Variable Metric Methods for Nonsmooth Optimization

Lukšan, Ladislav
2001

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Variable Metric Methods for Nonsmooth Optimization

Ladislav Lukšan and Jan Vlček

Technical report No. 837

May 2001

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodrenskou v 2, 182 07 Prague 8, Czech Republic
phone: (+4202) 6884244 fax: (+4202) 8585789
e-mail: uivt@uivt.cas.cz

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Variable Metric Methods for Nonsmooth Optimization [1]

Ladislav Lukšan and Jan Vlček[2]

Technical report No. 837
May 2001

Abstract

A variable metric method is introduced for nonsmooth unconstrained optimization, which has a feature of both standard variable metric methods and standard bundle methods. This method is competitive with standard bundle methods measured by number of iterations, but it is more efficient than these methods measured by computational time, since it does not use any time consuming quadratic programming subalgorithm. Numerical comparisons of this method with standard methods are given.

Keywords

Nonsmooth optimization, bundle methods, variable metric methods, computational experiments.

[1] Work supported by grant GA ČR 201/00/0080.

[2] Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic, E-mail: luksan@cs.cas.cz, vlcek@cs.cas.cz

VARIABLE METRIC METHODS FOR NONSMOOTH OPTIMIZATION [3]

Ladislav Lukšan, Jan Vlček
Institute of Computer Science, Academy of Sciences of the Czech Republic,
Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic and Technical University of Liberec, Hálkova 6, 461 17 Liberec, Czech Republic

1 Introduction

Nonsmooth optimization problems frequently occur in practice. In this contribution, we restrict our attention to unconstrained problems. We will assume that objective function $f : R^n \to R$ is locally Lipschitz and that we are able to compute a subgradient $g \in \partial f(x)$ at any point $x \in R^n$. Since a locally Lipschitz function is differentiable almost everywhere by the Rademacher theorem, then usually $g = \nabla f(x)$. A special feature of nonsmooth problems is the fact that the gradient $\nabla f(x)$ can change discontinuously and it need not be small in the neighbourhood of a local extremum of the objective function. For this reason, the usual optimization methods cannot be efficiently used. We show here that the variable metric methods mentioned in Section 2 are exceptional in a sense and that they can be successfully combined with bundle methods described briefly in Section 3. The resulting variable metric method for nonsmooth problems is introduced in Section 4. Computational experiments are reported in Section 5.

2 Standard variable metric methods

Variable metric methods were originally developed for smooth unconstrained optimization. These methods are iterative with the iteration step

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

where the direction vector d^k is computed by the formula

$$d^k = -H^k g^k$$

and the stepsize $t^k > 0$ is chosen by line search so that

$$f(x^k + t^k d^k) - f^k \leq \varepsilon_L t^k (d^k)^T g^k, \tag{2.1}$$

and

$$(d^k)^T g(x^k + t^k d^k) \geq \varepsilon_R (d^k)^T g^k, \tag{2.2}$$

[3] This work was supported by grant GA ČR 201/00/0080

with $0 < \varepsilon_L < 1/2$ and $\varepsilon_L < \varepsilon_R < 1$ (we use the notation $f^k = f(x^k)$, $g^k = g(x^k)$). The symmetric positive definite matrix H^k is computed recursively by the formula

$$\begin{aligned} H^{k+1} &= \gamma^k \left(H^k + \frac{\rho^k}{\gamma^k} \frac{1}{b^k} s^k (s^k)^T - \frac{1}{a^k} H^k u^k (H^k u^k)^T \right. \\ &\quad \left. + \frac{\eta^k}{a^k} \left(\frac{a^k}{b^k} s^k - H^k u^k \right) \left(\frac{a^k}{b^k} s^k - H^k u^k \right)^T \right), \end{aligned} \tag{2.3}$$

where $s^k = x^{k+1} - x^k$, $u^k = g^{k+1} - g^k$, $a^k = (u^k)^T H^k u^k$, $b^k = (u^k)^T s^k$, $c^k = (s^k)^T (H^k)^{-1} s^k$ and where $\rho^k > 0$, $\gamma^k > 0$, $\eta^k > 0$ are free parameters (we exclude the inefficient DFP method corresponding to $\eta^k = 0$). The values

$$\rho^k = \frac{(s^k)^T u^k}{2(f^k - f^{k+1} + (s^k)^T g^{k+1})}, \tag{2.4}$$

$$\gamma^k = \rho^k \sqrt{c^k / a^k}, \tag{2.5}$$

$$\eta^k = \frac{\max(0, \sqrt{c^k/a^k} - (b^k)^2/(a^k c^k))}{1 - (b^k)^2/(a^k c^k)} \tag{2.6}$$

are suitable choices for nonsmooth problems, but the value $\gamma^k = \rho^k \sqrt{c^k/a^k}$ should not be used in all iterations. Controlled scaling proposed in [7], which combines the above value with value $\gamma^k = 1$ is more advantageous.

