Yokohama Publishers
ISSN 1338-9151 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 2, No. 1, pp. 59–70, January 2006

VARIABLE METRIC METHOD FOR MINIMIZATION OF PARTIALLY SEPARABLE NONSMOOTH FUNCTIONS*

Ladislav Lukšan and Jan Vlček

Abstract: In this contribution, we propose a new partitioned variable metric method for minimizing nonsmooth partially separable functions. After a short introduction, the complete algorithm is introduced and some implementation details are given. We prove that this algorithm is globally convergent under standard mild assumptions. Computational experiments given confirm efficiency and robustness of the new method.

Key words: *unconstrained optimization, large-scale optimization, nonsmooth optimization, bundle-type methods, variable metric methods, nonlinear least squares, partially separable problems, computational experiments*

Mathematics Subject Classification: *65K05*

1 Introduction

Nonsmooth optimization methods can be used efficiently in many areas of industrial design. The most frequently used nonsmooth objective functions have the following forms

$$F(x) = \max_{1\le i\le m} f_i(x), \tag{1}$$

$$F(x) = \sum_{i=1}^{m} f_i(x), \tag{2}$$

where $f_i(x)$, $1 \le i \le m$, are locally Lipschitz nonsmooth functions (usually absolute values of smooth functions). The first function, which corresponds to l_∞ (minimax) approximation, can be used, e.g., for designing Chebyshev electrical filters. The second function, which corresponds to l_1 approximation, appears in image-restoration formulations as a term for recovering sharp edges. Both of these functions are locally Lipschitz and we are able to compute a (Clarke) 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 $\nabla F(x)$ changes discontinuously and is not small in the neighborhood of a local extremum. Thus the standard optimization methods cannot be used efficiently.

The most commonly used approach for solving nonsmooth optimization problems is based on the bundle principle. In this case, values $F(x^k)$, $g(x^k) \in \partial F(x^k)$ at a single point x^k

*This work was supported by the Grant Agency of the Czech Academy of Sciences, project code IAA1030405 and the institutional research plan No. AV0Z10300504

are replaced by a bundle of values $F^j = F(y^j)$, $g^j \in \partial F(y^j)$ obtained at trial points y^j, $j \in \mathcal{J}_k \subset \{1, \ldots, k\}$. This bundle of values serves for defining a piecewise quadratic function (with quadratic regularizing term), which is used for direction determination by solving a quadratic programming subproblem. The simplest proximal bundle methods use quadratic term with diagonal (usually scaled unit) matrix [7]. In this case, efficient methods require bundles containing approximately n elements, which is not practical in the large-scale case. Note that the quadratic programming subproblem corresponding to objective function (1) is sparse if functions $f_i(x)$, $1 \leq i \leq m$, have sparse subgradients. Thus the proximal bundle method can be used efficiently for function (1) if sparse quadratic programming solver is available. Unfortunately, it is not the case if objective function (2) is considered. In this case, the quadratic programming subproblem has dense constraints even if subgradients of functions $f_i(x)$, $1 \leq i \leq m$, are sparse.

To overcome difficulties concerning dense constraints, efficient nonsmooth variable metric methods [12],[14] were developed. In this case, variable metric updates accumulate information from previous iterations so that small-size bundles suffice for a rapid convergence. Thus we can use three-element bundles for an efficient implementation of the nonsmooth variable metric method and the solution of the corresponding quadratic programming subproblem can be obtained by simple formulas. Note that the nonsmooth variable metric method described in [14] uses three basic ideas of the bundle principle: quadratic programming subproblem with three bundle constraints, aggregation of subgradients and a special nonsmooth line search. These ideas will be mentioned below in connection with our new method.

The nonsmooth variable metric method described in [14] uses standard (dense) variable metric updates, which are not practical in the large-scale case. Therefore, additional possibilities motivated by the smooth case were studied. In [5], [6], efficient methods utilizing limited-memory variable metric updates are described and their convergence is studied.

