Alternative Implementations of The Griewangk Function

Artem Sokolov
sokolov@cs.colostate.edu

Darrell Whitley
whitley@cs.colostate.edu

Monte Lunacek
lunacek@cs.colostate.edu

Department of Computer Science
Colorado State University
Fort Collins, CO 80523

ABSTRACT

The well-known Griewangk function, used for evaluation of evolutionary algorithms, becomes easier as the number of dimensions grows. This paper suggests three alternative implementations that maintain function complexity for high-dimensional versions of the problem. Diagonal slices of the search landscape and local search are used to demonstrate and evaluate the difficulty of each function.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search; G.1.6 [**Numerical Analysis**]: Optimization—*Global Optimization*

General Terms

Keywords

Evolutionary Computation Benchmarks, Test Functions

1. INTRODUCTION

Complex and highly multi-modal functions are commonly used to evaluate evolutionary algorithms. Many of these functions can be scaled up to an arbitrary number of dimensions, either inherently through the function definition or by using an expansion technique [2]. Function behavior can, however, vary dramatically as the number of dimensions grows, and the original multi-modality, present in lower-dimensional versions of the problem, is sometimes lost. In this paper, we concern ourselves with one well-known benchmark function that suffers from this problem, namely the Griewangk function [1].

We address the loss of complexity in high dimensions by proposing three alternative implementations of the function. Twenty-dimensional diagonal slices of search landscapes are shown for visual comparison of multi-modality present in each implementation. We also make use of local search to perform empirical comparison among all implementations.

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

2. LOCAL SEARCH

In this paper, local search refers to a Gray coded *steepest ascent bit climber*. Each parameter is encoded as a Gray bit string and a neighborhood pattern forms around the current best solution by flipping one bit at a time. Local search evaluates all these neighborhood points before taking the best, or *steepest*, step. Local search restarts when no improving move is found.

3. GRIEWANGK TEST FUNCTIONS

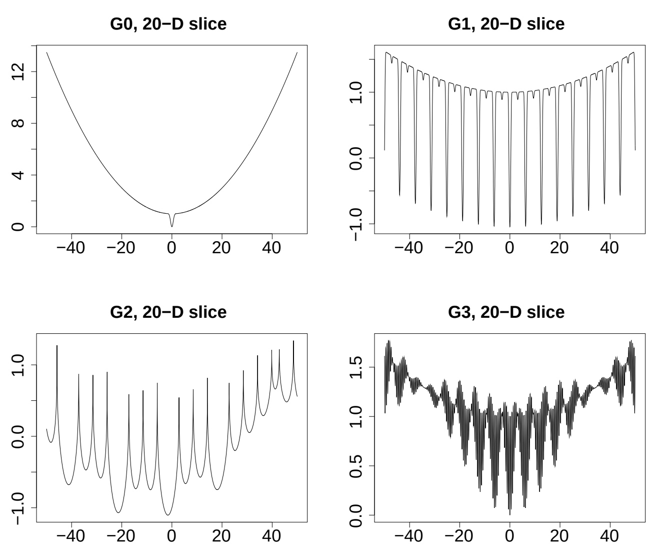

Figure 1: Diagonal slices of G0, G1, G2 and G3

The original Griewangk function is as follows [1]:

$$f(x_i|_{\ i=1,N}) = 1 + \sum_{i=1}^{N} \frac{x_i^2}{4000} - \prod_{i=1}^{N}(cos(x_i/\sqrt{i}))$$

The global optimum is at the origin and is equal to 0. The search space is bowl shaped due to the summation term, while local optima are created over the bowl through harmonic oscillation of a cosine function. The multiplication of cosines ensures that the problem is not separable.

The range of a cosine function is $[-1, 1]$. As the number of dimensions increase, repeated multiplication of values from this range cancels out the intended effect of harmonic oscillation. Higher dimensional versions of this problem become smoother and the problem becomes easier to solve [1].

