# Algorithm 811: NDA: Algorithms for Nondifferentiable Optimization

LADISLAV LUKŠAN and JAN VLČEK
Academy of Sciences of the Czech Republic

We present four basic Fortran subroutines for nondifferentiable optimization with simple bounds and general linear constraints. Subroutine PMIN, intended for minimax optimization, is based on a sequential quadratic programming variable metric algorithm. Subroutines PBUN and PNEW, intended for general nonsmooth problems, are based on bundle-type methods. Subroutine PVAR is based on special nonsmooth variable metric methods. Besides the description of methods and codes, we propose computational experiments which demonstrate the efficiency of this approach.

Categories and Subject Descriptors: D.3.2 [**Programming Languages**]: Language Classifications—*FORTRAN 77*; G.1.6 [**Numerical Analysis**]: Optimization; G.4 [**Mathematics of Computing**]: Mathematical Software

General Terms: Algorithms

Additional Key Words and Phrases: Minimax optimization, discrete Chebyshev approximation, sequential quadratic programming methods, variable metric methods, general linear constraints

## 1. INTRODUCTION

We propose four basic subroutines which implement selected nondifferentiable optimization algorithms. The efficiency of these subroutines is demonstrated in Lukšan and Vlček [1998; 1999] and Vlček and Lukšan [2001], by comparing them with similar (usually not free) codes known from literature.

The double-precision Fortran 77 subroutine PMIN is designed to find a close approximation to a local minimum of a special minimax objective function

$$F(x) = \max_{1 \le i \le n_a} f_i(x).$$

This work was supported under grant No. 201/00/0080 given by the Czech Republic Grant Agency.
Authors' address: Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 2, Prague 8, 182 07, Czech Republic; email: {luksan; vlcek}@cs.cas.cz.

Here $x \in R^n$ is a vector of $n$ variables, and $f_i : R^n \to R$, $1 \le i \le n_a$, are twice continuously differentiable functions. Subroutine PMIN is based on the sequential quadratic programming variable metric method described in Lukšan [1984] (see also Han [1981] and Pschenychny [1983] for theoretical foundation).

The double-precision Fortran 77 subroutines PBUN, PNEW, and PVAR are designed to find a close approximation to a local minimum of a nonlinear nonsmooth function $f(x)$. Here $x \in R^n$ is a vector of $n$ variables and function $f : R^n \to R$, assumed to be Lipschitz continuous. We assume for each $x \in R^n$ that we can compute $f(x)$, an arbitrary subgradient $g(x)$, i.e., one element of the subdifferential $\partial f(x)$ (called the generalized gradient in Clarke [1983]). Subroutine PBUN is based on the proximal bundle method described in Vlček [1995] (see also Kiwiel [1985], Lemarechal [1989], Lemarechal and Zowe [1994], Mäkelä and Neittaanmäki [1992], and Mifflin [1977; 1982] for theoretical foundation), which only uses first-order information. Subroutine PNEW is based on the bundle-Newton method described in Lukšan and Vlček [1998], which uses second-order information as well, i.e., an $n \times n$ symmetric matrix $G(x)$ as a substitute for the Hessian matrix. Subroutine PVAR is based on variable metric methods described in Lukšan and Vlček [1999] and Vlček and Lukšan [2001] which use variable metric updates for obtaining an approximation of the Hessian matrix.

All the above subroutines allow us to work with simple bounds and general linear constraints. Simple bounds are assumed in the form

$$x_i - \text{unbounded}, I_i^x = 0,$$

$$x_i^l \le x_i, I_i^x = 1,$$

$$x_i \le x_i^u, I_i^x = 2,$$

$$x_i^l \le x_i \le x_i^u, I_i^x = 3,$$

$$x_i = x_i^l = x_i^u, I_i^x = 5,$$

where $1 \le i \le n$. General linear constraints are assumed in the form

$$a_i^T x - \text{unbounded}, I_i^c = 0,$$

$$c_i^l \le a_i^T x, I_i^c = 1,$$

$$a_i^T x \le c_i^u, I_i^c = 2,$$

$$c_i^l \le a_i^T x \le c_i^u, I_i^c = 3,$$

$$a_i^T x = c_i^l = c_i^u, I_i^c = 5,$$

where $1 \leq i \leq n_c$ and where $n_c$ is the number of general linear constraints ($I^x$, $I^c$ correspond to arays IX, IC in the subroutines).

To simplify the user's work, additional easy-to-use subroutines are added. These subroutines call general subroutines PMIN, PBUN, PNEW, PVAR:

PMINU unconstrained minimax optimization.

PMINS minimax optimization with simple bounds.

PMINL minimax optimization with simple bounds and general linear constraints.

PBUNU, PNEWU, PVARU unconstrained nonsmooth optimization.

PBUNS, PNEWS, PVARS nonsmooth optimization with simple bounds.

PBUNL, PNEWL, PVARL nonsmooth optimization with simple bounds and general linear constraints.

Each subroutine contains a description of formal parameters and extensive comments. Moreover, text files PMIN.TXT, PBUN.TXT, PNEW.TXT, PVAR.TXT are added, which contain a detailed description of all important subroutines (including indications of required storage). Finally, test programs TMINU, TMINL, TBUNU, TBUNL, TNEWU, TNEWL, TVARU, TVARL are included, which contain sets of test problems. These test programs serve as examples for using the subroutines, verify their correctness, and demonstrate their efficiency.

## 2. SEQUENTIAL QUADRATIC PROGRAMMING METHODS FOR NONLINEAR MINIMAX OPTIMIZATION

To simplify the description of the method, we consider the particular linearly constrained problem written in the following form

$$x^\star = \arg \min_{x \in L_C} \{\max_{i \in M_1} f_i(x)\}, \tag{1}$$

where

$$L_C = \{x \in R^n : a_i^T x \leq b_i,\, i \in M_2\}$$

with $M_1 \cap M_2 = \emptyset$ and $M_1 \cup M_2 = M = \{1, \ldots, m\}$ ($L_C$ is a feasible set determined by linear constraints). It is clear that the application of the method described below to the problem with general linear constraints stated in Section 1 is straightforward, but this requires considering each type of constraints separately as is realized in subroutine PMIN.

### 2.1 Variable Metric Method for Nonlinear Minimax Optimization

If we introduce a new variable $u$, then the problem (1) can be reformulated as a nonlinear programming problem

$$(x^\star, u^\star) = \arg \min_{(x,\, u) \in N^{n+1}} \{u\}, \tag{2}$$

where

$$N^{n+1} = \{(x, u) \in R^{n+1} : f_i(x) \le e_i u,\ i \in M\}$$

with $e_i = 1$ for $i \in M_1$ and $e_i = 0$, $f_i(x) = a_i^T x - b_i$ for $i \in M_2$. This nonlinear programming problem can be solved by a sequential quadratic programming method that uses a quadratic approximation of the Lagrangian and a linear approximation of constraints in each iteration. Let $x^k \in R^n$ be a current approximation to the minimizer $x^\star$. Then the resulting quadratic programming subproblem has the form

$$(d^k, u^k) = \arg \min_{(d,u) \in L_k^{n+1}} \left\{ \frac{1}{2} d^T G^k d + u \right\}, \tag{3}$$

where $G^k$ is an approximation of the Hessian matrix of the Lagrangian and

$$L_k^{n+1} = \{(d, u) \in R^{n+1} : f_i^k + (a_i^k)^T d \le e_i u,\ i \in M\}$$

with $f_i^k = f_i(x^k)$, $a_i^k = \nabla f_i(x^k)$ for $i \in M_1$ and $f_i^k = a_i^T x^k - b_i$, $a_i^k = a_i$ for $i \in M_2$. The solution of the quadratic programming subproblem (3) has to satisfy the Karush-Kuhn-Tucker conditions

$$\begin{aligned} d^k &= -H^k g^k, \\ e^T \lambda^k &= 1, \\ \lambda^k &\ge 0, \\ \mu^k &\ge 0, \\ (\mu^k)^T \lambda^k &= 0, \end{aligned}$$

where $\lambda^k$ is the vector of Lagrange multipliers, $H^k = (G^k)^{-1}$, $A^k = [a_1^k, \ldots, a_m^k]$, $e = [e_1, \ldots, e_m]^T$, $f^k = [f_1^k, \ldots, f_m^k]^T$, $g^k = A^k \lambda^k$ is the gradient of the Lagrangian function, and $\mu^k = u^k e - f^k - (A^k)^T d^k$ is the vector of constraint violations. Note that if $g^k = 0$ then we obtain the Karush-Kuhn-Tucker conditions for the nonlinear programming problem (2) exactly so that the minimax problem (1) is solved if each $f_i$ is convex. Therefore, the condition $\|g^k\|_\infty \le$ `TOLG` is used in subroutine `PMIN` as the basic stopping criterion (when it is fulfilled, then `ITERM` = 4).

