CS1.404 (Spring 2024)
Optimization Methods
Deadline: 11.55 PM, May 6^{th}, 2024

Instructions

1. Attempting all questions is mandatory.
2. You are expected to solve all the questions using python programming language.
3. Use of any in-built libraries to solve the problem directly is not allowed.
4. Submission Format: Check assignment description or announcement post for more details.
5. Plagiarism is a strict No. We will pass all codes through the plagiarism checking tool to verify if the code is copied from somewhere. In that case, you get F grade in the course.
6. If any two students codes are found exactly same (if they copy from each other), both will get F grade.

1 Functions

1.1 Trid Function

$$f(\mathbf{x}) = \sum_{i=1}^{d}(x_i - 1)^2 - \sum_{i=2}^{d} x_{i-1}x_i$$

1.2 Matyas Function

$$f(\mathbf{x}) = 0.26(x_1^2 + x_2^2) - 0.48x_1 \times x_2$$

2 Algorithms

Implement the following algorithms

2.1 Projected Gradient Method

Refer to lecture 24 slides for pseudocode. Test cases will contain the constraints. There are two types of constraints-

- **linear:** $\mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub}$ where $\mathbf{lb}, \mathbf{x}, \mathbf{ub} \in \mathbb{R}^n$. In the testcase the constraints will be given in the form of a $2 \times n$ *numpy ndarray* where **lb** and **ub** are the first and second vectors respectively.
- $\mathbf{l_2}$: $\|\mathbf{x} - \mathbf{c}\| \leq r$ where $\mathbf{x}, \mathbf{c} \in \mathbb{R}^n, r \in \mathbb{R}$. In the testcase the constraint will be given in the form of a *tuple* containing its center ($1 \times n$ *numpy ndarray*) and radius respectively.

Stopping conditions

- Main algorithm *(refer to slide no. 5, lecture 24)*: Take $\epsilon = 10^{-2}$ or terminate after 1000 iterations.
- Backtracking *(refer to slide no. 16, lecture 24)*: Stop if

$$|f(\mathbf{x_k}) - f(P_c(\mathbf{x_k} - t_k \nabla f(\mathbf{x_k}))) - t_k \|G_{\frac{1}{t_k}}(\mathbf{x_k})\|^2| < 10^{-3}$$

2.2 Dual Ascent

For the given problem form the Lagrangian and optimize it using the dual ascent method. You will be provided the function, the derivative, a list of constraints and a list of the derivatives of the constraints. All the constraints are of the form $h_i(x) \leq 0$. You are not allowed to hard code the Lagrangian, you have to form it dynamically. Verify using KKT conditions that the point obtained is a local minima.

Algorithm 1 Dual Ascent

Initialization: Set $\alpha = 10^{-3}, k = 0$
while $k < 10000$ **do**
  $\mathbf{x_{k+1}} = \mathbf{x_k} - \alpha \nabla_x \mathcal{L}(\mathbf{x}_k, \lambda_k)$
  $\lambda_{\mathbf{k+1}} = \lambda_{\mathbf{k}} + \alpha \nabla_\lambda \mathcal{L}(\mathbf{x}_{k+1}, \lambda_k)$
  $k \leftarrow k + 1$
  $\lambda_{k+1,i} \leftarrow max(0, \lambda_{k+1,i}) \forall i$
end while
return $\mathbf{x_k}$

3 Submission Instructions

3.1 Allowed Packages

1. Python 3.11
2. NumPy
3. Matplotlib
4. PrettyTables

3.2 Boiler Plate

You have been provided with three files

1. *algos.py*: Where you have to implement the algorithms.
2. *functions.py*: Which contains all the functions used for testing.
3. *main.py*: This is the file to be executed to test your code.
4. *test_cases.py*

You are not allowed to create anymore files, nor are you allowed to modify *main.py* and *functions.py*. You can make changes to these files for your own test purposes but the final evaluations will use the original files for testing. In *algos.py* you are allowed to create your own new functions.

3.3 Report

You are required to submit a report (as a pdf) containing the KKT condition derivations from dual ascent.

3.4 Submission Format

Create a folder with your Roll No. as its name containing the four files *(algos.py, functions.py, main.py, Report.pdf)*. Zip it and name it Roll_No.zip
Example- *2019111013.zip*