A Review of Benchmark Functions used in Optimization Algorithms

Abstract
Benchmark functions play an important role in validating and comparing the performance of optimization algorithms. The benchmark functions should have some diverse properties, which can be useful in testing of any new algorithm. The efficiency, reliability, and validation of optimization algorithms can be done by using a set of standard benchmarks or test functions. For any new optimization, it is necessary to validate its performance and compare it with other existing algorithms using a good set of test functions. This paper presents some of the benchmarks used in testing the validating the optimization algorithms.

Keywords—benchmark functions, optimization, optimization algorithms, optimization problems, search space, optimum.

1. Introduction
1.1. Optimization—A Brief Overview
Optimization is the search for a set of variables that can either maximize or minimize a scalar cost function, \( f(x(n)) \). The n-dimensional decision vector, \( x \), consists of the n decision variables over which the decision maker has control. The cost function is multi-value variable since it depends on more than one decision variable. The decision maker desires a more efficient method than trial and error by which to obtain a quality decision vector, so the optimization techniques are employed[1]. Optimization problems are widely encountered in various fields in science and technology. Sometimes such problems can be very complex due to the actual and practical nature of the objective function or the model constraints[1].

1.2. Optimization Problems
An optimization problem is the problem of finding the best solution from all the feasible solutions. Optimization problems can be divided into two categories depending on the nature of variables i.e. whether they are continuous or discrete. Some optimization problems include: PSO (Particle Swarm Optimization), ACO (Ant Colony Optimization), ABC (Artificial Bee Colony Optimization), Genetic Algorithms.

1.3. Benchmark functions
Benchmark functions are used to evaluate characteristics of optimization algorithms, such as: velocity of convergence, precision, robustness, general performance. Whenever a new algorithm is to be evaluated, the benchmark functions are employed to check its reliability, efficiency and validity.
2. Benchmark Functions

Benchmark functions play an essential role in validating and comparing the performance of optimization algorithms. The benchmark functions should possess some diverse properties, which can be useful in testing of any new algorithm. The efficiency, reliability and validation of optimization algorithms can be done using a pair of standard benchmarks or benchmark functions. For almost any new optimization, it is required to validate its performance and compare it with other existing algorithms employing a good pair of benchmark functions. Benchmark functions are used to judge characteristics of optimization algorithms, such as: velocity of convergence, precision, robustness, general performance. Every time a new algorithm is usually to be evaluated, the benchmark functions are employed to check on its reliability, efficiency and validity. Some common characteristics of benchmark functions can be noticed to have a better view about benchmark functions.

2.1. Characteristics of benchmark functions

Benchmark functions can be classified in the terms of features like: Modality, Basins, Valleys, Separability, Dimensionality[2].

1. Modality - A number of ambiguous peaks in function corresponds to the modality of a function. If an algorithm encounters some peaks during a search process, the algorithm may get trapped in one of such peaks.

2. Basins - A steep decline around a large area is called a basin. If the optimization algorithm gets into these regions, the search process of an algorithm gets affected negatively[3].

3. Valleys - When a narrow area of little change surrounds the regions of steep descent, valley occurs. The progress of a search process gets slow down[3].

4. Separability - Separability is the measure of difficulty of benchmark functions. The separable functions are easier to solve than the inseparable functions[4].

5. Dimensionality - As the dimension or the number of parameters increases, the search space increases exponentially[5].

The properties like: continuous or differentiable functions, separable or non-separable, scalable or non-scalable, unimodal or multimodal, are presented here.

<table>
<thead>
<tr>
<th></th>
<th>Characteristics of benchmark functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ackley: Continuous, Differentiable, Non-Separable, Non-Scalable, Unimodal</td>
</tr>
<tr>
<td>2</td>
<td>Griewangk: Continuous, Differentiable, Non-Separable, Scalable, Multimodal</td>
</tr>
<tr>
<td>3</td>
<td>Rosenbrock: Continuous, Differentiable, Non-Separable, Scalable, Unimodal</td>
</tr>
<tr>
<td>4</td>
<td>Schwefel: Continuous, Differentiable, Partially-Separable, Scalable, Unimodal</td>
</tr>
<tr>
<td>5</td>
<td>Sphere: Continuous, Differentiable, Separable, Scalable, Multimodal</td>
</tr>
</tbody>
</table>
2.2. Comparison of Benchmark functions

