PANM 22

Ladislav Lukšan; Ctirad Matonoha; Jan Vlček
Nonsmooth equation method for nonlinear nonconvex optimization

In: Jan Chleboun and Jan Papež and Karel Segeth and Jakub Šístek and Tomáš Vejchodský (eds.): Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar. Hejnice, June 23-28, 2024. , Prague, 2025. pp. 105–114.

Persistent URL: http://dml.cz/dmlcz/703219

Programs and Algorithms of Numerical Mathematics 22
J. Chleboun, J. Papež, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics CAS, Prague 2025

NONSMOOTH EQUATION METHOD FOR NONLINEAR NONCONVEX OPTIMIZATION

Ladislav Lukšan, Ctirad Matonoha, Jan Vlček

Institute of Computer Science, Czech Academy of Sciences,
Pod vodárenskou věží 2, 182 00 Prague 8, Czech Republic
luksan@cs.cas.cz, matonoha@cs.cas.cz

Abstract: The contribution deals with the description of two nonsmooth equation methods for inequality constrained mathematical programming problems. Three algorithms are presented and their efficiency is demonstrated by numerical experiments.

Keywords: nonlinear programming, nonsmooth analysis, semismooth equations, KKT systems, algorithms

MSC: 90C30, 49M05, 49M37, 65K05

1. Introduction

In this contribution, we are concerned with a nonlinear programming problem (NP): Find the minimum of a function $F(x)$ on the set given by constraints $c(x) \le 0$, where $F\colon R^n \to R$, $c\colon R^n \to R^m$ are twice continuously differentiable mappings ($c(x) \le 0$ is considered by elements).

Necessary conditions (the KKT conditions) for the solution of problem (NP) (if the gradients of active constraints are linearly independent) have the following form

$$g(x,u) = 0, \quad c(x) \le 0, \quad u \ge 0, \quad UC(x)e = 0, \tag{1}$$

where

$$g(x,u) = \nabla F(x) + \sum_{k=1}^{m} u_k \nabla c_k(x) = \nabla F(x) + A(x)u$$

and $A(x) = [\nabla c_k(x)\colon 1 \le k \le m]$. Here $u \in R^m$ are the vectors of Lagrange multipliers, $U = \mathrm{diag}(u_k\colon 1 \le k \le m)$, $C(x) = \mathrm{diag}(c_k(x)\colon 1 \le k \le m)$ and e is the vector with unit elements.

Nonlinear programming problems are frequently solved by three types of methods:

DOI: 10.21136/panm.2024.10

- Sequential quadratic programming (SQP) methods: In this case, the quadratic programming subproblem

$$\text{Minimize} \quad Q(d) = \frac{1}{2} d^T B d + g^T d, \quad \text{where} \quad A^T d + c \leq 0,$$

is solved in every iteration.

- Interior points (IP) methods: In this case, we solve the sequence of equality constrained problems

$$\text{Minimize} \quad F(x) - \mu e^T \log(S) e, \quad \text{where} \quad c(x) + s = 0,$$

where $S = \operatorname{diag}(s_k \colon 1 \leq k \leq m) > 0$ and $\mu \to 0$. The constraints $s \geq 0$ are satisfied algorithmically using the bounds for stepsizes.

- Nonsmooth equation (NE) methods: In this case, we solve the equality constrained problem

$$\text{Minimize} \quad F(x), \quad \text{where} \quad h(x, u) = 0,$$

in every iteration. The set of equations $h(x, u) = 0$ is usually nonsmooth.

SQP methods require an efficient solution of the quadratic programming subproblem. In the large scale case it usually consumes a large computational time. IP and NE methods, which transform inequality constrained problems to equality constrained ones, are very efficient.

2. Nonsmooth equation methods

Inequalities in (1), so called complementarity conditions, can be transformed to equations using the Fischer-Burmeister function [2]

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

which is zero if and only if $a \geq 0$, $b \geq 0$ and $ab = 0$. The Fischer-Burmeister function $\psi(a, b)$ is continuously differentiable if $|a| + |b| \neq 0$ and semismooth if $|a| + |b| = 0$. Moreover, function $\psi^2(a, b)$ is continuously differentiable everywhere. The gradient and the Clarke subdifferential of the Fischer-Burmeister function are given by the formulas

$$\nabla \psi(a, b) = \begin{bmatrix} \frac{a}{\sqrt{a^2+b^2}} - 1 \\ \frac{b}{\sqrt{a^2+b^2}} - 1 \end{bmatrix}, \quad |a| + |b| \neq 0, \tag{2}$$

$$\partial \psi(0, 0) = \operatorname{conv} \bigcup_{\phi \in [0, 2\pi]} \begin{bmatrix} \cos\phi - 1 \\ \sin\phi - 1 \end{bmatrix}. \tag{3}$$

