Self-Adaptive Niching CMA-ES with Mahalanobis Metric

Ofer M. Shir, Michael Emmerich and Thomas Bäck

***Abstract*— Existing niching techniques commonly 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. Here we consider *niching* with the *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES), and introduce the *Mahalanobis distance metric* into the niching mechanism, aiming to allow a more accurate spatial classification, based on the ellipsoids of the distribution, rather than hyper-spheres of the *Euclidean metric*. This is tested with the CMA-$\binom{+}{,}$ routines, and compared to two niching frameworks - fixed niche radius as well as self-adaptive niche radius, which is based on the coupling to the step-size. The performance of the different variants is evaluated on a suite of theoretical test-functions. We thus present here the Mahalanobis-assisted CMA-niching as a state-of-the-art niching technique within Evolution Strategies (ES), and propose it as a solution to the so-called *niche radius problem*.**

I. Introduction

Niching methods are used to extend Evolutionary Algorithms (EAs) to multi-modal 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 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 - there is an increasing interest of the applications' community in providing the decision maker with multiple solutions with different *conceptual designs*, for single-criterion or multi-criteria search spaces (see, e.g., [1]).

Niching techniques are often subject to criticism due to the so-called *niche radius problem*. The majority of the niching methods holds 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. Obviously, there are landscapes for which this assumption is not applicable, and where this approach is most likely to fail. 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 were only few attempts to tackle it theoretically [2]. De facto, the niche-radius problem has been addressed at several directions, and a recent study offered a successful self-adaptive approach for an individual niche-radius, which is based on its coupling to the derandomized adaptation of the step-size [3].

Within Evolution Strategies (ES) [4], several niching methods have been proposed (see, e.g., [5]), and upon their successful application to high-dimensional theoretical functions, they were also successfully applied to a real-world physics challenging problem [6]. In that application, the niching technique was shown to be clearly qualitatively superior with respect to multiple restart runs with a single population, for locating highly-fit unique optima which had not been obtained otherwise, and represented different conceptual designs. The distance metric and the niche radius were tailored especially to that application, subject to theoretical justification.

This study considers *niching* with the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), and introduces the *Mahalanobis distance metric* into the niching mechanism, aiming to allow more accurate spatial classification, based on the ellipsoids of the evolving distribution, rather than the uniform hyper-spheres of the *Euclidean distance metric*. This idea can be easily implemented into the CMA-ES niching routines, since the covariance matrix of the distribution - an essential component of the Mahalanobis metric - is already learned by the algorithm. This new approach is tested with the CMA-$\binom{+}{,}$ routines, and compared to two niching frameworks - fixed niche radius mechanism, as well as self-adaptive niche radius mechanism. The latter exploits the self-adaptation of the step-size in the CMA-ES mechanism, and couples the individual niche radius to it. However, it uses the Euclidean distance metric for the classification. Our new proposed approach relies on the other self-adaptation component in the CMA-ES, the covariance matrix, and essentially, it offers an alternative self-adaptive niching approach with the Mahalanobis distance metric. The performance of the different variants is evaluated on a suite of theoretical test-functions, including problems with non-isotropic attractor basins.

The remainder of the paper is organized as follows. Section 2 presents the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), with its two selection variants. In section 3 we introduce the ES niching framework, and its MPR performance analysis. This is followed in section 4 by the description of the experimental setup, the numerical results and a discussion. In section 5 we draw conclusions, summarize our study, and propose future directions in the domain of our research.

Ofer M. Shir, Michael Emmerich and Thomas Bäck are with the Natural Computing Group at Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands; {oshir, emmerich, baeck}@liacs.nl, http://natcomp.liacs.nl.

II. 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 size. This is due to several disruptive effects [7] as well as to the fact that the selection of the *strategy parameters* setting is indirect, i.e. not the vector of a successful mutation is used to adapt the step size parameters, but 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 (CMA-ES) is considered the state-of-the-art derandomized-ES variant, following three generations of derandomized variants [8] [9] [10].

It is important to note, in particular in the context of this study, that the CMA-ES is consisted of two adaptation phases: the *cumulative step-size adaptation* (CSA), which is based on the path length control, as well as the actual *covariance matrix adaptation* (CMA), which is based on the evolution path. See [11] for an analysis of the two components.

A. $(1, \lambda)$-*CMA-ES*

We consider the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [7] *(rank-one update with cumulation)*. 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.