Name	Function	domain
G1	$f(x_i\|\ _{i=1,N}) = \sum_{i=1}^{N} \frac{x_i^2}{4000N} \ - log\left[\left[\prod_{i=1}^{N}(cos(x_i)+0.1)\right]+1.0\right]$	$[-512, 511]$
G2	$f(x_i\|\ _{i=1,N}) = \sum_{i=1}^{N} \frac{x_i^2}{4000N} \ - 1.5^{N/4}\left[\prod_{i=1}^{N}\sqrt{cos(x_i/N+i)+1.0}\right]^{1/4}$	$[-512, 511]$
G3	$f(x_i\|\ _{i=1,N}) = 1 + \sum_{i=1}^{N} \frac{x_i^2}{4000N} \ - cos(1.1^{N-1}x_1)\prod_{i=2}^{N} cos\left[\frac{1.1^{N-i}x_i}{10^{i-2}}\right]$	$[-512, 511]$

Table 1: Modified Griewangk functions

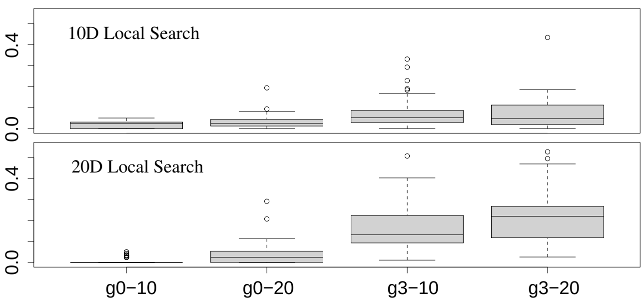

Figure 2: Local Search applied to ten- and twenty-dimensional versions of G0 and G3. Presented results are solutions found after 100,000 evaluations under 10 and 20 bits of precision.

We label the original Griewangk function as G0, and the three modified functions described below as G1, G2, and G3. Diagonal slices of each function for twenty-dimensional versions of the corresponding problems are given in Figure 1. Note the simplicity of the original G0.

Some general traits can be found across all the modifications. First, a scale factor is added to the summation term to stabilize the function range across dimensions. Second, the output from a cosine function is translated and/or scaled to offset the effect of multiplication. For G1 and G2 the global optimum does not have a consistent value as the number of dimensions is varied. However, G3 puts the global optimum at 0 independent of the number of dimensions. Finally, the product term may be passed to an additional function in order to change the shape of local optima. The new forms of Griewangk are given in table 1. All of the functions have the same basic form:

$$g(x_i|\ _{i=1,N}) = \sum_{i=1}^{N} \frac{x_i^2}{4000N} \ - X$$

The first modified function is characterized by taking a logarithm of the product term. As shown in Figure 1, the local optima become narrow and alternate in depth for G1. The overall bowl shape is retained.

The second modification, G2, involves a fourth order root. The optima are now rather wide with thin walls separating one from another. The search space also loses its symmetry due to a phase shift.

The last modified function, G3, is a fractal expansion that simplifies to the original Griewangk in 1-D. This results in a "cosine on cosine on cosine ..." structure. This function maintains a symmetrical bowl shape with the global optimum at the origin. The fractal expansion produces oscillations which are dependent on the number of dimensions N. As N increases, the smallest oscillation becomes more fine, and local optima continue to appear as one "zooms in" on an arbitrary interval of the search space. The function is very complex in high-dimensions.

4. EMPIRICAL COMPARISON

We ran Local Search for 100,000 evaluations on ten- and twenty-dimensional versions of G0, and G3. Our choice was guided by the fact that both functions have the global optimum at the origin and equal to zero, making comparison trivial. Figure 2 presents the results over 30 trials for 10 and 20 bits of precision. Each box-plot corresponds to the most optimal solution found after 100,000 evaluations.

There are two important things to note. The original G0 becomes easier as the number of dimensions is increased from 10 to 20. This is characterized by a smaller range of the box-plot. Similarly, the larger range of the box-plot corresponding to G3 suggests that the implementation is harder because the search does not converge to the same quality of solution in every trial.

The second thing to note is the problem difficulty with respect to precision. Because of the fractal nature of G3, a large number of local optima is introduced at a small scale making the search more difficult in twenty dimensions.

5. CONCLUSION

We have presented three alternative implementations to the well-known Griewangk function. These address the issue of the function becoming easier in a large number of dimensions. One of our proposed implementations also introduces a large number of local optima at a small scale as the number of dimension grows. This makes the function particularly appealing for evaluating evolutionary algorithms under high precision values.

6. REFERENCES

[1] D. Whitley, S. B. Rana, J. Dzubera, and K. E. Mathias. Evaluating evolutionary algorithms. *Artificial Intelligence*, 85(1-2):245–276, 1996.

[2] Whitley D., M. Lunacek, J. Knight. "Ruffled by Ridges: How Evolutionary Algorithms Can Fail". In K. Deb, ed., *GECCO (2) 2004: 294-306.* Springer-Verlag.