The above considerations are summarized in the following model algorithm. A detailed description of the standard variable metric methods can be found in [8].

Algorithm 2.1

Data $\varepsilon > 0$, $0 < \varepsilon_L < 1/2$, $\varepsilon_L < \varepsilon_R < 1$.

Step 1 (Initiation). Determine an initial point $x^1 \in R^n$ and an initial symmetric positive definite matrix H^1 (e.g. $H^1 = I$). Compute values $f^1 = f(x^1)$, $g^1 \in \partial f(x^1)$. Set $k = 1$.

Step 2 (Direction vector). If $\|g^k\| \leq \varepsilon$, then terminate the computation, otherwise set $d^k = -H^k g^k$.

Step 3 (Line search). Determine stepsizes t^k, to satisfy conditions (2.1)-(2.2) and compute values $f^{k+1} = f(x^{k+1})$, $g^{k+1} \in \partial f(x^{k+1})$. If $|f^{k+1} - f^k| \leq \varepsilon$ in several consecutive iterations, terminate the computation.

Step 4 (Update). Compute matrix H^{k+1} by (2.3) with parameters (2.4)-(2.6) and controlled scaling. Set $k := k + 1$ and go to Step 2.

It can be proved under mild assumptions (see [1]) that the variable metric method represented by Algorithm 2.1 is globally and superlinearly convergent in the smooth case. Even if this result cannot be generalized to the nonsmooth case, Algorithm 2.1 is surprisingly robust for solving nondifferentiable problems as is shown in [4], [9] and in Section 5. Notice that Algorithm 2.1 usually terminate in Step 3, since sequence $\{\|g^k\|\}$ does not converge to zero in the nonsmooth case.

3 Standard bundle methods

In the nonsmooth case, values $f(x^k)$, $g(x^k) \in \partial f(x^k)$ at single point x^k do not suffice for describing local behaviour of the objective function. A bundle of values $f^j = f(y^j)$, $g^j \in \partial f(y^j)$ obtained at trial points y^j, $j \in \mathcal{J}_k \subset \{1, \ldots, k\}$, gives much better information. These values serve for the construction of the piecewise linear function

$$f_L^k(x) = \max_{j \in \mathcal{J}_k} \{f^j + (x - y^j)^T g^j\} = \max_{j \in \mathcal{J}_k} \{f(x^k) + (x - x^k)^T g^j - \alpha_j^k\},$$

where $\alpha_j^k = f(x^k) - f_j^k$, $j \in \mathcal{J}_k$, are linearization errors and $f_j^k = f^j + (x^k - x^j)^T g^j$, $j \in \mathcal{J}_k$. This piecewise linear function is majorized by the objective function and $\alpha_j^k \geq 0$, $j \in \mathcal{J}_k$, in the convex case. To guarantee nonnegativity of these numbers in the nonconvex case, the subgradient locality measures

$$\alpha_j^k = \max \left\{ |f(x^k) - f_j^k|, \gamma (s_j^k)^\nu \right\},$$

where $\gamma > 0$, $\nu \geq 1$, $f_j^k = f_j(x^k)$ and

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

for $j \in \mathcal{J}_k$, are used in this section (see e.g. [3]). Since, from a practical point of view, we can only work with limited-size bundles, where $|\mathcal{J}_k| \leq m$, say ($|\mathcal{J}_k|$ is the cardinality of set $\mathcal{J}_k$), the set $\mathcal{J}_k$ is usually determined in such a way that $\mathcal{J}_k = \{1, \ldots, k\}$, whenever $k \leq m$, and $\mathcal{J}_{k+1} = \mathcal{J}_k \cup \{k+1\} \backslash \{k+1-m\}$, whenever $k \geq m$. If $\mathcal{J}_k \neq \{1, \ldots, k\}$, then one possibility guaranteeing the global convergence of the bundle method is the use of transformed aggregate values f_a^k, g_a^k, s_a^k and

$$\alpha_a^k = \max \left\{ |f(x^k) - f_a^k|, \gamma (s_a^k)^\nu \right\},$$

which accumulate information from previous iterations. These values represent a linear function which is added to the set of linear functions contained in the bundle. New aggregate values $\tilde{f}_a^k$, $\tilde{g}_a^k$, $\tilde{s}_a^k$, obtained by solving quadratic programming subproblem (3.1)-(3.3) (see (3.6)), are transformed to the next iteration by (3.12).