$\vec{p}_c^{(g)} \in \mathbb{R}^n$ 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 step size control. $\mathbf{C}^{(g)} \in \mathbb{R}^{n \times n}$, is the covariance matrix $\left(\mathbf{C}^{(g)} = \mathbf{B}^{(g)}\mathbf{D}^{(g)}\left(\mathbf{B}^{(g)}\mathbf{D}^{(g)}\right)^T\right)$:

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

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

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

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

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

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

where c_c, c_{cov}, c_σ and d_σ are learning/adaptation rates, and $H_{thresh} = \left(1.5 + \frac{1}{n-0.5}\right)\mathbf{E}\left(\left\|\mathcal{N}\left(0, \mathbf{I}\right)\right\|\right)$.

B. $(1 + \lambda)$-*CMA-ES*

This elitist version [12] [13] of the original CMA-ES algorithm combines the classical $(1+\lambda)$ ES strategy [14] [15] [4] with the Covariance Matrix Adaptation concept. The so-called *success rule based step size control* replaces the *path length control* of the CMA-Comma strategy:

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

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

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

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

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

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

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

All weighting variables and learning rates were applied as suggested in [7] and in [12].

III. CMA-ES Niching

The advent of modern Evolution Strategies allows successful global optimization with minimal settings, mostly without recombination, and with a low number of function evaluations. In particular, consider the $(1 \overset{+}{,} \lambda)$ CMA-ES presented in sections II-A and II-B. In the context of niching, the CMA-ES allows the construction of fairly simple and elegant niching algorithm. Here, we provide the reader with some details concerning our different niching approaches. Section III-A presents the general niching framework and the fixed-radius approach, section III-B presents the self-adaptive niche radius mechanism which is based on the coupling to the step-size, and finally section III-C introduces niching with the *Mahalanobis metric*.

A. The Niching Routine: Fixed Niche Radius

We consider a niching technique based on individual search points, which independently and simultaneously perform a derandomized $(1, \lambda)$ or $(1 + \lambda)$ search in different locations of the space. 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

Algorithm 1 Dynamic Peak Identification

input: Pop, q, ρ

Sort Pop in decreasing fitness order
$i := 1$
$NumPeaks := 0$
$DPS := \emptyset$
while $NumPeaks \neq q$ and $i \leq popSize$ **do**
  if $Pop[i]$ is not within ρ of peak in DPS **then**
    $DPS := DPS \cup \{Pop[i]\}$
    $NumPeaks := NumPeaks + 1$
  end if
  $i := i + 1$
end while

output: DPS

Algorithm 2 CMA-ES Niching with Fixed Niche Radius

for $i = 1...(q+p)$ 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 with the DPI Algorithm
for all elements of DPS **do**
  Set peak as a search point
  Inherit the D-set and update it respectively
end for
if N_{DPS}=size of $DPS < q$ **then**
  Generate $q - N_{dps}$ new search points, reset D-sets
end if
if $mod(gen, kappa) = 0$ **then**
  Reset the $(q+1)^{th}...(q+p)^{th}$ search points
end if

defined as the collection of all the dynamic variables of the derandomized algorithm which uniquely define the search at a given point of time. Such dynamic variables 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 $(q + 1)^{th}...(q + p)^{th}$ D-sets are individuals which are randomly re-generated in every cycle of generations (denoted κ) 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 proportionate to q, so setting the value of p reflects a dilemma between applying a wide restart approach for exploring further the search space and exploiting computational resources for the existing niches. In any case, due to the *curse of dimensionality*, p loses its significance as the dimension of the problem gets higher.

Until stopping criteria are 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 [16] (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 respectively.

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

B. Self-Adaptive Niche-Radius Approach

A recent study offered a self-adaptive approach for an individual niche-radius, which is based on the derandomized adaptation of the step-size in the CMA-ES mechanism[3]. The niche radius is coupled to the step-size, and hence relies on the *cumulative step-size adaptation* (CSA) mechanism. Since the step-size does not hold any further spatial information concerning the landscape, the classification into niches uses hyper-spheres, based on the *Euclidean metric*. We provide the reader with the basic idea of this approach.

