Niche Radius Adaptation in the CMA-ES Niching Algorithm

Ofer M. Shir[1] and Thomas Bäck[1,2]

[1] Leiden Institute of Advanced Computer Science
Universiteit Leiden
Niels Bohrweg 1, 2333 CA Leiden
The Netherlands

[2] NuTech Solutions
Martin-Schmeisser-Weg 15
44227 Dortmund
Germany

Abstract. Following the introduction of two *niching methods* within Evolution Strategies (ES), which have been presented recently and have been successfully applied to theoretical high-dimensional test functions, as well as to a real-life high-dimensional physics problem, the purpose of this study is to address the so-called *niche radius problem*.
A new concept of *adaptive individual niche radius*, introduced here for the first time, is applied to the ES Niching with Covariance Matrix Adaptation (CMA) method. The proposed method is described in detail, and then tested on high-dimensional theoretical test functions.
It is shown to be robust and to achieve satisfying results.

1 Introduction

Evolutionary Algorithms (EAs) have the tendency to converge quickly into a single solution [1–3], i.e. all the individuals of the artificial population evolve to become nearly identical. Given a problem with multiple solutions, the traditional EAs will locate a single solution. This is the desired result for many complex tasks, but a problem arises when multimodal domains are considered and multiple optima are required. For instance, consider an optimization problem for a high-dimensional real-world application, which requires the location of highly-fit multiple solutions with high diversity among them - a result which a sequential multiple-restart algorithm doesn't aim for. *Niching methods*, the extension of EAs to multi-modal optimization, address this problem by maintaining the diversity of certain properties within the population - and this way they allow parallel convergence into multiple good solutions. Up to date, *niching methods* have been studied mainly within the field of Genetic Algorithms (GAs). The research in this direction has yielded various successful methods which have been shown to find multiple solutions efficiently [1], but naturally were limited to low-dimensional real-valued problems. Evolution Strategies (ES) are a *canonical*

EA for real-valued 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. The higher the dimensionality of the search space, the more suitable a task becomes for an ES (see, e.g. [3], pp. 149-159). Two ES niching methods have been proposed lately [4, 5]. Upon their successful application to high-dimensional theoretical functions, those methods were successfully applied to a real-world high-dimensional physics problem, namely the *optimization of dynamic molecular alignment by shaped laser pulses* [6]. In that application, the niching technique was shown to be clearly qualitatively inferior with respect to multiple restart runs with a single population, for locating highly-fit unique optima which had not been obtained otherwise.
The ES niching methods, as the majority of the GA niching methods, hold an assumption concerning the fitness landscape, stating that the peaks 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 those niching methods are most likely to fail (for example see Fig. 1, 2). There were several GA-oriented studies which addressed

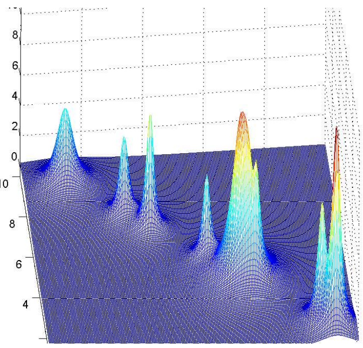

Fig. 1. $\mathcal{S},\ n=2$

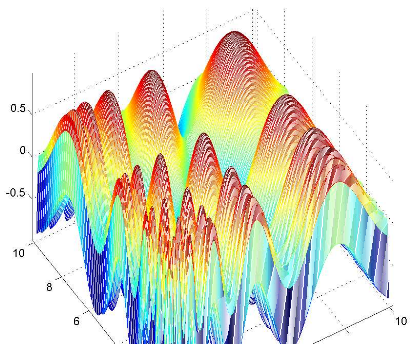

Fig. 2. $\mathcal{V},\ n=2$

this so-called *niche radius problem*, aiming to drop this assumption, such as the cooling-based UEGO [7] or the clustering-based DNC [8]. A more theoretical study of a clustering-based niching can be found in [9]. Moreover, an iterative statistical-based approach was introduced lately [10] for learning an optimal niche radius (without relaxing the fitness landscape assumption).
Our proposed method introduces a new concept to the niche radius problem, inspired by the ES self-adaptation concept - an **adaptive individual niche radius**. The idea is that each individual carries and adapts a niche radius along with its adaptive strategy parameters. This method is an "adaptive extension"

to the *CMA-ES dynamic niching algorithm* [5], as will be explained.
The remainder of the paper is organized as follows: In section 2 we introduce the background for the various components of our proposed algorithm. Section 3 introduces our proposed algorithm. In section 4 the test functions as well as the methodology for the performance study are outlined, where the numerical results are presented and analyzed in section 5. Section 6 provides summary and conclusion.