Formula (3) implies that $[-1,-1]^T \in \partial\psi(0,0)$. Therefore, setting $r(a,b)=\sqrt{a^2+b^2}$ for $|a|+|b|\neq 0$ and $r(a,b)=1$ for $|a|+|b|=0$ we obtain

$$\left[\begin{array}{c} \frac{a}{r(a,b)}-1 \\ \frac{b}{r(a,b)}-1 \end{array}\right] \in \partial\psi(a,b). \tag{4}$$

Complementarity conditions in (1) are satisfied if and only if $\psi(u_k,-c_k(x))=0$, $1\le k\le m$, so (1) can be replaced by the system of nonlinear equations

$$f(z)=f(x,u)=\left[\begin{array}{c} g(x,u) \\ h(x,u) \end{array}\right]=0, \tag{5}$$

where $h(x,u)=[\psi(u_k,-c_k(x)) : 1\le k\le m]^T$. The mapping $f(z)$ is semismooth at every point $z\in R^{n+m}$. Therefore

$$f'(z,d)=Jd+o(\|d\|) \quad \text{if} \quad \|d\|\to 0 \quad \text{and} \quad J\in\partial f(z+d)$$

and

$$f(z+d)-f(z)=f'(z,d)+o(\|d\|)=Jd+o(\|d\|). \tag{6}$$

Linearizing system (5) by using (6), we obtain a step of the Newton method

$$x_+=x+d_x, \qquad u_+=u+d_u,$$

where

$$\left[\begin{array}{cc} B & A \\ (R+C)R^{-1}A^T & -(R-U)R^{-1} \end{array}\right]\left[\begin{array}{c} d_x \\ d_u \end{array}\right]=-\left[\begin{array}{c} g(x,u) \\ h(x,u) \end{array}\right], \tag{7}$$

and where

$$B\approx G(x,u)=\nabla^2F(x)+\sum_{k=1}^m u_k\nabla^2c_k(x),$$

$A=A(x)$, $C=C(x)$, $U=U(x)$, $R=\mathrm{diag}(r_k : 1\le k\le m)$, $r_k=\sqrt{c_k(x)^2+u_k^2}$.

The algorithm of a nonsmooth equation method can be roughly described in the following way. For given vectors $x\in R^n$, $u\in R^m$ we determine direction vectors d_x, d_u by solving a linear system equivalent to (7). Furthermore, we choose new vectors x_{i+1}, u_{i+1} by using a suitable merit function (Section 4) or by using a combined filter (Section 5).

3. Determination of a direction vector

Linear system (7) is not suitable for iterative solvers in general since it is nonsymmetric and can have unsuitable diagonal elements. A symmetric linear system can be obtained by multiplying the second row of (7) by the matrix $(R+C)^{-1}R$. Then

$$\left[\begin{array}{cc} B & A \\ A^T & -M \end{array}\right]\left[\begin{array}{c} d_x \\ d_u \end{array}\right]=-\left[\begin{array}{c} g(x,u) \\ (R+C)^{-1}R\,h(x,u) \end{array}\right],$$

where $M = (R+C)^{-1}(R-U)$ is a diagonal positive definite matrix. Diagonal elements of M can be very large in general. Therefore, we eliminate direction vectors corresponding to inactive constraints.

Definition 1. *A constraint with index k is active if*

$$-\frac{\partial \psi_k}{\partial u_k} \le \hat{\varepsilon}\, \frac{\partial \psi_k}{\partial c_k} \qquad \Longleftrightarrow \qquad r_k - u_k \le \hat{\varepsilon}(r_k + c_k),$$

where $\psi_k = \psi(u_k, -c_k)$ and $\hat{\varepsilon} > 0$ (usually $0.01 \le \hat{\varepsilon} \le 1$). Active quantities are denoted by $\hat{c}_k$, $\hat{u}_k$, $\hat{r}_k$, $\hat{M}$ and inactive quantities are denoted by $\check{c}_k$, $\check{u}_k$, $\check{r}_k$, $\check{M}$.

