Performance Analysis of Niching Algorithms Based on Derandomized-ES Variants

Ofer M. Shir
Natural Computing Group, Leiden University
Niels Bohrweg 1, Leiden, The Netherlands
oshir@liacs.nl

Thomas Bäck*
Natural Computing Group, Leiden University
Niels Bohrweg 1, Leiden, The Netherlands
baeck@liacs.nl

ABSTRACT

A survey of niching algorithms, based on 5 variants of *derandomized* Evolution Strategies (ES), is introduced. This set of niching algorithms, ranging from the very first *derandomized approach to self-adaptation of ES* to the sophisticated $(1 \overset{+}{,} \lambda)$ Covariance Matrix Adaptation (CMA), is applied to multimodal continuous theoretical test functions, of different levels of difficulty and various dimensions, and compared with the MPR performance analysis tool. While characterizing the performance of the different derandomized variants in the context of niching, some conclusions concerning the niching formation process of the different mechanisms are drawn, and the hypothesis of a tradeoff between learning time and niching acceleration is numerically confirmed. Niching with $(1 + \lambda)$-CMA core mechanism is shown to experimentally outperform all the other variants. Some theoretical arguments supporting the advantage of a plus-strategy for niching are discussed.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: ARTIFICIAL INTELLIGENCE—*Problem Solving, Control Methods, and Search*

General Terms

Algorithms, Experimentation, Performance

Keywords

Niching, Derandomized Evolution Strategies, MPR Analysis

1. INTRODUCTION

Evolutionary Algorithms (EAs), popular population-based stochastic search-methods, have the tendency to lose diversity within their population of feasible solutions and to converge into a single solution [11, 2, 3]. *Niching methods*, the extension of EAs to multi-modal optimization, address this issue by maintaining the diversity of certain properties within the population - and this way they allow parallel convergence into multiple good solutions in multimodal domains. 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*, as will be explained. The majority of the niching methods hold 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 isn't 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 [20]. 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 [17].

Evolution Strategies (ES) [4] are a *canonical EA for continuous function optimization*, due to their straightforward encoding, their specific variation operators, the self-adaptation of their mutation distribution as well as to their high performance in this domain in comparison with other methods on benchmark problems. Even for large dimensions, an ES is a suitable method, and was shown to outperform other competing methods [3]. However, the standard ES approaches are exposed to several disruptive effects, especially concerning the individual mutative step-size control. The family of derandomized Evolution Strategies [6] offers an improved mutative ES mechanism, and are considered as the state of the art strategies.

Several ES niching methods have been proposed (see, e.g., [16]), and upon their successful application to high-dimensional theoretical functions, they were also successfully applied

*NuTech Solutions, Martin-Schmeisser-Weg 15, 44227 Dortmund, Germany.

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

to a real-world physics challenging problem [18]. 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 paper presents a survey of ES niching techniques, based on 5 variants of *derandomized* Evolution Strategies, applied to a set of continuous theoretical test functions of different levels of difficulty. As far as we know this is the first comparison of these ES variants, in particular in the context of niching. The performance of the algorithms is evaluated based on the so-called MPR analysis tool [16], which allows to characterize to some degree the learning behavior, the niching formation process and the saturation profile of the different mechanisms. Niching with $(1+\lambda)$-CMA core mechanism is shown to experimentally outperform all the other variants. The numerical results are consistent, and support our experimental conclusions. Some theoretical arguments supporting the advantage of a plus-strategy for niching are discussed.

The remainder of the paper is organized as follows. Section 2 presents the various evolutionary core mechanisms in use for this survey - the so-called family of derandomized algorithms. 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.

2. THE FAMILY OF DERANDOMIZED EVOLUTION STRATEGIES

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 [6] 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 first versions of *derandomized ES algorithms* introduced a controlled global step-size in order to monitor the individual step-sizes by decreasing the stochastic effects of a probabilistic sampling. The selection disturbance was completely removed with later versions by omitting the adaptation of strategy parameters by means of probabilistic sampling. This was combined with individual information from the last generation (the successful mutations, i.e. of selected offspring), and then adjusted to *correlated mutations*. Later on, the concept of *adaptation by accumulated information* was introduced, aiming to use wisely the past information for the purpose of step-size adaptation: instead of using the information from the last generation only, it was successfully generalized to a weighted average of the previous generations.