The direction vector $d^k \in R^n$ obtained as the solution to the quadratic programming subproblem (3) is used for the definition of the new approximation $x^{k+1}$ to the minimizer $x^\star$ by the formula

$$x^{k+1} = x^k + t^k d^k,$$

where $0 < t^k \le 1$ is a step length, which is chosen in such a way that

$$F(x^k + t^k d^k) - F(x^k) \le m_L t^k (d^k)^T g^k,$$

where $0 < m_L < 1/2$ is a tolerance for function decrease in the line search (parameter TOLS in subroutine PMIN). The step length $t^k$ is chosen iteratively either by bisection (MES = 1), or by a two-point quadratic interpolation (MES = 2), or by a three-point quadratic interpolation (MES = 3), or by a three-point cubic interpolation (MES = 4) (MES is a parameter of subroutine PMIN).

Having the new approximation $x^{k+1}$ to the minimum $x^\star$, we can compute the new matrix $A^{k+1} = [a_1^{k+1}, \ldots, a_m^{k+1}]$, where $a_i^{k+1} = \nabla f_i(x^{k+1})$ for $i \in M_1$ and $a_i^{k+1} = a_i$ for $i \in M_2$. If we denote $s^k = x^{k+1} - x^k$ and $z^k = A^{k+1}\lambda^k - A^k\lambda^k = A^{k+1}\lambda^k - g^k$, then the BFGS method [Fletcher 1987] consists in the following update

$$G^{k+1} = \frac{1}{\gamma^k}\left(G^k + \gamma^k \frac{z^k(z^k)^T}{(s^k)^T z^k} - \frac{G^k s^k (G^k s^k)^T}{(s^k)^T G^k s^k}\right)$$

$$= \frac{1}{\gamma^k}\left(G^k + \gamma^k \frac{z^k(z^k)^T}{(s^k)^T z^k} + t^k \frac{g^k (g^k)^T}{(s^k)^T g^k}\right),$$

where $\gamma^k > 0$ is a self-scaling parameter. This parameter is equal to one if MET = 1 (MET is a parameter of subroutine PMIN). If MET = 2, the special value $\gamma^k = (s^k)^T G^k s^k / (s^k)^T z^k = -t^k (s^k)^T g^k / (s^k)^T z^k$ is used in the first iteration (or in the iteration after a restart). If MET = 3, this special value is also used whenever it lies in the interval $[0.5, 4.0]$. The BFGS method requires condition $(s^k)^T z^k > 0$ to be satisfied, which guarantees a positive definiteness of matrix $G^{k+1}$. Unfortunately, this condition does not hold in the minimax optimization automatically. If $(s^k)^T z^k \le 0$ and MEC = 1, we set either $G^{k+1} = G^k$ if MER = 0 or $G^{k+1} = I$ if MER = 1. If $(s^k)^T z^k \le 0$ and MEC = 2, we use the Powell correction [Powell 1977] (MEC and MER are parameters of subroutine PMIN). At the first iteration, we set $H^1 = I$ (the unit matrix). This setting is also performed if $-(d^k)^T g^k \le \underline{\varepsilon} \|d^k\| \|g^k\|$, where $\underline{\varepsilon}$ is a restart tolerance (parameter TOLD of the subroutine PMIN).

### 2.2 Dual Range Space Method for a Special Quadratic Programming Subproblem

Consider a quadratic programming problem in which we seek a pair $(d^\star, u^\star) \in R^{n+1}$ in such a way that

$$(d^\star, u^\star) = \arg \min_{(d,u) \in L^{n+1}} \phi(s, u), \qquad (4)$$

where

$$\phi(d, u) = \frac{1}{2} d^T G d + u$$

and

$$L^{n+1} = \{(d, u) \in R^{n+1} : f_i + a_i^T d \leq e_i u,\ i \in M\}$$

(see (3)). The assumption that the matrix $G$ is positive definite implies that problem (4) is convex and that we can apply a duality theory to obtain a dual quadratic programming problem which consists in seeking a vector $\lambda^\star \in R^m$ (vector of Lagrange multipliers of (4)) so that

$$\psi(\lambda^\star) = \min_{\lambda \in L_D} \psi(\lambda), \tag{5}$$

where

$$\psi(\lambda) = \frac{1}{2} \lambda^T A^T H A \lambda - f^T \lambda$$

and

$$L_D = \{\lambda \in R^m : e^T \lambda = 1,\ \lambda \geq 0\}.$$

Here $H = G^{-1}$,

$$A = [a_1, \ldots, a_m],\ f = [f_1, \ldots, f_m]^T,\ e = [e_1, \ldots, e_m]^T.$$

The solution of (4) can be obtained from the solution of (5) by the formulas

$$d^\star = -HA\lambda^\star \tag{6}$$

and

$$u^\star = f^T \lambda^\star - (\lambda^\star)^T A^T H A \lambda^\star. \tag{7}$$

The solution $\lambda^\star$ of (5) is the optimal Lagrange multiplier vector of (4). Since problem (5) is convex, $\lambda^\star$ is its solution if and only if the Karush-Kuhn-Tucker conditions are valid, i.e., if and only if

$$e^T \lambda^\star = 1,\ \lambda^\star \geq 0, \tag{8}$$

and there exists a scalar $u^\star$ such that

$$\mu^\star = A^T H A \lambda^\star - f + u^\star e \geq 0,\ (\mu^\star)^T \lambda^\star = 0. \tag{9}$$

Vector $\mu^\star$ is the Lagrange multiplier vector of problem (5). Conditions (6) and (9) imply that $u^\star$ in (9) is identical with $u^\star$ in (7). This in turn implies that $\mu^\star$ is, at the same time, the vector of constraint values of problem (4).

Consider any subset $I \subset M$, and denote the vectors of elements $\lambda_i$, $f_i$, $e_i$, $i \in I$ by $\lambda$, $f$, $e$, respectively. Similarly, let $A$ be the matrix of columns $a_i$, $i \in I$. To simplify the investigation of the dual range space method, we denote

$$\tilde{A} = \begin{bmatrix} A \\ -e^T \end{bmatrix}, \tilde{H} = \begin{bmatrix} H & 0 \\ 0 & 1 \end{bmatrix}$$

and assume that the subset $I \subset M$ was chosen in such a way that the columns of $\tilde{A}$ are linearly independent.