Eliminating inactive directions we obtain

$$\check{d}_u = \check{M}^{-1}(\check{A}^T d_x + \check{c}) - \check{u}, \tag{8}$$

$$\begin{bmatrix} \hat{B} & \hat{A} \\ \hat{A}^T & -\hat{M} \end{bmatrix} \begin{bmatrix} d_x \\ \hat{d}_u \end{bmatrix} + \begin{bmatrix} \hat{g}(x,u) \\ (\hat{R}+\hat{C})^{-1}\hat{R}\hat{h}(x,u) \end{bmatrix} = \begin{bmatrix} r_x \\ \hat{r}_u \end{bmatrix}, \tag{9}$$

where

$$\hat{B} = B + \check{A}\check{M}^{-1}\check{A}^T, \qquad \hat{g}(x,u) = g(x,u) + \check{A}\check{M}^{-1}\check{c}.$$

To obtain direction vectors d_x, $\hat{d}_u$, we solve linear equations (9) with sufficient precisions r_x, $\hat{r}_u$ and compute $\check{d}_u$ by (8). Note that $\|\hat{M}\| \le \hat{\varepsilon}$, $\|\check{M}^{-1}\| < 1/\hat{\varepsilon}$ and

$$\|\hat{M}\| \to 0, \quad \|\check{M}^{-1}\| \to 0 \quad \text{if} \quad \hat{g}(x.u) \to 0, \quad \hat{h}(x,u) \to 0.$$

Symmetric matrix $\hat{B}$ has a bounded norm and is positive definite if B is positive definite. For this reason we use a positive definite matrix $B = G + E$ obtained by using the Gill-Murray decomposition [3] of $G = G(x,u)$ (B is positive definite if it is obtained by the quasi-Newton method).

Nonsmooth equation methods for nonlinear programming problems are realized by the following algorithm.

Algorithm 1. *Line search method.*

Data: *Parameter for active constraint determination $\hat{\varepsilon}$. Precisions $0 < \overline{\omega}_x < 1$, $0 < \overline{\omega}_u < 1$. Maximum stepsize $\overline{\Delta} > 0$.*

Input: *Initial approximation of a KKT point x.*

Step 1: Initiation. *Choose initial Lagrange multipliers u_k, $1 \le k \le m$, such that $u_k \ne 0$. Compute value $F(x)$ and vector $c(x)$. If a filter is used, set $n_F = 1$ and $\mathcal{F} = \{F(x), \Phi(x,u)\}$. Set $i := 0$.*

Step 2: Termination. *Compute matrix $A := A(x)$ and vector $g := g(x,u)$. If (8) holds with a required precision, terminate computation, else set $i := i + 1$.*

Step 3: Hessian matrix approximation. *Determine positive definite matrix B as an approximation of the Hessian matrix $G(x,u)$.*

Step 4: Determination of direction vectors. *Divide constraints into active and inactive parts using parameter $\hat{\varepsilon}$ to obtain system (9). Determine vectors d_x, $\hat{d}_u$ as approximate solutions of (9) with precisions r_x, $\hat{r}_u$ and compute vector $\check{d}_u$ by (8). If a merit function is used, determine value $\sigma \geq 0$ by (12) and compute derivative $\varphi'(0)$ by (11).*

Step 5: Stepsize selection. *Determine stepsize $t > 0$ using Algorithm 2 or Algorithm 3 and set $x := x + td_x$, $u := u + td_u$. Compute value $F(x)$, vector $c(x)$ and go to Step 2.*

4. Line search with a merit function

After obtaining direction vectors d_x, d_u, we seek a stepsize t to decrease the value of the merit function

$$\varphi(t) = F_j(x + td_x) + \sigma P_j(x + td_x, u + td_u), \quad \sigma \geq 0, \quad j = 1, 2,$$

where

$$\begin{aligned}
F_1(x + td_x) &= F(x + td_x), \\
F_2(x + td_x) &= F(x + td_x) + (u + d_u)^T c(x + td_x), \\
P_1(x + td_x, u + td_u) &= \|h(x + td_x, u + td_u)\|_1, \\
P_2(x + td_x, u + td_u) &= \frac{1}{2}\|h(x + td_x, u + td_u)\|^2.
\end{aligned}$$

