 ± 18.077
$\mathcal{S}: n=5$	-0.006 ± 0.008	-0.006 ± 0.004	-0.030 ± 0.012	0.021 ± 0.004
$\mathcal{S}: n=10$	-0.002 ± 0.001	-0.002 ± 0.001	-0.009 ± 0.010	-0.005 ± 0.003
$\mathcal{V}: n=3$	-0.063 ± 0.008	-0.065 ± 0.010	-0.015 ± 0.005	-0.040 ± 0.010
$\mathcal{V}: n=10$	-0.027 ± 0.002	-0.020 ± 0.003	-0.025 ± 0.001	-0.025 ± 0.001
$\mathcal{R}_{SR}: n=3$	-0.055 ± 0.006	-0.056 ± 0.009	-0.037 ± 0.010	-0.045 ± 0.012
$\mathcal{R}_{SR}: n=10$	-0.021 ± 0.002	-0.021 ± 0.002	-0.018 ± 0.001	-0.018 ± 0.001
$\mathcal{G}_{SR}: n=3$	-0.176 ± 0.150	-0.156 ± 0.050	-0.152 ± 0.069	-0.181 ± 0.206
$\mathcal{G}_{SR}: n=10$	-0.031 ± 0.011	-0.031 ± 0.016	-0.032 ± 0.013	-0.034 ± 0.011

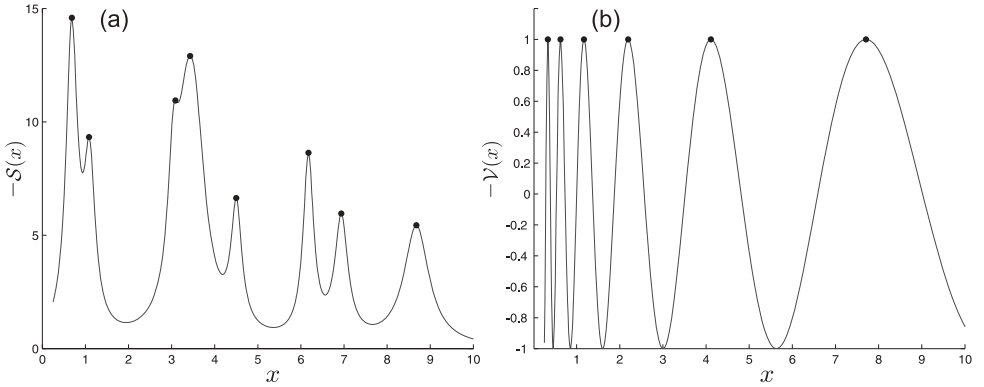

Figure 5: Final population of the CMA-(1, 10) with self-adaptive niche radius on the 1D Shekel [panel (a)] and the 1D Vincent [panel (b)] test cases. Note the negative scaling of the function values.

allows the niching mechanism in most cases to form appropriate niches and to converge faster.

5.5 General Behavior

The proposed self-adaptive niche radius routine performed well on the landscapes with a deceptive distribution of optima, that is, $\mathcal{V}$ and $\mathcal{S}$, and managed to successfully tackle the niche radius problem. Typical runs on $\mathcal{V}$ and $\mathcal{S}$ for $n = 1$ are depicted in Figure 5. Moreover, Figures 6, 7, and 8 illustrate the adaptation of the classification-ellipses by