2 From Fitness Sharing to the CMA-ES Niching Method

2.1 Fitness Sharing

The *fitness sharing* approach [11] was the pioneering GA niching method. Its idea is to consider the fitness as a shared resource and by that to aim to decrease redundancy in the population. Given the similarity metric of the population, which can be genotype or phenotype based, the *sharing function* is given by:

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

where d_{ij} is the distance between individuals i and j, ρ (traditionally noted as σ_{sh}) is the fixed radius of every niche, and α_{sh} is a control parameter, usually set to 1. Using the *sharing function*, the *niche count* is then defined as follows:

$$m_i = \sum_{j=1}^{N} sh(d_{ij}) \tag{2}$$

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

$$f_i^{sh} = \frac{f_i}{m_i} \tag{3}$$

2.2 Dynamic Niche Sharing

The *dynamic niche sharing* method [12], 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*:

$$m_i^{dyn} = \begin{cases} n_j & \text{if individual } i \text{ is within dynamic niche } j \\ m_i & \text{otherwise (non-peak individual)} \end{cases} \tag{4}$$

where n_j is the size of the jth dynamic niche, and m_i is the standard *niche count*, as defined in Eq. 2. The shared fitness is then defined respectively:

$$f_i^{dyn} = \frac{f_i}{m_i^{dyn}} \tag{5}$$

The identification of the dynamic niches can be done in the greedy approach, as proposed in [12] as the Dynamic Peak Identification (DPI) algorithm.

2.3 Dynamic Niching in Evolution Strategies

The ES dynamic niching algorithm [4] was introduced recently as a niching method for the Evolution Strategies framework. The inspiration for this algorithm was given by various niching algorithms from the GA field, and in particular by the *fitness sharing* [11] and its dynamic extension [12], as well as by the *crowding* concept [13]. The basic idea of the algorithm is to *dynamically identify the various fitness-peaks of every generation* that define the niches, *classify all the individuals into those niches*, and apply *a mating restriction scheme which allows competitive mating only within the niches*: in order to prevent *genetic drift*, every niche can produce the same number of offspring, following a *fixed mating resources* concept.
For more details we refer the reader to [14].

2.4 Dynamic Niching with Covariance Matrix Adaptation ES

The dynamic niching with CMA-ES algorithm [5] was the successor of *the ES dynamic niching algorithm*, where the CMA replaces the mutative step-size control. We provide here a short overview of the CMA-ES method, followed by a description of the algorithm.

The $(1, \lambda)$-CMA-ES: A Brief Overview The *covariance matrix adaptation evolution strategy* [15], is a variant of ES that has been successful for treating correlations among object variables. This method tackles the critical element of Evolution Strategies, the adaptation of the mutation parameters. We provide here a short description of the principal elements of the $(1, \lambda)$-CMA-ES.
The fundamental property of this method is the exploitation of information obtained from previous successful mutation operations. Given an initial search point $\boldsymbol{x}^0$, λ offspring are sampled from it by applying the mutation operator. The best search point out of those λ offspring is chosen to become the parent of the next generation.
Explicitly, the action of the *mutation operator* for generating the λ samples of search points in generation $g + 1$ is defined as follows:

$$\boldsymbol{x}^{g+1} \sim \mathcal{N}\left(\boldsymbol{x}_k^{(g)}, {\sigma^{(g)}}^2 \mathbf{C}^{(g)}\right), \qquad k = 1, ..., \lambda \tag{6}$$

where $\mathcal{N}(\boldsymbol{m}, \mathbf{C})$ denotes a normally distributed random vector with mean $\boldsymbol{m}$ and covariance matrix $\mathbf{C}$. The matrix $\mathbf{C}$, the crucial element of this process, is initialized as the *unity matrix* and is learned during the course of evolution, based on cumulative information of successful mutations (the so-called *evolution path*). The global step size, $\sigma^{(g)}$, is based on information from the *principal component analysis* of $\mathbf{C}^{(g)}$ (the so-called *"conjugate" evolution path*). We omit most of the details and refer the reader to Hansen and Ostermeier [15].

