Niching in Evolution Strategies * †

Ofer M. Shir
Leiden Institute of Advanced Computer Science
Universiteit Leiden
Niels Bohrweg 1, 2333 CA Leiden
The Netherlands
oshir@liacs.nl

Thomas Bäck‡
Leiden Institute of Advanced Computer Science
Universiteit Leiden
Niels Bohrweg 1, 2333 CA Leiden
The Netherlands
baeck@liacs.nl

ABSTRACT

EAs have the tendency to converge quickly into a single solution. *Niching methods*, the extension of EAs to address this issue, have been investigated up to date mainly within the field of Genetic Algorithms (GAs). In our study we investigate the basis for *niching methods* within Evolution Strategies (ES), and propose the first ES niching method. Results show that this method can reliably find and maintain multiple niches even for high-dimensional problems.

Categories and Subject Descriptors: I.2.8 [Computing Methodologies]: Problem Solving, Control Methods, and Search.

General Terms: Algorithms, Performance.

Keywords: Evolution Strategies, Niching Methods, Multiple Optima Search.

1. INTRODUCTION

It has been shown that traditional GAs lose their diversity and converge into a single solution [2]. Likewise, the standard ES is exposed to several strong effects which interrupt the formation and maintenance of multiple solutions and push the evolution process towards a rapid convergence into a single solution. In our research we have extended the ES by a dynamic niching approach to overcome these limitations.

*A full version of this paper is available as a *Technical Report*: TR 2005-01, LIACS, Leiden University, 2005. Available at `http://www.liacs.nl/~oshir`.

†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)'.

‡Nutech Solutions,
Martin-Schmeisser-Weg 15,
44227 Dortmund, Germany.

GECCO'05, June 25–29, 2005, Washington, DC, USA.
ACM 1-59593-010-8/05/0006.

Algorithm 1 Greedy Dynamic Peak Identification (DPI)

```
input:    Pop - array of population members
          N - population size
          q - number of peaks to identify
          ρ - niche radius.
Sort Pop in decreasing fitness order
i := 1
NumPeaks := 0
DPS := ∅ (Dynamic Peak Set)
loop until NumPeaks = q or i = N + 1
   if Pop[i] is not within ρ of peak in DPS
      DPS := DPS ∪ {Pop[i]}
      NumPeaks := NumPeaks + 1
   endif
   i := i + 1
endloop
output:  Dynamic Peak Set
```

2. ES DYNAMIC NICHING

A single generation loop of our proposed algorithm can be described as follows. The *mutation operator* is applied, using a single step-size per individual (self-adaptation is done in the traditional way [1]). Fitness is then evaluated. The various fitness-peaks are identified dynamically using the *dynamic peak identification* (DPI) algorithm [3], with the *euclidean distance* in the decision parameters space as a distance metric (the method is given as algorithm 1). Using an estimated so-called niche radius ρ, all the individuals are classified into those peaks and populate those niches. At this point *a mating restriction scheme* is applied, *which allows competitive mating only within the niches*: every niche can produce a defined number of offspring, following a *fixed mating resources* concept. In particular, a uniform distribution of the resources to q niches is considered: $\tilde{\mu} = \frac{\mu}{q}$ $\tilde{\lambda} = \frac{\lambda}{q}$, meaning that each niche has $\tilde{\mu}$ parents and produces $\tilde{\lambda}$ offspring in every generation. The $\tilde{\lambda}$ individuals are produced within every niche in the following manner - the first parent is chosen via *tournament selection*, where the second parent is the best individual in the niche which is different than the first parent (this is known as the *line breeding mechanism*). Given those $\tilde{\lambda}$ pairs of parents, the *recombination operator* is applied: *intermediate recombination* for the strategy parameters and *discrete recombination* for the decision parameters. The $\tilde{\mu}$ parents of the next generation are selected as follows: the best η of the $\tilde{\lambda}$ offspring along with the best $\delta = \tilde{\mu} - \eta$ individuals of the current niche. At this point, additional $\omega = \tilde{\mu}$ random individuals are newly added to the whole population, and they will take part in

Algorithm 2 ES Dynamic Niching: Generation Loop

```
Apply Mutation on the population
Evaluate fitness of population and Sort
Compute the Dynamic Peak Set using the DPI (Algo-1)
for every niche i = 1..q produce the next generation:
   Generate λ̃ offspring as follows:
      Choose 1st parent via Tour-Selec. of the niche
      Choose the best indiv. of that niche as the 2nd parent
      Apply standard recombination
   Select the best η out of the λ̃ offspring and the best
   μ̃−η indiv. of the current niche to form the next generation
endfor
Generate additional ω = μ̃ random indiv.,
Join all q niches, to yield the new population
```

the next round of the *dynamic peak identification* algorithm. The algorithm holds two assumptions: q, the expected/desired number of peaks, is given or can be estimated, and all peaks are at least in distance 2ρ from each other, where ρ is the fixed radius of every niche. The formula for determining the value of the niche radius depends on q, the number of peaks of the target function:

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

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 follows:

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

Our algorithm is summarized as algorithm 2.