Note that the different variants of *derandomized-ES* hold different numbers of strategy parameters to be adapted, and this is a factor in the learning speed of the optimization routine. The different algorithms hold a number of strategy parameters in either a *linear* or *quadratic* order in terms of the dimensionality of the search problem n, and there seems to be a trade-off between the number of strategy parameters and the time needed for the adaptation/learning process of the step-sizes. We hereby present briefly different derandomized-ES algorithms that are used in our niching framework.

DR1

The first derandomized attempt [13] couples the successful mutations to the selection of decision parameters, and learns the mutation step-size directly from the difference vectors between parents and selected offspring:

$$\vec{x}^{g+1} = \vec{x}^{g} + \xi^{k}\delta^{g}\vec{\xi}^{k}_{scal}\vec{\delta}^{g}_{scal}\vec{Z}^{k} \tag{1}$$

$$\delta^{g+1} = \left(\xi_{sel}\right)^{\beta} \cdot \delta^{g} \qquad \vec{\delta}^{g+1}_{scal} = \left(\vec{\xi}^{sel}_{scal} + b\right)^{\beta_{scal}} \cdot \vec{\delta}^{g}_{scal} \tag{2}$$

where $\vec{\xi}_{scal} = \vec{\mathcal{N}}\left(0,1\right)^{+}$, $\vec{Z} \in \left\{-1,+1\right\}^{n}$, and β, β_{scal}, b and ξ^{k} are constants.

DR2

This variant [14] aims to accumulate information about the correlation or anti-correlation of past mutation vectors in order to adapt the step-size:

$$\vec{x}^{g+1} = \vec{x}^{g} + \delta^{g}\vec{\delta}^{g}_{scal}\vec{Z}^{k} \qquad \vec{Z}^{k} = \vec{\mathcal{N}}\left(0,1\right) \tag{3}$$

$$\vec{Z}^{g} = c\vec{Z}_{sel} + \left(1-c\right)\vec{Z}^{g-1} \tag{4}$$

$$\delta^{g+1} = \delta^{g} \cdot \left(exp\left(\frac{\left|\vec{Z}^{g}\right|}{\sqrt{n}\sqrt{\frac{c}{2-c}}} - 1 + \frac{1}{5n}\right)\right)^{\beta} \tag{5}$$

$$\vec{\delta}^{g+1}_{scal} = \vec{\delta}^{g}_{scal} \cdot \left(\frac{\left|\vec{Z}^{g}\right|}{\sqrt{\frac{c}{2-c}}} + 0.35\right)^{\beta_{scal}} \tag{6}$$

DR3

This third generation [7] achieved invariance w.r.t. the scaling of variables and the rotation of the coordinate system:

$$\vec{x}^{g+1} = \vec{x}^{g} + \delta^{g}\xi^{k}\vec{y}^{k}, \qquad \vec{y}^{k} = c_{m}\mathbf{B}\cdot\vec{z}^{k} \tag{7}$$

$$\mathbf{B} = \left(\vec{b}_{1},...,\vec{b}_{m}\right), \qquad \delta^{g+1} = \delta^{g}\left(\xi^{k}\right)^{\beta} \tag{8}$$

$$\vec{b}^{g+1}_{1} = \left(1-c\right)\cdot\vec{b}^{g}_{1} + c\cdot\left(c_{u}\xi^{k}\vec{y}^{k}\right), \qquad \vec{b}^{g+1}_{i+1} = \vec{b}^{g}_{i} \tag{9}$$

where $\vec{z} = \vec{\mathcal{N}}\left(0,1\right)$, $\mathbf{B} \in \mathbb{R}^{m\times n}$ and m is between n^{2} and $2n^{2}$.