Dynamic Niching with CMA The algorithm uses the $(1, \lambda)$-CMA ES as its evolutionary core mechanism. A brief description of the algorithm follows. Given q, the estimated/expected number of peaks, $q + p$ "CMA-sets" are initialized, where a CMA-set is defined as the collection of all the dynamic variables of the CMA 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 $q + p$ CMA-sets, which are associated with $q + p$ search points: q for the peaks and p for the "non-peaks domain". The $(q + 1)^{th}...(q + p)^{th}$ CMA-sets are individuals which are randomly re-generated in every generation as potential candidates for niche formation. Until stopping criteria are met, the following procedure takes place. Each search point samples λ offspring, based on its evolving CMA-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 algorithm, and the peaks become the new search points. Their CMA-sets are inherited from their parents and updated according to the CMA method.

2.5 The Niche Radius Problem

The traditional formula for the niche radius for *phenotypic sharing* in GAs was derived by Deb and Goldberg [16]. By following the trivial analogy and considering the decision parameters as the decoded parameter space of the GA, the same formula was applied to the ES niching methods. It is important to note that this formula depends on q, the expected/desired number of peaks in the solution space:

$$\rho = \frac{r}{\sqrt[n]{q}} \tag{7}$$

where given lower and upper boundary values $x_{k,min}$, $x_{k,max}$ of each coordinate in the decision parameters space, r is defined as $r = \frac{1}{2}\sqrt{\sum_{k=1}^{n}(x_{k,max} - x_{k,min})^2}$. For the complete derivation see, e.g., [6].

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

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

3 The Proposed Algorithm: Niche Radius Adaptation in the CMA-ES Niching Algorithm

Our new algorithm tackles the *niche radius problem*, in particular the assumption regarding the fitness landscape: it introduces the concept of an individual niche radius which adapts during the course of evolution. The idea is to *couple* the niche radius to the global step size σ, whereas the *indirect selection* of the niche radius is applied through the demand for λ individuals per niche. This is implemented through a quasi *dynamic fitness sharing* mechanism.

The *CMA-ES Niching method* is used as outlined earlier (Sec. 2.4), with the following modifications. q is given as an input to the algorithm, but it's now merely a prediction or a demand for the number of solutions, with no effect on the nature of the search. 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:

$$\rho_i^{g+1} = \left(1 - c_i^{g+1}\right) \cdot \rho_{parent}^{g} + c_i^{g+1} \cdot \sigma_{parent}^{g+1} \tag{8}$$

where c_i^g is the individual learning coefficient, which is updated according to the *delta of the step size* σ:

$$c_i^{g+1} = \frac{1}{5} \cdot \left(1 - \exp\left\{\alpha \cdot \Delta\sigma_i^{g+1}\right\}\right) \qquad \Delta\sigma_i^{g+1} = \left|\sigma_{parent}^{g+1} - \sigma_{parent}^{g}\right| \tag{9}$$

This profile is chosen in order to keep the learning coefficient close to $\frac{1}{5}$ for big changes in the global step size, but make it exponentially approach 0 as the global step size vanishes, i.e. convergence is achieved. The parameter α determines the nature of this profile, and it seems to become problem dependent for some landscapes (a discussion concerning this parameter will follow).
The DPI algorithm is run using the **individual niche radii**, for the identification of the peaks and the classification of the population.
Furthermore, introduce:

$$g(x,\lambda) = 1 + \Theta(\lambda - x) \cdot \frac{(\lambda - x)^2}{\lambda} + \Theta(x - \lambda) \cdot (\lambda - x)^2 \tag{10}$$

where $\Theta(y)$ is the *Heaviside step function.* Given a fixed λ, $g(x,\lambda)$ is a parabola with unequal branches, centered at $(x = \lambda,\ g = 1)$. An explanation will follow. Then, by applying the calculation of the *dynamic niche count* m_i^{dyn} (Eq. 4), based on the appropriate radii, we **define** the *niche fitness* of individual i by:

$$f_i^{niche} = \frac{f_i}{g\left(m_i^{dyn}, \lambda\right)} \tag{11}$$

The selection of the next parent in each niche is based on this *niche fitness.*
Eq. 11 enforces the requirement for having a fixed resource of λ individuals per niche, since $g(x,\lambda)$ obtains values greater than 1 for any niche count different than λ. The anti-symmetry of $g(x,\lambda)$ is therefore meant to penalize more the niches which exceeded λ members, in comparison to those with less than λ members. This equation is a variant of the dynamic shared fitness (Eq. 5), and is used now in the context of niche radius adaptation.
A single generation of the method is summarized as Algorithm 1.

