technische universität dortmund

Fakultät für Informatik
Lehrstuhl für Algorithm Engineering

Übungen zur Vorlesung
Praktische Optimierung, SoSe 2024

Prof. Dr. Günter Rudolph, Dr. Marco Pleines
http://ls11-www.cs.tu-dortmund.de/people/rudolph/teaching/lectures/POKS/SS2024/lecture.jsp

Blatt 2, Block A

22.04.2024

Abgabe: 02.05.2024

Aufgabe 2.1: Optimierung mithilfe von *scipy* (4 Punkte)

Verwenden Sie aus dem `Python`-Modul `scipy.optimize` die Funktionen `minimize` (bzw. `minimize_scalar` für `Brent`), um die beiden Funktionen

(a) $f_1(x) = (x+5)^2, \quad x \in [-10, 10], \quad$ und

(b) $f_2(x) = (x+3)^2 - 5\cos(5x), \quad x \in [-10, 10]$

zu minimieren. Nutzen Sie dafür die Verfahren `Brent`, `BFGS` und `L-BFGS-B`, die in *scipy* implementiert sind. Verwenden Sie als Startwert $x_0 = 0$ und nutzen Sie ansonsten die *default*-Einstellungen. Plotten Sie mithilfe von *matplotlib* sowohl die Funktionen (ein Plot pro Funktion) als auch die Optimierungsresultate (je Plot drei Punkte) und sorgen Sie mithilfe von Beschriftungen dafür, dass die Grafiken verständlich sind. Interpretieren Sie Ihre Ergebnisse.

Aufgabe 2.2: Kompasssuche (6 Punkte)

Implementieren Sie die Kompasssuche, wie sie in der Vorlesung beschrieben wurde. Verwenden Sie für die Menge D erst die positiven und dann die negativen Einheitsvektoren, also beispielsweise $D = \{e_1, e_2, -e_1, -e_2\}$. Die Kompasssuche soll, wie in der Vorlesung beschrieben, für n Dimensionen einsetzbar sein. Der Funktion soll den Namen `kompasssuche` sowie die Argumente `f`, `x0`, `s0`, `theta` und `iters` haben. Speichern Sie in jedem Iterationsschritt der Kompasssuche die Werte von $x^{(k)}, s^{(k)}$ und $f(x^{(k)})$ ab, um später den „Weg“ der Suche nachvollziehen und ausgeben zu können. Für die volle Punktzahl darf pro Iteration der Kompasssuche nur ein Funktionsaufruf pro betrachteter Dimension verwendet werden.

Nutzen Sie Ihre Implementierung, um die Funktionen

(a) $f(x) = (x+5)^2, \quad x \in [-10, 10],$

(b) $f(x) = (x+3)^2 - 5\cos(5x), \quad x \in [-10, 10],$

(c) $f(x, y) = x^2 + y^2, \quad x, y \in [-10, 10],$

(d) $f(x, y) = x\sin(x) + 3y^2, \quad x, y \in [-10, 10]$

zu minimieren.

Wenden sie dabei die Kompasssuche für alle möglichen Kombinationen der folgenden Parameter an:

- Funktionen (a) und (b): $x^{(0)} \in \{3, 9\}, s^{(0)} \in \{0.5, 4\}, \theta \in \{0.3, 0.8\}$, iters $= 20$,
- Funktionen (c) und (d): $x^{(0)} \in \{(3,3), (9,9)\}, s^{(0)} \in \{0.5, 4\}, \theta \in \{0.3, 0.8\}$, iters $= 20$.

Geben Sie die Kombinationen von Parametern an, für die jeweils die beste Approximation des Optimums erreicht wird. Wird für mehrere Parameterkombinationen dieselbe Approximationsgüte erzielt, wählen Sie diejenige Parameterkombination aus, für die die Kompasssuche die geringste Anzahl an Iterationsschritten benötigt.

Plotten Sie die vier Zielfunktionen f und den „Weg“ der Kompasssuche mit den von Ihnen ausgewählten Parametern in vier separaten Plots. Zum Plotten des 2D-Plots kann das Paket *matplotlib* empfohlen werden (vgl. *Einführung in Python*). Zum Nummerieren der Wegpunkte im Plot eignet sich die Funktion `plt.text()`. Für den 3D-Plot bieten sich *matplotlib* oder *plotly* an (vgl. ebenda). Interpretieren Sie die Ergebnisse in einem kurzen Erklärungstext.