Direction vector d^k is usually obtained as a minimum of the piecewise quadratic function

$$f_Q^k(x) = \frac{1}{2}(x - x^k)^T G^k (x - x^k) + \max\{f_L^k(x), f(x^k) + (x - x^k)^T g_a^k - \alpha_a^k\},$$

where $(1/2)(x - x^k)^T G^k (x - x^k)$ with G^k positive definite is the regularizing term. This minimization problem is equivalent to the quadratic programming problem: Minimize function

$$\frac{1}{2} d^T G^k d + v \tag{3.1}$$

on the set determined by the constraints

$$-\alpha_j^k + d^T g^j \leq v, \quad j \in \mathcal{J}_k, \tag{3.2}$$

$$-\alpha_a^k + d^T g_a^k \leq v \tag{3.3}$$

(minimization proceeds over all pairs $(d, v) \in R^{n+1}$ satisfying inequalities (3.2), (3.3)). Solution of the problem (3.1)-(3.3) can be expressed in the form

$$\begin{aligned} d^k &= -(G^k)^{-1}\tilde{g}_a^k, & (3.4)\\ v^k &= -(d^k)^T G^k d^k - \tilde{\alpha}_a^k, & (3.5) \end{aligned}$$

where

$$\left.\begin{aligned} \tilde{g}_a^k &= \sum_{j\in\mathcal{J}_k} \lambda_j^k g^j + \lambda_a^k g_a^k, \\ (\tilde{\alpha}_a^k, \tilde{f}_a^k, \tilde{s}_a^k) &= \sum_{j\in\mathcal{J}_k} \lambda_j^k(\alpha_j^k, f_j^k, s_j^k) + \lambda_a^k(\alpha_a^k, f_a^k, s_a^k,) \end{aligned}\right\} \quad (3.6)$$

and where the Lagrange multipliers λ_j^k, $j \in \mathcal{J}_k$, λ_a^k, are solutions of the dual quadratic programming problem: Minimize function

$$\frac{1}{2}\left(\sum_{j\in\mathcal{J}_k} \lambda_j g^j + \lambda_a g_a^k\right)^T (G^k)^{-1}\left(\sum_{j\in\mathcal{J}_k} \lambda_j g^j + \lambda_a g_a^k\right) + \sum_{j\in\mathcal{J}_k} \lambda_j \alpha_j^k + \lambda_a \alpha_a^k \quad (3.7)$$

on the set determined by the constraints

$$\left.\begin{aligned} &\lambda_j \geq 0, \quad j \in \mathcal{J}_k, \quad \lambda_a \geq 0, \\ &\textstyle\sum_{j\in\mathcal{J}_k} \lambda_j + \lambda_a = 1. \end{aligned}\right\} \quad (3.8)$$

The minimum value of function (3.7), which corresponds to the solution of problem (3.7)-(3.8), is

$$w^k = \frac{1}{2}(\tilde{g}_a^k)^T (G^k)^{-1}\tilde{g}_a^k + \tilde{\alpha}_a^k = -v^k - \frac{1}{2}(\tilde{g}_a^k)^T (G^k)^{-1}\tilde{g}_a^k. \quad (3.9)$$

Having direction vector d^k determined, we compute a new approximation of the minimum of the objective function. It is usually not possible to just set $x^{k+1} = x^k + d^k$. To guarantee the global convergence of the bundle method, we use a line search procedure which generates two points

$$\begin{aligned} x^{k+1} &= x^k + t_L^k d^k, \\ y^{k+1} &= x^k + t_R^k d^k, \end{aligned}$$

where $0 \leq t_L^k \leq t_R^k \leq 1$ are stepsizes, in such a way, that exactly one of the two possibilities, descent step and zero step, occurs. Several line search procedures suitable for bundle methods are introduced in [3]. We prefer here the line search procedure described in [16], since it is relevant for the variable metric method described in the next section as well. Thus the descent step implies the conditions

