Improving population-based algorithms used in Global Optimization with Fitness Deterioration techniques

Robert Schaefer, Adrian Wolny

Department of Computer Science
AGH University of Science and Technology

September 22, 2011 / XIII National Conference on Evolutionary Computation and Global Optimization

Outline

It is impossible for any optimization algorithm to outperform random walks on all possible problems

. . . a conclusion from No Free Lunch Theorem

Statement of a Problem

- Optimization of single criterion problems Finding multiple robust solutions of a multimodal, real-valued function $f : \mathbb{X} \to Y$ with $Y \subseteq \mathbb{R}$
- Population-based metaheuristics, specifically Evolutionary Algorithms
- Evolutionary algorithms have the tendency to lose diversity within their population of feasible solutions and to converge into a single solution.
- Problem: Location and maintenance of multiple solutions. Common solution: incorporating niching methods like crowding or fitness sharing
- Different approach: *Sequential Niching* with *Fitness Deterioration*
- Related Work: *Evolutionary Search with Soft Selection with Deterioration of Objective Function (ESSS-DOF)* [3] by Prof. A. Obuchowicz

Definition

Sequential niching - the process of iterative degradation of fitness landscape in the areas occupied by clusters of individuals which are assumed to agglomerate inside basins of attraction.

while !$terminationCriterion()$ **do**
  $execute(evolutionaryAlgorithm)$
  $clusters = cluster(population)$
  $detFitness = performCrunching(clusters, currentFitness)$
  $updateFitness(detFitness)$
end while

Clustering

- An assumption: clusters lie inside basins of attraction. If so the distribution of individuals inside the cluster provides useful information about the about its size and shape
- Density based clustering: *Ordering Points To Identify the Clustering Structure* (OPTICS) [2]

Definition

Crunching function. For a given cluster of individuals crunching function is a real-valued non-negative function K which is created by estimating the density function of individuals in the cluster and then by adapting it in order to approximate the basin of attraction occupied by the cluster.

The *OPTICS ordering* of the population returned by the EA allows us to extract clusters with higher densities efficiently ($O(n)$) and to create more accurate crunching functions.

Clustering

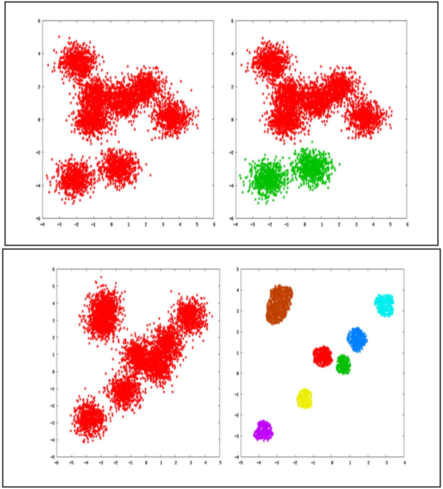


Figure: The result of extracting clusters with different densities from a random data set.

Properties of a good deterioration algorithm:

- It has to be cheap
- It has to discourage the population from visiting the neighborhood of solutions found in previous iterations
- It has to be easy to extend for subsequent solutions

Fitness Deterioration

For each cluster generate a multi-dimensional Gaussian function (crunching function):

$$g(x) = -F_k(x_{max})exp(\frac{-1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)) \quad (1)$$

where F_k is a fitness function in kth iteration of the algorithm, Σ is an **unbiased sample covariance matrix** estimated from the cluster population:

$$\Sigma = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\mu)(x_i-\mu)^T \quad (2)$$

Fitness Deterioration

Fitness function in $k+1$-th iteration is of the form:

$$F_{k+1}(x) = F_k(x) + \sum_{i=1}^{M} \alpha_i(x) g_i(x) \tag{3}$$

where the α-coefficients are given by the following equations:

$$\alpha_1 + \ldots + \alpha_M = 1 \tag{4}$$

$$\alpha_i = \xi\left(\frac{1}{r_i}\right) \tag{5}$$

To reduce the cost of α-coefficeints computation we may substitute $\alpha_1 = \ldots = \alpha_M = 1$ which also yields very good results

Crunching Function Adjustment

- sample covariance matrix estimator is extremely sensitive to outliers
- we perform eigenvalue decomposition which gives use the d orthogonal directions (eigenvectors)
- for each eigenvector v_i we generate two points (so called *outliers*) $p_{i1} = \overline{m} + \sqrt{\lambda_i}\overline{v_i}$ and $p_{i2} = \overline{m} - \sqrt{\lambda_i}\overline{v_i}$
- then we add $2d$ generated outliers to the initial population which constitute a cluster, and compute new covariance matrix