In addition to an adaptation mechanism of the niche radius, it is necessary to have a secondary selection mechanism, which will take into account 'better' niche radii. A niche radius is initialized for each individual in the population, noted as ρ_i^0. The update step of the niche radius of individual i in generation $g + 1$ is based on the parent's radius and on its step-size:

$$\rho_i^{g+1} = \left(1 - c_i^{g+1}\right) \cdot \rho_{parent}^{g} + c_i^{g+1} \cdot \sqrt{n} \cdot \sigma_{parent}^{g+1} \quad (12)$$

$$c_i^{g+1} = \gamma \cdot \left(1 - \exp\left\{\alpha \cdot \Delta\sigma_i^{g+1}\right\}\right) \quad (13)$$

$$\Delta\sigma_i^{g+1} = \left|\sigma_{parent}^{g+1} - \sigma_{parent}^{g}\right| \quad (14)$$

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

$$\gamma = \begin{cases} \frac{1}{5} & (1,\lambda) \\ \frac{4}{5} & (1+\lambda) \end{cases} \qquad \alpha = \begin{cases} -10 & (1,\lambda) \\ -100 & (1+\lambda) \end{cases} \quad (15)$$

The *dynamic niche count* is defined as:

$$m_i^{dyn} = \begin{cases} n_k & \text{if indiv. } i \text{ is within niche } k \\ \sum_{j=1}^{\lambda} sh(d_{ij}) & \text{otherwise (non-peak individual)} \end{cases} \quad (16)$$

where n_k is the size of the k^{th} dynamic niche, and $sh(d_{ij})$ is the *sharing function*.

The selection of the next parent in each niche is based on the *niche fitness*, after the calculation of the *dynamic niche count* m_i^{dyn}:

$$f_i^{niche} = \frac{f_i}{g\left(m_i^{dyn}, \lambda\right)} \quad (17)$$

with the *dynamic niche count* m_i^{dyn}, and with $g(x, \lambda)$ defined as:

$$g(x,\lambda) = 1 + \Theta(\lambda - x) \cdot \frac{(\lambda - x)^2}{\lambda} + \Theta(x - \lambda) \cdot (\lambda - x)^2 \quad (18)$$

A pseudo-code for the *self-adaptive* niching routine is presented as Algorithm 3.

Algorithm 3 CMA-ES Niching with Adaptive Niche Radius

for $i = 1...(q+p)$ search points **do**
  Generate λ samples based on the D-set of i
  Update the niche radius ρ_i^{g+1} according to Eq.12
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 (Eq. 17)
  Set indiv. with best niche fitness as a search point
  Inherit the D-set and update it respectively
end for
...

C. CMA-ES Niching with Mahalanobis Metric

Existing niching techniques, and in particular those presented in sections III-A and III-B, 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 distribution of the decision space, it is worthwhile to use it for a better spatial classification mechanism within the niching framework. In essence, this is an upgraded niching mechanism, so to say, as it captures a more accurate spatial classification into niches, and it is also self-adaptive by definition.

The careful reader should note that this idea 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*, for details see [6]). After giving this motivation, we proceed with discussing the details of this idea.

1) The Metric: Consider the *Mahalanobis 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* is defined as:

$$d(\vec{v}, \vec{m}) = \sqrt{(\vec{v} - \vec{m})^T \Sigma^{-1} (\vec{v} - \vec{m})} \tag{19}$$

It can be shown that the *iso-surfaces* of this metric are ellipsoids which 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 reduces to the normalized *Euclidean* distance, and the iso-surfaces become hyper-spheres.

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 of an individual $\vec{y}$ to the niche by $d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \mathbf{C_x}^{-1} (\vec{x} - \vec{y})}$. In essence, since different individuals have different covariance matrices, this operation is not symmetric. Thus the classification into niches depends on the so-called *peak individuals*, which are selected based on their higher fitness.

Note that the proposed mechanism lacks the secondary selection mechanism, which was necessary for the self-adaptive niche radius approach, but it does not seem to be required here. In first approximation, we offer the following explanation for that: In the general context of global optimization, a successful unimodal search relies more on the adaptive covariance matrix, rather than on the mutative step-size, alone. Thus, the selected individuals must carry a well-adapted covariance matrix, whereas a bad step-size may actually be tolerated. Apparently, this is not the case for the process of forming the niches with the Euclidean metric - a bad niche radius, coupled to the step-size, can harm dramatically the process. Hence, the secondary selection mechanism was required.

