technische universität dortmund

Fakultät für Informatik
Lehrstuhl für Algorithm Engineering

Übungen zur Vorlesung
Praktische Optimierung, SoSe 2017

Günter Rudolph, Simon Wessing
http://ls11-www.cs.tu-dortmund.de/people/rudolph/teaching/lectures/POKS/SS2017/lecture.jsp

Blatt 3, Block A

08.05.2016

Abgabe: 16.05.2016, 12:30 Uhr

Aufgabe 3.1: Gradientenverfahren (10 Punkte)
In der Datei `https://ls11-www.cs.uni-dortmund.de/_media/de/rudolph/lehre/po17/po17_blatt3_code.zip` wird ein herkömmliches Gradientenabstiegsverfahren nach dem Schema

$$x^{(k+1)} = x^{(k)} - s(k)\frac{\nabla f(x^{(k)})}{\|\nabla f(x^{(k)})\|}$$

samt einer Testfunktion

$$f(\vec{x}) = \sum_{i=1}^{n}\Big(\sum_{j=1}^{i} x_j\Big)^2.$$

zur Verfügung gestellt. In dieser Implementierung wird eine Schrittweite vorab gewählt und verwendet, solange in jeder Iteration eine Verbesserung erzielt wird. Im Falle einer Verschlechterung wird die Schrittweite mit einem Faktor $g \in]0,1[$ multipliziert.

(a) (3 Punkte) Versuchen Sie den Faktor g so einzustellen, dass der finale Zielfunktionswert optimal wird. Belassen Sie alle anderen Parameter auf ihren Voreinstellungen. Beschreiben Sie, was Sie getan haben und interpretieren Sie die Ergebnisse.

(b) (3 Punkte) Ersetzen Sie die Gradientenfunktion durch eine Approximation per Differenzenquotient. Vergleichen Sie die Laufzeit und Qualität mit der Variante mit analytischem Gradienten.

(c) (4 Punkte) Ersetzen Sie die Funktion `optimizationStep()` durch eine Funktion, die die Liniensuche „Brent“ aus der Funktion `optim` verwendet. Der Parameter `stepSize` wird in diesem Fall zur oberen Schranke. Untersuchen Sie die Auswirkungen der Änderung hinsichtlich des finalen Zielfunktionswertes und der verbrauchten Zeit. Optimieren Sie dann wieder den Schrumpfungsfaktor g und untersuchen Sie erneut.