Introduction

In this project, you will learn how to use performance profiles to compare different optimisation methods when applied to a collection of benchmark instances. You are expected to implement some of the unconstrained optimisation algorithms presented in the course and analyse their performance using the knowledge acquired so far.

You are asked to solve four problems. The first three problems consist of 2-dimensional functions. In the last problem, you will solve 100 random instances of a large-scale quadratic problem. You are expected to implement six variants of three optimisation methods: each method uses two distinct line searches. In addition, you must write a report with a detailed analysis of the performance of the optimisation methods when solving the four problems.

Learning objectives

1. Learn by practice some of the algorithms for unconstrained optimisation to obtain a deeper understanding of the methods and challenges concerning their implementation;
2. Learn how to perform a structured comparison of different algorithms using performance profiles;
3. Learn how to report numerical results that are scientifically sound and reproducible.

General requirements

1. Students can work individually or in pairs to complete this project. Groups of three or more individuals will not be accepted;
2. Students are expected to submit a report (see details in the "Report format" section) and **all Julia code files** developed for the project. The codes will be tested and, therefore, should be commented accordingly.
3. **Deadline** for submitting the project report and the code files is **25/11/2018**. Late submissions will not be accepted.

Specific project requirements

1. You must implement and compare the performance of the following algorithms, with and without exact line searches:
   (a) Gradient method
   (b) Newton's method
   (c) BFGS quasi-Newton method

   The exact line search method must be the univariate Newton's method with $\lambda = 0$ as an initial point. The inexact line search must be Armijo's rule, parameterised by the students. The parameterisation must be clearly justified in the report.
2. The following four functions must be optimised:
   (a) $f(x_1, x_2) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$ with starting point $(x_1, x_2) = (7, 3)$
   (b) $f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$ with starting point $(x_1, x_2) = (1, 1.5)$
   (c) $f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$ with starting point $(x_1, x_2) = (-2, 1)$
   (d) $f(x) = (1/2)x^\top Ax - b^\top x$ with $x \in \mathbf{R}^{150}$ (100 problem instances with random A and b)

   For problems (a), (b), and (c), the report must include a critical assessment of the performance of each algorithm for a maximum of 10000 iterations (i.e., if the solution obtained is optimal, number of iterations taken, time per iteration, and total time).

   For the set of problems (d), the report must include performance profiles comparing the 6 algorithm variants. Information concerning performance profiles can be found in this link: https://link.springer.com/article/10.1007/s101070100263. A skeleton code that constructs the random instances and performance profiles can be downloaded here.

Report format

A Latex template for the report is provided and can be downloaded here. Font sizes, margins, and other layout settings will be set for the template and are not to be tampered with.

The report has a strict page limit of five pages, which includes figures. References are not necessary, but if used, they are not included in the page limit.

The report will consist of the following sections:

1. **Project description**

   Describe the project setting, such as the computational platform used (operating system, processor type and clock speed, memory), the algorithmic variants implemented and tested, the parameterisation (for example, multipliers in the Armijo rule and tolerance values used), how derivatives are determined (using ForwardDiff package or calculated beforehand). All details that allow replicating the numerical experiments must be described in this section.

2. **Numerical results**

   For the 2-dimensional functions (a), (b), and (c), the algorithms must be compared in terms of their convergence, iteration count, and solution time. You are expected to give reasoning for the observed performance based on the information presented in this course. A structured comparison between all variants is expected for the 2-dimensional functions. Plots showing the convergence behavior are also expected. Example codes used in class will be provided.

   For the set of problems (d), present the performance profiles for the two different condition number settings, and provide a detailed comparison between the algorithms based on these profiles. Discuss also how the two different line searches and condition number settings affect the performance of the algorithms.

3. **Discussion and conclusions**

   Rank the algorithms according to their performance in each problem (a), (b), (c), and (d). Discuss aspects that affect the behavior of each algorithm, and if the obtained solutions can be classified as global optima (instead of local optima).

Assessment

The assessment consists of two components: a report component comprising 80% and a feedback component comprising 20% of the final grade.

1. **Report component**

   The report will be assessed by the lecturer and the TA according to the following criteria:

   (1) Correctness of the algorithms
   (2) Technical quality of analysis
   (3) Format, clarity, and presentation

   Each criterion (1), (2), and (3) will be graded by both the lecturer and the TA, and the final grade will be the average of 5 best criterion grades.

2. **Feedback component**

   Each (pair of) student(s) are supposed to provide feedback on the report of another (pair of) student(s). The reports will be distributed randomly. The feedback must be provided on a one-page report, commenting on both the positive aspects and those that could be improved based on the assessment criteria. No scores need to to be provided.

   The quality of the feedback report will be accessed by the lecturer and the TA by a grade. All students will receive the feedback report concerning their own work without name disclosure.