It is necessary that $\varphi'(0) < 0$ holds and that the stepsize t satisfies the Armijo condition

$$\varphi(t) - \varphi(0) \leq \varepsilon_1 t \varphi'(0), \quad \text{where} \quad 0 < \varepsilon_1 < 1/2. \tag{10}$$

For subsequent investigations, we use the notation

$$\begin{aligned}
F_1 : &\quad \chi(r) = d_x^T r_x - (\hat{u} + \hat{d}_u)^T \hat{r}_u, \\
F_1 : &\quad \gamma_0 = (u + d_u)^T M (u + d_u) - (u + d_u)^T c, \\
P_1 : &\quad \gamma_1 = \|h\|_1 - \|(\hat{R} + \hat{C})\hat{R}^{-1}\hat{r}_u\|_1,
\end{aligned}$$

$$\begin{aligned}
F_2 : &\quad \chi(r) = d_x^T r_x, \\
F_2 : &\quad \gamma_0 = 0, \\
P_2 : &\quad \gamma_1 = \|h\|^2 - \hat{h}^T(\hat{R} + \hat{C})\hat{R}^{-1}\hat{r}_u.
\end{aligned}$$

It is necessary that $\gamma_1 > 0$ holds, which is satisfied if

$$P_1 : \ \|\hat{r}_u\|_1 \leq \frac{\overline{\omega}_u}{2}\|h\|_1, \quad P_2 : \ \|\hat{r}_u\| \leq \frac{\overline{\omega}_u}{2}\|h\|, \quad \text{where} \quad 0 \leq \overline{\omega}_u < 1.$$

Theorem 1 ([4]). *Let vectors d_x, $\hat{d}_u$ be obtained as an approximate solution of* (9) *and vector $\check{d}_u$ be obtained by* (8). *Then*

$$\varphi'(0) = -d_x^T B d_x - \gamma_0 - \gamma_1 \sigma + \chi(r). \tag{11}$$

If $\gamma_1 > 0$,

$$\sigma \geq \underline{\sigma} > -\frac{d_x^T B d_x + \gamma_0}{\gamma_1}, \tag{12}$$

and if system (9) *is solved with the precision*

$$\chi(r) < d_x^T B d_x + \gamma_0 + \gamma_1 \sigma, \tag{13}$$

then $\varphi'(0) < 0$.

Algorithm 2. *Line search with a merit function.*

Data: *Parameters $0 < \beta < 1$, $0 < \varepsilon_1 < 1/2$, minimum stepsize $0 < \underline{t} < 1$. Derivative $\varphi'(0)$ obtained from* (11)

Input: *Pair (x, u), values $F(x)$, $c(x)$ and direction pair (d_x, d_u) obtained as a solution of equations* (8)–(9).

Step 1: *Choose initial stepsize $t > 0$ (usually $t = 1$). If $\varphi'(0) \geq 0$ go to Step* 5.

Step 2: *If $t < \underline{t}$, go to Step* 5, *else compute new values $F(x+td_x)$ and $c_k(x+td_x)$, $1 \leq k \leq m$.*

Step 3: Minimization of the objective function. *If the Armijo condition* (7) *is satisfied, go to Step* 6.

Step 4: *Set $t := \beta t$ and go to Step* 2.

Step 5: Restart. *Choose well positive diagonal matrix D (usually $D = I$). Solve precisely equations* (8)–(9) *with B replaced by D. Set $\sigma = 0$ and compute derivative $\varphi'(0) < 0$ from* (11). *Find stepsize $0 < t < 1$ such that $F(x+td_x) < F(x)$.*

Step 6: *Terminate stepsize selection ($t > 0$ is an obtained stepsize).*

The line search methods with a merit function are very efficient, namely if we use the Lagrangian function $F_2(x, u)$ and if the penalty parameter can decrease. Unfortunately, in this case the global convergence cannot be proved.

5. Line search with a filter

Denote for simplicity $z = (x, u)$, $\Phi(z) = (1/2)\|h(x, u)\|^2$ and $g(z) = g(x, u)$. At the same time, although F does not depend on u, let for consistency $F(z) = F(x)$.