Results of implementing unimodal, multimodal and rotating test functions in optimization problems can be compared as following:

<table>
<thead>
<tr>
<th>Features</th>
<th>Unimodal functions</th>
<th>Multimodal functions</th>
<th>Shifting and rotating functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>Easy to locate the global optimum</td>
<td>Difficult to locate the global optimum</td>
<td>By rotating the functions, the dimensions of these functions become non-separable</td>
</tr>
<tr>
<td></td>
<td>Sphere, schwefel, rosenbrock</td>
<td>rastrigin, Ackley, griewangk</td>
<td>- the resulting problems become more difficult for a search algorithm to solve.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>rotated rastrigin, rotated griewangk, rotated weierstrass</td>
</tr>
</tbody>
</table>

3. Generating a Benchmark Function

A general procedure to generate a new benchmark function using linear transformations and function composition as follows:

1. We begin with a point P in the search space.
2. A point Q in homogeneous coordinates is computed using P.
3. The point Q is multiplied by a matrix representing a sequence of linear transformations to obtain Q₀.
4. Using Q₀, we compute P₀, which gives us the point P transformed in the search space.
5. The value of the test function fi is obtained passing the point P₀ as the argument to the test function, and f(P₀) is computed.
6. Before computing the composite function F, other linear transformations can be applied to fi(P₀) and obtain a fi(P₀).
7. Finally, we compute the composite function F by adding the fi(P₀) values or by computing the max of them[6].

4. Benchmarks and Experimental Settings

The benchmark functions are widely used to compare an algorithm’s performance with other algorithms.
Some commonly used benchmark functions and their properties like: search range, number of optimums, global minima, optimum point, are presented in a tabular form.
### Table 3- Standard features of benchmark functions

<table>
<thead>
<tr>
<th>Benchmark function</th>
<th>Search range</th>
<th>Number of optimums</th>
<th>Global Minima</th>
<th>Optimum point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere</td>
<td>[-100, 100]</td>
<td>1</td>
<td>((0, ..., 0))</td>
<td>(o = (0, 0, \ldots, 0)).</td>
</tr>
<tr>
<td>Rosenbrock</td>
<td>[-30,30]</td>
<td>1</td>
<td>(x^{*} = (l, ..., l))</td>
<td>(f(x^{*}) = 0). (o = (1, 1, \ldots, 1)).</td>
</tr>
<tr>
<td>schwefel</td>
<td>[-500,500]</td>
<td>Multiple local optimums</td>
<td>(x^{*} = (l, ..., l))</td>
<td>(f(x^{*}) = 0). (o = (420.96, 420.96, \ldots, 420.96)).</td>
</tr>
<tr>
<td>rastrigin</td>
<td>[-5.12, 5.12]</td>
<td>1</td>
<td>(x^{*} = (0, ..., 0))</td>
<td>(f(x^{*}) = 0). (o = (0, 0, \ldots, 0)).</td>
</tr>
<tr>
<td>Ackley</td>
<td>[-32, 32]</td>
<td>Multiple local optima</td>
<td>(x^{*} = (0, ..., 0))</td>
<td>(f(x^{*}) = 0). (o = (0, 0, \ldots, 0)).</td>
</tr>
<tr>
<td>Griewangk</td>
<td>[-600,600]</td>
<td>Multiple local optima</td>
<td>(x^{*} = (0, ..., 0))</td>
<td>(f(x^{*}) = 0). (o = (0, 0, \ldots, 0)).</td>
</tr>
</tbody>
</table>

**Heat map** is the graphical representation of data where individual values contained in a matrix are represented in the form of colors. The heatmaps of some functions are presented here.

![Heat map of Ackley function](image)

![Heat map of Griewangk function](image)
Rastrigin

Rosenbrock

Schwefel
5. Conclusion and Future Scope

Benchmark functions are important in testing or evaluating any algorithm. These functions are well-suited to evaluate a new algorithm, by comparing its efficiency with other algorithms and testing its validity using different parameters can be done. This paper presents the details about the characteristics of these benchmark functions and some features like: search space, global optimum, optimal point, number of optimums etc. These properties are helpful in differentiating the benchmark functions from each other. Some functions are presented here, and many more can be worked upon in future.

6. References

[1] Woo Nam Lee and Jong Bae Park, "Educational Simulator for Particle Swarm Optimization and Economic Dispatch Applications".