If $I = I^*$ were the set of active constraints at the solution of problem (4), then we could compute the dual variables $u^\star$ and $\lambda^\star$ from (8)–(9). Unfortunately, this set is not known a priori. Therefore, we start with the set $I = \{k\}$, where $k \in M_1$ is arbitrary. Then $u = f_k - a_k^T H a_k$ and $\lambda = [1]$. Suppose that $I \subset M$ is a current subset and $u$, $\lambda$ are corresponding dual variables. Then we can proceed in the following way. First we compute the direction vector $d = -HA\lambda$ and the value of the most violated primal constraint

$$\mu_k = ue_k - f_k - a_k^T d = \min_{i \in M \setminus I} \{ue_i - f_i - a_i^T d\}.$$

If $\mu_k \geq 0$ then the set of active constraints has been detected and the solutions of (4) and (5) have been found. Otherwise, we set $\lambda_k = 0$ and compute the primal and dual step lengths

$$t_k^P = -\frac{\mu_k}{\beta_k \gamma_k + \delta_k}$$

$$t_k^D = \frac{\lambda_j}{q_{kj} + \gamma_k p_j} = \min_{i \in \bar{I}} \frac{\lambda_i}{q_{ki} + \gamma_k p_i},$$

where $p = (\tilde{A}^T \tilde{H} \tilde{A})^{-1} e$, $q_k = (\tilde{A}^T \tilde{H} \tilde{A})^{-1} \tilde{A}^T \tilde{H} \tilde{a}_k$, $\beta_k = e_k - e^T q_k$, $\gamma_k = \beta_k / p^T e$, $\delta_k = \tilde{a}_k^T (\tilde{H} - \tilde{H}\tilde{A}(\tilde{A}^T \tilde{H} \tilde{A})^{-1} \tilde{A}^T \tilde{H}) \tilde{a}_k$ (with $\tilde{a}_k = [a_k, -e_k]^T$) and $\bar{I} = \{i \in I : q_{ki} + \gamma_k p_i > 0\}$. If $\beta_k \gamma_k + \delta_k = 0$ then we set $t_k^P = \infty$. If $\bar{I} = \emptyset$, we set $t_k^D = \infty$. If simultaneously $t_k^P = \infty$ and $t_k^D = \infty$, the problem has no feasible solution. Otherwise we set $t_k = \min\{t_k^P, t_k^D\}$ and compute $u := u + t_k \gamma_k$, $\lambda := \lambda - t_k(q_k + \gamma_k p)$, $\lambda_k := \lambda_k + t_k$, $\mu_k := (1 - t_k / t_k^P)\mu_k$.

If $t_k^P \leq t_k^D$, then the primal step is realized, i.e., we set $I := I \cup \{k\}$, $\lambda := [\lambda^T, \lambda_k]^T$, $e := [e^T, e_k]^T$, $A := [A, a_k]$, $\tilde{A} := [\tilde{A}, \tilde{a}_k]$, recompute $d = -HA\lambda$, and determine a new value of the most violated primal constraint and a new index $k$.

If $t_k^P > t_k^D$, then the dual step is realized, i.e., we set $I := I \backslash \{j\}$, $\lambda := \lambda^{(j)}$, $e := e^{(j)}$, $A := A^{(j)}$, $\tilde{A} := \tilde{A}^{(j)}$, where the upper index in parentheses denotes an element or column which is deleted. Now, two cases can occur. If $I \cap M_1 \neq \emptyset$, we recompute the primal and dual step lengths and repeat the process with the same index $k$. If $I \cap M_1 = \emptyset$, then we compute $u := u - \mu_k$, set $I := I \cup \{k\}$, $\lambda := [\lambda^T, \lambda_k]^T$, $e := [e^T, e_k]^T$, $A := [A, a_k]$, $\tilde{A} := [\tilde{A}, \tilde{a}_k]$, recompute $d = -HA\lambda$, and determine the new value of the most violated primal constraint and the new index $k$.

In Lukšan [1985] it has been proved that the above dual range space method finds solutions of quadratic programming problems (4) and (5) after a finite number of steps. This method is also used for solving quadratic programming subproblems in the bundle-type methods described in the next section.

## 3. BUNDLE-TYPE METHODS FOR NONSMOOTH OPTIMIZATION

To simplify the description of the method, we consider the particular linearly constrained problem written in the following form

$$x^\star = \arg\min_{x \in L_C} \{f(x)\}, \tag{10}$$

where

$$L_C = \{x \in R^n : a_j^T x \leq b_j, j \in M_2\}.$$

It is clear that the application of the methods described below to the problem with general linear constraints stated in Section 1 is straightforward, but this requires considering each type of constraint separately as is realized in subroutines PBUN and PNEW.

The idea behind the bundle methods is that they use a bundle of information obtained at points $y^j \in L_C$, $j \in J_k$, where $J_k \subset \{1, \ldots, k\}$. The bundle of information serves for building a simple nonsmooth model which is utilized for direction determination. Having the direction vector $d \in R^n$, a special line search procedure which produces either serious or short or null steps is used in such a way that

$$x^{k+1} = x^k + t_L^k d^k, y^{k+1} = x^k + t_R^k d^k, \tag{11}$$

where $0 \leq t_L^k \leq t_R^k \leq 1$. Serious steps, characterized by the relation $t_R^k = t_L^k$, i.e., $y^{k+1} = x^{k+1}$, are typical for classical optimization methods. For nonsmooth minimization, special null steps are essential. Both short and null steps with $t_R^k \neq t_L^k$, i.e., $y^{k+1} \neq x^{k+1}$, obtain bundle information from a larger domain which can include points lying on the opposite sides of a possible discontinuity of the objective gradient. The difference between the bundle methods described below consists in the choice of a nonsmooth model. The proximal bundle method uses a piecewise linear function with a special quadratic penalty term while the bundle-Newton method uses a piecewise quadratic function.

Notice that bundle methods can only handle locally Lipschitz, weakly upper semismooth objectives.

### 3.1 The Proximal Bundle Method

The piecewise linear function used in the proximal bundle method is based on the cutting-plane model

$$\hat{f}_k(x) = \max_{j \in J_k} \{f(y^j) + (g^j)^T(x - y^j)\} = \max_{j \in J_k} \{f(x^k) + (g^j)^T(x - x^k) - \beta_j^k\},$$

where $g^j \in \partial f(y^j)$, $j \in J_k$, are subgradients and $\beta_j^k = f(x^k) - f(y^j) - (g^j)^T(x^k - y^j)$, $j \in J_k$, are linearization errors. If the objective function were convex, then the cutting plane model would underestimate it, i.e., $\hat{f}_k(x) \leq f(x)$ for all $x \in L_C$. This is not valid in general, since $\beta_j^k$ may be negative in a nonconvex case. Therefore, the linearization error $\beta_j^k$ is replaced by the so-called subgradient locality measure

$$\alpha_j^k = \max\{|\beta_j^k|, \gamma(s_j^k)^\omega\}, \tag{12}$$

where

$$s_j^k = \|x^j - y^j\| + \sum_{i=j}^{k-1} \|x^{i+1} - x^i\|$$

is the distance measure approximating $\|x^k - y^j\|$ without the need to store the bundle point $y^j$, $\gamma \geq 0$ is the distance measure parameter (parameter ETA of subroutine PBUN), and $\omega$ is the distance measure exponent (equal to 2 in subroutine PBUN). We can set $\gamma = 0$ in the convex case. Obviously, now $\min_{L_C} \hat{f}_k \leq f(x^k)$ from $\alpha_j^k \geq 0$ and $x^k \in L_C$. In order to respect the above considerations, we can define the following local subproblem for the direction determination

$$d^k = \arg \min_{x^k + d \in L_C} \left\{\hat{f}_k(x^k + d) + \frac{1}{2}\sigma^k d^T d\right\},$$

where the regularizing quadratic penalty term $(1/2)\sigma^k d^T d$ is added to guarantee the existence of the solution $d^k$ and to keep the approximation local enough.

The choice of weights $\sigma^k$ is very important. Weights which are too large imply a small $\|d^k\|$, almost all serious steps and a slow descent. Weights which are too small imply a large $\|d^k\|$ and many null steps. The weight updating method depends on the parameter `MOT` of subroutine `PBUN`:

—Quadratic interpolation (`MOT` = 1): The idea is based on a simplified case $n = 1$ and $f$ quadratic, where $\sigma^k$ estimates the second-order derivative of $f$ (see Kiwiel [1990]). By letting

$$\sigma^{k+1} = \min\{\max\{\sigma_{int}^{k+1}, \sigma^k/10, \sigma_{min}\}, 1/\sigma_{min}, 10\sigma^k\},$$

where $\sigma_{min}$ is a small positive constant, we safeguard the value $\sigma_{int}^{k+1}$ obtained by quadratic interpolation (see Vlček [1995] for details).

