Assignment I

Anonymous Student, Hardworking Student

*Aalto University School of Science, Department of Mathematics and Systems Analysis*

*Aalto University School of Science, Department of Computer Science*

*{anonymous.student,hardworking.student}@aalto.fi*

Read carefully the comments below and delete all the text before starting to fill in with your report.

1. **Project Description**

In this section, the students should provide an **overview** of the **goal** of this project.

For example:

*In this project, we explore the numerical performance between different methods commonly presented as possible solvers for unconstrained optimisation problems. In our experiments, we tested and reported the performance profile of the gradient (standard and ”heavy ball” variant) and Newton and quasi-Newton (BFGS variant) with exact and inexact line search methods in three different examples. The first two consist of two-dimensional functions where the convergence rate, local or global optimality, number of iterations and average runtime (per iteration) were reported. In the final problem, we ...*

In this section, it is crucial to mention the **setting** of the computational platform used (either local or a server) and details on each method/algorithm that will be implemented and tested. Note that details or pseudo-codes of those algorithms are not necessary. In addition, you should **describe** the experiments that will be performed and the subject (type of functions).

Suggestion: any and all visual aid is always welcomed (contour curve (<https://docs.juliaplots.org/latest/series_types/contour/> ), plots (<https://docs.juliaplots.org/latest/tutorial/)>)

1. **Numerical Results**

In this section, the **core** component of this project should be thoroughly described. Start with dividing the section into **three** subsections: one for each problem, **provide** a description of the numerical results. To aid your description, create a table (such as below) and create a structured comparison between each method. For example, a comparison regarding which methods have the best convergence rate, the smallest amount of iterations, and the smallest average runtime should be included in our text. Such comparisons can be visually aided by corresponding plots.

|  |  |  |  |
| --- | --- | --- | --- |
| Value of | Time | Iterations | Solution |
| 10 | 111.02 | 2 | 12.09 |
| 100 | 98.09 | 4 | 12.09 |
| 200 | 124.34 | 7 | 999.92 |
| 500 | 10.70 | 4 | 12.09 |
| 1000 | 234.01 | 10 | 12.34 |
| 10000 | 592.12 | 9 | 12,40 |

Suggestion: Each **column** in your table could also be visualised as part of a **plot**. For example, a relevant comparison could be between the gradient and gradient ”heavy-ball” runtime, each with an inexact and exact search, resulting in four different sets of values. A **bar plot** could easily be used to compare all four methods.



**Figure 1** - Plotting with Julia is fun when it works

For problem III, the comparison should be pushed further to explore the differences between the two profiles.

Suggestion: Given that 100 distinct problem instances will be randomly generated, we can express the variation between each instance performance by creating a boxplot (more details in <https://plotly.com/julia/box-plots/> ).

2.1) Problem I

In this section, we are reporting the numerical results from the following task I ...

2.2) Problem II

In this section, we are reporting the numerical results from the following task II ...

2.3) Problem III

In this section, we are reporting the numerical results from the following task III ….

1. Discussion and conclusions

While the numerical results were presented in the previous sections, in this section, we should present the general aspect of all four algorithms. Callbacks to the previous section are encouraged, and a final analysis regarding the behaviour of the results and the convexity and type of function (for example) could be expressed.

Suggestion: In this section, we answered the question: *out of all the experiments, why did problem X have the best performance using method Y ?*