3) Numerical Implementation: As for the technical details, we discuss here the numerical implementation of the Mahalanobis 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 *eigen-decomposition* of the covariance matrix $\mathbf{C}$, which is calculated every generation, yields

$$\mathbf{C} = \mathbf{BD}(\mathbf{BD})^T \tag{20}$$

where $\mathbf{D} = diag\left(\sqrt{\Lambda_1}, \sqrt{\Lambda_2}, ..., \sqrt{\Lambda_n}\right)$, with the *eigenvalues* $\{\Lambda_i\}_{i=1}^n$. In order to obtain $\mathbf{C}^{-1}$, one can derive,

$$\mathbf{C}^{-1} = \left[\mathbf{BD}(\mathbf{BD})^T\right]^{-1} = \mathbf{B}^{T\ -1}\mathbf{D}^{T\ -1}\mathbf{D}^{-1}\mathbf{B}^{-1} = \mathbf{B} \cdot diag\left(\frac{1}{\Lambda_1}, \frac{1}{\Lambda_2}, ..., \frac{1}{\Lambda_n}\right) \cdot \mathbf{B}^T \tag{21}$$

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., [17]), we observe in practice[1] 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 calculation time in comparison to the inversion procedure. Hence, it pays off to follow the derivation given here.

Due to numerical features of the eigen-decomposition, which were also discussed by Hansen et. al. ([7] pp. 20), but are crucial here for the inversion operation of the covariance matrix, we choose to introduce a lower bound to the eigenvalues: $\Lambda_{min} = 10^{-10}$.

IV. 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.

[1]The calculations are done with MATLAB 7.0.

A. 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 [16], we adopt the performance metric called the *maximum peak ratio statistic*. This metric measures the quality as well as the number of optima given as a final result by the evolutionary algorithm. Explicitly, given the fitness of the optima in the final population $\left\{\tilde{f}_i\right\}_{i=1}^{q}$, and the real optima of the objective function $\left\{\hat{\mathcal{F}}_i\right\}_{i=1}^{q}$, the *maximum peak ratio* is defined for a *minimization problem* as follows:

$$MPR = \frac{\sum_{i=1}^{q} \hat{\mathcal{F}}_i}{\sum_{i=1}^{q} \tilde{f}_i} \tag{22}$$

Also, given a maximization problem the MPR is defined as the obtained optima divided by the real optima. The real optima of the objective function cannot always be derived analytically, particularly in complex problems. Hence, some optima are obtained numerically when necessary.

Although this metric was originally introduced to be analyzed by means of the saturation MPR value for performance evaluation, a new perspective was introduced in [5]. 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*:

$$y(t) = \frac{a}{1 + \exp\left\{c\left(t - T\right)\right\}} \tag{23}$$

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 [5] that the logistic parameters should be interpreted in the following way - T as the *learning period* of the algorithm and the absolute value of c as its *niching formation acceleration*. That study compared the behavior of two ES niching mechanisms, and concluded with the claim that there was a clear *trade-off*: either a long learning period followed by a high niching acceleration or a short learning period followed by a low niching acceleration.

Here we shall focus in 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.

B. Test Functions

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 hyper-sphere, followed by an exponentially increasing number of minima in the up-going hyper-spheres . Ackley's function has been widely investigated in the context of *evolutionary computation* [15].
- $\mathcal{L}$ - also known as $F2$, as had been originally introduced in [18] - is a sinusoid trapped in an exponent envelope. The parameter k determines the sharpness of the peaks in the function landscape (we set $k = 6$). $\mathcal{L}$ has one global minimum, regardless of n and k. It has been a popular test function for GA niching methods.
- The Rastrigin function [2] has one global minimum, surrounded by a large number of local minima arranged in a lattice configuration.
  We also consider its shifted-rotated variant [19].
- The Griewank function [2] has its global minimum $(f^* = 0)$ at the origin, with several thousand global minima in the area of interest. There are 4 sub-optimal minima $f \approx 0.0074$ with $\vec{x}^* \approx \left(\pm\pi, \pm\pi\sqrt{2}, 0, 0, 0, ...0\right)$.
  We also consider its shifted-rotated variant [19].
- The function after Fletcher and Powell [15] is a non-separable *non-linear parameter estimation problem*, which has a non-uniform distribution of 2^n minima. It has non-isotropic attractor basins.

Table I summarizes the unconstrained multimodal test functions as well as their initialization intervals.

C. Modus Operandi

We consider *four* niching approaches, subject to *two* selection strategies - comma and plus: fixed-radius (CMA), self-adaptive niche radius (SA-CMA), Mahalanobis (M-CMA), and Mahalanobis with self-adaptive niche radius (M-SA-CMA). The 8 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 core mechanism of a $(1 \overset{+}{,} 10)$-strategy per niche and initial points are sampled uniformly within the initialization intervals. Initial step sizes are set to $\frac{1}{4}$ of the intervals, and initial niche radii for the self-adaptive routines are set to $\sqrt{n} \cdot \sigma_{init}$. 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 $p = 0$.