—Minimum localization (`MOT` = 2): The quadratic interpolation is not suitable for $f$ of the polyhedral type. Since the second-order derivative of the single-variable quadratic function $ax^2 + bx + c$, $b$ fixed, is inversely proportional to the coordinate of the minimum, we set $\sigma_{loc}^{k+1} = \sigma^k/x_{min}$, where $x_{min}$ is a computed estimate of the minimum of $f$ in the direction $d^k$. We again safeguard $\sigma_{loc}^{k+1}$ similarly as $\sigma_{int}^{k+1}$.

—Quasi-Newton condition (`MOT` = 3): If we approximate the Hessian matrix of $f$ by $\sigma_{con}^{k+1} \cdot I$, then the quasi-Newton condition with aggregate subgradient $g_0^{k+1}$ (see below) can be written in the form $\sigma_{con}^{k+1}\|d^k\|^2 = (d^k)^T(g_0^{k+1} - g_0^k)$. We safeguard $\sigma_{con}^{k+1}$ by setting $\sigma^{k+1} = \min\{\max\{\sigma_{con}^{k+1}, 10^{-3}\}, 10^3\}$.

The above local subproblem is still a nonsmooth optimization problem. However, due to the piecewise linear nature it can be rewritten as a (smooth) quadratic programming subproblem

$$(d^k, u^k) = \arg\min_{(d,u)\in L_k} \left\{u + \frac{1}{2}\sigma^k d^T d\right\}, \tag{13}$$

where

$$L_k = \{(d, u) : -\alpha_j^k + (g^j)^T d \le e_j u, j \in J_k \cup M_2\}$$

with $\alpha_j^k$ given by (12), $g^j \in \partial f(y^j)$, $e_j = 1$ for $j \in J_k$, and $\alpha_j^k = b_j - a_j^T x$, $g^j = a_j$, $e_j = 0$ for $j \in M_2$ (we suppose that $J_k \cap M_2 = \emptyset$, which is easily ensured in our implementation). This quadratic programming subproblem can be efficiently solved by the dual range space method described in Section 2.2.

The above derivation was slightly simplified, since the aggregation of constraints was not included. In fact we add the element $\{0\}$ to $J_k$, letting

$$\tilde{f}_0^{k-1} = \sum_{j\in J_{k-1}\setminus\{0\}} \lambda_j^{k-1}(f(y^j) + (g^j)^T(x^{k-1} - y^j)) + \lambda_0^{k-1} f_0^{k-1},$$

$$\tilde{s}_0^{k-1} = \sum_{j \in J_{k-1}} \lambda_j^{k-1} s_j^{k-1},$$

$$f_0^k = \tilde{f}_0^{k-1} + (g_0^k)^T (x^k - x^{k-1}),$$

$$s_0^k = \tilde{s}_0^{k-1} + |x^k - x^{k-1}|,$$

$$g_0^k = \sum_{j \in J_{k-1} \setminus \{0\}} \lambda_j^{k-1} g^j + \lambda_0^{k-1} g_0^{k-1},$$

$$\alpha_0^k = \max\{|f(x^k) - f_0^k|, \gamma(s_0^k)^\omega\}$$

and $e_0 = 1$, $f_0^1 = f(x^1)$, $s_0^1 = 0$, $g_0^1 = g^1$. The values $\lambda_j^{k-1}$, $j \in J_{k-1}$ are Lagrange multipliers of the quadratic programming subproblem from the previous iteration.

Having the pair $(d^k, u^k)$ determined as a solution to the quadratic programming subproblem (13), we can obtain points (11) using a suitable line search (such a line search is guaranteed to be finite only in the weakly upper semismooth case). The line search consists in the initial setting $t_L^k = 0$ and the construction of the sequence $t_i^k > 0$, $i \in N$ ($N$ is the set of natural numbers), $t_1^k = 1$, using an interpolation method (bisection if `MES` $= 1$ or two-point quadratic interpolation if `MES` $= 2$, where `MES` is the parameter of subroutines `PBUN` and `PNEW`) and suitable backtracking. Let $0 < m_L < 1/2$, $m_L < m_R < 1$ be line search tolerances (parameters `TOLS` and `TOLP` in subroutines `PBUN` and `PNEW`) and $0 < \underline{t} < 1$. If

$$f(x^k + t_i^k d^k) \leq f(x^k) + m_L t_i^k v^k, \tag{14}$$

where $v^k = u^k + \sum_{j \in J_k} \lambda_j^k \alpha_j^k - \tilde{\alpha}_0^k$, $\tilde{\alpha}_0^k = \max\{|\tilde{f}_0^k - f(x^k)|, \gamma(\tilde{s}_0^k)^\omega\}$, then we set $t_L^k = t_i^k$. If $t_L^k \geq \underline{t}$, then we set $t_R^k = t_L^k$ and terminate the line search (serious step). Otherwise, if

$$-\alpha_{k+1}^{k+1} + (g^{k+1})^T d^k \geq m_R v^k, \tag{15}$$

where

$$\alpha_{k+1}^{k+1} = \max\{|\beta_{k+1}^{k+1}|, \gamma(s_{k+1}^{k+1})^\omega\},$$

$$\beta_{k+1}^{k+1} = f(x^k + t_L^k d^k) - f(x^k + t_i^k d^k) - (t_L^k - t_i^k)(g^{k+1})^T d^k,$$

$$s_{k+1}^{k+1} = \|(t_L^k - t_i^k) d^k\|$$

and $g^{k+1} \in \partial f(x^k + t_i^k d^k)$, then we set $t_R^k = t_i^k$ and terminate the line search. Otherwise, the line search continues with $i$ increased by 1.

The iteration is terminated if $w^k \leq$ `TOLG`, where $w^k = (1/2)|g_0^k|^2 + \tilde{\alpha}_0^k$ (`TOLG` is a parameter of subroutines `PBUN` and `PNEW`).

### 3.2 The Bundle-Newton Method

The bundle-Newton method is based on the following piecewise quadratic model

$$\tilde{f}_k(x) = \max_{j \in J_k} \left\{ f(y^j) + (g^j)^T(x - y^j) + \frac{1}{2}\rho^j(x - y^j)^T G^j (x - y^j) \right\}$$

$$= \max_{j \in J_k} \left\{ f(x^k) + (g_j^k)^T(x - x^k) + \frac{1}{2}\rho^j(x - x^k)^T G^j (x - x^k) - \beta_j^k \right\},$$

where $G^j$ are symmetric positive definite matrices, $\rho^j \in [0, 1]$ are values defined below, $g_j^k = g^j + \rho^j G^j(x^k - y^j)$, and

$$\beta_j^k = f(x^k) - f(y^j) - (g^j)^T(x^k - y^j) - \frac{1}{2}\rho^j(x^k - y^j)^T G^j(x^k - y^j)$$

for $j \in J_k$. Note that even in the convex case $\beta_j^k$ might be negative. Therefore, we replace the error $\beta_j^k$ by the locality measure (12) again so that $\min_{L_C} \tilde{f}_k \le f(x^k)$. But $\gamma > 0$ is now required for the distance measure parameter (parameter `ETA` of the subroutines `PNEW`), and $\omega$ can be set to 1 or 2 (parameter `MOS` of the subroutines `PNEW`). The local subproblem for direction determination has the form

$$d^k = \arg \min_{x^k + d \in L_C} \{\tilde{f}_k(x^k + d)\}.$$

This local subproblem is in fact a nonlinear minimax problem which can be solved approximately by the Lagrange-Newton method (see Fletcher [1987]). Thus, we solve the following (smooth) quadratic programming subproblem

$$(d^k, v^k) = \arg \min_{(d, v) \in L_k} \left\{ v + \frac{1}{2} d^T W^k d \right\}, \tag{16}$$

where $W^k = \sum_{j \in J_{k-1}} \lambda_j^{k-1} \rho^j G^j$ and $\lambda_j^{k-1}, j \in J_{k-1}$ are Lagrange multipliers of the quadratic programming subproblem from the previous iteration and

$$L_k = \{(d, v) : -\alpha_j^k + (g_j^k)^T d \le e_j v, j \in J_k \cup M_2\}$$

with $\alpha_j^k$ given by (12), $g_j^k = g^j + \rho^j G^j(x^k - y^j)$, $e_j = 1$ for $j \in J_k$, and $\alpha_j^k = b_j - a_j^T x$, $g_j^k = a_j$, $e_j = 0$ for $j \in M_2$. This quadratic programming subproblem can be efficiently solved by the dual range space method described in Section 2.2.