Definition 2. *Let $F(z_1) \leq F(z_2)$ and $\Phi(z_1) \leq \Phi(z_2)$. Then we say that the pair $(F(z_2), \Phi(z_2))$ is dominated by the pair $(F(z_1), \Phi(z_1))$. A filter $\mathcal{F} = \{(F_j, \Phi_j) \colon 1 \leq j \leq n_F\}$ is a set of pairs where no pair is dominated by another pair (n_F is a number of pairs in the filter).*

The line search with a filter procedure uses three strategies for obtaining new trial points. If $t < \underline{t}$, where $\underline{t} > 0$ is a computed lower bound, we use a feasibility restoration phase. In this case, we determine a new vector $d_z \in R^{n+m}$ and a suitable stepsize $t > 0$ by minimizing $\Phi(z)$ to satisfy (17). If $t \geq \underline{t}$, we first check whether

$$F(z + td_z) < F_j \quad \text{or} \quad \Phi(z + td_z) < \Phi_j \tag{14}$$

holds for $1 \leq j \leq n_F$ (otherwise, the stepsize is shortened). If $t \geq \underline{t}$ and

$$d_z^T \nabla F(z) < 0, \qquad -d_z^T \nabla F(z) t > \delta_3 \Phi^\nu(z), \tag{15}$$

where $\delta_3 > 0$ a $\nu > 1$, the stepsize selection is terminated if

$$F(z + td_z) - F(z) \leq \varepsilon_1 t d_z^T \nabla F(z), \tag{16}$$

where $0 < \varepsilon_1 < 1/2$ (the Armijo condition). If $t \geq \underline{t}$ and (15) does not hold, the stepsize selection is terminated if

$$F(z + td_z) < F(z) - \delta_1 \Phi(z) \quad \text{or} \quad \Phi(z + td_z) < \Phi(z) - \delta_2 \Phi(z), \tag{17}$$

where $0 < \delta_1 < 1$ and $0 < \delta_2 < 1$ (the filter condition).

Algorithm 3. *Line search with a filter.*

Data: *Parameters* $0 < \beta < 1$, $0 < \varepsilon_1 < 1/2$, $0 < \delta_1 < 1$, $0 < \delta_2 < 1$, $\delta_3 > 0$, $0 < \delta_4 < 1$, *size of filter* $n_F \geq 1$, *maximum size of filter* $m_F > 1$, *filter* $\mathcal{F} = \{(F_j, \Phi_j) : 1 \leq j \leq n_F\}$ *(usually* $n_F = 1$ *and* $\mathcal{F} = \{F(z), \Phi(z)\}$*).*

Input: *Pair* $z = (x, u)$, *values* $F(z)$, $\Phi(z)$ *and direction vector* $d_z = (d_x, d_u)$ *obtained as a solution of equations* (8)–(9).

Step 1: *Compute minimum stepsize* $\underline{t} > 0$ *by* (18). *Choose initial stepsize* $t > 0$ *(usually* $t = 1$*).*

Step 2: *If* $t < \underline{t}$, *go to Step* 6. *If* $t \geq \underline{t}$, *compute new values* $F := F(z + td_z)$ *and* $\Phi := \Phi(z + td_z)$. *If* $(F, \Phi) \in \mathcal{F}$ *(i.e.,* (14) *does not hold), go to Step* 5.

Step 3: Minimization of the objective function. *If* (15) *holds and Armijo condition* (16) *is satisfied, go to Step* 8. *If* (15) *holds and Armijo condition* (16) *is not satisfied, go to Step* 5.

Step 4: Utilization of the filter. *If* (15) *does not hold and condition* (17) *is satisfied, go to Step* 7. *If* (15) *does not hold and condition* (17) *is not satisfied, go to Step* 5.

Step 5: *Set* $t := \beta t$ *and go to Step* 2.

Step 6: Feasibility restoration. *Find a new direction vector* d_z *and a suitable stepsize* $t > 0$ *in such a way that the values* $F := F(z + td_z)$, $\Phi := \Phi(z + td_z)$ *satisfy conditions* $(F, \Phi) \notin \mathcal{F}$ *and* $\Phi < \Phi(z) - \delta_2' \Phi(z)$, *where* $\delta_2' > 0$.

Step 7: Filter update. *Compute values* $F = F(z) - \delta_1 \Phi(z)$, $\Phi = \Phi(z) - \delta_2 \Phi(z)$. *Remove from the filter pairs* (F_j, Φ_j) *dominated by* (F, Φ) *and add* (F, Φ) *into the filter.*

Step 8: *Terminate stepsize selection* ($t > 0$ *is an obtained stepsize).*

The minimum stepsize is computed by the rule [8]

$$\begin{aligned} \underline{t} &= \delta_4 \min\left(\varepsilon_0, \frac{\delta_1 \Phi(z)}{|d_z^T \nabla F(z)|}, \frac{\delta_3 \Phi^\nu(z)}{|d_z^T \nabla F(z)|}\right), & d_z^T \nabla F(z) < 0, \\ \underline{t} &= \delta_4 \varepsilon_0, & d_z^T \nabla F(z) \geq 0, \end{aligned} \tag{18}$$

where $0 < \delta_4 < 1$.

