Blockwise direct search methods

LI Haitian, The Hong Kong Polytechnic University

✉ *hai-tian.li@connect.polyu.hk*

Introduction

Direct search methods are one class of derivative-free optimization algorithms that evaluate the objective function only based on the comparison of function values. We introduce a new framework of direct search method called blockwise direct search (BDS), which divides the searching set into blocks. For every iteration, each block has its step size. We develop this framework into a solver, which is open-source and easy to use. In numerical experiments, we observe that BDS can also be compared with model-based methods in some cases. In addition, our solver shows its efficiency and stability under noise without introducing specific techniques. BDS can also be used to tune the hyperparameters itself to improve the performance.

Problem Definition

Solve an optimization problem

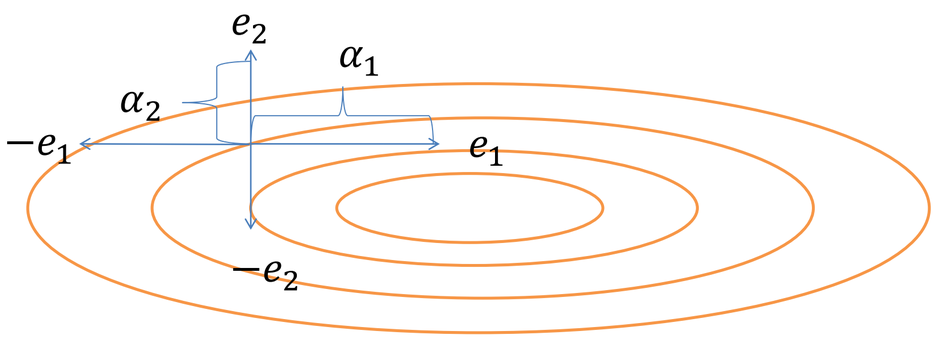
$$min_{x\in\mathbb{R}^n} f(x)$$

using function values but not derivatives (classical or generalized).

In some occasions, we may encounter the above problem when

- Derivatives are not available even though f may be smooth.
- "not available": the evaluation is impossible or too expensive.

Methods



$\mathcal{D}_1 = \{e_1, -e_1\}$ and $\mathcal{D}_2 = \{e_2, -e_2\}$

- One specific case:

Algorithm 1: Cyclic Blockwise Direct Search (CBDS)

Input: $x_0 \in \mathbb{R}^n$, $0 < \theta < 1 \leq \gamma$, $\alpha_0^1, \ldots, \alpha_0^m \in (0, \infty)$, a forcing function ρ, and a search direction set $\mathcal{D} = \cup_{i=1}^m \mathcal{D}^i \subset \mathbb{R}^n$.

for $k = 0, 1, \ldots$ **do**
  Set $y_k^1 = x_k$
  for $i = 1, \ldots, m$ **do**
    if $f(y_k^i + \alpha_k^i d_k^i) < f(y_k^i) - \rho(\alpha_k^i)$ for some $d_k^i \in \mathcal{D}^i$ **then**
      Set $y_k^{i+1} = y_k^i + \alpha_k^i d_k^i$ and $\alpha_{k+1}^i = \gamma \alpha_k^i$
    else
      Set $y_k^{i+1} = y_k^i$ and $\alpha_{k+1}^i = \theta \alpha_k^i$
  Set $x_{k+1} = y_k^{m+1}$

Results

- Blockwise Direct Search (BDS) is a substantial improvement over the classical direct search method based on sufficient decrease
- BDS is robust under noise without any noise-handling techniques

- open-source and easy to use
- tested continuously via GitHub Actions
- tested under different platforms

BDS on GitHub

Discussion

- Convergence and worst-case complexity (an adapted framework?)
- Make use of the existing iterates (finite difference or interpolation)
- Extend our implementation to other languages (Python, Julia, etc.)

References

1. M. J. D. Powell. On search directions for minimization algorithms. Math. Program., 4:193–201, 1973.
2. T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. SIAM Rev., 45:385–482, 2003.