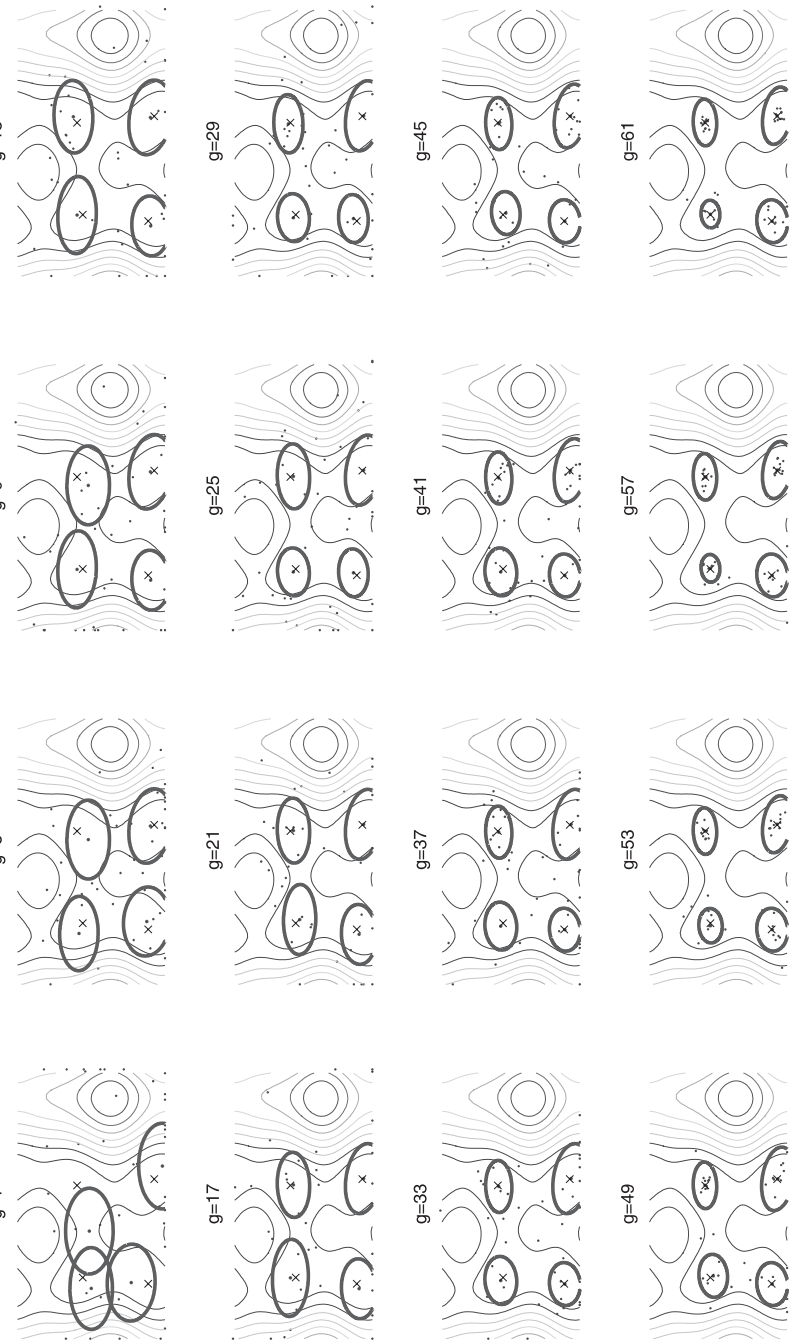

Figure 6: A snapshot gallery: the adaptation of the classification-ellipses, subject to the Mahalanobis metric with the updating covariance matrix, in the CMA+ strategy for a 2D Fletcher-Powell problem. Images are taken in the box $[-\pi, \pi]^2$. Contours of the landscape are given as the background, where the X's indicate the real optima, the dots are the evolving individuals, and the ellipses are plotted centered about the peak individual. A snapshot is taken every four generations (i.e., every 160 function evaluations), as indicated by the counter.