$(1,\lambda)$-CMA-ES

We consider the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [6] *(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 $\left(\mathbf{C}^{(g)} = \mathbf{B}^{(g)}\mathbf{D}^{(g)}\left(\mathbf{B}^{(g)}\mathbf{D}^{(g)}\right)^T\right)$:

$$\vec{x}^{g+1} = \vec{x}^g + \sigma_g \mathbf{B}^g \mathbf{D}^g \vec{z}_k^{g+1} \tag{10}$$

$$H_\sigma^{g+1} = \begin{cases} 1 & \text{if } \frac{\left\|\vec{p}_\sigma^{g+1}\right\|}{\sqrt{1-(1-c_\sigma)^{2\ (g+1)}}} < H_{thresh} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

$$\vec{p}_c^{g+1} = (1-c_c) \cdot \vec{p}_c^g + H_\sigma^{g+1}\sqrt{c_c(2-c_c)} \cdot \mathbf{B}^g \mathbf{D}^g \vec{z}_{sel}^{g+1} \tag{12}$$

$$\mathbf{C}^{g+1} = (1-c_{cov}) \cdot \mathbf{C}^g + c_{cov} \cdot \vec{p}_c^{g+1}\left(\vec{p}_c^{g+1}\right)^T \tag{13}$$

$$\vec{p}_\sigma^{g+1} = (1-c_\sigma) \cdot \vec{p}_\sigma^g + \sqrt{c_\sigma(2-c_\sigma)} \cdot \mathbf{B}^g \vec{z}_{sel}^{g+1} \tag{14}$$

$$\sigma^{g+1} = \sigma^g \cdot exp\left(\frac{c_\sigma}{d_\sigma} \cdot \left(\frac{\left\|\vec{p}_\sigma^{g+1}\right\|}{\mathbf{E}\left(\left\|\mathcal{N}\left(0, \mathbf{I}\right)\right\|\right)} - 1\right)\right) \tag{15}$$

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

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

This elitist version [8] [9] of the original CMA-ES algorithm combines the classical $(1+\lambda)$ ES strategy [15] [3] [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:

$$\vec{x}^{g+1} = \vec{x}^g + \sigma_g \mathbf{B}^g \mathbf{D}^g \vec{z}_k^{g+1} \tag{16}$$

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

$$\bar{p}_{succ} = (1-c_p) \cdot \bar{p}_{succ} + c_p \cdot p_{succ} \tag{17}$$

$$\sigma^{g+1} = \sigma^g \cdot \exp\left(\frac{1}{d} \cdot \left(\bar{p}_{succ} - \frac{p_{succ}^{target}}{1 - p_{succ}^{target}}\left(1 - \bar{p}_{succ}\right)\right)\right) \tag{18}$$

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

$$\vec{p}_c = \begin{cases} \left(1-c_c\right)\vec{p}_c + \sqrt{c_c\left(2-c_c\right)} \cdot \frac{\vec{x}_{sel}^{g+1} - \vec{x}^g}{\sigma_{parent}^g} & \text{if } \bar{p}_{succ} < p_\Theta \\ \left(1-c_c\right)\vec{p}_c & \text{otherwise} \end{cases} \tag{19}$$

$$\mathbf{C}^{g+1} = \begin{cases} \left(1-c_{cov}\right) \cdot \mathbf{C}^g + c_{cov} \cdot \vec{p}_c \vec{p}_c^T & \text{if } \bar{p}_{succ} < p_\Theta \\ \left(1-c_{cov}\right) \cdot \mathbf{C}^g + c_{cov} \cdot \left(\vec{p}_c \vec{p}_c^T + c_c\left(2-c_c\right)\mathbf{C}^g\right) & \text{otherwise} \end{cases} \tag{20}$$

Here, all weighting variables and learning rates are as suggested in the given citations, and especially in [6] and in [8].

3. ES DYNAMIC 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 $(1 \overset{+}{,} \lambda)$ derandomized ES variants presented in the previous section. In the context of niching, this generation of modern ES variants allows the construction of fairly simple and elegant niching algorithms. We provide the reader with some details concerning our niching framework.

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 $\rho$ of peak in $DPS$ then
        $DPS := DPS \cup \{Pop[i]\}$
        $NumPeaks := NumPeaks + 1$
    end if
    $i := i + 1$
end while
output: DPS
```

For the sake of simplicity of the comparison between the different core mechanisms, we limit this study to a niching approach based on a fixed niche radius, and without recombination.

3.1 The Niching Routine

We consider a niching technique with 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 the niches' representatives for the next iteration, or simply the next search points, based on the rank of their fitness and their spatial 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 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 mutation vector / 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 quasi-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

Algorithm 2 Dynamic ES Niching: A Single Generation