The line search method with a filter is globally convergent (i.e, the process, started from an arbitrary point, converges to the KKT point) if the following standard assumptions are satisfied:

- Functions $F(x)$ and $c_k(x)$, $1 \leq k \leq m$, are twice continuously differentiable. Function values and derivatives are uniformly bounded.
- Matrices appearing in (9) are uniformly nonsingular.
- Matrices B in (7) are uniformly bounded and uniformly positive definite.
- Conditions $|u_k| + |c_k(x)| \geq \varepsilon$ (strict complementarity) and $r_k + c_k(x) \neq 0$ are satisfied.

Theorem 2 ([8])**.** *Consider a nonsmooth equation line search method realized by Algorithm 1 and Algorithm 2. If standard assumptions for global convergence are satisfied, then* $\|h(z)\| \to 0$.

Theorem 3 ([7])**.** *Consider a nonsmooth equation method, where equations (8)–(9) are solved with the precisions*

$$d_x^T r_x \leq \overline{\omega}_x d_x^T \hat{B} d_x, \qquad \|\hat{r}_u\| \leq \overline{\omega}_u \|\hat{c}(x)\|,$$

where $0 \leq \overline{\omega}_x < 1$, $0 \leq \overline{\omega}_u < 1$. *Let the stepsizes be determined by Algorithm 3. If standard assumptions for global convergence are satisfied, then the method is globally convergent.*

6. Computational experiments

The computational comparisons were preformed using the system for universal functional optimization UFO [5] on the collection of test problems TEST21. This collection contains 18 problems with 1000 variables and is a modification of the collection TEST20 described in [6]. The comparisons were made using the performance

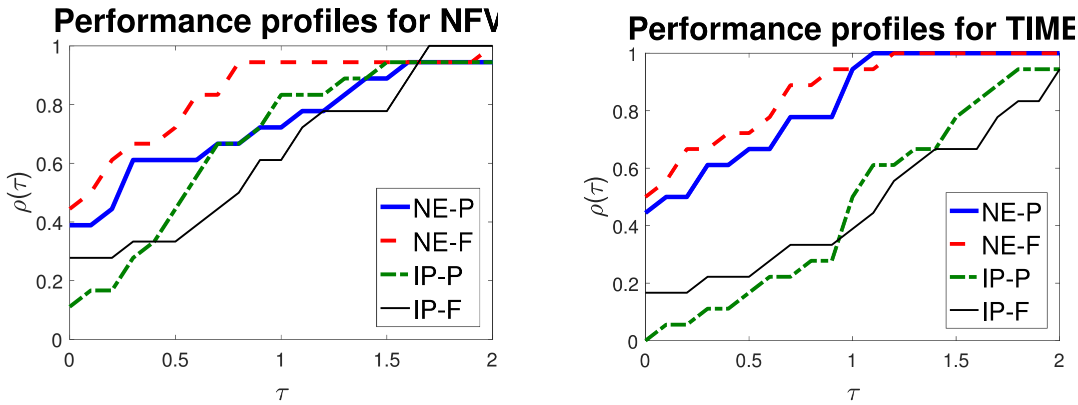

Figure 1: Comparison of Newton's methods.