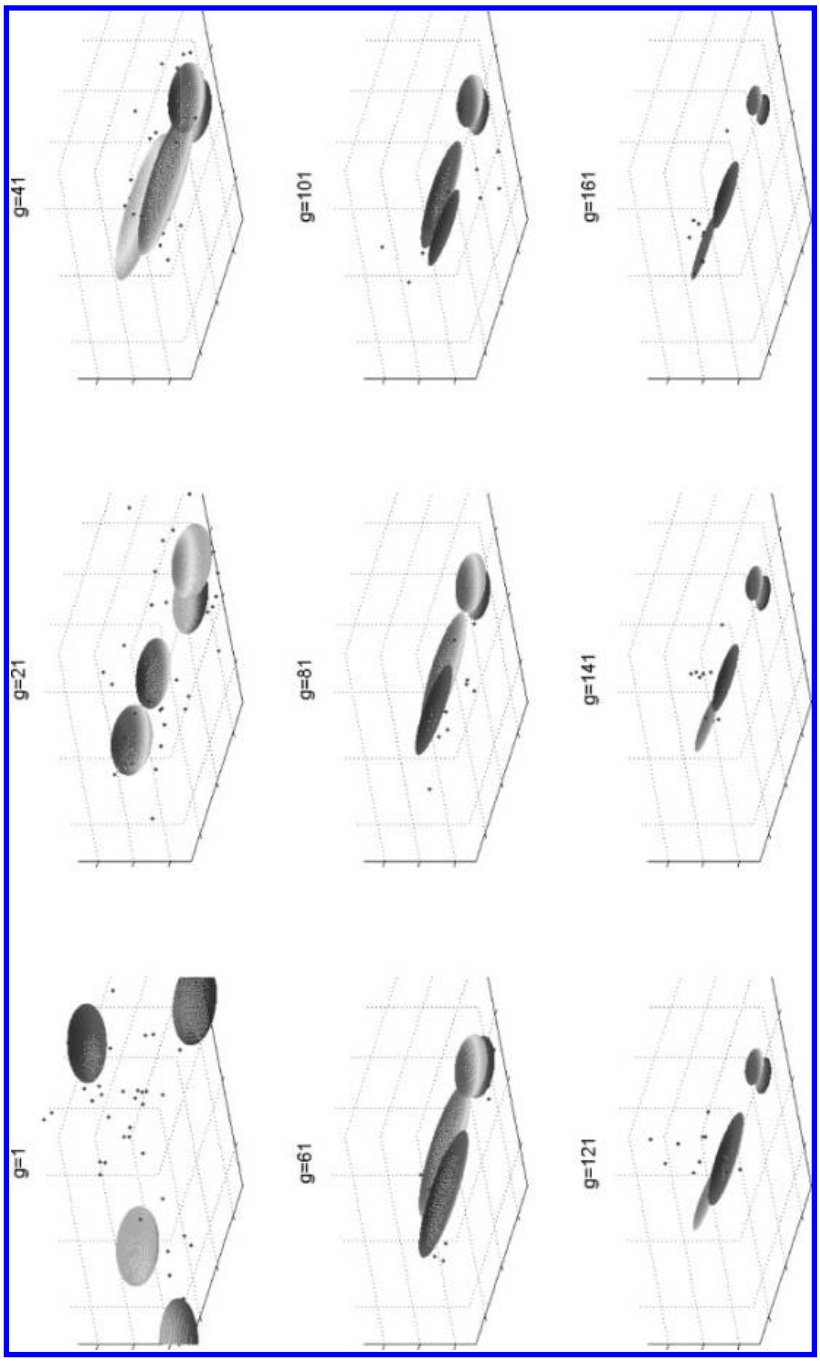

Figure 7: A 3D snapshot gallery: the adaptation of the classification-ellipses, subject to the Mahalanobis metric with the updating covariance matrix, in the CMA+ strategy for a 3D Fletcher-Powell problem. Images are taken in the box $[-\pi, \pi]^3$. The ellipses are centered about the evolving peak individuals. A snapshot is taken every 20 generations (i.e., every 800 function evaluations), as indicated by the counter.

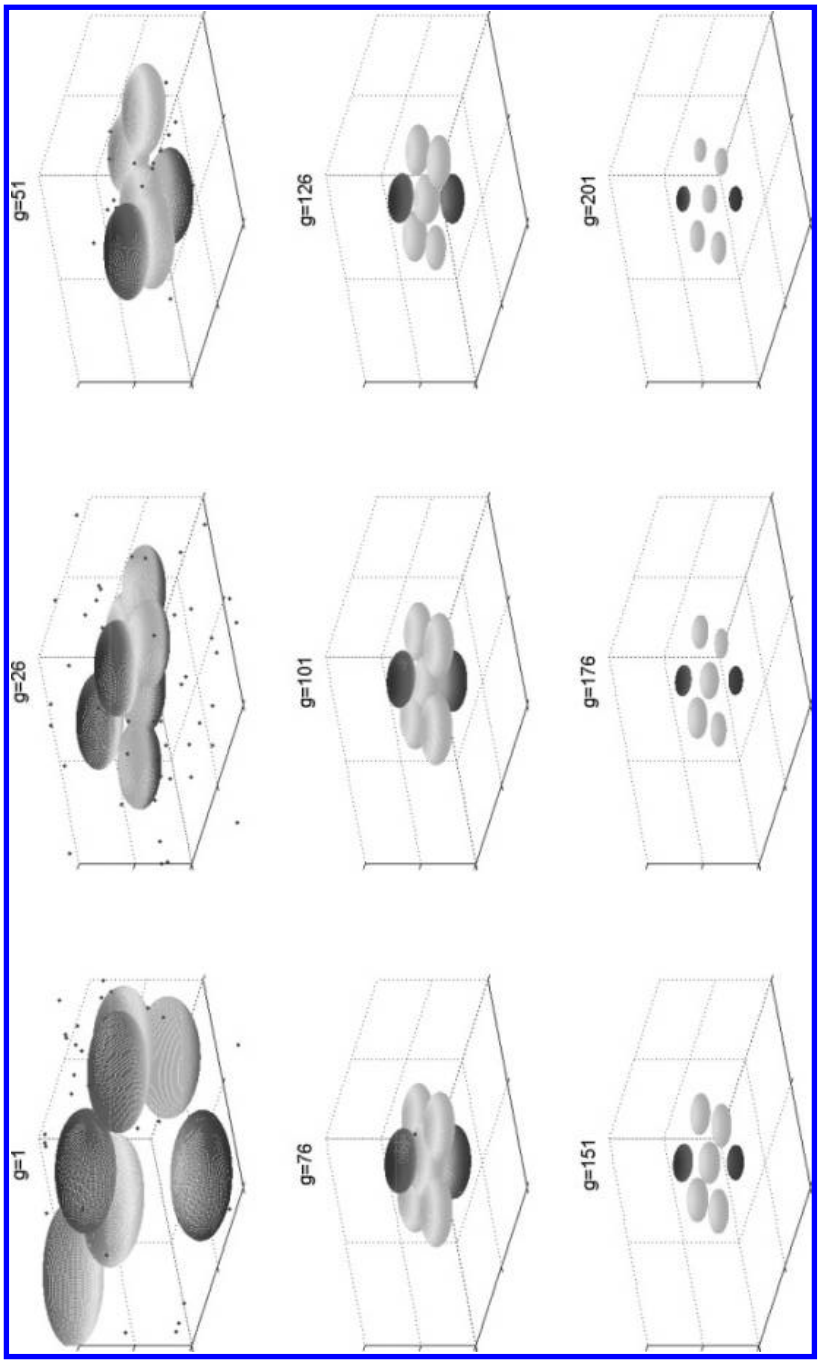

Figure 8: A 3D snapshot gallery: the adaptation of the classification-ellipses, subject to the Mahalanobis metric with the updating covariance matrix, in the CMA+ strategy for a 3D Ackley problem. Images are taken in the box $[-2, 2]^3$. The ellipses are centered about the evolving peak individuals, and are observed to adapt simultaneously. A snapshot is taken every 25 generations (i.e., every 1,750 function evaluations), as indicated by the counter.

the M-CMA+ routine for the 2D Fletcher-Powell, 3D Fletcher-Powell, and 3D Ackley problems, respectively. It can be observed in the Fletcher-Powell case that each niche has its own characteristic matrix and convergence profile, whereas the convergence in the Ackley problem seems to be simultaneous, as expected from the landscape symmetry.

