

Homework 2

Finding multiple extrema of a function

Complicated functions generally have more local minima

[https://en.wikipedia.org/wiki/Test_functions_for_optimization]. Even if the gradient method converges, it will find only one of these minima. But this minimum can be local and can be far from the global minimum. For this reason, the gradient method is often started from different starting points and then the best point to which the method has converged is selected. In this task, we will consider the function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, an admissible set of the form of the multidimensional interval (box) $[x_{\min}, x_{\max}] \subset \mathbb{R}^n$ and the optimization problem

$$\begin{aligned} &\text{minimize} && f(x) && (1) \\ &\text{under conditions} && x \in [x_{\min}, x_{\max}]. && (2) \end{aligned}$$

Recall that $x \in [x_{\min}, x_{\max}]$ is satisfied if it holds for all components. As one of the test functions we will consider the Griewank function [https://en.wikipedia.org/wiki/Griewank_function]

$$f_{\text{griewank}}(x) = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \frac{x_i}{\sqrt{i}}.$$

Input

Implement a `generate_solutions` function that finds all local minima of a function on a given interval. The `generate_solutions` function must have the following input arguments (in the order listed):

- `f` : the f function that will be used for minimization,
- `g` : its derivative ∇f ,
- `x_min` : lower limit of the admissible set,
- `x_max` : upper limit of the admissible set.

The `generate_solutions` function somehow generates a larger number of admissible points in the admissible set and then runs the gradient method with projection for each of them. Use the (slightly modified) code from the lecture.

```
function optim(f, g, P, x; α=0.01, max_iter=10000)
    for _ in 1:max_iter
        y = x - α*g(x)
        x = P(y)
```

```

    end
    return x
end

```

```

P(x, x_min, x_max) = min.(max.(x, x_min), x_max)

```

Use all outputs of the `optim` function. So do not check if the found point is a local minimum, a global minimum or a stationary point. Then filter out the unique solution. Remember that the gradient method does not converge exactly to the solution, but only very close to the solution. In other words, the same solution can have a different representation, which differs only by a small error.

The `generate_solutions` function must also meet the following properties:

- We can assume that the interval is entered correctly, i.e. $x_{\min} \leq x_{\max}$.
- If $f : \mathbb{R} \rightarrow \mathbb{R}$, the solution must operate on both scalar and vector inputs. So if we call `f(0)`, the solution can be called via `generate_solutions(f,g,-1,1)`. If, on the other hand, we call `f([0])`, the solution can be called via `generate_solutions(f,g,[-1],[1])`.
- Its output is an $n \times k$ matrix, where n is the dimension of the vector x and k is the number of found points.
- The solution must not run for too long. A timeout of 150 seconds is set for evaluating the solution.

Next, implement the function `f_griewank` described above and its derivative `g_griewank`. Remember that this function depends on the length n of its argument. If $n = 1$, this function must be called by both `f_griewank(0)` and `f_griewank([0])`, as well as its derivative.

courses/b0b36jul/en/hw/hw2.txt · Last modified: 2022/10/24 15:10 by adamluk3