4 Test Functions and Experimental Procedure

Table 1 summarizes the unconstrained multimodal test functions [3, 5, 17, 18], as well as their initialization intervals. Some of the functions have a symmetric or

Algorithm 1 $(1,\lambda)$-CMA-ES Dynamic Niching with Adaptive Niche Radius

```
for all i = 1..q + p search points
   Generate λ samples based on the CMA distribution of i
   Update the niche radius ρ_i^{g+1} according to Eq.8
endfor
Evaluate Fitness of the population.
Compute the Dynamic Peak Set of the population using the DPI, based on individual radii
Compute the Dynamic Niche Count (Eq.4) of every individual
for every given peak of the dynamic-peak-set do:
   Compute the Niche Fitness (Eq. 11)
   Set indiv. with best niche fitness as a search point of the next generation
   Inherit the CMA-set and update it respectively
endfor
if N_dps =size of dynamic-peak-set < q
   Generate q − N_dps new search points, reset CMA-sets
endif
Reset the (q + 1)^th...(q + p)^th search points
```

equal distribution of optima ($\mathcal{A}$, $\mathcal{L}$, $\mathcal{B}$, $\mathcal{M}$, $\mathcal{G}$), and some do not ($\mathcal{F}$, $\mathcal{V}$, $\mathcal{S}$). Some of the functions are *non-separable*. The *CMA-ES dynamic niching* algorithm (with a fixed niche radius) was tested on $\{\mathcal{A},\ \mathcal{L},\ \mathcal{F}\}$ [5], whereas it was not applied to the rest of the functions given here. Some of those additional test functions $\{\mathcal{M},\ \mathcal{B},\ \mathcal{G}\}$ are benchmark multimodal functions that will further test the robustness of the algorithm as a niching method, and the others $\{\mathcal{V},\ \mathcal{S}\}$ focus on the *niche radius problem*.

$\mathcal{M}$ is meant to test the stability of a particularly large number of niches: In the interval $[0,1]^n$ this function has 5^n maxima, equally distributed as a *hyper-grid*, with equal function values of 1. $\mathcal{V}$ is a sine function with decreasing frequency (6^n optima in the interval $[0.25,10]^n$). $\mathcal{S}$, suggested in [18], introduces a landscape with dramatically uneven spread of optima. Both $\mathcal{V}$ and $\mathcal{S}$ are not likely to be tackled by a niching method with a fixed niche radius.

The algorithm is tested on the specified functions for various dimensions. Each test case includes 100 runs. All runs are performed with a core mechanism of a $(1,10)$-strategy per niche and initial points are sampled uniformly within the initialization intervals. Initial step sizes, as well as initial niche radii, are set to $\frac{1}{6}$ of the intervals. The parameter q is set based on a-priori knowledge when available, or arbitrarily otherwise; p is set to 1. The default value of α is -10, but it becomes problem dependent for some cases, and has to be tuned. Each run is stopped after 10^5 generations ($(q+1)\cdot 10^6$ evaluations).

We consider three measures as the performance criteria: the saturation M.P.R. (*maximum peak ratio*; see, e.g., [5]), the global optimum location percentage, and the number of optima found (with respect to the desired value, q).

5 Numerical Results

The results of the simulations are summarized in table 2. As reflected by those results, our method performs in a satisfying manner. A comparison shows that the performance of the new niching method is not harmed by the introduction of the niche radius adaptation mechanism with respect to the same multimodal