6 Discussion and Outlook

We have introduced new concepts of adaptive niche radii and niche shapes into the framework of niching with the covariance matrix adaptation evolution strategy. The main goal was to tackle the so-called niche radius problem, and to offer an efficient niching mechanism with less presumptions on the search landscape. It was successfully achieved at two levels: The construction of the self-adaptive niche radius, and the application of the Mahalanobis distance metric for the adaptation of the niche shapes. We have described both approaches in detail.

The careful reader should note that niching with CMA subject to the Mahalanobis distance metric is applicable only when the niching distance is calculated in the decision space. Sometimes this is not the case, and other spaces are used for that (e.g., the second-derivative space in functional optimization; for details see Shir and Bäck, 2009).

We would like to discuss here the important issue of parameters in light of our proposed approaches. The discussion is done at two levels. The first is the relaxation of existing parameters in the fixed-radius CMA niching algorithm, and more specifically the parameter q. The parameter q is reduced in this study, for the first time, from being a critical niching parameter in the fixed radius approach into being the estimated/desired target number of niches/peaks in the self-adaptive approaches without any influence on the algorithmic behavior. However, a possibly wrong estimation of q would simply result in the proposed approaches wasting CPU cycles when too large, or missing good optima when too small. The second level is the introduction of new parameters, that is, α and γ [Equation (24)], for the function of the learning coefficients [Equation (23)]. Although this is an undesirable situation, one should keep in mind that by setting only two parameters, we are allowing the application of a niching method to landscapes with a large number of optima with possibly different basin sizes, that would require different niche radii, respectively. We would like to stress that if these parameters had not been introduced, the application to such landscapes would not have been feasible with the fixed radius approach, or would have required setting as many parameters as the number of peaks when staying in the same framework of radius based niching. Thus, by setting only these two parameters, we achieve a lot. Moreover, the proposed settings apply to a wide range of practically relevant landscapes, and do not have to be chosen for each new problem by means of additional experiments.

Given the CMA-ES-$(1 \dagger \lambda)$ search engines, four variants of niching were considered per routine, and tested on a suite of artificial landscapes. The new approaches were shown to perform in a satisfying manner, on landscapes with evenly and unevenly spread optima. The niche radius problem was tackled successfully by the self-adaptive approach, as demonstrated on landscapes with unevenly spread optima, both separable and nonseparable. The application of the Mahalanobis distance metric achieved its goal of improving the niching process, in terms of obtaining on average higher quality suboptima, subject to higher niching acceleration in multimodal landscapes with non-isotropic attractor shapes. It does not seem to improve nor to hamper, on average, the identification of the location of global minimum in landscapes with isotropic attractor basins, as expected.

Regarding the implementation of the Mahalanobis distance metric, we have offered here a numerical simplification of the required calculation, which was observed to pay off in terms of computation time. By applying this numerical implementation, the Mahalanobis approach shares the same computational complexity as the previously discussed approaches.

We thus present here the self-adaptive niche radius CMA niching with Mahalanobis metric as a state of the art niching technique within evolution strategies, and propose it as a solution to the so-called niche-radius problem.

As future research, we would like to consider the following directions:

- Carrying out a comparison study of these new approaches to classical GA niching techniques, sharing- or non-sharing based (e.g., crowding, or clearing, as introduced by Petrowski, 1996).
- Integrating non-sharing niching methods into the CMA framework.
- Employing these methods in the presented CMA niching algorithm as the secondary selection mechanisms for the niche radius self-adaptation subroutine.
- Transferring this generation of self-adaptive niching algorithms into real-world applications, and possibly to experimental optimization.
- Investigating the performance of the self-adaptive niche-radius niching approach on dynamic landscapes, with time-varying optima in location and shape.
- Studying the influence of the population size, especially in higher dimensional search landscapes, on the performance of the proposed techniques. Also, (μ, λ) search engines may be taken into account within these self-adaptive niching frameworks (see Shir et al., 2009).

Acknowledgments

This work is part of the research programme of the "Stichting voor Fundamenteel Onderzoek de Materie (FOM)," which is financially supported by the "Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO)."

References

Aichholzer, O., Aurenhammer, F., Brandtstätter, B., Ebner, T., Krammer, H., and Magele, C. (2000). Niching evolution strategy with cluster algorithms. In *Proceedings of the 9th Biennial IEEE Conference on Electromagnetic Field Computations*, p. 137.

Ando, S., Sakuma, J., and Kobayashi, S. (2005). Adaptive isolation model using data clustering for multimodal function optimization. In *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, GECCO 2005*, pp. 1417–1424.

Avigad, G., Moshaiov, A., and Brauner, N. (2004). Concept-based interactive brainstorming in engineering design. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 8(5):454–459.

Bäck, T. (1994). Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In *Proceedings of the First IEEE Conference on Evolutionary Computation (ICEC'94)*, pp. 57–62.