The above derivation is not full, since the aggregation of constraints is not included. The aggregation of constraints is based on the same principle that was used in the proximal bundle method. We refer to Lukšan and Vlček [1998] for details.

Having the pair $(d^k, v^k)$ determined as a solution to the quadratic programming subproblem (16), we can obtain the points (11) using a line search which is in fact the same as in the proximal bundle method. Again, conditions (14) and (15) are used, where

$$\alpha_{k+1}^{k+1} = \max\{|\beta_{k+1}^{k+1}|, \gamma(s_{k+1}^{k+1})^{\omega}\},$$

$$\beta_{k+1}^{k+1} = f(x^k + t_L^k d^k) - f(x^k + t_i^k d^k) - (t_L^k - t_i^k)(g_{k+1}^{k+1})^T d^k$$

$$- (\rho^{k+1}/2)(t_L^k - t_i^k)^2 (d^k)^T G(x^k + t_i^k d^k) d^k,$$

$$s_{k+1}^{k+1} = \|(t_L^k - t_i^k) d^k\|$$

and where $g^{k+1}$ is replaced by $g_{k+1}^{k+1} = g(x^k + t_i^k d^k) + \rho^{k+1}(t_L^k - t_i^k) G(x^k + t_i^k d^k) d^k$. Here $g(x^k + t_i^k d^k) \in \partial f(x^k + t_i^k d^k)$, and $G(x^k + t_i^k d^k)$ is a second-order matrix computed at the point $x^k + t_i^k d^k$. The stopping criterion is in fact the same as in the proximal bundle method.

The damping parameters $\rho^j$, $j \in J_k$, have unit values on most iterations and are zeroed if many short or null steps occur, since a quadratic model is inefficient in this case.

## 4. VARIABLE METRIC METHODS FOR NONSMOOTH OPTIMIZATION

The main deficiency of standard bundle methods is the necessity of solving a rather extensive QP subproblem in every iteration, which is a time-consuming procedure. On the other hand, standard variable metric methods are relatively robust and efficient when they are applied to nonsmooth convex problems (e.g., see Lemarechal [1978]). This fact indicates that special nonsmooth modifications of variable metric methods, not containing time-consuming operations, could be developed. Roughly speaking, three basic ideas of bundle methods can be applied to variable metric methods for improving their efficiency and robustness. The essential feature is the utilization of null steps for obtaining sufficient information about a nondifferentiable function. Furthermore, a simple aggregation of subgradients and application of modified linearization errors are used that guarantee convergence of subgradients to zero and allow us to evaluate a termination criterion.

### 4.1 Globally Convergent Variable Metric Methods

Globally convergent variable metric methods based on these ideas are proposed in Lukšan and Vlček [1999] and Vlček and Lukšan [2001]. These methods, which utilize a simple three-term aggregation at null steps can be described by the following simplified procedure (we consider the unconstrained case in this subsection).

Starting with $x^1$, $f(x^1)$, $g^1 \in \partial f(x^1)$, $H^1$ positive definite (e.g., $H^1 = I$), $\tilde{g}_1 = g^1$, $\tilde{\alpha}_1 = 0$, the $k$th iteration begins by testing whether matrix $H^k$ is sufficiently positive definite (if not, correction $\varrho^k I$, $\varrho^k > 0$ is added to $H^k$). Then the determination of the direction vector $d^k = -H^k \tilde{g}_k$ and the computation of the stationarity measure $w^k = (\tilde{g}^k)^T H^k \tilde{g}^k + 2\tilde{\alpha}^k$ follow. If $w^k \leq$ TOLG (TOLG is a parameter of subroutine PVAR), then $x^k$ is a good approximation of a stationary point. Otherwise, a step length $t^k$ is selected, e.g., using a piecewise linear approximation (moreover, in the nonconvex case [Vlček and Lukšan 2001], a special line-search procedure is used), together with $y^{k+1} = x^k + t^k d^k$, $f(y^{k+1})$ and $g^{k+1} \in \partial f(y^{k+1})$. Let $0 < m_L < 1/2$ be a line search tolerance (we use the value $m_L = 10^{-4}$ in subroutine PVAR). If

$$f(y^{k+1}) - f(x^k) \leq -m_L t^k w^k \tag{17}$$

(descent step), then we set $x^{k+1} = y^{k+1}$, $\tilde{g}^{k+1} = g^{k+1}$, compute $s^k = x^{k+1} - x^k$, $z^k = g^{k+1} - g^m$, where $m$ is the index of the iteration after the latest serious step, and determine $H^{k+1}$ from $H^k$ by the BFGS update [Fletcher 1987]

$$H^{k+1} = H^k + \left(1 + \frac{(z^k)^T H^k z^k}{(z^k)^T s^k}\right)\frac{s^k (s^k)^T}{(z^k)^T s^k} - \frac{H^k z^k (s^k)^T + s^k (z^k)^T H^k}{(z^k)^T s^k}$$

finishing the $k$th iteration. If (17) is not satisfied (null step), then we set $x^{k+1} = x^k$, compute $\alpha^{k+1} = (f(x^k) - f(y^{k+1}))/t^k + (d^k)^T g^{k+1}$ in the convex case [Lukšan and Vlček 1999] or $\alpha^{k+1} = \max[(f(x^k) - f(y^{k+1})) + t^k (d^k)^T g^{k+1}, \gamma |t^k d^k|^{\omega}]$, $\gamma > 0$, $\omega \geq 1$ in the nonconvex case [Vlček and Lukšan 2001], determine multipliers $\lambda_j^k \geq 0$, $j \in \{1, 2, 3\}$, $\lambda_1^k + \lambda_2^k + \lambda_3^k = 1$, which minimize the function

$$\varphi(\lambda_1, \lambda_2, \lambda_3) = |\lambda_1 W^k g^m + \lambda_2 W^k g^{k+1} + \lambda_3 W^k \tilde{g}^k|^2 + 2[\lambda_2 \alpha^{k+1} + \lambda_3 \tilde{\alpha}^k],$$

where $W^k = (H^k)^{1/2}$, and set

$$\tilde{g}^{k+1} = \lambda_1^k g^m + \lambda_2^k g^{k+1} + \lambda_3^k \tilde{g}^k, \; \tilde{\alpha}^{k+1} = \lambda_2^k \alpha^{k+1} + \lambda_3^k \tilde{\alpha}^k.$$

After this simple aggregation we compute $s^k = y^{k+1} - x^k$, $z^k = g^{k+1} - g^m$. If $(\tilde{g}^k)^T (s^k - H^k z^k) > 0$, then we construct $H^{k+1}$ from $H^k$ by the SR1 update [Fletcher 1987]

$$H^{k+1} = H^k + \frac{(s^k - H^k z^k)(s^k - H^k z^k)^T}{(z^k)^T (s^k - H^k z^k)}$$

finishing the $k$th iteration.

More details concerning these methods are given in Lukšan and Vlček [1999] and Vlček and Lukšan [2001].

### 4.2 Active Set Strategy for Linear Constraints

Since variable metric methods do not use quadratic programing subalgorithms, which allow us to handle linear constraints automatically, we have to apply the active set strategy to these methods directly (see Panier [1987]). To simplify the description of active set strategy, we again consider the particular problem (10). Suppose $x^k \in R^n$ is a feasible point so that $a_i^T x^k = b_i$, $i \in I_k$ and $a_i^T x^k < b_i$, $i \in M_2 \backslash I_k$, where $I_k \subset M_2$ is a set of indices of active constraints. Then we can restrict to minimization on the manifold

$$L_k = \{x : (A^k)^T x = b^k\} = \{x : x^k + Z^k \hat{s}\} \tag{18}$$

where $A^k = [a_i]_{i \in I_k}$, $(b^k)^T = [b_i]_{i \in I_k}$, $(A^k)^T Z^k = 0$, $(Z^k)^T Z^k = I$ and Range$([A^k, Z^k]) = R^n$. Considering this reduced problem, we can easily see that the reduced subgradient and the reduced Hessian matrix are given by $\hat{g}(\hat{s}) = (Z^k)^T g(x)$ and $\hat{G}(\hat{s}) = (Z^k)^T G(x) Z^k$, respectively. If we have an approximation $\hat{H}^k$ of $[(Z^k)^T G(x^k) Z^k]^{-1}$, we can improve it by a variable metric update where the vectors $z^k$ and $s^k$ are replaced by the reduced vectors $\hat{z}^k = (Z^k)^T z^k$ and $\hat{s}^k = t^k \hat{d}^k = -t^k \hat{H}^k \hat{g}^k$.