```
for $i = 1...(q + p)$ search points do
    Generate $\lambda$ samples based on the D-set of $i$
end for
Evaluate fitness of the population
Compute the Dynamic Peak Set with Algo. 1
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
```

the entire population is done using the DPI routine [12] (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.

3.2 MPR Analysis

Our research focuses on the ability to identify global as well as local optima, and to converge in these directions through time, with no particular interest in the distribution of the population. Thus, as has been done in earlier studies of GA niching [12], 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 niches 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:

$$MPR = \frac{\sum_{i=1}^{q} \hat{\mathcal{F}}_i}{\sum_{i=1}^{q} \tilde{f}_i} \quad \in [0, 1] \tag{21}$$

i.e., $MPR = 1$ represents a perfect niching process, where the real optima of the objective function were all located and are within the population. Also, given a *maximization* problem, the MPR is defined as the obtained fitness of the niches divided by the real optima. The real optima of the objective function cannot always be obtained analytically, particularly in complex problems. Hence, some optima are computed numerically when necessary.

Although this metric was originally introduced to be analyzed by means of the saturation MPR value, a new perspective was introduced in [16]. That recent study investigated the MPR as a function of time, focusing on the early stages of the run. It was shown experimentally that the time-dependent MPR data fits a theoretical function: *the logistic curve.*

The Logistic Equation

A simple modeling of the human population growth is often described by the following differential equation:

$$\frac{dy}{dt} = cy\left(1 - \frac{y}{a}\right), \tag{22}$$

with the solution

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

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

This equation, known as the *logistic equation*, describes many processes in nature. All those processes share the same pattern of behavior - growth with *acceleration*, followed by *deceleration* and then a *saturation* phase.

In the context of evolutionary niching methods, it was argued in [16] 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.*

3.3 Previous Results

In [16] this MPR time-dependent analysis was applied to two ES-based niching techniques: the Standard-ES Schwefel-approach niching, and the CMA-ES 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, got negative as well as positive values. Negative values mean that the niches 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).

4. EXPERIMENTAL PROCEDURE

In the following section we shall describe an experimental setup for comparing the 5 derandomized variants with respect to the MPR analysis, and present the numerical results. We emphasize again the fact that our set of core mechanisms is composed of two classes:

- Mechanisms with a linear number of strategy parameters in n: DR1, DR2.

- Mechanisms with a quadratic number of strategy parameters in n, which aim to achieve invariance with respect to translation and rotation operations: DR3, CMA, CMA+.

Thus, the CPU time profile differs, respectively, among the different variants. We, in any case, are interested in the convergence behavior subject to the same number of function evaluations and population settings.

4.1 Test Functions

We consider the following multimodal test functions:

- $\mathcal{M}$: 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.

- $\mathcal{A}$: 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 the up-going hyper-spheres . Ackley's function has been widely investigated in the context of *evolutionary computation* [3].

- $\mathcal{L}$: also known as $F2$, as had been originally introduced in [5], 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.

- $\mathcal{R}$: the Rastrigin function [20] 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].

- $\mathcal{G}$: the Griewank function [20] 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].

- $\mathcal{F}$: the function after Fletcher and Powell [3] is a non-separable *non-linear parameter estimation problem*, which has a non-uniform distribution of 2^n minima.

Table 1 summarizes the unconstrained multimodal test functions as well as their initialization intervals.

4.2 Modus Operandi

The 5 niching algorithms are tested on the specified functions for various dimensions[1]. Each test case includes 100 runs per algorithm. 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. The parameter q is set based on a-priori knowledge when available, or arbitrarily otherwise.

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.

As mentioned earlier, setting the parameter p reflects the trade-off between further sampling the search-space, on the expense of exploiting the function evaluations at the existing niches. Here, we set $p = 1$.

A curve fitting routine is applied to each run in order to retrieve the characteristic parameters of its logistic curve. This routine uses the least-squared-error method, and runs an optimization procedure to minimize it.

Table 4: Global minimum reached in 100 runs.

Test-Case	**DR1**	**DR2**	**DR3**	**CMA**	**CMA+**
$\mathcal{A}:\ n=3$	100%	100%	100%	100%	100%
$\mathcal{A}:\ n=10$	90%	91%	90%	92%	95%
$\mathcal{L}:\ n=3$	93%	74%	92%	97%	100%
$\mathcal{L}:\ n=10$	9%	2%	0%	17%	13%
$\mathcal{R}:\ n=3$	20%	19%	13%	16%	48%
$\mathcal{R}:\ n=10$	0%	0%	0%	0%	0%
$\mathcal{G}:\ n=3$	13%	21%	32%	13%	88%
$\mathcal{G}:\ n=10$	8%	16%	4%	16%	2%
$\mathcal{F}:\ n=3$	100%	100%	100%	100%	100%
$\mathcal{F}:\ n=10$	14%	12%	15%	23%	15%
$\mathcal{R}_{RS}:\ n=3$	45%	40%	39%	54%	72%
$\mathcal{R}_{RS}:\ n=10$	0%	0%	0%	0%	0%
$\mathcal{G}_{RS}:\ n=3$	4%	2%	4%	12%	8%
$\mathcal{G}_{RS}:\ n=10$	6%	1%	3%	14%	0%

4.3 Numerical Results

The numerical results are presented at several levels:

Niching Acceleration

Table 2 presents the mean and the standard deviations for the parameter c over the 100 runs, as obtained by the curve fitting routine. There is a clear trend in the given numerical results - in the vast majority of the test cases, the DR2 algorithm has the highest absolute values of c, whereas the CMA+ has the lowest absolute values. This trend corresponds to having the highest niching acceleration and the lowest niching acceleration, respectively. Moreover, the 4 comma strategies have c values in the same order of magnitude, where the CMA usually has the lowest absolute value among them.

MPR Saturation

This scalar value represents, to some degree, the quality of the obtained minima, and thus the final result of the niching process. Table 3 presents the mean and the standard deviation of the saturation MPR values for the different test cases. As can be seen in this table, the CMA-$\left(\overset{+}{,}\right)$ algorithms achieve the highest MPR values, and as far as the niching process is concerned - together they outperform the other methods. However, for the given test cases, there is no clear winner for the MPR value.