Bäck, T. (1996). *Evolutionary algorithms in theory and practice*. New York: Oxford University Press.

Beyer, H.-G., and Arnold, D. V. (2003). Qualms regarding the optimality of cumulative path length control in CSA/CMA-evolution strategies. *Evolutionary Computation*, 11(1):19–28.

Beyer, H.-G., and Schwefel, H.-P. (2002). Evolution strategies: A comprehensive introduction. *Natural Computing: An International Journal*, 1(1):3–52.

Branke, J. (2001). *Evolutionary optimization in dynamic environments*. Norwell, MA: Kluwer Academic Publishers.

Cioppa, A. D., Stefano, C. D., and Marcelli, A. (2004). On the role of population size and niche radius in fitness sharing. *IEEE Transactions on Evolutionary Computation*, 8(6):580–592.

Deb, K., and Goldberg, D. E. (1989). An investigation of niche and species formation in genetic function optimization. In *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 42–50.

Deb, K., and Spears, W. M. (1997). Speciation methods. In T. Bäck, D. Fogel, and Z. Michalewicz (Eds.), *The Handbook of Evolutionary Computation*. Oxford, UK: IOP Publishing and Oxford University Press.

Deb, K., and Tiwari, S. (2005). Omni-optimizer: A procedure for single and multi-objective optimization. In *Evolutionary Multi-Criterion Optimization, Third International Conference, EMO 2005*, vol. 3410 of *Lecture Notes in Computer Science*, pp. 47–61. Berlin: Springer.

Gan, J., and Warwick, K. (2001). Dynamic niche clustering: A fuzzy variable radius niching technique for multimodal optimisation in GAs. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pp. 215–222.

Giorgi, P., Jeannerod, C.-P., and Villard, G. (2003). On the complexity of polynomial matrix computations. In *ISSAC '03: Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, pp. 135–142.

Goldberg, D. E., and Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms and Their Application*, pp. 41–49.

Hanagandi, V., and Nikolaou, M. (1998). A hybrid approach to global optimization using a clustering algorithm in a genetic search framework. *Computers and Chemical Engineering*, 22(12):1913–1925.

Hansen, N., Gawelczyk, A., and Ostermeier, A. (1995a). Sizing the population with respect to the local progress in $(1, \lambda)$-evolution strategies—A theoretical analysis. In *Proceedings of the 1995 IEEE International Conference on Evolutionary Computation*, pp. 312–317.

Hansen, N., and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195.

Hansen, N., Ostermeier, A., and Gawelczyk, A. (1995b). On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA6)*, pp. 57–64.

Igel, C., Hansen, N., and Roth, S. (2007). Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1):1–28.

Igel, C., Suttorp, T., and Hansen, N. (2006). A computational efficient covariance matrix update and a (1+1)-CMA for evolution strategies. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2006*, pp. 453–460.

Jelasity, M. (1998). UEGO, an abstract niching technique for global optimization. In *Parallel Problem Solving from Nature - PPSN V*, vol. 1498 of *Lecture Notes in Computer Science*, pp. 378–387.

Mahfoud, S. (1995). *Niching methods for genetic algorithms*. PhD thesis, University of Illinois at Champaign-Urbana.

Miller, B., and Shaw, M. (1996). Genetic algorithms with dynamic niche sharing for multimodal function optimization. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC'96)*, pp. 786–791.

Ostermeier, A., Gawelczyk, A., and Hansen, N. (1993). A derandomized approach to self adaptation of evolution strategies. Technical Report TR-93-003, TU Berlin.

Ostermeier, A., Gawelczyk, A., and Hansen, N. (1994a). A derandomized approach to self adaptation of evolution strategies. *Evolutionary Computation*, 2(4):369–380.

Ostermeier, A., Gawelczyk, A., and Hansen, N. (1994b). Step-size adaptation based on non-local use of selection information. In *Parallel Problem Solving from Nature - PPSN III*, vol. 866 of *Lecture Notes in Computer Science*, pp. 189–198. Berlin: Springer.

Petrowski, A. (1996). A clearing procedure as a niching method for genetic algorithms. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC'96)*, pp. 798–803.

Preuss, M., Schönemann, L., and Emmerich, M. (2005). Counteracting genetic drift and disruptive recombination in $(\mu + /, \lambda)$-EA on multimodal fitness landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2005*, pp. 865–872.

Rechenberg, I. (1973). *Evolutionsstrategies: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart, Germany: Frommann-Holzboog Verlag.

Schönemann, L., Emmerich, M., and Preuss, M. (2004). On the extinction of sub-populations on multimodal landscapes. In *Proceedings of the International Conference on Bioinspired Optimization Methods and their Applications, BIOMA 2004*, pp. 31–40.

Shir, O. M., and Bäck, T. (2005). Dynamic niching in evolution strategies with covariance matrix adaptation. In *Proceedings of the 2005 Congress on Evolutionary Computation CEC-2005*, pp. 2584–2591.

Shir, O. M., and Bäck, T. (2006). Niche radius adaptation in the CMA-ES niching algorithm. In *Parallel Problem Solving from Nature, PPSN IX*, vol. 4193 of *Lecture Notes in Computer Science*, pp. 142–151. Berlin: Springer.