Function evaluations: the idea is to allocate a fixed number of evaluations per peak $\left(n \cdot 10^4\right)$, 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-squared-error method).

D. Performance Analysis

A recent study evaluated the performance of niching techniques, based on derandomized ES variants, on the same suite of theoretical test functions [20]. Notice, however, that it applied a fixed radius approach, and set $p = 1$.

TABLE I

TEST FUNCTIONS TO BE *minimized* AND INITIALIZATION DOMAINS. FOR SOME OF THE NON-SEPARABLE FUNCTIONS, WE APPLY TRANSLATION AND ROTATION: $\vec{y} = \mathcal{O}\left(\vec{x} - \vec{r}\right)$ WHERE $\mathcal{O}$ IS AN ORTHOGONAL ROTATION MATRIX, AND $\vec{r}$ IS A SHIFTING VECTOR.

Separable:

Name	Function	Init	Niches
$\mathcal{M}$	$\mathcal{M}\left(\vec{x}\right) = -\frac{1}{n}\sum_{i=1}^{n}\sin^{\alpha}\left(5\pi x_i\right)$	$[0,1]^n$	100
Ackley	$\mathcal{A}(\vec{x}) = -c_1 \cdot \exp\left(-c_2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)$ $-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(c_3 x_i)\right) + c_1 + e$	$[-10,10]^n$	$2 \cdot n + 1$
$\mathcal{L}$	$\mathcal{L}(\vec{x}) = -\prod_{i=1}^{n}\sin^k\left(l_1 \pi x_i + l_2\right) \cdot \exp\left(-l_3\left(\frac{x_i - l_4}{l_5}\right)^2\right)$	$[0,1]^n$	$n+1$
Rastrigin	$\mathcal{R}(\vec{x}) = 10n + \sum_{i=1}^{n}\left(x_i^2 - 10\cos\left(2\pi x_i\right)\right)$	$[-1,5]^n$	$n+1$
Griewank	$\mathcal{G}\left(\vec{x}\right) = 1 + \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)$	$[-10,10]^n$	5

Non-separable:

Name	Function	Init	Niches
Fletcher-Powell	$\mathcal{F}(\vec{x}) = \sum_{i=1}^{n}\left(A_i - B_i\right)^2$ $A_i = \sum_{j=1}^{n}\left(a_{ij}\cdot\sin(\alpha_j) + b_{ij}\cdot\cos(\alpha_j)\right)$ $B_i = \sum_{j=1}^{n}\left(a_{ij}\cdot\sin(x_j) + b_{ij}\cdot\cos(x_j)\right)$ $a_{ij},\, b_{ij} \in [-100,100]\,;\quad \vec{\alpha} \in [-\pi,\pi]^n$	$[-\pi,\pi]^n$	10
Shifted Rotated Rastrigin	$\tilde{\mathcal{R}}(\vec{x}) = 10n + \sum_{i=1}^{n}\left(y_i^2 - 10\cos\left(2\pi y_i\right)\right)$	$[-5,5]^n$	$n+1$
Shifted Rotated Griewank	$\tilde{\mathcal{G}}\left(\vec{x}\right) = 1 + \sum_{i=1}^{n}\frac{y_i^2}{4000} - \prod_{i=1}^{n}\cos\left(\frac{y_i}{\sqrt{i}}\right)$	$[0,600]^n$	5

TABLE II

GLOBAL MINIMUM REACHED IN 100 RUNS (CMA DENOTES A $(1,\lambda)$-STRATEGY, CMA+ DENOTES A $(1+\lambda)$-STRATEGY). THE BEST RESULT, PER STRATEGY, IS HIGHLIGHTED.

