A Simple Line Search Operator for Ridged Landscapes

Andrea Soltoggio
School of Computer Science
University of Birmingham
Birmingham, B15 2TT, UK
A.Soltoggio@cs.bham.ac.uk

ABSTRACT

This paper describes a new simple operator for Evolutionary Algorithms (EA) to climb ridged landscapes.

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

General Terms: Algorithms, Performance.

Keywords: Directional mutation, Biased mutation, Evolution Strategies, Optimization.

1. INTRODUCTION

The general idea that EAs adapt well to several optimisation problems is given by the fact that little or no modification in the algorithm is needed when applying it to a different problem and performance remain robust. However, if EAs are robust when dealing with many local minima, they are not when the fitness landscape presents ridges. With this respect, Salomon [2] showed that algorithms that are effective on certain problems perform poorly or fail when applying coordinate rotation to the search points. Ridges that in most test functions run along the axes will diverge from them after a coordinate rotation.

Current methods to tackle the problem include line-search approaches, like the simplex method, hybrid or memetic EAs with local gradient search or heuristics. A general method is based on adaptive variations and covariance matrices that evolve a set of rotations to realign the ridges with the axes.

In this paper, we discuss the use of a particular line-search method usually referred to as *directed* or *directional mutation* based on the landscape topology. The basic idea is that information retrieved from a set of search points can be used to identify a promising climbing direction, which is consequently applied as directional mutation. The proposed algorithm simply computes a climbing direction comparing the best individual with all the others.

2. GLOBAL DIRECTIONAL MUTATION

The operator called *global directional mutation* was originally introduced in a GA for a real world application to tune control systems [3].

Here we use the following implementation. For a population of μ parents and λ offspring ordered by fitness, in a generic ES, the vector gdm is computed as

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.
ACM 1-59593-186-4/06/0007.

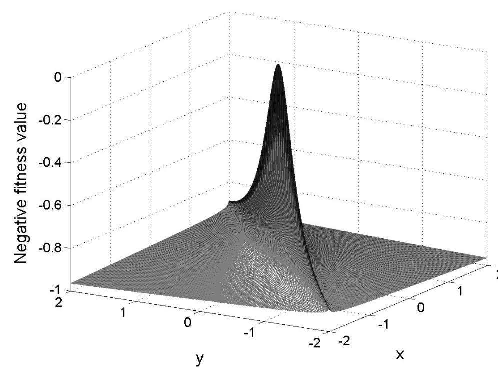

Figure 1: Ridged landscape: function of equation 3.

$$\overrightarrow{gdm} = \sum_{i=2}^{\mu+\lambda} \frac{(\vec{x}_1 - \vec{x}_i)}{\|\vec{x}_1 - \vec{x}_i\|} \quad . \tag{1}$$

where x_i are the search points of the union of population of λ parents and μ offspring. The summation is applied to the normalised vectors $x_1 - x_i$ in order to equally consider the orientation of each vector without bias on its modulus.

3. THE EVOLUTION STRATEGY

The EA used for experimental verification of *global directional mutation* is an evolution strategy with adaptive mutation rate as described in [1]. The genotype is composed by $2n$ values representing the n-dimensional vector plus n adaptive variances. The population sizes for the parents and the offspring are $\mu = 50$ and $\lambda = 100$. A $(\mu + \lambda)$-selection was used. No recombination was applied. To avoid stagnation a lower bound to the adaptive variances was set to 10^{-4}. The variances were randomly initialised between 0.1 and the lower bound.

3.1 Adding the global directional mutation

The standard evolution strategy was executed with the addition of the new operator, while the other settings remained unchanged. An extra variance σ_{n+1} was added to the genotype to adjust the length of the vector gdm described by equation 1. With probability $0 < p < 1$, each offspring already mutated underwent the following further

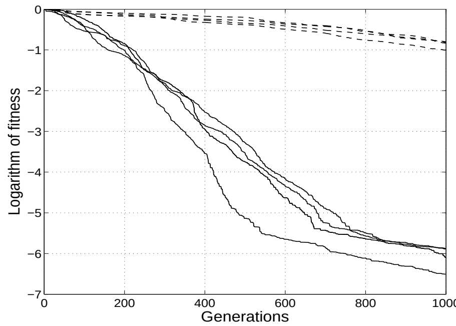

Figure 2: Performance of ES (dashed lines) and ES + gdm (continuous lines) for function f_1.