Shir, O. M., and Bäck, T. (2009). Niching with derandomized evolution strategies in artificial and real-world landscapes. *Natural Computing: An International Journal*, 8(1):171–196.

Shir, O. M., Preuss, M., Naujoks, B., and Emmerich, M. (2009). Enhancing decision space diversity in evolutionary multiobjective algorithms. In *Proceedings of Evolutionary Multi-Criterion Optimization: Fifth International Conference (EMO 2009)*, vol. 5467 of *Lecture Notes in Computer Science*, pp. 95–109. Berlin: Springer.

Shir, O. M., Siedschlag, C., Bäck, T., and Vrakking, M. J. (2006). Niching in evolution strategies and its application to laser pulse shaping. In *Proceedings of the International Conference Evolution Artificielle*, vol. 3871 of *Lecture Notes in Computer Science*, pp. 85–96. Berlin: Springer.

Stoean, C., Preuss, M., Gorunescu, R., and Dumitrescu, D. (2005). Elitist generational genetic chromodynamics—A new radii-based evolutionary algorithm for multimodal optimization. In *Proceedings of the 2005 Congress on Evolutionary Computation (CEC'05)*, pp. 1839–1846.

Stoean, C., Preuss, M., Stoean, R., and Dumitrescu, D. (2007). Disburdening the species conservation evolutionary algorithm of arguing with radii. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2007*, pp. 1420–1427.

Streichert, F., Stein, G., Ulmer, H., and Zell, A. (2003). A clustering based niching EA for multimodal search spaces. In *Proceedings of the International Conference Evolution Artificielle*, vol. 2936 of *Lecture Notes in Computer Science*, pp. 293–304. Berlin: Springer.

Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y. P., Auger, A., and Tiwari, S. (2005). Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical report, Nanyang Technological University, Singapore.

Törn, A., and Zilinskas, A. (1987). *Global optimization*, vol. 350 of *Lecture Notes in Computer Science*. Berlin: Springer.

Ursem, R. K. (1999). Multinational evolutionary algorithms. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC 1999)*, pp. 1633–1640.

van der Goes, V., Shir, O. M., and Bäck, T. (2008). Niche radius adaptation with asymmetric sharing. In *Parallel Problem Solving from Nature - PPSN X*, vol. 5199 of *Lecture Notes in Computer Science*, pp. 195–204. Berlin: Springer.

Whitley, D., Mathias, K. E., Rana, S. B., and Dzubera, J. (1996). Evaluating evolutionary algorithms. *Artificial Intelligence*, 85(1–2):245–276.

Yin, X., and Germany, N. (1993). 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*, pp. 450–457. Berlin: Springer.

This article has been cited by:

1. D T Muhamediyeva. 2020. Particle swarm method for solving the global optimization problem using the equilibrium coefficient. *Journal of Physics: Conference Series* **1441**, 012153. [Crossref]
2. Yajun Liang, Xiaofei Wang, Hui Zhao, Tong Han, Zhenglei Wei, Yintong Li. 2019. A covariance matrix adaptation evolution strategy variant and its engineering application. *Applied Soft Computing* **83**, 105680. [Crossref]
3. Yang Lou, Shiu Yin Yuen, Guanrong Chen, Xin Zhang. On-line Search History-assisted Restart Strategy for Covariance Matrix Adaptation Evolution Strategy 3142-3149. [Crossref]
4. Jakub Sawicki, Marcin Łoś, Maciej Smołka, Julen Alvarez-Aramberri. 2019. Using Covariance Matrix Adaptation Evolutionary Strategy to boost the search accuracy in hierarchic memetic computations. *Journal of Computational Science* **34**, 48-54. [Crossref]
5. Mohanna Orujpour, Mohammad-Reza Feizi-Derakhshi, Taymaz Rahkar-Farshi. 2019. Multi-modal forest optimization algorithm. *Neural Computing and Applications* **41**. . [Crossref]
6. Wei Li. 2019. Matrix Adaptation Evolution Strategy with Multi-Objective Optimization for Multimodal Optimization. *Algorithms* **12**:3, 56. [Crossref]
7. Michael Emmerich, Ofer M. Shir, Hao Wang. Evolution Strategies 89-119. [Crossref]
8. Michael Emmerich, Ofer M. Shir, Hao Wang. Evolution Strategies 1-31. [Crossref]
9. Vassiiis Vassiliades, Jean-Baptiste Mouret. Discovering the elite hypervolume by leveraging interspecies correlation 149-156. [Crossref]
10. Peng Wang, Kun Cheng, Yan Huang, Bo Li, Xinggui Ye, Xiuhong Chen. 2018. Multiscale Quantum Harmonic Oscillator Algorithm for Multimodal Optimization. *Computational Intelligence and Neuroscience* **2018**, 1-12. [Crossref]
11. Ruwang Jiao, Sanyou Zeng, Jawdat S. Alkasassbeh, Changhe Li. 2017. Dynamic multi-objective evolutionary algorithms for single-objective optimization. *Applied Soft Computing* **61**, 793-805. [Crossref]
12. Debao Chen, Renquan Lu, Feng Zou, Suwen Li, Peng Wang. 2017. A learning and niching based backtracking search optimisation algorithm and its applications in global optimisation and ANN training. *Neurocomputing* **266**, 579-594. [Crossref]
13. Ali Ahrari, Kalyanmoy Deb, Mike Preuss. 2017. Multimodal Optimization by Covariance Matrix Self-Adaptation Evolution Strategy with Repelling Subpopulations. *Evolutionary Computation* **25**:3, 439-471. [Abstract] [Full Text] [PDF] [PDF Plus]
14. Zhuoran Xu, Hiroyuki Iizuka, Masahito Yamamoto. 2017. Attraction basin sphere estimation approach for niching CMA-ES. *Soft Computing* **21**:5, 1327-1345. [Crossref]
15. Da-Zhao Tan, Wei-Neng Chen, Jun Zhang, Wei-Jie Yu. Fast pedestrian detection using multimodal estimation of distribution algorithms 1248-1255. [Crossref]
16. Mario Garza-Fabre, Shaun M. Kandathil, Julia Handl, Joshua Knowles, Simon C. Lovell. 2016. Generating, Maintaining, and Exploiting Diversity in a Memetic Algorithm for Protein Structure Prediction. *Evolutionary Computation* **24**:4, 577-607. [Abstract] [Full Text] [PDF] [PDF Plus] [Supplemental Material]
17. Ke-Lin Du, M. N. S. Swamy. Dynamic, Multimodal, and Constrained Optimizations 347-369. [Crossref]
18. Sheldon Hui, Ponnuthurai N. Suganthan. 2016. Ensemble and Arithmetic Recombination-Based Speciation Differential Evolution for Multimodal Optimization. *IEEE Transactions on Cybernetics* **46**:1, 64-74. [Crossref]