Test-Case	**CMA**	**M-CMA**	**SA-CMA**	**M-SA-CMA**	**CMA+**	**M-CMA+**	**SA-CMA+**	**M-SA-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%
$\mathcal{F}:\ n=4$	100%	100%	100%	100%	100%	100%	100%	100%
$\mathcal{F}:\ n=10$	25%	40%	36%	**46%**	17%	22%	34%	**37%**
$\tilde{\mathcal{R}}:\ n=3$	46%	**54%**	14%	24%	50%	**66%**	10%	26%
$\tilde{\mathcal{R}}:\ n=10$	**8%**	2%	0%	0%	0%	0%	0%	0%
$\tilde{\mathcal{G}}:\ n=3$	**10%**	0%	0%	0%	**9%**	0%	0%	0%
$\tilde{\mathcal{G}}:\ n=10$	0%	0%	0%	0%	0%	0%	0%	0%

We discuss here the performance analysis at two levels:

Global Minimum: Table II contains the percentage of runs in which the global minimum was located. $\mathcal{M}$ is discarded from the table, as its global minimum was always found, by all algorithms, for every dimension n under investigation. For the *comma* strategy (four left columns), 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 metric does not improve the global optimization. We may conclude that there is no clear 'winner', and that the Mahalanobis-assisted routines do not achieve a dramatic improvement in global optimization. This is an expected result, as this metric focuses in the formation of the niches.

MPR Saturation: This scalar value represents, to some degree, the quality of the obtained minima, and thus the final result of the niching process. Tables III and IV present the mean and the standard deviation of the saturation MPR values for the different test cases.

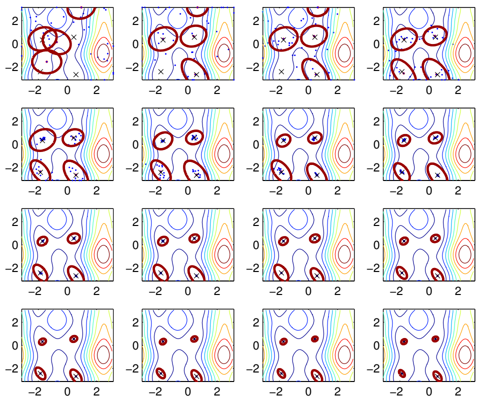

Fig. 1. A snapshot gallery: the adaptation of the *classification-ellipses*, subject to the Mahalanobis metric with the updating covariance matrix, in the CMA+ strategy for a $2D$ Fletcher-Powell problem. The contour of the landscape is given as the background, where the X's indicate the real optima, the dots are the evolving individuals, and the ellipses are plotted based on the peak individual. A snapshot is taken every 2 cycles (i.e., every 80 function evaluations).

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

Note that the niching routines, except for the fixed niche radius case, fail on the Ackley function - i.e., they locate only the global minimum (all niches are located in the global attractor's basin). This effect has been observed in the past [3], and was explained by the strong basin of attraction of the global minimum, in comparison to the sub-optimal minima.

E. Behavior Analysis

The MPR analysis allows us to compare the *niching acceleration* of the different routines. Due to space limitations, we omit here the tables with the numerical results of this parameter.

The comma strategy has higher niching acceleration values (the parameter c in Eq. 23), as expected from past observations [20]. Within each strategy, there is a clear trend of higher niching acceleration for the Mahalanobis-assisted routines. This result is pretty much intuitive - a more accurate spatial classification, as obtained by the Mahalanobis metric, allows the niching to form and to converge faster.

Figure 1 illustrates the adaptation of the classification-ellipses, in the Mahalanobis-assisted CMA+ strategy for a $2D$ Fletcher-Powell problem. It can be observed that each niche has its own characteristic matrix and convergence profile.

V. Discussion and Outlook

We have introduced the Mahalanobis distance metric into the framework of ES niching with covariance matrix adaptation, aiming to improve the process of niche formation by a more accurate spatial classification. This new idea relies on the successful learning of the covariance matrix, and it is self-adaptive by nature. We have offered a numerical simplification of the required calculation, which was observed to pay off in terms of computation time.

Mahalanobis-assisted niching routines, based on CMA-$(\overset{+}{,})$ with fixed as well as self-adaptive niche radius approaches, were tested on a suite of theoretical test functions. The new metric seems to achieve its goal and improve the niching process, in terms of obtaining on average higher quality sub-optima, subject to higher *niching acceleration*. It does not seem to improve nor to harm, on average, the location of global minimum, as expected.

We thus present here the Mahalanobis-assisted CMA-niching as a state-of-the-art niching technique within Evolution Strategies, and propose it as a solution to the so-called *niche-radius problem*.

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

[1] G. Avigad, A. Moshaiov, and N. Brauner, "Concept-based interactive brainstorming in engineering design," *JACIII*, vol. 8, no. 5, pp. 454–459, 2004.

[2] A. Törn and A. Zilinskas, *Global Optimization*. Springer, 1987, vol. 350.