Table 1. Test functions to be *minimized* and initialization domains

Name	Function	Init
Ackley	$\mathcal{A}(\boldsymbol{x}) = -c_1 \cdot \exp\left(-c_2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)$ $-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(c_3 x_i)\right) + c_1 + e$	$[-10, 10]^n$
$\mathcal{L}$	$\mathcal{L}(\boldsymbol{x}) = -\prod_{i=1}^{n} \sin^k\left(l_1 \pi x_i + l_2\right) \cdot \exp\left(-l_3\left(\frac{x_i - l_4}{l_5}\right)^2\right)$	$[0, 1]^n$
Fletcher-Powell	$\mathcal{F}(\boldsymbol{x}) = \sum_{i=1}^{n}\left(A_i - B_i\right)^2$ $A_i = \sum_{j=1}^{n}\left(a_{ij} \cdot \sin(\alpha_j) + b_{ij} \cdot \cos(\alpha_j)\right)$ $B_i = \sum_{j=1}^{n}\left(a_{ij} \cdot \sin(x_j) + b_{ij} \cdot \cos(x_j)\right)$	$[-\pi, \pi]^n$
$\mathcal{M}$	$\mathcal{M}\left(\boldsymbol{x}\right) = -\frac{1}{n}\sum_{i=1}^{n}\sin^{\alpha}\left(5\pi x_i\right)$	$[0, 1]^n$
Bohachevsky	$\mathcal{B}\left(\boldsymbol{x}\right) = \sum_{i=1}^{n-1}(x_i^2 + 2x_{i+1}^2$ $-0.3 \cdot \cos(3\pi x_i) - 0.4 \cdot \cos(4\pi x_{i+1}) + 0.7)$	$[-10, 10]^n$
Grienwank	$\mathcal{G}\left(\boldsymbol{x}\right) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-10, 10]^n$
Shekel	$\mathcal{S}\left(\boldsymbol{x}\right) = -\sum_{i=1}^{10}\frac{1}{k_i(\boldsymbol{x}-a_i)(\boldsymbol{x}-a_i)^T + c_i}$	$[0, 10]^n$
Vincent	$\mathcal{V}\left(\boldsymbol{x}\right) = -\frac{1}{n}\sum_{i=1}^{n}\sin\left(10 \cdot \log(x_i)\right)$	$[0.25, 10]^n$

test functions reported in [5], except for the *Ackley* function in high dimensions. The latter seems to become deceptive for the adaptation mechanism as the dimensions go up: it requires the tuning of the parameter α, but no longer obtains satisfying results for $n > 15$. This occurs since the global minimum has a far stronger basin of attraction in comparison to the local minima, and many niches are formed in this basin. However, our confidence in the method is further reassured by the results on the functions $\{\mathcal{M},\ \mathcal{B},\ \mathcal{G}\}$ which are quite satisfying. Concerning the landscapes with the "deceptive" distribution of optima, i.e. $\mathcal{V}$ and $\mathcal{S}$, our method performed well, and managed to tackle the niche radius problem successfully. The tuning of α is also required for $\mathcal{S}$.
A visualizations of the runs on $\mathcal{V}$ and $\mathcal{S}$ for $n = 1$ are given as Fig. 3 and Fig. 4.

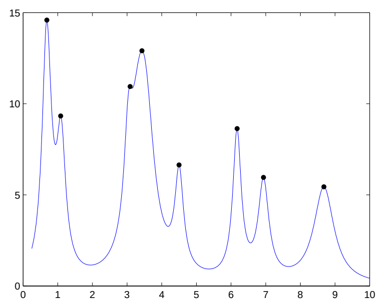

Fig. 3. $\mathcal{S}$ $(n = 1)$: final population

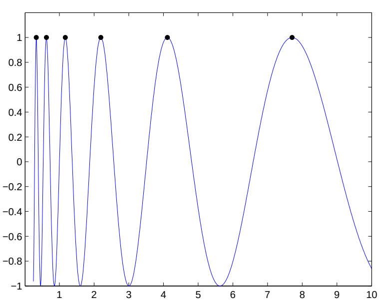

Fig. 4. $\mathcal{V}$ $(n = 1)$: final population

Table 2. Performance Results

Function	M.P.R.	Global	Optima/q	Function	M.P.R.	Global	Optima/q
$\mathcal{A}:\ n=3$	1	100%	7/7	$\mathcal{M}:\ n=3$	1	100%	100/100
$\mathcal{A}:\ n=20$	0.6984	59%	22.6/41	$\mathcal{M}:\ n=10$	0.9981	100%	99.1/100
$\mathcal{A}:\ n=40$	0.3186	43%	20.8/81	$\mathcal{M}:\ n=40$	0.7752	100%	87.2/100
$\mathcal{L}:\ n=4$	0.9832	100%	4.4/5	$\mathcal{B}:\ n=3$	0.9726	100%	3.96/5
$\mathcal{L}:\ n=10$	0.7288	47%	3.4/11	$\mathcal{B}:\ n=10$	0.5698	82%	2.21/5
$\mathcal{F}:\ n=2$	1	100%	4/4	$\mathcal{B}:\ n=20$	0.1655	61%	1.21/5
$\mathcal{F}:\ n=4$	0.881	100%	3.0/4	$\mathcal{G}:\ n=2$	0.7288	100%	3.96/5
$\mathcal{F}:\ n=10$	0.783	67%	2.3/4	$\mathcal{G}:\ n=10$	0.398	53%	2.2/5
$\mathcal{V}:\ n=1$	0.8385	100%	5.05/6	$\mathcal{S}:\ n=1$	0.9676	100%	7.833/8
$\mathcal{V}:\ n=2$	0.8060	100%	17.86/36	$\mathcal{S}:\ n=2$	0.8060	100%	6.33/8
$\mathcal{V}:\ n=5$	0.9714	100%	39.16/50	$\mathcal{S}:\ n=5$	0.7311	91%	4.37/8
$\mathcal{V}:\ n=10$	0.9649	100%	36.9/50	$\mathcal{S}:\ n=10$	0.7288	79%	3.41/8