19. Weiguo Sheng, Shenyong Chen, Mengmeng Sheng, Gang Xiao, Jiafa Mao, Yujun Zheng. 2016. Adaptive Multi-Subpopulation Competition and Multi-Niche Crowding based Memetic Algorithm for Automatic Data Clustering. *IEEE Transactions on Evolutionary Computation* 1-1. [Crossref]
20. Parisa Molavi Damanahi, Gelareh Veisi, Seyyed Javad Seyyed Mahdavi Chabok. Improved differential evolution algorithm based on chaotic theory and a novel Hill-Valley method for large-scale multimodal optimization problems 268-275. [Crossref]
21. Yong Wang, Han-Xiong Li, Gary G. Yen, Wu Song. 2015. MOMMOP: Multiobjective Optimization for Locating Multiple Optimal Solutions of Multimodal Optimization Problems. *IEEE Transactions on Cybernetics* **45**:4, 830-843. [Crossref]
22. Subhodip Biswas, Souvik Kundu, Swagatam Das. 2015. Inducing Niching Behavior in Differential Evolution Through Local Information Sharing. *IEEE Transactions on Evolutionary Computation* **19**:2, 246-263. [Crossref]
23. Lingxi Li, Ke Tang. 2015. History-Based Topological Speciation for Multimodal Optimization. *IEEE Transactions on Evolutionary Computation* **19**:1, 136-150. [Crossref]
24. Subhodip Biswas, Souvik Kundu, Swagatam Das. 2014. An Improved Parent-Centric Mutation With Normalized Neighborhoods for Inducing Niching Behavior in Differential Evolution. *IEEE Transactions on Cybernetics* **44**:10, 1726-1737. [Crossref]
25. Weiguo Sheng, Shengyong Chen, Michael Fairhurst, Gang Xiao, Jiafa Mao. 2014. Multilocal Search and Adaptive Niching Based Memetic Algorithm With a Consensus Criterion for Data Clustering. *IEEE Transactions on Evolutionary Computation* **18**:5, 721-741. [Crossref]
26. Weifeng Gao, Gary G. Yen, Sanyang Liu. 2014. A Cluster-Based Differential Evolution With Self-Adaptive Strategy for Multimodal Optimization. *IEEE Transactions on Cybernetics* **44**:8, 1314-1327. [Crossref]
27. Marcio Weck Pereira, Guenther Schwedersky Neto, Mauro Roisenberg. A topological niching covariance matrix adaptation for multimodal optimization 2562-2569. [Crossref]
28. Miqing Li, Shengxiang Yang, Jinhua Zheng, Xiaohui Liu. 2014. ETEA: A Euclidean Minimum Spanning Tree-Based Evolutionary Algorithm for Multi-Objective Optimization. *Evolutionary Computation* **22**:2, 189-230. [Abstract] [Full Text] [PDF] [PDF Plus]
29. Ofer M. Shir, Jonathan Roslund, Darrell Whitley, Herschel Rabitz. 2014. Efficient retrieval of landscape Hessian: Forced optimal covariance adaptive learning. *Physical Review E* **89**:6. . [Crossref]
30. 2014. Gaussian Classifier-Based Evolutionary Strategy for Multimodal Optimization. *IEEE Transactions on Neural Networks and Learning Systems* **25**:6, 1200-1216. [Crossref]
31. Wei Hong Lim, Nor Ashidi Mat Isa. 2014. Teaching and peer-learning particle swarm optimization. *Applied Soft Computing* **18**, 39-58. [Crossref]
32. Mostafa Z. Ali, Noor H. Awad. 2014. A novel class of niche hybrid Cultural Algorithms for continuous engineering optimization. *Information Sciences* **267**, 158-190. [Crossref]
33. Serguei Vassiliev, Doug Bruce. Water and Oxygen Diffusion Pathways Within Photosystem II. Computational Studies of Controlled Substrate Access and Product Release 351-380. [Crossref]
34. Dongcheng Fan, Weiguo Sheng, Shengyong Chen. A diverse niche radii niching technique for multimodal function optimization 70-74. [Crossref]
35. Aniruddha Basak, Swagatam Das, Kay Chen Tan. 2013. Multimodal Optimization Using a Biobjective Differential Evolution Algorithm Enhanced With Mean Distance-Based Selection. *IEEE Transactions on Evolutionary Computation* **17**:5, 666-685. [Crossref]
36. Michael G. Epitropakis, Xiaodong Li, Edmund K. Burke. A dynamic archive niching differential evolution algorithm for multimodal optimization 79-86. [Crossref]