Global Minimum

Table 4 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. Generally speaking, the CMA-$\left(\overset{+}{,}\right)$ routines, and in particular the CMA+ strategy, is superior with respect to the other derandomized variants.

One can also observe a strong correlation between tables 3 and 4: routines that obtain a high MPR saturation value, i.e. locate the high-quality peaks, usually perform well globally and locate the global minimum in high percentage of the runs.

[1] Matlab source-code of the 5 routines is available at `http://www.liacs.nl/home/oshir/NichingES/`

Table 1: Test functions to be *minimized*, initialization domains and number of desired peaks. 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	$\mathcal{R}_{RS}(\vec{x}) = 10n + \sum_{i=1}^{n}\left(y_i^2 - 10\cos\left(2\pi y_i\right)\right)$	$[-5,5]^n$	$n+1$
Shifted Rotated Griewank	$\mathcal{G}_{RS}\left(\vec{x}\right) = 1 + \sum_{i=1}^{n}\frac{y_i^2}{4000} - \prod_{i=1}^{n}\cos\left(\frac{y_i}{\sqrt{i}}\right)$	$[0,600]^n$	5

Table 2: The parameter c obtained from the curve fitting: mean and standard deviation over 100 runs.

Test-Case	**DR1**	**DR2**	**DR3**	**CMA**	**CMA+**
$\mathcal{M}:\ n=3$	-0.1067 ± 0.0059	-0.1379 ± 0.0087	-0.1059 ± 0.0096	-0.0694 ± 0.0046	-0.0537 ± 0.0026
$\mathcal{M}:\ n=10$	-0.0592 ± 0.0017	-0.0723 ± 0.0023	-0.0713 ± 0.0031	-0.0402 ± 0.0014	-0.0153 ± 0.0004
$\mathcal{M}:\ n=40$	-0.0272 ± 0.0006	-0.0327 ± 0.0010	-0.0239 ± 0.0007	-0.0129 ± 0.0005	-0.0031 ± 0.0002
$\mathcal{A}:\ n=3$	-0.1530 ± 0.0380	-0.2264 ± 0.0577	-0.1667 ± 0.0056	-0.1353 ± 0.0332	-0.0475 ± 0.0063
$\mathcal{A}:\ n=10$	-0.0631 ± 0.0088	-0.0794 ± 0.0127	-0.0712 ± 0.0110	-0.0547 ± 0.0105	-0.0172 ± 0.0010
$\mathcal{L}:\ n=3$	-0.1637 ± 0.0703	-0.1942 ± 0.1235	-0.1510 ± 0.0637	-0.1479 ± 0.0470	-0.0631 ± 0.0301
$\mathcal{L}:\ n=10$	-0.1503 ± 0.0145	-0.1856 ± 0.0236	-0.1433 ± 0.0572	-0.1466 ± 0.0160	-0.0397 ± 0.0025
$\mathcal{R}:\ n=3$	-0.0218 ± 0.0323	-0.0346 ± 0.0420	-0.0086 ± 0.0119	-0.0298 ± 0.0235	-0.0099 ± 0.0105
$\mathcal{R}:\ n=10$	-0.0462 ± 0.0073	-0.0492 ± 0.0097	-0.0389 ± 0.0172	-0.0222 ± 0.0070	-0.0160 ± 0.0019
$\mathcal{G}:\ n=3$	-0.0121 ± 0.0137	-0.0245 ± 0.0169	-0.0121 ± 0.0032	-0.0234 ± 0.0403	-0.0056 ± 0.0118
$\mathcal{G}:\ n=10$	-0.0312 ± 0.0267	-0.1019 ± 0.0197	-0.0308 ± 0.0301	-0.0227 ± 0.0031	-0.0191 ± 0.0153
$\mathcal{F}:\ n=3$	-0.0219 ± 0.0228	-0.0419 ± 0.0172	-0.0243 ± 0.0238	-0.0227 ± 0.0247	-0.0151 ± 0.0123
$\mathcal{F}:\ n=10$	-0.0540 ± 0.0928	-0.0873 ± 0.1052	-0.0775 ± 0.1230	-0.0438 ± 0.0825	-0.0216 ± 0.0205
$\mathcal{R}_{RS}:\ n=3$	-0.1569 ± 0.0360	-0.2537 ± 0.0529	-0.1779 ± 0.0473	-0.2003 ± 0.0409	-0.0546 ± 0.0083
$\mathcal{R}_{RS}:\ n=10$	-0.0715 ± 0.0261	-0.0948 ± 0.0186	-0.0825 ± 0.0249	-0.0724 ± 0.0274	-0.0204 ± 0.0016
$\mathcal{G}_{RS}:\ n=3$	-0.1081 ± 0.0665	-0.1258 ± 0.0743	-0.1182 ± 0.0638	-0.1133 ± 0.0693	-0.0503 ± 0.0067
$\mathcal{G}_{RS}:\ n=10$	-0.0564 ± 0.0153	-0.0722 ± 0.0146	-0.0850 ± 0.0196	-0.0900 ± 0.0123	-0.0202 ± 0.0041

Table 3: **The saturation MPR value: mean and standard deviation over 100 runs.**

Test-Case	**DR1**	**DR2**	**DR3**	**CMA**	**CMA+**
$\mathcal{M}:\ n = 3$	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
$\mathcal{M}:\ n = 10$	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
$\mathcal{M}:\ n = 40$	0.9967 ± 0.0018	1 ± 0	0.9879 ± 0.0030	1 ± 0	1 ± 0
$\mathcal{A}:\ n = 3$	0.9711 ± 0.0285	0.9662 ± 0.0275	0.9595 ± 0.0301	0.9768 ± 0.0238	0.9924 ± 0.0167