3. EXPERIMENTAL RESULTS

3.1 The Test Functions

1. Himmelblau's function (*minimization*; $x_1, x_2 \in [-6, 6]$):

$$\mathcal{H}(x_1, x_2) = (x_1^2 + x_2 - h_1)^2 + (x_1 + x_2^2 - h_2)^2$$

2. $\mathcal{L}$ (*maximization*; $\vec{x} \in [0, 1]^n$):

$$\mathcal{L}(\vec{x}) = \prod_{i=1}^{n} \sin^k(l_1\pi x_i + l_2) \cdot \exp\left(-l_3\left(\frac{x_i - l_4}{l_5}\right)^2\right)$$

3. Ackley's function (*minimization*; $\vec{x} \in [-10, 10]^n$)

$$\mathcal{A}(\vec{x}) = -c_1 \cdot \exp\left(-c_2\sqrt{\tfrac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \\ - \exp\left(\tfrac{1}{n}\sum_{i=1}^{n} cos(c_3 x_i)\right) + c_1 + e$$

4. The function after Fletcher and Powell:

$$\mathcal{F}(\vec{x}) = \sum_{i=1}^{n}(A_i - B_i)^2 \\ A_i = \sum_{j=1}^{n}(a_{ij} \cdot sin(\alpha_j) + b_{ij} \cdot cos(\alpha_j)) \\ B_i = \sum_{j=1}^{n}(a_{ij} \cdot sin(x_j) + b_{ij} \cdot cos(x_j))$$

where $\mathbf{A} = (a_{ij})$, $\mathbf{B} = (b_{ij})$, and $\vec{\alpha} = (\alpha_j)$ have random elements:

$$a_{ij},\, b_{ij} \in [-100, 100]\,; \quad \vec{\alpha} \in [-\pi, \pi]^n$$

We consider minimization with $\vec{x} \in [-\pi, \pi]^n$.

Table 1: Performance Results

Function	M.P.R	Global	Optima/q
$\mathcal{H}$	1	100%	4/4
$\mathcal{L}:\ n=1$	1	100%	5/5
$\mathcal{L}:\ n=2$	1	100%	5/5
$\mathcal{L}:\ n=3$	1	100%	7/7
$\mathcal{L}:\ n=4$	0.9974	100%	5/5
$\mathcal{L}:\ n=10$	0.8612	80%	7.2/11
$\mathcal{A}:\ n=2$	1	100%	5/5
$\mathcal{A}:\ n=3$	1	100%	7/7
$\mathcal{A}:\ n=20$	0.9999	100%	41/41
$\mathcal{A}:\ n=30$	0.9681	100%	61/61
$\mathcal{A}:\ n=40$	0.9940	100%	81/81
$\mathcal{F}:\ n=2$	1	100%	4/4
$\mathcal{F}:\ n=4$	0.9321	100%	3.4/4
$\mathcal{F}:\ n=10$	0.9141	70%	2.8/4

3.2 Performance Criteria

We consider the *maximum peak ratio statistic* as our niching performance criterion, which has also been in use in GA niching methods [3]. Given the fitness of the optima in the final population $\left\{\tilde{f}_i\right\}_{i=1}^{q}$, and the actual optima of the objective function $\left\{\hat{\mathcal{F}}_i\right\}_{i=1}^{q}$, the *maximum peak ratio* is defined for a *maximization problem* as follows:

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

Also, given a minimization problem, the MPR is defined as the actual optima divided by the obtained optima.

3.3 Experimental Results

The results (table 1) refer to an average over multiple runs on each test function. All simulations were run up to an upper bound of $10,000$ generations. Three measures are introduced in the table for each test case (those are mean values): the MPR, the global optimum location percentage, and the number of optima found (with respect to q).

4. CONCLUSION

The experimental results show clearly that our method has achieved its goal of locating multiple solutions for the given optimization problems. The algorithm performed well on all test functions, where the excellent results for the high-dimensional Ackley test-cases should be emphasized.

5. REFERENCES

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

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

[3] B. L. Miller and M. J. 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.