[3] O. M. Shir and T. Bäck, "Niche radius adaptation in the cma-es niching algorithm." in *Parallel Problem Solving from Nature - PPSN IX, 9th International Conference, Reykjavik, Iceland, September 9-13, 2006, Procedings*, ser. Lecture Notes in Computer Science, vol. 4193. Springer, 2006, pp. 142–151.

[4] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies a comprehensive introduction," *Natural Computing: an international journal*, vol. 1, no. 1, pp. 3–52, 2002.

[5] O. M. Shir and T. Bäck, "Dynamic niching in evolution strategies with covariance matrix adaptation," in *Proceedings of the 2005 Congress on Evolutionary Computation CEC-2005*. Piscataway, NJ, USA: IEEE Press, 2005.

[6] O. M. Shir, C. Siedschlag, T. Bäck, and M. J. Vrakking, "Niching in evolution strategies and its application to laser pulse shaping," in *Lecture Notes in Computer Science*, vol. 3871. Springer, 2006.

[7] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[8] A. Ostermeier, A. Gawelczyk, and N. Hansen, "A derandomized approach to self adaptation of evolution strategies." Tech. Rep., 1993.

[9] ——, "Step-size adaptation based on non-local use of selection information." in *PPSN*, ser. Lecture Notes in Computer Science, vol. 866. Springer, 1994.

[10] N. Hansen, A. Ostermeier, and A. Gawelczyk, "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)*, 1995.

[11] H.-G. Beyer and D. V. Arnold, "Qualms regarding the optimality of cumulative path length control in csa/cma-evolution strategies," *Evolutionary Computation*, vol. 11, no. 1, pp. 19–28, 2003.

[12] C. Igel, N. Hansen, and S. Roth, "The multi-objective variable metric evolution strategy," Ruhr-Universität Bochum, Tech. Rep. 2005-04, 2005.

[13] C. Igel, T. Suttorp, and N. Hansen, "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)*. ACM Press, 2006, pp. 453–460.

TABLE III

THE SATURATION MPR VALUE, $(1,\lambda)$-STRATEGY: MEAN AND STANDARD DEVIATION OVER 100 RUNS.

Test-Case	**CMA**	**M-CMA**	**SA-CMA**	**M-SA-CMA**
$\mathcal{M}:\ n=3$	1 ± 0	1 ± 0	1 ± 0	1 ± 0
$\mathcal{M}:\ n=10$	0.9943 ± 0.0019	0.9670 ± 0.0030	1 ± 0	1 ± 0
$\mathcal{M}:\ n=40$	0.9560 ± 0.0058	0.9527 ± 0.0080	0.9935 ± 0.0012	**0.9946 ± 0.0018**
$\mathcal{A}:\ n=3$	0.9378 ± 0.0435	N.A.	0.8600 ± 0.1428	N.A.
$\mathcal{A}:\ n=10$	0.9094 ± 0.0332	N.A.	N.A.	N.A.
$\mathcal{L}:\ n=3$	0.8638 ± 0.0924	**0.8700 ± 0.1064**	0.7133 ± 0.0833	0.8337 ± 0.0994
$\mathcal{L}:\ n=10$	0.2403 ± 0.0863	0.3885 ± 0.1137	0.4776 ± 0.0803	**0.5635 ± 0.1048**
$\mathcal{R}:\ n=3$	0.3010 ± 0.0805	0.2283 ± 0.0628	0.1585 ± 0.0411	**0.3052 ± 0.1025**
$\mathcal{R}:\ n=10$	**0.1033 ± 0.0446**	0.0621 ± 0.0112	0.0820 ± 0.0188	0.0944 ± 0.0216
$\mathcal{G}:\ n=3$	0.2485 ± 0.1256	0.2339 ± 0.0451	**0.2830 ± 0.0920**	0.2551 ± 0.0640
$\mathcal{G}:\ n=10$	**0.2524 ± 0.1687**	0.1951 ± 0.0400	0.1858 ± 0.0917	0.1900 ± 0.0412
$\mathcal{F}:\ n=4$	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**
$\tilde{\mathcal{R}}:\ n=3$	**0.3305 ± 0.1026**	0.2309 ± 0.0407	0.1383 ± 0.0508	0.2684 ± 0.0740
$\tilde{\mathcal{R}}:\ n=10$	**0.1304 ± 0.0394**	0.0873 ± 0.0422	0.0694 ± 0.0186	0.0930 ± 0.0175
$\tilde{\mathcal{G}}:\ n=3$	0.0720 ± 0.0430	0.0779 ± 0.0435	0.0846 ± 0.0475	**0.0875 ± 0.0361**
$\tilde{\mathcal{G}}:\ n=10$	0.1336 ± 0.0376	0.1441 ± 0.0365	0.1220 ± 0.0348	**0.1610 ± 0.0337**

TABLE IV

THE SATURATION MPR VALUE, $(1+\lambda)$-STRATEGY: MEAN AND STANDARD DEVIATION OVER 100 RUNS.