$$t_R^k = t_L^k > 0, \qquad f(x^k + t_L^k d^k) \leq f(x^k) - \varepsilon_L t_L^k w^k, \quad (3.10)$$

while the zero step implies the conditions

$$t_R^k > t_L^k = 0, \qquad (d^k)^T g(x^k + t_R^k d^k) \geq \alpha^{k+1} - \varepsilon_R w^k \quad (3.11)$$

with

$$\alpha^{k+1} = \max\left\{|f(x^k) - f(x^k + t_R^k d^k) + t_R^k (d^k)^T g(x^k + t_R^k d^k)|, \gamma |t_R^k d^k|^\nu\right\}.$$

Here $0 < \varepsilon_L < 1/2$ and $\varepsilon_L < \varepsilon_R < 1$.

Having point x_{k+1} determined, it is necessary to transform all values to this point. This is realized by the formulas

$$\left.\begin{array}{ll} f_j^{k+1} = f_j^k + (x^{k+1} - x^k)^T g^j, & j \in J_k \\ f_a^{k+1} = \tilde{f}_a^k + (x^{k+1} - x^k)^T \tilde{g}_a^k & \\ f_{k+1}^{k+1} = f^{k+1} + (x^{k+1} - y^{k+1}) g^{k+1} & \\ g_a^{k+1} = \tilde{g}_a^k & \\ s_j^{k+1} = s_j^k + \|x^{k+1} - x^k\|, & j \in J_k \\ s_a^{k+1} = \tilde{s}_a^k + \|x^{k+1} - x^k\| & \\ s_{k+1}^{k+1} = \|x^{k+1} - y^{k+1}\| & \end{array}\right\} \tag{3.12}$$

It remains to specify the way for determining matrices G^k. To ensure the global convergence of a bundle method, we assume for simplicity that matrices G^k are uniformly positive definite and uniformly bounded (their eigenvalues are positive and lie in the compact interval that does not contain zero). Moreover, if the k-th step is a zero step, then we assume that $h^T (G^{k+1})^{-1} h \leq h^T (G^k)^{-1} h$, $\forall h \in R^n$. These assumptions are relatively strong, but they can be weakened for individual bundle methods.

In the most frequently used proximal bundle method, where matrix G^k is diagonal of the form $G^k = \sigma^k I$, the above assumptions are satisfied if weights σ^k are positive and lie in the compact interval that does not contain zero and $\sigma^{k+1} \geq \sigma^k$ holds in the zero step. The choice of weight σ^k is very important. Weights which are too large imply small direction vectors, almost all serious steps and a slow descent. Weights which are too small imply large direction vectors and many zero steps.

The above considerations are summarized in the following model algorithm. Detailed descriptions of standard bundle methods can be found in [2], [3], [5], [6], [13], [14], [15].

Algorithm 3.1

Data $\varepsilon > 0$, $0 < \varepsilon_L < 1/2$, $\varepsilon_L < \varepsilon_R < 1$, $\gamma > 0$, $\nu \geq 1$, $m \geq 1$.

Step 1 (Initiation). Determine an initial point $x^1 \in R^n$ and an initial symmetric positive definite matrix G^1. Set $y^1 = x^1$ and compute values $f^1 = f(y^1)$, $g^1 \in \partial f(y^1)$. Set $s_1^1 = s_a^1 = 0$, $f_1^1 = f_a^1 = f^1$, $g_1^1 = g_a^1 = g^1$, $J_1 = \{1\}$ and $k = 1$.

Step 2 (Direction vector). Find the solution of quadratic programming subproblem (3.1)-(3.3) (constraint (3.3) is used only if $J_k \neq \{1, \ldots, k\}$). This defines the Lagrange multipliers λ_j^k, $j \in J_k$ and λ_a^k ($\lambda_a^k \neq 0$ only if $J_k \neq \{1, \ldots, k\}$), aggregate values $\tilde{g}_a^k$, $\tilde{\alpha}_a^k$, $\tilde{f}_a^k$, $\tilde{s}_a^k$, direction vector d^k and values v^k, w^k. If $w^k \leq \varepsilon$, terminate the computation.

Step 3 (Line search). Determine stepsizes t_L^k, t_R^k to obtain either a descent step or a zero step (i.e. either (3.10) or (3.11) holds). Set $x^{k+1} = x^k + t_L^k d^k$, $y^{k+1} = x^k + t_R^k d^k$ and $f^{k+1} = f(x^k + t_R^k d^k)$, $g^{k+1} = g(x^k + t_R^k d^k)$.

