Univ.-Prof. Dr. Radu Ioan Boţ

Vienna, January 17, 2020
To be discussed on January 24, 2020

Nonlinear Optimization

Exercise session 7

46. Consider the optimization problem

$$\begin{array}{ll} \min & f(x) := (x_1+1)^2 + (x_2+2)^2 \\ \text{s.t.} & g_1(x) := -x_1 \leq 0 \\ & g_2(x) := -x_2 \leq 0 \end{array}$$

with $x = (x_1, x_2)^T$. For $\alpha > 0$, find the minimum $x^*(\alpha)$ of the penalty function

$$P(x;\alpha) := f(x) + \frac{\alpha}{2}\|g_+(x)\|^2,$$

and the limits $x^* := \lim_{\alpha \to +\infty} x^*(\alpha)$ and $\lambda^* := \lim_{\alpha \to +\infty} \alpha g_+(x^*(\alpha))$. Find out if (x^*, λ^*) is a KKT point of the optimization problem. (3 points)

47. Consider the optimization problem

$$(P) \qquad \begin{array}{ll} \min & f(x) := x^2 \\ \text{s.t.} & g(x) := 1 - \ln(x) \leq 0 \end{array}$$

and the penalized optimization problem

$$\min_{x \in \mathbb{R}} P(x;\alpha) := f(x) + \alpha\varphi\left(\frac{g(x)}{\alpha}\right)$$

with $\varphi(t) = e^t - 1$ (*exponential penalty function*). For $\alpha > 0$, find the optimal solution $x^*(\alpha)$ of the penalty optimizaton problem and prove that $x^* := \lim_{\alpha \downarrow 0} x^*(\alpha)$ is an optimal solution of the problem (P). (3 point)

48. Consider the optimization problem

$$\begin{array}{ll} \min & x^2 \\ \text{s.t.} & x - 1 = 0 \end{array}$$

and its optimal solution $x^* = 1$. Find $\bar{\alpha} > 0$ such that the ℓ_1-penalty function $P_1(\cdot;\alpha)$ is exact at x^* for every $\alpha \geq \bar{\alpha}$. (3 points)

49. Consider the optimization problem (P) in Exercise 44.

(a) Prove that

$$\mu^* := \lim_{\alpha\to\infty} \alpha h(x^*(\alpha))$$

is the Lagrange multiplier which corresponds to the optimal solution x^*.

(b) Consider the penalized optimization problem

$$\min_{x\in\mathbb{R}^n} P_1(x;\alpha) := f(x) + \alpha|h(x)|.$$

Find $\bar{\alpha} > 0$ such that $P_1(\cdot;\alpha)$ is exact at x^* for every $\alpha \geq \bar{\alpha}$.

Hint. Use μ^* to find $\bar{\alpha}$. (4 points)

50. Prove that the following functions are NCP-functions:

(a) *the minimum function*:

$$\varphi(a,b) = \min\{a,b\}.$$

(b) *the Fischer-Burmeister function*:

$$\varphi(a,b) = \sqrt{a^2+b^2} - a - b.$$

(c) *the penalized minimum function*:

$$\varphi(a,b) = 2\lambda \min\{a,b\} + (1-\lambda)a_+b_+,$$

where $a_+ := \max\{a,0\}, b_+ := \max\{b,0\}$ and $\lambda \in (0,1)$.

(3 points)

51. Let $(x^*,\lambda^*,\mu^*) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p$ be a KKT point of the optimization problem (all functions are assumed to be twice continuously differentiable)

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_i(x) \leq 0, i = 1,...,m \\ & h_j(x) = 0, j = 1,...,p \end{array}$$

fulfilling:

(i) $g_i(x^*) + \lambda_i^* \neq 0$ for all $i = 1,...,m$ (*strict complementarity*);

(ii) the gradients $\nabla g_i(x^*), i \in \mathcal{A}(x^*) = \{i = 1,...,m : g_i(x^*) = 0\}$, and $\nabla h_j(x^*), j = 1,...,p$, are linearly independent (*LICQ*);

(c) it holds $d^T\nabla^2_{xx}L(x^*,\lambda^*,\mu^*)d > 0$ for all $d \neq 0$ with $\nabla g_i(x^*)^Td = 0$, $i \in \mathcal{A}(x^*)$, and $\nabla h_j(x^*)^Td = 0, j = 1,...,p$ (*second order sufficient optimality condition*).

Further, let $\Phi : \mathbb{R}^{n+m+p} \to \mathbb{R}^{n+m+p}$ be defined by

$$\Phi(x,\lambda,\mu) := \begin{pmatrix} \nabla_x L(x,\lambda,\mu) \\ h(x) \\ \phi(-g(x),\lambda) \end{pmatrix}$$

and

$$\phi(-g(x), \lambda) = (\varphi(-g_1(x), \lambda_1), ..., \varphi(-g_m(x), \lambda_m))^T \in \mathbb{R}^m,$$

where $\varphi : \mathbb{R}^2 \to \mathbb{R}$ is the minimum function

$$\varphi(a, b) = \min\{a, b\}.$$

Prove that the matrix $\nabla\Phi(x^*, \lambda^*, \mu^*)$ is well-defined and regular. (4 points)

52. Implement the *Lagrange-Newton algorithm* in MATLAB. Use

```
function [LN, Iter] = lagrange_newton(func, x0, μ0, kmax, epsilon)
```

as first line. Here, `func` denotes a function handle, `x0` and $\mu0$ denote the starting vectors, `kmax` the maximal number of allowed iterations, and `epsilon` the parameter for the stopping criterion. A matrix $LN = [x0, \mu0, x1, \mu1, x2, \mu2, ...]$ containing the iteration history and the number of performed iterations `Iter` should be returned.

The implemented algorithm should be tested on the following functions, starting values, and parameters:

(a) $f(x1, x_2) = 2x_1^4 + x_2^4 + 4x_1^2 - x_1x_2 + 6x_2^2$, $h(x_1, x_2) = 2x_1 - x_2 + 4$ with `x0`$=(0, 0)^T$, $\mu0 = 0$, `kmax`=200 and `epsilon`$=10^{-3}$.

(b) $f(x_1, x_2, x_3) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3$, $h_1(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2 - 25$, $h_2(x_1, x_2, x_3) = 8x_1 + 14x_2 + 7x_3 - 56$ with `x0`$=(3, 0.2, 3)^T$, $\mu0 = (0, 0)^T$, `kmax`=200 and `epsilon`$=10^{-5}$.

(5 points)

The generated files should be sent by January 31, 2020 to `radu.bot@univie.ac.at`.