Test-Case	**CMA+**	**M-CMA+**	**SA-CMA+**	**M-SA-CMA+**
$\mathcal{M}:\ n=3$	1 ± 0	1 ± 0	1 ± 0	1 ± 0
$\mathcal{M}:\ n=10$	0.9911 ± 0.0027	0.9860 ± 0.0027	1 ± 0	1 ± 0
$\mathcal{M}:\ n=40$	0.9745 ± 0.0076	0.9802 ± 0.0066	1 ± 0	1 ± 0
$\mathcal{A}:\ n=3$	0.9893 ± 0.0257	**0.9995 ± 0.0088**	0.9295 ± 0.0301	0.9370 ± 0.1592
$\mathcal{A}:\ n=10$	0.9455 ± 0.0174	**0.9871 ± 0.0191**	N.A.	N.A.
$\mathcal{L}:\ n=3$	0.9592 ± 0.0329	**0.9618 ± 0.0364**	0.8185 ± 0.0786	0.9187 ± 0.0654
$\mathcal{L}:\ n=10$	**0.4543 ± 0.1160**	0.3732 ± 0.1148	0.4229 ± 0.1080	0.4324 ± 0.0900
$\mathcal{R}:\ n=3$	0.5279 ± 0.1183	**0.5519 ± 0.1067**	0.1631 ± 0.0718	0.2497 ± 0.0893
$\mathcal{R}:\ n=10$	**0.1024 ± 0.0399**	0.0769 ± 0.0271	0.0486 ± 0.0088	0.0533 ± 0.0111
$\mathcal{G}:\ n=3$	0.3255 ± 0.0941	0.3343 ± 0.1006	0.3053 ± 0.1138	**0.4936 ± 0.2338**
$\mathcal{G}:\ n=10$	0.0372 ± 0.0082	0.0530 ± 0.0146	**0.0624 ± 0.0188**	0.0601 ± 0.0149
$\mathcal{F}:\ n=4$	0.0007 ± 0.0003	0.8615 ± 0.3847	0.0044 ± 0.0024	**0.9905 ± 0.0376**
$\mathcal{F}:\ n=10$	0.0001 ± 0.0001	0.0001 ± 0.0001	**0.0005 ± 0.0005**	0.0001 ± 0.0001
$\tilde{\mathcal{R}}:\ n=3$	0.4864 ± 0.1373	**0.5634 ± 0.1398**	0.1346 ± 0.0506	0.2491 ± 0.1288
$\tilde{\mathcal{R}}:\ n=10$	**0.0808 ± 0.0304**	0.0804 ± 0.0175	0.0440 ± 0.0060	0.0408 ± 0.0059
$\tilde{\mathcal{G}}:\ n=3$	0.0720 ± 0.0430	0.0779 ± 0.0435	**0.0846 ± 0.0475**	0.0815 ± 0.0361
$\tilde{\mathcal{G}}:\ n=10$	0.1336 ± 0.0376	0.1441 ± 0.0365	0.1220 ± 0.0348	**0.1610 ± 0.0337**

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

[15] T. Bäck, *Evolutionary algorithms in theory and practice*. New York, NY, USA: Oxford University Press, 1996.

[16] B. Miller and M. Shaw, "Genetic algorithms with dynamic niche sharing for multimodal function optimization," in *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC'96)*, New York, NY, USA, 1996.

[17] P. Giorgi, C.-P. Jeannerod, and G. Villard, "On the complexity of polynomial matrix computations," in *ISSAC '03: Proceedings of the 2003 international symposium on Symbolic and algebraic computation*. New York, NY, USA: ACM Press, 2003.

[18] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*. Mahwah, NJ, USA: Lawrence Erlbaum Associates, Inc., 1987, pp. 41–49.

[19] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization," `http://www.ntu.edu.sg/home/EPNSugan/`, Tech. Rep., 2005.

[20] O. M. Shir and T. Bäck, "Performance analysis of niching algorithms based on derandomized-ES variants," to appear in the GECCO-2007 Proceedings.