Step 4 (Update). Compute transformed values by (3.12) and determine matrix G^{k+1} satisfying assumptions discussed above. If $|\mathcal{J}_k| < m$, set $\mathcal{J}_{k+1} = \mathcal{J}_k + \{k+1\}$. If $|\mathcal{J}_k| = m$, set $\mathcal{J}_{k+1} = \mathcal{J}_k + \{k+1\} \backslash \{k+1-m\}$. Set $k := k+1$ and go to Step 2.

It can be proved under mild assumptions (see e.g. [3]), that the number of consecutive zero steps in Algorithm 3.1 is finite and that every cluster point of the sequence $\{x^k\}$ is a stationary point of the objective function. Notice that Algorithm 3.1 requires relatively large bundles ($m \sim n$) to be computationally efficient so that the solution of the quadratic programming subproblem (3.1)-(3.3) is time consuming.

4 Variable metric methods for nonsmooth problems

In this section, we refer to the hybrid method which keeps good properties of both the standard variable metric method and the standard bundle method. We apply variable metric updates which use original subgradients to the matrix $H^k = (G^k)^{-1}$ in (3.4), which allows us to decrease the bundle dimension drastically. At the same time, we use aggregate subgradients after zero steps and a line search described in the previous section to guarantee the global convergence.

The variable metric methods for nonsmooth problems described in this section use, for direction determination, the original subgradient after a descent step and the aggregated subgradient after a zero step. The aggregation procedure uses three subgradients $g^m \in \partial f(x^k)$, $g^{k+1} \in \partial f(y^{k+1})$, $\tilde{g}^k$ and three subgradient locality measures $\alpha_m = 0$, $\alpha_{k+1} \geq 0$, $\tilde{\alpha}_k \geq 0$. The quadratic programming subproblem (3.7)-(3.8) reduces to the minimization of the function

$$\begin{aligned} \varphi(\lambda_1, \lambda_2, \lambda_3) &= \frac{1}{2}(\lambda_1 g^m + \lambda_2 g^{k+1} + \lambda_3 \tilde{g}^k)^T H^k (\lambda_1 g^m + \lambda_2 g^{k+1} + \lambda_3 \tilde{g}^k) \\ &\quad + \lambda_2 \alpha^{k+1} + \lambda_3 \tilde{\alpha}^k, \end{aligned} \tag{4.1}$$

where $\lambda_i \geq 0$, $i \in \{1,2,3\}$ and $\lambda_1 + \lambda_2 + \lambda_3 = 1$. The optimal values $\lambda_i^k \geq 0$, $i \in \{1,2,3\}$ can be computed by simple formulas. The resulting aggregate subgradient and aggregate subgradient locality measure are computed from the formulas

$$\tilde{g}^{k+1} = \lambda_1^k g^m + \lambda_2^k g^{k+1} + \lambda_3^k \tilde{g}^k, \quad \tilde{\alpha}^{k+1} = \lambda_2^k \alpha^{k+1} + \lambda_3^k \tilde{\alpha}^k. \tag{4.2}$$

To ensure the global convergence, we assume that matrices H^k are uniformly positive definite. This requires some modification of the basic algorithm, namely corrections of matrices H^k if necessary. In such a way we obtain more complex algorithms which are described in [9] and [16] in detail. The assumption that $h^T (G^{k+1})^{-1} h \leq h^T (G^k)^{-1} h$, $\forall h \in R^n$ after a zero step is usually guaranteed by the SR1 update. Therefore, we use the BFGS update after a descent step and the SR1 update after a zero step.

We summarize our considerations in the following model algorithm. Detailed description of variable metric methods for nonsmooth problems can be found in [9] and [16].

Algorithm 4.1

Data $\varepsilon \geq 0$, $0 < \varepsilon_L < 1/2$, $\varepsilon_L < \varepsilon_R < 1$, $\gamma > 0$, $\nu \geq 1$.

Step 1 (Initiation). Choose the starting point $x^1 \in \mathcal{R}^n$ and positive definite matrix H^1 (e.g. $H^1 = I$), set $y^1 = x^1$ and $\alpha^1 = 0$ and compute $f^1 = f(x^1)$ and $g^1 \in \partial f(x^1)$. Set $k = 1$.