If the solution to the problem (10) lies on $L_k$, we can find it by using methods described in Section 4.1, remembering that all vectors have to be replaced by corresponding reduced vectors. If the solution to the problem (10) does not lie on $L_k$, the active constraints have to be changed subsequently. The test for constraint deletion at iteration $k$ employs the Lagrange multiplier vector $\lambda^k = ((A^k)^T A^k)^{-1} (A^k)^T \hat{g}^k$. Let $e_{i_k}^T \lambda^k = \min_{i \in I_k} e_i^T \lambda^k$. Then $i_k$ is deleted from $I_k$ if $e_{i_k}^T \lambda^k \le -c \|\hat{g}^k\|$, where $c > 0$ (parameter EPS in subroutine PVAR). On the other hand, constraint addition is performed at the end of the iteration. For this purpose, the upper bound

$$t_{max}^k = \min_{\substack{i \in M_2 \backslash I_k \\ a_i^T s^k > 0}} \frac{b_i - a_i^T x^k}{a_i^T s^k}$$

is determined, and the suitable stepsize $t^k \le t_{max}^k$ is found. After the variable metric update is carried out, all indices $i \in M_2 \backslash I_k$ satisfying $|b_i - a_i^T x^{k+1}| \le 10^{-8}$ are added to $I_k$.

If the set of active constraints is changed, then the representation of the corresponding linear manifold has to be updated, e.g., see Gill and Murray [1974]. To simplify the notation, we omit index $k$ in the rest of this section and denote by + and − quantities after addition and deletion of an active constraint, respectively. Then the current (active) linear manifold is represented by the set of constraint indices $I$, the matrix of constraint normals $A$, the upper triangular matrix $R$ satisfying $R^T R = A^T A$, and the orthonormal basis $Z$.

Table I. Results Obtained by Program TMINU

| Problem | NIT | NFV | NFG | F | G | ITERM |
|---|---|---|---|---|---|---|
| 1 | 7 | 8 | 8 | 0.19522245D+01 | 0.1041D−07 | 4 |
| 2 | 7 | 8 | 8 | 0.21607942D−09 | 0.1776D−13 | 4 |
| 3 | 93 | 180 | 94 | 0.25068549D−10 | 0.6509D−06 | 4 |
| 4 | 13 | 15 | 14 | 0.35997193D+01 | 0.2498D−07 | 4 |
| 5 | 11 | 16 | 12 | −0.44000000D+02 | 0.2507D−06 | 4 |
| 6 | 12 | 21 | 13 | −0.44000000D+02 | 0.8543D−06 | 4 |
| 7 | 8 | 9 | 9 | 0.42021427D−02 | 0.6561D−09 | 4 |
| 8 | 5 | 6 | 6 | 0.50816327D−01 | 0.1502D−06 | 4 |
| 9 | 10 | 12 | 11 | 0.80843684D−02 | 0.1874D−08 | 4 |
| 10 | 11 | 11 | 11 | 0.11570644D+03 | 0.7077D−08 | 4 |
| 11 | 35 | 113 | 36 | 0.26359735D−02 | 0.1488D−07 | 4 |
| 12 | 34 | 86 | 35 | 0.20160755D−02 | 0.1562D−08 | 4 |
| 13 | 7 | 8 | 8 | 0.99665144D−05 | 0.4525D−06 | 4 |
| 14 | 6 | 8 | 7 | 0.12237126D−03 | 0.6839D−07 | 4 |
| 15 | 17 | 57 | 17 | 0.22340496D−01 | 0.3261D−12 | 4 |
| 16 | 21 | 53 | 22 | 0.34904927D−01 | 0.2523D−07 | 4 |
| 17 | 11 | 16 | 12 | 0.19729062D+00 | 0.4825D−06 | 4 |
| 18 | 19 | 107 | 20 | 0.61852848D−02 | 0.1888D−06 | 4 |
| 19 | 19 | 45 | 20 | 0.68063006D+03 | 0.5743D−06 | 4 |
| 20 | 13 | 19 | 14 | 0.24306209D+02 | 0.9366D−07 | 4 |
| 21 | 19 | 30 | 20 | 0.13372828D+03 | 0.5547D−06 | 4 |
| 22 | 36 | 88 | 36 | 0.54598150D+02 | 0.1964D−05 | −6 |
| 23 | 2 | 38 | 22 | 0.26108258D+03 | 0.7934D−06 | 4 |
| 24 | 18 | 20 | 19 | 0.91153896D−07 | 0.5473D−06 | 4 |
| 25 | 67 | 286 | 68 | 0.48029699D−01 | 0.8649D−06 | 4 |

After addition of the constraint normal $a_+$ to matrix $A$, we obtain

$$R_+ = \begin{bmatrix} R & r_+ \\ 0 & \rho_+ \end{bmatrix},$$

where $R^T r_+ = A^T a_+$ and $\rho_+ = \|Z^T a_+\| = \sqrt{a_+^T a_+ - r_+^T r_+}$. Furthermore, let $P$ be the orthogonal matrix (we use a product of the Givens rotation matrices) such that $P^T Z^T a_+ = \|Z^T a_+\| e_1$, where $e_1$ is the first column of the unit matrix. Then $Z_+$ is obtained from $ZP$ by deletion of its first column and

$$\hat{H}_+ = \hat{H}_* - \frac{\hat{h}\hat{h}^T}{\hat{\eta}}$$

where

$$P^T \hat{H} P = \begin{bmatrix} \hat{H}_* & \hat{h} \\ \hat{h}^T & \hat{\eta} \end{bmatrix}.$$

After deletion of the constraint normal $a$ from matrix $A$, let $M$ be the permutation matrix which interchanges column $a$ with the last column of $A$

Table II. Results Obtained by Program TMINL

| Problem | NIT | NFV | NF | F | G | ITER |
|---|---|---|---|---|---|---|
| 1 | 6 | 7 | 7 | −0.38965952D+00 | 0.6129D−08 | 4 |
| 2 | 5 | 5 | 5 | −0.33035714D+00 | 0.2220D−15 | 4 |
| 3 | 8 | 8 | 8 | −0.44891079D+00 | 0.2032D−10 | 4 |
| 4 | 75 | 75 | 75 | −0.42928061D+00 | 0.4449D−10 | 4 |
| 5 | 9 | 9 | 9 | −0.18596187D+01 | 0.8299D−12 | 4 |
| 6 | 7 | 9 | 8 | 0.10183089D+00 | 0.8211D−06 | 4 |
| 7 | 7 | 10 | 8 | 0.71054274D−14 | 0.6600D−06 | 4 |
| 8 | 15 | 23 | 16 | 0.24306209D+02 | 0.3499D−06 | 4 |
| 9 | 23 | 37 | 24 | 0.13372828D+03 | 0.2859D−07 | 4 |
| 10 | 15 | 16 | 15 | 0.50694800D+00 | 0.1487D−10 | 4 |
| 11 | 15 | 20 | 16 | 0.27608128D−03 | 0.1610D−09 | 4 |
| 12 | 157 | 864 | 158 | −0.17688070D+04 | 0.3565D−07 | 4 |
| 13 | 15 | 22 | 16 | 0.12272261D+04 | 0.1706D−06 | 4 |
| 14 | 147 | 270 | 148 | 0.70492480D+04 | 0.1030D−06 | 4 |
| 15 | 65 | 109 | 65 | 0.17478699D+03 | 0.2174D−08 | 4 |

