cocoa

Release 0.1

Mike Gimelfarb

Dec 22, 2023

CONTENTS:

CHAPTER

ONE

INTRODUCTION

COCOA is a suite of algorithms written in C++ for the optimization of continuous black-box functions (mostly without using derivative information). Main advantages:

- a single unified interface for all algorithms
- a variety of classical algorithms with state-of-the-art improvements (e.g. automatic parameter adaptation)
- convenient wrappers for Python with a user-friendly API

CHAPTER

TWO

INSTALLATION

To use this library in a Python project, you will need:

- C++ compiler (e.g., MS Build Tools)
- git
- pybind11

Then install directly from source:

```
pip install git+https://github.com/mike-gimelfarb/cocoa
```

CHAPTER

THREE

BASIC EXAMPLE

The following example optimizes the 10-dimensional <Rosenbrock function *https://en.wikipedia.org/wiki/Rosenbrock_function*>_ using the active variant of the CMA-ES evolutionary strategy optimizer

```
import numpy as np
from cocoaopt import ActiveCMAES

# function to optimize
def fx(x):
    return sum((100 * (x2 - x1 ** 2) ** 2 + (1 - x1) ** 2) for x1, x2 in zip(x[:-1],␣
↪x[1:]))

n = 10  # dimension of problem
alg = ActiveCMAES(mfev=10000, tol=1e-4, np=20)
sol = alg.optimize(fx,
                   lower=-10 * np.ones(n),
                   upper=10 * np.ones(n),
                   guess=np.random.uniform(low=-10., high=10., size=n))
print(sol)
```

This will print the following output:

```
x*: 0.999989 0.999999 1.000001 1.000007 1.000020 1.000029 1.000102 1.000183 1.000357 1.
↪000689
objective calls: 6980
constraint calls: 0
B/B constraint calls: 0
converged: yes
```