$\mathcal{A}:\ n = 10$	0.9013 ± 0.0239	0.9049 ± 0.0254	0.9013 ± 0.0250	0.9195 ± 0.0232	0.9417 ± 0.0225
$\mathcal{L}:\ n = 3$	0.9625 ± 0.0283	0.9448 ± 0.0383	0.9527 ± 0.0290	0.9621 ± 0.0267	0.9957 ± 0.0059
$\mathcal{L}:\ n = 10$	0.5054 ± 0.1625	0.3791 ± 0.1526	0.1665 ± 0.1287	0.5960 ± 0.1480	0.5619 ± 0.1089
$\mathcal{R}:\ n = 3$	0.2631 ± 0.3140	0.2452 ± 0.0361	0.2334 ± 0.0416	0.1428 ± 0.0458	0.4806 ± 0.1237
$\mathcal{R}:\ n = 10$	0.0515 ± 0.0072	0.0629 ± 0.0068	0.0546 ± 0.0053	0.0567 ± 0.0089	0.0526 ± 0.0054
$\mathcal{G}:\ n = 3$	0.1145 ± 0.1682	0.5257 ± 0.4696	0.3664 ± 0.0502	0.2231 ± 0.2883	0.7609 ± 0.0981
$\mathcal{G}:\ n = 10$	0.0243 ± 0.0421	0.0261 ± 0.0473	0.0664 ± 0.0179	0.0145 ± 0.0171	0.0788 ± 0.0289
$\mathcal{F}:\ n = 3$	0.0022 ± 0.0024	0.0018 ± 0.0018	0.0019 ± 0.0015	0.0028 ± 0.0040	0.0016 ± 0.0004
$\mathcal{F}:\ n = 10$	0.0002 ± 0.0004	0.0003 ± 0.0006	0.0001 ± 0.0003	0.0005 ± 0.0013	0.0003 ± 0.0004
$\mathcal{R}_{RS}:\ n = 3$	0.4088 ± 0.1114	0.4627 ± 0.0669	0.4225 ± 0.1171	0.4692 ± 0.1026	0.5628 ± 0.0981
$\mathcal{R}_{RS}:\ n = 10$	0.0847 ± 0.0153	0.0991 ± 0.0192	0.0781 ± 0.0146	0.1075 ± 0.0166	0.0713 ± 0.0138
$\mathcal{G}_{RS}:\ n = 3$	0.0720 ± 0.0430	0.0779 ± 0.0435	0.0846 ± 0.0475	0.0815 ± 0.0361	0.1076 ± 0.0406
$\mathcal{G}_{RS}:\ n = 10$	0.1336 ± 0.0376	0.1441 ± 0.0365	0.1220 ± 0.0348	0.1610 ± 0.0337	0.0447 ± 0.0126

4.4 The $c - T$ Tradeoff Hypothesis

We would like to numerically assess the hypothesis claiming the existence of a tradeoff between the learning period T and the niching acceleration c [16], with respect to the 5 algorithms under investigation.

We consider two test functions of the suite, one per class: the *separable* $\mathcal{M}$ and the *non-separable* $\mathcal{G}_{RS}$ (the *Shifted Rotated Griewank*). For each we run the algorithms for an increasing dimensionality of $n = 3, 4, ..., 30$, and obtain the MPR parameters for 100 runs - in order to plot c as a function of T.

Figures 1 and 2 present the $c - T$ curves for $\mathcal{M}$ and $\mathcal{G}_{RS}$, respectively. The curves reflect a clear trade-off between c and T over the dimensions for the algorithms for both cases (an exception - the DR3 over $\mathcal{M}$). We consider this a numerical assessment for the hypothesis - the longer the learning period, the lower the niching acceleration.

5. DISCUSSION AND OUTLOOK

We have presented a survey of advanced derandomized Evolution Strategies variants to a suite of theoretical test problems.

Generally speaking, the CMA-$\left(\overset{+}{,}\right)$ routines, and in particular the CMA+ strategy, were found to be superior with respect to the other derandomized variants.

The low niching acceleration of the *plus-strategy* seems to be the key for the successful niching, allowing it to obtain the best location of the global minimum, and to reach the highest MPR saturation values. The *niching acceleration* seems to originate, to our best understanding, in the adaptation profile of the step-size, and apparently the CMA+ mechanism offers a profile which suits niching very well.

In addition, we suggest an explanation for the advantage of a plus strategy for niching. The niching problem can be considered as an optimization task with constraints, i.e., the formation of niches that restricts competing niches and their optimization routine of exploring the search space freely. It has been suggested in previous studies (see, e.g., [10]) that ES self-adaptation in constrained problems will tend to fail with a comma-strategy, and thus a plus-strategy is preferable for such problems. We might link this argumentation to the observation of our numerical results here, and suggest that a plus-strategy is preferable for niching.