37. Ilya Loshchilov. CMA-ES with restarts for solving CEC 2013 benchmark problems 369-376. [Crossref]
38. B. Y. Qu, Ponnuthurai Nagaratnam Suganthan, Swagatam Das. 2013. A Distance-Based Locally Informed Particle Swarm Model for Multimodal Optimization. *IEEE Transactions on Evolutionary Computation* **17**:3, 387-402. [Crossref]
39. Subhrajit Roy, Sk. Minhazul Islam, Swagatam Das, Saurav Ghosh, Athanasios V. Vasilakos. 2013. A simulated weed colony system with subregional differential evolution for multimodal optimization. *Engineering Optimization* **45**:4, 459-481. [Crossref]
40. Subhrajit Roy, Sk. Minhazul Islam, Swagatam Das, Saurav Ghosh. 2013. Multimodal optimization by artificial weed colonies enhanced with localized group search optimizers. *Applied Soft Computing* **13**:1, 27-46. [Crossref]
41. B. Y. Qu, P. N. Suganthan, J. J. Liang. 2012. Differential Evolution With Neighborhood Mutation for Multimodal Optimization. *IEEE Transactions on Evolutionary Computation* **16**:5, 601-614. [Crossref]
42. MOSTAFA M. H. ELLABAAN, YEW SOON ONG, Q. C. NGUYEN, JER-LAI KUO. 2012. EVOLUTIONARY DISCOVERY OF TRANSITION STATES IN WATER CLUSTERS. *Journal of Theoretical and Computational Chemistry* **11**:05, 965-995. [Crossref]
43. B.Y. Qu, J.J. Liang, P.N. Suganthan. 2012. Niching particle swarm optimization with local search for multi-modal optimization. *Information Sciences* **197**, 131-143. [Crossref]
44. Ka-Chun Wong, Chun-Ho Wu, Ricky K.P. Mok, Chengbin Peng, Zhaolei Zhang. 2012. Evolutionary multimodal optimization using the principle of locality. *Information Sciences* **194**, 138-170. [Crossref]
45. J.J. Liang, S.T. Ma, B.Y. Qu, B. Niu. Strategy Adaptative Memetic Crowding differential evolution for multimodal optimization 1-7. [Crossref]
46. Ofer M. Shir. Niching in Evolutionary Algorithms 1035-1069. [Crossref]
47. Mike Preuss, Paolo Burelli, Georgios N. Yannakakis. Diversified Virtual Camera Composition 265-274. [Crossref]
48. Mike Preuss. Improved Topological Niching for Real-Valued Global Optimization 386-395. [Crossref]
49. Mostafa Ellabaan, Xianshun Chen, Nguyen Quang Huy. Multi-modal Valley-Adaptive Memetic Algorithm for Efficient Discovery of First-Order Saddle Points 83-92. [Crossref]
50. Souvik Kundu, Subhodip Biswas, Swagatam Das, Digbalay Bose. A Selective Teaching-Learning Based Niching Technique with Local Diversification Strategy 160-168. [Crossref]
51. Rohan Mukherjee, Rupam Kundu, Swagatam Das. Clustered Parent Centric Normal Cross-Over for Multimodal Optimization 276-284. [Crossref]
52. Ratul Majumdar, Ankur Ghosh, Aveek Kumar Das, Souvik Raha, Koushik Laha, Swagatam Das, Ajith Abraham. Artificial Weed Colonies with Neighbourhood Crowding Scheme for Multimodal Optimization 779-787. [Crossref]
53. Pascal Comte, Sergei Vassiliev, Sheridan Houghten, Doug Bruce. 2011. Genetic algorithm with alternating selection pressure for protein side-chain packing and pK(a) prediction. *Biosystems* **105**:3, 263-270. [Crossref]
54. Swagatam Das, Sayan Maity, Bo-Yang Qu, P.N. Suganthan. 2011. Real-parameter evolutionary multimodal optimization — A survey of the state-of-the-art. *Swarm and Evolutionary Computation* **1**:2, 71-88. [Crossref]
55. Jaroslaw Arabas, Lukasz Bartnik, Karol Opara. DMEA — An algorithm that combines differential mutation with the fitness proportionate selection 1-8. [Crossref]