Table III. Results Obtained by Program TBUNU

| Problem | NIT | NFV | NFG | F | G | ITERM |
|---|---|---|---|---|---|---|
| 1 | 42 | 45 | 45 | 0.38117065D−06 | 0.1135D−02 | 2 |
| 2 | 18 | 20 | 20 | 0.29892889D−15 | 0.5439D−08 | 2 |
| 3 | 31 | 33 | 33 | 0.19522245D+01 | 0.3085D−03 | 2 |
| 4 | 14 | 16 | 16 | 0.20000000D+01 | 0.1921D−06 | 2 |
| 5 | 17 | 19 | 19 | −0.30000000D+01 | 0.5564D−08 | 4 |
| 6 | 13 | 15 | 15 | 0.72000015D+01 | 0.2212D−02 | 4 |
| 7 | 11 | 12 | 12 | −0.14142136D+01 | 0.1437D−04 | 4 |
| 8 | 66 | 68 | 68 | −0.99999941D+00 | 0.1089D−02 | 4 |
| 9 | 13 | 15 | 15 | −0.10000000D+01 | 0.9859D−07 | 4 |
| 10 | 43 | 46 | 46 | −0.80000000D+01 | 0.1282D−02 | 4 |
| 11 | 43 | 45 | 45 | −0.43999999D+02 | 0.3734D−02 | 2 |
| 12 | 27 | 29 | 29 | 0.22600162D+02 | 0.1451D−03 | 4 |
| 13 | 60 | 62 | 62 | −0.32348679D+02 | 0.2190D−02 | 2 |
| 14 | 115 | 116 | 116 | −0.29196928D+01 | 0.1318D−02 | 2 |
| 15 | 92 | 93 | 93 | 0.55981567D+00 | 0.8266D−03 | 4 |
| 16 | 74 | 75 | 75 | −0.84140829D+00 | 0.7236D−03 | 2 |
| 17 | 160 | 162 | 162 | 0.97857724D+01 | 0.3625D−03 | 2 |
| 18 | 143 | 157 | 157 | 0.16703852D+02 | 0.4138D−02 | 2 |
| 19 | 150 | 151 | 151 | 0.16712381D−06 | 0.7782D−04 | 2 |
| 20 | 39 | 40 | 40 | 0.27274665D−12 | 0.1250D+00 | 2 |

so that $AM = [A_{-}a]$ and so that the matrix $RM$ is upper Hessenberg. Let $Q$ be the orthogonal matrix (product of the Givens rotation matrices) which anihilates the subdiagonal elements of $RM$ so that

$$QRM = \begin{bmatrix} R_{-} & r \\ 0 & \rho \end{bmatrix}.$$

Then $R_{-}$ is a part of the upper triangular matrix $QRM$. Furthermore, $Z_{-} = [Zz_{-}]$ where

Table IV. Results Obtained by Program TBUNL

| Problem | NIT | NFV | NFG | F | G | ITERM |
|---|---|---|---|---|---|---|
| 1 | 10 | 11 | 11 | −0.38965952D+00 | 0.4532D−04 | 4 |
| 2 | 4 | 5 | 5 | −0.33035714D+00 | 0.3886D−14 | 4 |
| 3 | 8 | 10 | 10 | −0.44891079D+00 | 0.6982D−03 | 4 |
| 4 | 79 | 80 | 80 | −0.42928061D+00 | 0.7703D−05 | 2 |
| 5 | 16 | 17 | 17 | −0.18596138D+01 | 0.2017D−09 | 2 |
| 6 | 16 | 17 | 17 | 0.10183089D+00 | 0.3284D−05 | 2 |
| 7 | 43 | 44 | 44 | 0.28724453D−08 | 0.1966D−07 | 2 |
| 8 | 74 | 76 | 76 | 0.24306219D+02 | 0.5835D−02 | 4 |
| 9 | 140 | 143 | 143 | 0.13372840D+03 | 0.2872D−01 | 2 |
| 10 | 65 | 68 | 68 | 0.50694798D+00 | 0.3576D−04 | 2 |

Table V. Results Obtained by Program TNEWU

| Problem | NIT | NFV | NFG | F | G | ITERM |
|---|---|---|---|---|---|---|
| 1 | 58 | 59 | 59 | 0.22533111D−15 | 0.8624D−05 | 2 |
| 2 | 7 | 8 | 8 | 0.16765701D−10 | 0.5792D−05 | 4 |
| 3 | 9 | 10 | 10 | 0.19522245D+01 | 0.2172D−05 | 4 |
| 4 | 10 | 11 | 11 | 0.20000068D+01 | 0.2161D−04 | 4 |
| 5 | 14 | 15 | 15 | −0.30000000D+01 | 0.5398D−08 | 2 |
| 6 | 4 | 6 | 6 | 0.72000000D+01 | 0.1445D−08 | 4 |
| 7 | 16 | 17 | 17 | −0.14142136D+01 | 0.5653D−07 | 4 |
| 8 | 11 | 13 | 13 | −0.10000000D+01 | 0.4158D−07 | 4 |
| 9 | 10 | 11 | 11 | −0.10000000D+01 | 0.4562D−06 | 4 |
| 10 | 29 | 30 | 30 | −0.80000000D+01 | 0.1066D−04 | 4 |
| 11 | 13 | 15 | 15 | −0.44000000D+02 | 0.4215D−05 | 4 |
| 12 | 7 | 8 | 8 | 0.22600173D+02 | 0.1263D−02 | 4 |
| 13 | 22 | 24 | 24 | −0.32348679D+02 | 0.3409D−02 | 4 |
| 14 | 76 | 77 | 77 | −0.29197002D+01 | 0.1061D−02 | 4 |
| 15 | 89 | 91 | 91 | 0.55981330D+00 | 0.1528D−05 | 4 |
| 16 | 12 | 14 | 14 | −0.84140833D+00 | 0.6734D−06 | 4 |
| 17 | 52 | 53 | 53 | 0.97857721D+01 | 0.2964D−03 | 4 |
| 18 | 40 | 42 | 42 | 0.16703855D+02 | 0.1781D+00 | 4 |
| 19 | 36 | 37 | 37 | 0.38373702D−08 | 0.5758D−08 | 2 |
| 20 | 24 | 25 | 25 | 0.45289427D−08 | 0.1100D−09 | 2 |

$$z_- = AM\begin{bmatrix} R_- & r \\ 0 & \rho \end{bmatrix}^{-1}\begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Since the second-order information along the direction $z_-$ is not contained in matrix $\hat{H}$, we set

$$\hat{H}_- = \begin{bmatrix} \hat{H} & 0 \\ 0 & 1 \end{bmatrix}.$$

## 5. VERIFICATION OF SUBROUTINES

Tables I–VIII give the results obtained by using test programs TMINU, TMINL, TBUNU, TBUNL, TNEWU, TNEWL, TVARU, TVARL, which serve for demonstration, verification, and testing of subroutines PMINU, PMINL, PBUNU,

Table VI. Results Obtained by Program TNEWL

| Problem | NIT | NFV | NFG | F | G | ITERM |
|---|---|---|---|---|---|---|
| 1 | 6 | 7 | 7 | −0.38965952D+00 | 0.1650D−07 | 4 |
| 2 | 3 | 4 | 4 | −0.33035714D+00 | 0.6661D−15 | 4 |
| 3 | 49 | 50 | 50 | −0.44891079D+00 | 0.1014D−06 | 4 |
| 4 | 9 | 10 | 10 | −0.42928019D+00 | 0.2923D−04 | 4 |
| 5 | 28 | 29 | 29 | −0.18596187D+01 | 0.8903D−05 | 4 |
| 6 | 9 | 10 | 10 | 0.10183089D+00 | 0.1136D−06 | 4 |
| 7 | 54 | 55 | 55 | 0.28329335D−08 | 0.5279D−06 | 2 |
| 8 | 18 | 19 | 19 | 0.24306209D+02 | 0.5075D−03 | 4 |
| 9 | 47 | 49 | 49 | 0.13372920D+03 | 0.4180D−04 | 4 |
| 10 | 94 | 98 | 98 | 0.50694800D+00 | 0.4072D−06 | 4 |