Step 2 (Descent step initiation). Initialize aggregate subgradient $\tilde{g}^k = g^k$ and aggregate subgradient locality measure $\tilde{\alpha}^k = 0$ and set $m = k$.

Step 3 (Direction determination). Set $w^k = (1/2)(\tilde{g}^k)^T H^k \tilde{g}^k + \tilde{\alpha}^k$. If $w^k \leq \varepsilon$, then terminate the computation, otherwise set $d^k = -H^k \tilde{g}^k$.

Step 4 (Line search). Determine stepsizes t_L^k, t_R^k to obtain either a descent step or a zero step (i.e. either (3.10) or (3.11) holds). Set $x^{k+1} = x^k + t_L^k d^k$, $y^{k+1} = x^k + t_R^k d^k$, $f^{k+1} = f(x^k + t_R^k d^k)$, $g^{k+1} = g(x^k + t_R^k d^k)$ and compute α^{k+1}. Set $u^k = g^{k+1} - g^m$. If $t_L^k > 0$ (descent step), go to Step 7.

Step 5 (Aggregation). Determine multipliers $\lambda_i^k \geq 0$, $i \in \{1, 2, 3\}$, $\lambda_1^k + \lambda_2^k + \lambda_3^k = 1$,. which minimize the function (4.1) and compute the aggregate quantities (4.2)

Step 6 (SR1 update). Let $v^k = H^k u^k - t_R^k d^k$. If $\tilde{g}_k^T v_k < 0$, then set

$$H^{k+1} = H^k - v^k (v^k)^T / (u^k)^T v^k,$$

otherwise set $H^{k+1} = H^k$. Set $k = k + 1$ and go to Step 3.

Step 7 (BFGS update). If $(u^k)^T d^k > 0$, set

$$H^{k+1} = H^k + \left(t_L^k + \frac{(u^k)^T H^k u^k}{(u^k)^T d^k} \right) \frac{d^k (d^k)^T}{(u^k)^T d^k} - \frac{H^k u^k (d^k)^T + d^k (u^k)^T H^k}{(u^k)^T d^k},$$

otherwise set $H^{k+1} = H^k$. Set $k = k + 1$ and go to Step 2.

The above algorithm is not only interesting theoretically but it is surprisingly very efficient computationally. It is competitive with standard bundle methods measured by the number of iterations, even if it uses the bundle of dimension at most 2. Moreover, it is more efficient than standard bundle methods measured by the computational time, since it does not use the time consuming quadratic programming subproblem (with dimension $m \sim n$).

5 Computational experiments

In this section, we give a numerical comparison of three algorithms (VM - Algorithm 2.1, BM - Algorithm 3.1, VN - algorithm 4.1) described in previous sections. The test problems given in [10] were used in our computational experiments (report [10] together with FORTRAN codes can be obtained from the page `http://www.cs.cas.cz/~luksan/test.html`). All algorithms were implemented in the UFO 2000 system (see [12]). Computations were performed on an PC-Pentium 3 (800 MHz) computer in double precision arithmetic. The results are reported in the following table, where `NIT` is the number of iterations, `NFV` is the number of function evaluations and `F` is the minimum function value. The last row contains the summary values and the total computational time (in seconds).

The results in Table 5.1 imply several conclusions. The standard variable metric method solved 24 problems successfully (it fails only in Problem 8). On the other hand, it was computationally less efficient in Problems 17, 21, 22, 25. The standard bundle method is very robust but its computational efficiency does not correspond to the decrease of the number of function evaluations. The variable metric method for nonsmooth problems is robust and also computationally efficient (it requires the minimum CPU time).

