Niche Radius Adaptation in the CMA-ES Niching Algorithm

Ofer M. Shir and Thomas Bäck*

Natural Computing Group, Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands

Motivation

Niching methods are the extension of Evolutionary Algorithms (EAs) to multi-modal optimization: they allow parallel convergence into multiple good solutions by maintaining the diversity of certain properties within the population. The majority of the EAs Niching methods holds 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. This is the so-called *niche radius problem*.

Niching Background: Fundamental Concepts

The *fitness sharing* approach considers the fitness as a shared resource and by that aims to decrease redundancy in the population. Given d_{ij}, the distance between individuals i and j, ρ (traditionally noted as σ_{sh}), the radius of every niche, and α_{sh}, a control parameter usually set to 1 - the *sharing function* is:

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

Based on this *sharing function*, the *niche count* is then defined as following:

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

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

$$f_i^{sh} = \frac{f_i}{m_i} \qquad (3)$$

The *dynamic niche sharing* method recognizes the q peaks of the forming niches and classifies the individuals accordingly. Introduce the *dynamic niche count*:

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

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:

$$f_i^{dyn} = \frac{f_i}{m_i^{dyn}} \qquad (5)$$

The identification of the niches can be done with the Dynamic Peak Identification (DPI) algorithm [2].

Dynamic Niching with Covariance Matrix Adaptation Evolution Strategy

The *dynamic niching with CMA-ES* algorithm [2] is a niching method which uses the Covariance Matrix Adaptation Evolution Strategy [1] as its core evolutionary mechanism. The aim of this approach is to find multiple local optima simultaneously, within one run of the ES. 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.

The Niche Radius Problem

The traditional formula for the niche radius for *phenotypic sharing* in GAs and for ES niching is given by $\rho = \frac{r}{\sqrt[n]{q}}$, 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 $r = \frac{1}{2}\sqrt{\sum_{k=1}^{n}(x_{k,max} - x_{k,min})^2}$.

Hence, by applying this niche radius approach, two assumptions used to be 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.

Niche Radius Adaptation in the CMA-ES Niching Algorithm

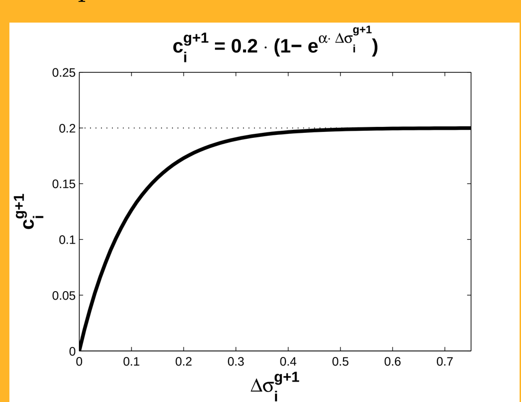
Our new algorithm tackles the *niche radius problem*, in particular the assumption regarding the fitness landscape: it introduces the concept of an individual niche radius which adapts during the course of evolution. The idea is to *couple* the niche radius to the global step size σ, whereas the *indirect 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.

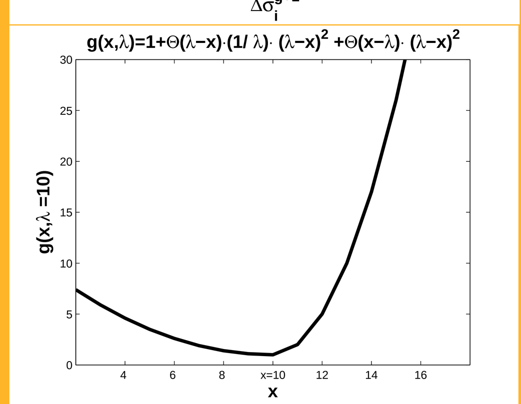
The *CMA-ES Niching method* is used as outlined earlier, with the following modifications. q is given as an input to the algorithm, but it's now merely a prediction or a demand for the number of solutions, with no effect on the nature of the search. A niche radius is initialized for each individual in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

$$\rho_i^{g+1} = \left(1 - c_i^{g+1}\right)\cdot\rho_{parent}^{g} + c_i^{g+1}\cdot\sigma_{parent}^{g+1} \qquad (6)$$

c_i^g is the individual learning coefficient, which is updated according to the *delta of the step size* σ $\left(\Delta\sigma_i^{g+1} = \left|\sigma_{parent}^{g+1} - \sigma_{parent}^{g}\right|\right)$:

$$c_i^{g+1} = \frac{1}{5}\cdot\left(1 - \exp\left\{\alpha\cdot\Delta\sigma_i^{g+1}\right\}\right) \qquad (7)$$





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

$$g(x,\lambda) = 1 + \Theta(\lambda - x)\cdot\frac{(\lambda - x)^2}{\lambda} + \Theta(x - \lambda)\cdot(\lambda - x)^2 \qquad (8)$$

where $\Theta(y)$ is the *Heaviside step function*. Given a fixed λ, $g(x,\lambda)$ is a parabola with unequal branches, centered at $(x = \lambda,\ g = 1)$; see plot.