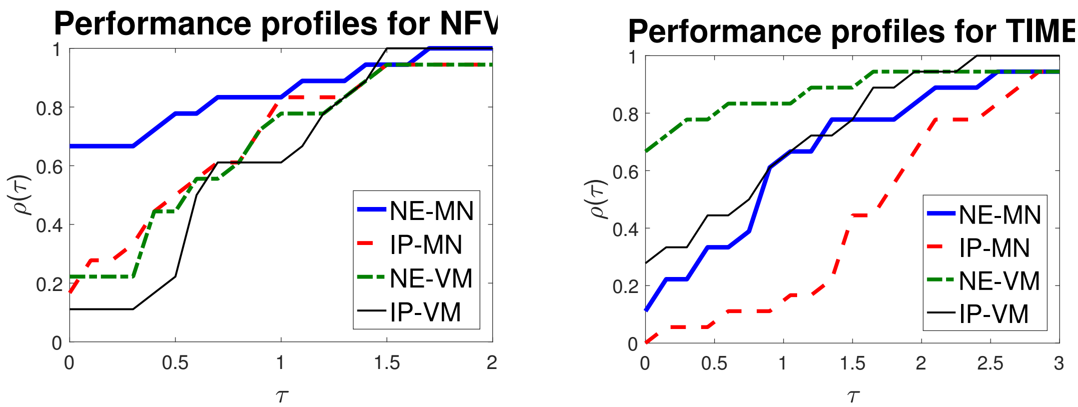

Figure 2: Comparison of variable metric methods.

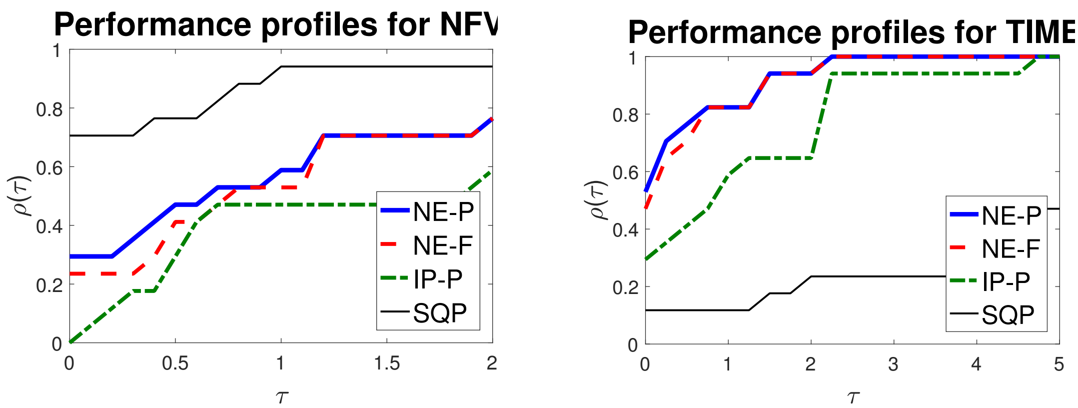

Figure 3: Comparison with the sequential quadratic programming method (SQP).

profiles for the number of function evaluations (NFV) and for the total computational time (TIME). The details about performance profiles as well as the meaning of τ and $\rho(\tau)$ used in Figures 1–3 can be found in [1]. The following notation is used:

NE	– nonsmooth equation methods,	IP	– interior point methods,
P	– merit function,	F	– filter,
MN	– Newton methods,	VM	– variable metric methods.

Acknowledgements

This work was supported by the long-term strategic development financing of the Institute of Computer Science (RVO:67985807).

References

[1] Dolan, E. D. and Moré, J. J.: Benchmarking optimization software with performance profiles. Math. Program. **91** (2002), 201–213.

[2] Fischer, A.: A special Newton-type optimization method. Optimization **24** (1992), 269–284.

[3] Gill, P. E. and Murray, W.: Newton type methods for unconstrained and linearly constrained optimization. Math. Program. **7** (1974), 311–350.

[4] Lukšan, L., Matonoha, C., and Vlček, J.: Nonsmooth equation method for nonlinear nonconvex optimization. In: M. Křížek, P. Neittaanmäki, R. Glovinski, S. Korotov (eds.), *Conjugate Gradient Algorithms and Finite Element Methods*, pp. 131–145. Springer Verlag, Berlin, 2004.

[5] Lukšan, L., Tůma, M., Matonoha, C., Vlček J., Ramešová, N., Šiška, M., and Hartman, J.: UFO 2024. Interactive System for Universal Functional Optimization. Technical Report V-1289. Prague, ICS AS CR, 2024.

[6] Lukšan, L. and Vlček, J.: Sparse and partially separable test problems for unconstrained and equality constrained optimization. Technical Report V-767. Prague, ICS AS CR 1999.

[7] Stuchlý, J.: Řešení rozsáhlých úloh nelineárního programování metodami vnitřního bodu. Diplomová práce, Univerzita Karlova, Praha 2004.

[8] Wächter, A. and Biegler, L. T.: Line search filter methods for nonlinear programming - motivation and global convergence. SIAM J. on Computation **16** (2005), 1–31.