P	NI	NF	VM - F	NI	NF	BM - F	NI	NF	VN - F
1	40	57	.59452613D-15	42	45	.38117064D-06	34	34	.27598807D-10
2	31	69	.41762449D-08	18	20	.46154993D-08	15	16	.94894120D-10
3	22	40	1.9522244	31	33	1.9522245	17	17	1.9522247
4	29	85	2.0000000	14	16	2.0000000	17	17	2.0000000
5	38	88	-2.9999999	17	19	-3.0000000	20	20	-2.9999996
6	22	69	7.2000000	13	15	7.2000014	19	19	7.2000000
7	14	47	-1.4142135	11	12	-1.4142135	10	10	-1.4142133
8	1	42	-	66	68	-.99999940	55	59	-.99999247
9	30	106	-.99999997	13	15	-1.0000000	37	37	-.99999979
10	22	39	-7.9999999	43	46	-7.9999999	14	14	-7.9999998
11	40	87	-43.999999	43	45	-43.999999	38	38	-43.999999
12	43	93	22.600162	27	29	22.600162	40	40	22.600162
13	59	193	-32.348675	60	62	-32.348678	52	53	-32.348678
14	216	442	-2.9196985	154	155	-2.9196975	32	32	-2.9197003
15	82	191	.55981310	92	93	.55981566	81	83	.55981553
16	87	239	-.84140832	74	75	-.84140828	89	89	-.84140570
17	851	1663	9.7857799	160	162	9.7857723	241	241	9.7858732
18	86	162	16.703837	128	143	16.703861	88	89	16.703838
19	90	162	.43449088D-06	150	151	.16712381D-06	123	123	.14683215D-05
20	135	283	.25905459D-07	39	40	.12440972D-12	23	23	.00000000
21	493	940	-638564.96	245	251	-638530.48	357	359	-638564.91
22	453	1090	.11091672D-05	52	53	.11665945D-11	358	360	.41534959D-05
23	64	112	.35595885D-07	19	20	.51313988D-08	65	66	.32729678D-05
24	72	157	.30888921D-07	27	28	.23412735D-07	67	67	.94570857D-06
25	427	1534	32.348815	428	450	32.349182	313	315	32.349159
Σ	3447	7990	TIME = 2.80	1966	2046	TIME = 1.48	2205	2221	TIME = 0.93

Table 5.1 - Comparison of optimization methods.

Bibliography

[1] Byrd R.H., Liu D.C., Nocedal J.: On the behavior of Broyden's class of quasi-Newton methods. Report No. NAM 01, Dept. of Electrical Engn. and Computer Science, Northwestern University, Evanston, 1990.

[2] J.B.Hiriart-Urruty, C.Lemarechal: Convex Analysis and Minimization Algorithms. Springer Verlag, Berlin, 1993.

[3] Kiwiel K.C. Methods of Descent for Nondifferentiable Optimization, Lecture Notes in Mathematics 1133, Springer-Verlag, Berlin, 1985.

[4] Lemarechal C.: Numerical experiments in nonsmooth optimization. Proc. IIASA workshop "Progress in Nondifferentiable Optimization", December 1978.

[5] Lemarechal C.: Nondifferentiable Optimization. In: Optimization (G.L.Nemhauser, A.H.G.Rinnooy Kan, M.J.Todd, eds.), Elsevier Science Publishers, North-Holland, Amsterdam 1989.

[6] Lemarechal C., Zowe J.: A condensed introduction to bundle methods in nonsmooth optimization. In: Algorithms for Continuous Optimization (E.Spedicato, ed.), Kluwer Academic Publishers, Dordrecht, 1994.

[7] L.Lukšan L.: Computational experience with improved variable metric methods for unconstrained minimization. Kybernetika 26 (1990) 415-431.

[8] Lukšan L., Spedicato E.: Variable metric methods for unconstrained optimization and nonlinear least squares. Journal of Computational and Applied Mathematics 124 (2000) 61-93.

[9] Lukšan L., Vlček J.: Globally convergent variable metric method for convex nonsmooth unconstrained minimization. Journal of Optimization Theory and Applications Vol.102, 1999, pp.593-613.

[10] Lukšan L., Vlček J.: Test problems for nonsmooth unconstrained and linearly constrained optimization. Technical Report V-798. Prague, ICS AS CR 2000.

[11] Lukšan L., Vlček J.: NDA: Algorithms for nondifferentiable optimization. To appear in Transactions on Mathematical Software.

[12] Lukšan L., Tůma M., Šiška M., Vlček J., Ramešová N.: UFO 2000. Interactive system for universal functional optimization. Technical Report V-826. Prague, ICS AS CR 2000.

[13] Mäkelä M.M., Neittaanmäki P.: Nonsmooth Optimization. World Scientific Publishing Co., London, 1992.

[14] Mifflin R.: A modification and an extension of Lemarechal's algorithm for nonsmooth minimization, Mathematical Programming Study 17 (1982) 77-90.

[15] J.Outrata, M.Kočvara, J.Zowe: Nonsmooth Approach to Optimization Problems with Equilibrium Constraints, Kluwer Academic Publishers, London, 1998.

[16] Vlček J., Lukšan L.: Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization. To appear in Journal of Optimization Theory and Applications.