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:

$$f_i^{niche} = \frac{f_i}{g\left(m_i^{dyn}, \lambda\right)} \qquad (9)$$

The selection of the next parent in each niche is based on this *niche fitness*.
A single generation of the method is summarized as Algorithm 1.

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. 6
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. 9 )
   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
```

Numerical Results

Table 1 summarizes the unconstrained multimodal test functions. 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., [2]), the global optimum location percentage, and the number of optima found (with respect to the desired value, q). The results of the simulations are summarized in table 2.

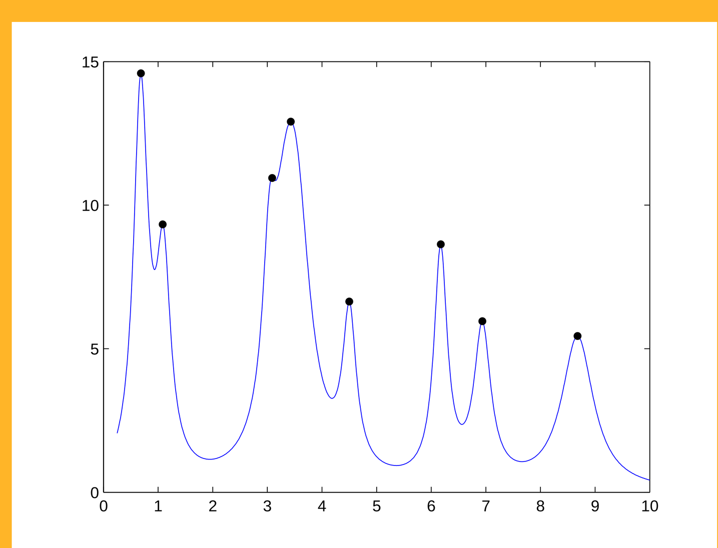
Name	Function	Init
Ackley	$\mathcal{A}(\vec{x}) = -c_1\cdot\exp\left(-c_2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(c_3 x_i)\right) + c_1 + e$	$[-10,10]^n$
$\mathcal{L}$	$\mathcal{L}(\vec{x}) = -\prod_{i=1}^n \sin^k(l_1\pi x_i + l_2)\cdot\exp\left(-l_3\left(\frac{x_i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

Test Functions

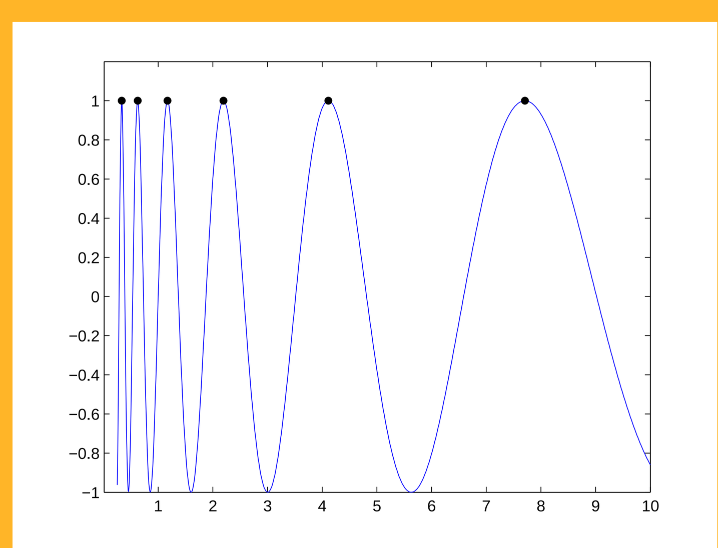
Function	M.P.R.	Global	Optima/q	Function	M.P.R.	Global	Optima/q
$\mathcal{A}:\ n=3$	1	100%	7/7	$\mathcal{M}:\ n=3$	1	100%	100/100
$\mathcal{A}:\ n=20$	0.6984	59%	22.6/41	$\mathcal{M}:\ n=10$	0.9981	100%	99.1/100
$\mathcal{A}:\ n=40$	0.3186	43%	20.8/81	$\mathcal{M}:\ n=40$	0.7752	100%	87.2/100
$\mathcal{L}:\ n=4$	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

Numerical Results

As reflected by those results, our method performs in a satisfying manner. 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 test functions reported in [2], except for the *Ackley* function in high dimensions. Our confidence is further reassured by the results on the functions $\{\mathcal{V},\ \mathcal{S}\}$ with the unevenly spread optima, which are satisfying (plots are given below).



$\mathcal{S}$ $(n=1)$: Final Population



$\mathcal{V}$ $(n=1)$: Final Population

Summary and Conclusion

The proposed method 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. More importantly, the *niche radius problem* is tackled successfully, as demonstrated on functions with unevenly spread optima. 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

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

References

[1] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2), 2001.

[2] Ofer M. Shir and Thomas 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.

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