Table VII. Results Obtained by Program TVARU

| Problem | NIT | NFV | NFG | F | G | ITERM |
|---|---|---|---|---|---|---|
| 1 | 34 | 34 | 34 | 0.27598804D−10 | 0.5517D−10 | 4 |
| 2 | 15 | 16 | 16 | 0.94894120D−10 | 0.4745D−10 | 4 |
| 3 | 17 | 17 | 17 | 0.19522247D+01 | 0.9348D−06 | 4 |
| 4 | 17 | 17 | 17 | 0.20000000D+01 | 0.2918D−07 | 4 |
| 5 | 20 | 20 | 20 | −0.29999997D+01 | 0.4258D−06 | 4 |
| 6 | 19 | 19 | 19 | 0.72000001D+01 | 0.1901D−06 | 4 |
| 7 | 10 | 10 | 10 | −0.14142133D+01 | 0.1414D−06 | 4 |
| 8 | 55 | 59 | 59 | −0.99999247D+00 | 0.4052D−06 | 4 |
| 9 | 37 | 37 | 37 | −0.99999979D+00 | 0.4851D−06 | 4 |
| 10 | 14 | 14 | 14 | −0.79999998D+01 | 0.4535D−06 | 4 |
| 11 | 38 | 38 | 38 | −0.43999999D+02 | 0.6473D−06 | 4 |
| 12 | 30 | 31 | 31 | 0.22600163D+02 | 0.6268D−06 | 4 |
| 13 | 46 | 47 | 47 | −0.32348678D+02 | 0.1001D−05 | 2 |
| 14 | 32 | 32 | 32 | −0.29197004D+01 | 0.3444D−06 | 4 |
| 15 | 74 | 76 | 76 | 0.55981553D+00 | 0.2796D−06 | 4 |
| 16 | 89 | 89 | 89 | −0.84140570D+00 | 0.3911D−06 | 4 |
| 17 | 158 | 158 | 158 | 0.97860064D+01 | 0.8896D−06 | 4 |
| 18 | 93 | 93 | 93 | 0.16703838D+02 | 0.3408D−06 | 4 |
| 19 | 123 | 123 | 123 | 0.14683216D−05 | 0.9099D−06 | 4 |
| 20 | 23 | 23 | 23 | 0.00000000D+00 | 0.3200D−22 | 4 |

PBUNL, PNEWU, PNEWL, PVARU, PVARL. In the tables, rows corresponding to individual test problems contain the number of iterations NIT, the number of function evaluations NFV, the number of gradient evaluations NFG, the final value of the objective function F, the value of the termination criterion G, and the cause of termination ITERM. All computations reported were performed on a Pentium PC computer, under the Windows 2000 system using the Digital Visual Fortran (Version 6) compiler, in double-precision arithmetic. All subroutines were checked with a Fortran verifier and implemented and tested on various UNIX workstations (Digital, Silicon Graphics, Hewlett Packard).

Computational comparisons of subroutines PBUNU, PNEWU, PVARU with other algorithms described in literature can be found in Lukšan and Vlček [1998; 1999] and Vlček and Lukšan [2001].

Table VIII. Results Obtained by Program TVARL

| Problem | NIT | NFV | NFG | F | G | ITERM |
|---|---|---|---|---|---|---|
| 1 | 11 | 11 | 11 | −0.38965952D+00 | 0.1632D−07 | 4 |
| 2 | 5 | 9 | 9 | −0.33035714D+00 | 0.4872D−30 | 2 |
| 3 | 15 | 15 | 15 | −0.44891079D+00 | 0.3518D−07 | 4 |
| 4 | 88 | 89 | 89 | −0.42928061D+00 | 0.2323D+01 | 2 |
| 5 | 24 | 24 | 24 | −0.18596186D+01 | 0.2939D−06 | 4 |
| 6 | 25 | 25 | 25 | 0.10183094D+00 | 0.9428D−07 | 4 |
| 7 | 93 | 93 | 93 | 0.37907345D−05 | 0.7469D−06 | 4 |
| 8 | 159 | 159 | 159 | 0.24306464D+02 | 0.6516D−06 | 4 |
| 9 | 226 | 227 | 227 | 0.13372829D+03 | 0.1738D−05 | 2 |
| 10 | 144 | 145 | 145 | 0.50695075D+00 | 0.5979D−05 | 2 |

## REFERENCES


CLARKE, F. H. 1983. *Optimization and Nonsmooth Analysis*. Wiley-Interscience, New York, NY.

FLETCHER, R. 1987. *Practical Methods of Optimization*. 2nd ed. Wiley-Interscience, New York, NY.

GILL, P. E. AND MURRAY, W. 1974. Newton-type methods for unconstrained and linearly constrained optimization. *Math. Program. 7*, 311–350.

HAN, S. P. 1981. Variable metric methods for minimizing a class of nondifferentiable functions. *Math. Program. 20*, 1, 1–13.

KIWIEL, K. C. 1985. *Methods of Descent for Nondifferentiable Optimization*. Springer Lecture Notes in Mathematics, vol. 1133. Springer-Verlag, Vienna, Austria.

KIWIEL, K. C. 1990. Proximity control in bundle methods for convex nondifferentiable minimization. *Math. Program. 46*, 1 (Jan.), 105–122.

LEMARECHAL, C. 1978. Numerical experiments in nonsmooth optimization. In *Proceedings of the IIASA Workshop on Progress in Nondifferentiable Optimization* (Dec.).

LEMARECHAL, C. 1989. Nondifferentiable optimization. In *Optimization*, G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, Eds. Elsevier Science Publishers Ltd., Essex, UK.

LEMARECHAL, C. AND ZOWE, J. 1994. A condensed introduction to bundle methods in nonsmooth optimization. In *Algorithms for Continuous Optimization*, E. Spedicato, Ed. Kluwer Academic Publishers, Hingham, MA.

LUKŠAN, L. 1984. Dual method for solving a special problem of quadratic programming as a subproblem at linearly constrained nonlinear minimax approximation. *Kybernetica 20*, 6, 445–457.

LUKŠAN, L. 1985. An implementation of recursive quadratic programming variable metric methods for linearly constrained nonlinear minimax approximation. *Kybernetica 21*, 1, 22–40.

LUKŠAN, L. AND VLČEK, J. 1998. A Bundle-Newton method for nonsmooth unconstrained minimization. *Math. Program. 83*, 3, 373–391.

LUKŠAN, L. AND VLČEK, J. 1999. Globally convergent variable metric method for convex nonsmooth unconstrained minimization. *J. Optim. Theory Appl. 102*, 3, 593–613.

MÄKELÄ, M. M. AND NEITTAANMÄKI, P. 1992. *Nonsmooth Optimization*. World Scientific Publishing Co., Inc., River Edge, NJ.

MIFFLIN, R. 1977. An algorithm for constrained optimization with semismooth functions. *Math. Oper. Res. 2*, 191–207.

MIFFLIN, R. 1982. A modification and an extension of Lemarechal's algorithm for nonsmooth minimization. *Math. Program. Study 17*, 77–90.

PANIER, E. R. 1987. An active set method for solving linearly constrained nonsmooth optimization problems. *Math. Program. 37*, 3 (May 1), 269–292.

POWELL, M. J. D. 1977. A fast algorithm for nonlinearly constrained optimization calculations. In *Numerical Analysis*, G. A. Watson, Ed. Springer Lecture Notes in Computer Science, vol. 630. Springer-Verlag, New York, NY.

PSCHENYCHNY, B. N. 1983. *Method of Linearization*. Nauka, Moscow, Russia.

VLČEK, J. 1995. Bundle algorithms for nonsmooth unconstrained minimization. Res. Rep. V-608. Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, Czech Republic.

VLČEK, J. AND LUKŠAN, L. 2001. Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization. To appear in J. Optim. Theory Appl. 111, 2.

Received: February 2000; revised: April 2000; accepted: February 2001