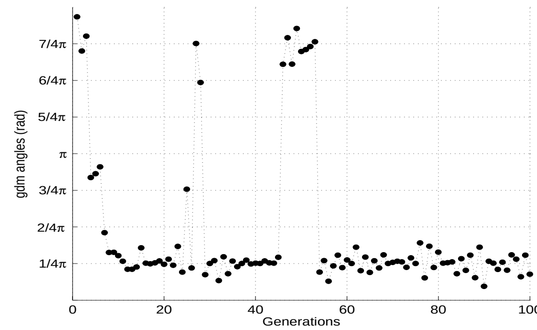

Figure 3: Angles of the *gdm* vector in a run for a 2-dimensional function f_1.

mutation

$$\begin{aligned}\sigma'_{n+1} &= \sigma_{n+1} \cdot \exp\big(\tau' \cdot N(0,1) + \tau \cdot N_{n+1}(0,1)\big) \\ \vec{x}' &= \vec{x} + (\sigma_{n+1} \cdot N(1,1) \cdot \overrightarrow{gdm}) \quad ,\end{aligned} \qquad (2)$$

where $N(1,1)$ has the purpose of distributing the new point over the line determined by *gdm* with a bias to the positive direction of the vector. The Gaussian noise acts like an additional scaling factor that randomises the position of the offspring on the supposed ridge-line. Note that when the probability p of applying the vector *gdm* is equal to 0, the algorithm is a traditional ES.

4. TEST PROBLEMS

As a test for ridge climbing algorithms, we propose the following function

$$f_1^{(n)}(x_1,\ldots,x_n) = 1 - \Big[\sum_{i=1}^{n-1} 10(|x_i - x_{i+1}|)^l + \sum_{i=1}^{n} x_i^m + 1\Big]^{-1} \quad . \qquad (3)$$

where l determines the sharpness of the ridge and m the profile. The minimum is $f_3(\vec{0}) = 0$, and the surface approaches asymptotically the value 1 for $\vec{x} \to \infty$. By changing the coefficients of x and y in the differential term, it is possible to vary the angle of the ridge. Figure 1 shows a two-dimension plot for l = 0.5 and m = 2.

A second test was done on the Rosenbrock's function in the generalised form

$$f_2(\vec{x}) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right] \quad . \qquad (4)$$

This function is characterised by a curved narrow valley that cannot be followed by changing only one variable a time. A third function used in the benchmarks is the sphere function: The test was introduced to assess the impact of the *gdm* operator when the landscape does not present ridges.

5. EXPERIMENTAL RESULTS

Figure 2 shows the minimum fitness during the evolutionary search for function f_1. Similar results were obtained for fuction f_2. On the sphere function, the *gdm* operator did not significantly affected the search, showing a small performance improvement. In all test, *gdm* was applied with probability p = 0.25.

5.1 Analysis

When *gdm* is applied, some of the offspring are cast along a line. Observing the search through generations, it is possible to notice that the angle of *gdm* moves and rotates searching for a ridge. While new points generated by random mutation are distributed in a hyper-sphere or hyper-ellipsoid, *gmd* generates points that stretch with different lengths along this line. When the population converges, the ridge is found and the *gdm* vector starts climbing it.

Once the population has converged on a ridge, and is therefore unable to proceed with a standard ES, the *gdm* vectors seem to explore the ridge-line and pull the whole population along it. This resembles swarm behaviour and suggests analogies with particle swarm optimisation.

Figure 3 shows the angles of *gdm* for each generation in a run on function f_1. Note that at the beginning, and twice during the run, the vector approaches angles of $7/4\pi$, that are normal to the ridge. This means that the vector drives the population towards the ridge before it starts climbing it, or needs to realign the population during climbing (generations 26-27, and 46-53). When the search points are aligned with the ridge, the angle approaches $1/4\pi$, the population has converged and climbs the ridge. The duplex role of *gdm* of first converging and aligning the population and then climbing the ridge is well illustrated in figure 3.

6. CONCLUSION

The new operator proved to enhance effectively the search on ridged landscapes. *Gdm* is simply implemented and does affect negatively the search on non-ridged landscapes. A real world application [3] and this study have shown its advantages. Its use is not limited to ES, *gmd* can be implemented for any EAs.

7. REFERENCES

[1] T. Bäck and H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutinary Computation*, 1(1):1–23, Spring 1993.

[2] R. Salomon. Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions; a survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39(3):263–278, 1996.

[3] A. Soltoggio. An Enhanced GA to Improve the Search Process Reliability in Tuning of Control Systems. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 05)*, 2005.