Test functions and results

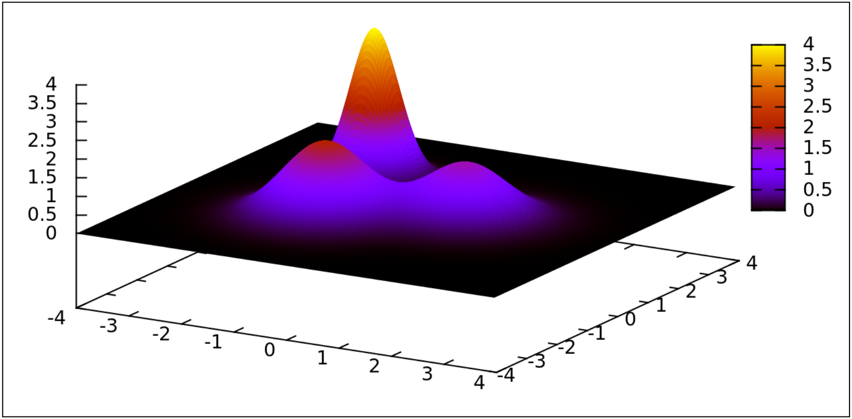
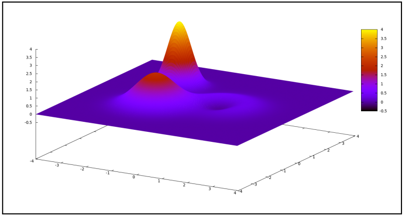
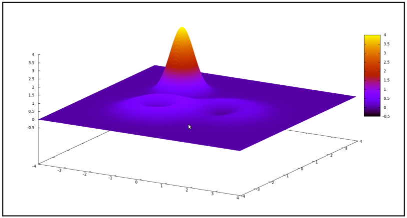
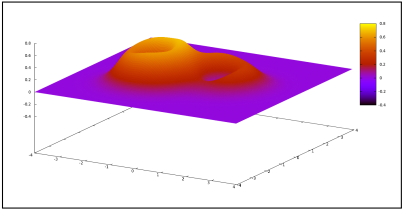


Figure: Test function:
$f(X) = 2e^{-((x+1.1)^2+(y+1.1)^2)} + 1.5e^{-((x-1)^2+y^2)} + 4e^{-3((x+1.5)^2+(y-1.5)^2)}$,
where $X \in \mathbb{R}^2$

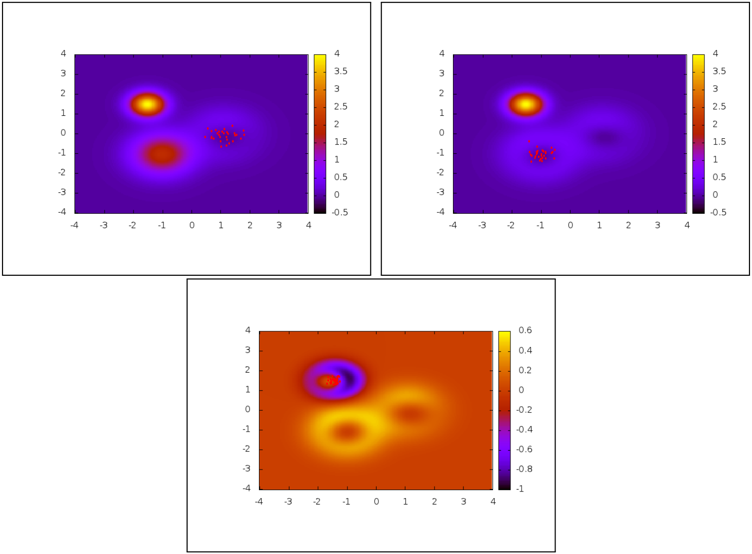
Test functions and results







Test functions and results



Summary

- *Sequential niching* algorithm which may be used as an effective metaheuristic for solving multimodal optimization problems
- clustering algorithms may be used as a reasonable stop criterion for real-valued evolutionary algorithms
- The assumption that distribution of individuals inside a cluster provides information about its shape oversimplifies the real nature of the evolutionary algorithms

Implementation in Java:
https://github.com/wolny/Fitness-Deterioration - lightweight framework which implements all of the algorithms and ideas presented in this work and which can be used for further research

Bibliography

- *Global Optimization Algorithms - Theory and Application* by Thomas Weise
- *OPTICS: Ordering Points To Identify the Clustering Structure* by Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, Jrg Sander
- *Evolutionary search with soft selection* by A. Obuchowicz
- *Foundations of global genetic optimization* by Robert Schaefer, Henryk Telega
- *No Free Lunch Theorems for Optimization* by David Wolpert, William Macready
- *Niching Methods for Genetic Algorithms* by Samir W. Mahfound
- *Hierarchical Genetic Strategy with real number encoding* by Robert Schaefer and Joanna Kolodziej