Notes on benchmark defintion for optimization algorithms

December 1, 2025

Abstract

To evaluate performance increasements of variouse optimization algorithms a benchmark should be definied, which first measures all the given algorithms running on a previously definied workload without an enhancement of performance and afterwards with an enhancement. This makes the improvement comparable. Further the target of the benchmark should be mostly independent on the computation enviroment.

1 Domain independent-definition

In general performance measures fall into 4 categories; efficiency, reliability, and quality of algorithmic output and parameter tuning and stopping conditions.

1. **Efficiency** - *Number of fundamental evaluation*: Times the algorithms is calling a subroutine which gives information about the optimization problem, e.g.: the evaluation of the objective function. *Running Time*, Either measure wall clock or CPU time but keep the background operations to a minimum.
2. **Reliability** - *Success rate*: counting the number of test problems that are successfully solved within a pre-selected tolerance. *Constraint violations*: The average objective function values and the average constraint violation values have also been reported to measure reliability. *Multiple Startingpoints* reliability can be based on fixed starting points, but its is often better to use multiple starting points.
3. **Quality of Solution** - *Known solution available*: If a solution is available we can use different metrics to measure against the soltuion. *No known solution available*: Use the best performed output so far or another approach is to estimate optimal solution using statistical techniques.
4. **Parameter tuning and stopping conditions** If different choices of input parameters are allowed in an algorithm, researchers should mention the parameter settings used and how they were selected.

Category	Description	Equation
Efficiency	Number of evaluations until solution	EvalCount
Efficiency	CPU time until solution obtained	T_{CPU}
Reliability	Success rate	$\frac{\text{\#successful runs}}{\text{\#total runs}}$
Quality of Solution	Final error	$f(\bar{x}) - f(x^*)$
Quality of Solution	Normalized final error	$\frac{f(\bar{x})-f(x^*)}{f(x_0)-f(x^*)}$
Quality of Solution	Variability	$\text{std}\big(f(\bar{x})\big)$

Table 1: Benchmarking metrics for optimization algorithms

2 Domain dependent-definition

The following presents the application of the given 4 categories on a real use case. The workload we consider is training a DCGAN. We will measure the performance of various optimization algorithms running only in parallel on CPU and GPU.

1. **Efficiency** - *Running Time*: GPU/CPU operation time until solution obtained (Computation). Data Transfer Time e.g.: time spent in next(batch) (I/O).

2. **Reliability** - *Success rate*: Accuracy / Success rate for every 10th epoch.

3. **Quality of Solution** - *Known solution available*: Normalized final error and Variability.