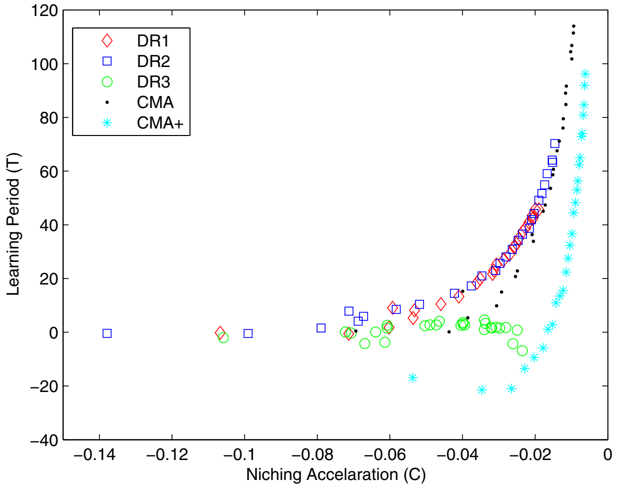

Figure 1: The $c - T$ curve for $\mathcal{M}$: clear trade-off for the different algorithms, except for DR3, which has a flat curve.

Moreover, the hypothesis claiming that there exists a trade-off between the learning period and the niching acceleration has been numerically assessed in this study.

In the future we will recommend the application of the proposed niching variants to multimodal real-world problems, as well as the construction of self-adaptive niche radius routines with the derandomized-ES algorithms which were studied in this paper.

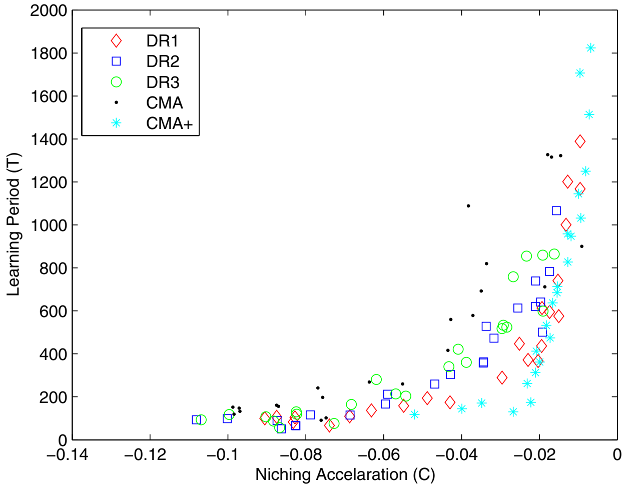

Figure 2: The $c-T$ curve for $\mathcal{G}_{RS}$: clear trade-off for the 5 different algorithms.

Acknowledgments

Ofer Shir would like to thank Daniel Berkovic and Nicola Lipman for their priceless contribution that greatly improved this paper.
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)'.

6. REFERENCES

[1] G. Avigad, A. Moshaiov, and N. Brauner. Concept-based interactive brainstorming in engineering design. *JACIII*, 8(5):454–459, 2004.

[2] T. Bäck. Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In Z. Michalewicz, J. D. Schaffer, H.-P. Schwefel, D. B. Fogel, and H. Kitano, editors, *Proc. First IEEE Conf. Evolutionary Computation (ICEC'94), Orlando FL*, volume 1, pages 57–62, Piscataway, NJ, USA, 1994. IEEE Press.

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

[4] H.-G. Beyer and H.-P. Schwefel. Evolution strategies a comprehensive introduction. *Natural Computing: an international journal*, 1(1):3–52, 2002.

[5] 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*, pages 41–49, Mahwah, NJ, USA, 1987. Lawrence Erlbaum Associates, Inc.

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

[7] 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.

[8] C. Igel, N. Hansen, and S. Roth. The multi-objective variable metric evolution strategy. Technical Report 2005-04, Ruhr-Universität Bochum, 2005.

[9] 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)*, pages 453–460. ACM Press, 2006.

[10] O. Kramer and H.-P. Schwefel. On three new approaches to handle constraints within evolution strategies. *Natural Computing: an international journal*, 5(4):363–385, 2006.

[11] S. Mahfoud. *Niching Methods for Genetic Algorithms*. PhD thesis, University of Illinois at Urbana Champaign, 1995.

[12] 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.

[13] A. Ostermeier, A. Gawelczyk, and N. Hansen. A derandomized approach to self adaptation of evolution strategies. Technical report, 1993.

[14] A. Ostermeier, A. Gawelczyk, and N. Hansen. Step-size adaption based on non-local use of selection information. In *PPSN*, volume 866 of *Lecture Notes in Computer Science*. Springer, 1994.

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

[16] 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, 2005. IEEE Press.

[17] 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*, volume 4193 of *Lecture Notes in Computer Science*, pages 142–151. Springer, 2006.

[18] 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*, volume 3871. Springer, 2006.

[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. Technical report, 2005.

[20] A. Törn and A. Zilinskas. *Global Optimization*, volume 350. Springer, 1987.