6 Summary and Conclusion

We have proposed a new niching algorithm, with a niche-radius adaptation mechanism, for Evolution Strategies. In particular, this method relies on the CMA-ES algorithm, and couples the individual niche radius to the individual step size. The method is tested on a set of highly multimodal theoretical functions for various dimensions. It is shown to perform in a satisfying manner in the location of the desired optima of functions which were tested in the past on the predecessor of this method, using a fixed niche radius. The *Ackley* function in high dimensions seems to be deceptive for this method. More importantly, the *niche radius problem* is tackled successfully, as demonstrated on functions with unevenly spread optima. In these cases the performance was satisfying as well.
The function of the learning coefficients has to be tuned (through the parameter α) in some cases. Although this is an undesired situation, i.e., the adaptation mechanism is problem dependent, this method makes it possible to locate all desired optima on landscapes which could not be handled by the old methods of fixed niche radii, or would require the tuning of q parameters.

Acknowledgments

The authors would like to thank Michael Emmerich for the fruitful discussions. 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. Mahfoud, S.: Niching Methods for Genetic Algorithms. PhD thesis, University of Illinois at Urbana Champaign (1995)

2. Bäck, T.: Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In Michalewicz, Z., Schaffer, J.D., Schwefel, H.P., Fogel, D.B., Kitano, H., eds.: Proc. First IEEE Conf. Evolutionary Computation (ICEC'94), Orlando FL. Volume 1., Piscataway, NJ, USA, IEEE Press (1994)
3. Bäck, T.: Evolutionary algorithms in theory and practice. Oxford University Press, New York, NY, USA (1996)
4. Shir, O.M., Bäck, T.: Niching in evolution strategies. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2005, New York, NY, USA, ACM Press (2005)
5. Shir, O.M., Bäck, T.: 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. Shir, O.M., Siedschlag, C., Bäck, T., Vrakking, M.J.: Niching in evolution strategies and its application to laser pulse shaping. In: Lecture Notes in Computer Science. Volume 3871., Springer (2006)
7. Jelasity, M.: Uego, an abstract niching technique for global optimization. In: Parallel Problem Solving from Nature - PPSN V. Volume 1498., Amsterdam, Springer (1998)
8. Gan, J., Warwick, K.: Dynamic niche clustering: A fuzzy variable radius niching technique for multimodal optimisation in GAs. In: Proceedings of the 2001 Congress on Evolutionary Computation CEC2001, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, IEEE Press (2001)
9. Preuss, M., Schönemann, L., Emmerich, M.: Counteracting genetic drift and disruptive recombination in (μ, λ)-ea on multimodal fitness landscapes. In: GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation, New York, NY, USA, ACM Press (2005)
10. Cioppa, A.D., Stefano, C.D., Marcelli, A.: On the role of population size and niche radius in fitness sharing. IEEE Trans. Evolutionary Computation **8** (2004)
11. Goldberg, D.E., Richardson, J.: 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)
12. Miller, B., Shaw, M.: 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. Jong, K.A.D.: An analysis of the behavior of a class of genetic adaptive systems. PhD thesis (1975)
14. Shir, O.M., Bäck, T.: Niching in evolution strategies. Technical report (2005)
15. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation **9** (2001)
16. Deb, K., Goldberg, D.E.: An investigation of niche and species formation in genetic function optimization. In: Proceedings of the third international conference on Genetic algorithms, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1989)
17. Hansen, N., Kern, S.: Evaluating the cma evolution strategy on multimodal test functions. In: Parallel Problem Solving from Nature - PPSN V. Volume 1498., Amsterdam, Springer (1998)
18. Törn, A., Zilinskas, A.: Global Optimization. Volume 350. Springer (1987)