In this paper we focus our attention on partially separable functions of the form (2), where $f_i(x)$, $1 \leq i \leq m$, are nonsmooth locally Lipschitz particular functions depending on n_i variables, where all numbers n_i, $1 \leq i \leq m$, are small comparing with n. Partially separable functions are very popular in the smooth case, since efficient variable metric methods exist for seeking their minima [3]. Before description of the new method, we shortly describe the main ideas of variable metric methods for partially separable functions.

Let $R_i^n \subset R^n$ be the subspace defined by n_i variables appearing in f_i and $Z_i \in R^{n \times n_i}$ be the matrix whose columns form the canonical orthonormal basis in R_i^n (i.e., they are columns of the unit matrix). Then we can define reduced gradients $g_i = Z_i^T \nabla f_i$ and reduced Hessian matrices $G_i = Z_i^T \nabla^2 f_i Z_i$. The k-th iteration of variable metric methods for partially separable functions starts in the point x^k with reduced gradients g_i^k and approximations of reduced Hessian matrices B_i^k, $1 \leq i \leq m$. Then gradient g^k and matrix B^k are constructed in such a way that

$$g^k = \sum_{i=1}^m Z_i g_i^k, \qquad B^k = \sum_{i=1}^m Z_i B_i^k Z_i^T \tag{3}$$

and the direction vector d^k is computed by solving linear system $B^k d^k = -g^k$. The new point $x^{k+1} = x^k + \alpha^k d^k$ is determined by line search to satisfy the weak Wolfe conditions

$$\begin{aligned} F(x^k + \alpha^k d^k) - F(x^k) &\leq \varepsilon_1 \alpha^k (d^k)^T g^k, \\ (d^k)^T g(x^k + \alpha^k d^k) &\geq \varepsilon_2 (d^k)^T g^k, \end{aligned}$$

with $0 < \varepsilon_1 < 1/2$ and $\varepsilon_1 < \varepsilon_2 < 1$. Finally, new reduced gradients g_i^{k+1} are computed and new approximations of reduced Hessian matrices B_i^{k+1}, $1 \leq i \leq m$ are obtained by variable

metric updates using differences $s_i^k = Z_i^T(x^{k+1} - x^k)$, $u_i^k = g_i^{k+1} - g_i^k$, $1 \le i \le m$. We describe these updates in the next section.

The paper is organized as follows. In Section 2, we introduce a new variable metric method for minimizing partially separable nonsmooth functions and describe the corresponding algorithm in detail. In Section 3 we study theoretical properties of this partitioned nonsmooth variable metric method. Namely, we prove that this method is globally convergent under mild assumptions. Finally, in Section 4 we present results of our experiments confirming efficiency of the new method.

2 Description of the new method

The algorithm given below generates a sequence of basic points $\{x^k\} \subset R^n$ which should converge to a minimizer of $F : R^n \to R$ and a sequence of trial points $\{y^k\} \subset R^n$ satisfying $x^{k+1} = x^k + t_L^k d^k$, $y^{k+1} = x^k + t_R^k d^k$ for $k \ge 1$ with $y^1 = x^1$, where $t_R^k > 0$, $t_R^k \ge t_L^k \ge 0$ are appropriately chosen stepsizes, $B^k d^k = -\tilde{g}^k$ is a direction vector, $\tilde{g}^k$ is an aggregate subgradient and the matrix B^k obtained by partitioned variable metric updates accumulates information about previous subgradients and represents an approximation of the Hessian matrix if function f is smooth. Stepsizes t_R^k and t_L^k are chosen by a special line-search procedure described in [14]. If the descent condition

$$F(y^{k+1}) \le F(x^k) - \varepsilon_L t_R^k w_k \tag{4}$$

is satisfied with suitable t_R^k, where $0 < \varepsilon_L < 1/2$ is fixed and $-w_k < 0$ represents the desirable amount of descent, then $x^{k+1} = y^{k+1}$ and $t_L^k = t_R^k$ (descent step). In this case, the line-search procedure guarantees that either $t_L^k \ge \underline{t}$ or $\beta_{k+1} > \varepsilon_A w^k$, where

$$\beta_{k+1} = \max\left(|F(x^k) - F(y^{k+1}) - (x^k - y^{k+1})^T g^{k+1}|, \gamma|x^k - y^{k+1}|^\nu\right) \tag{5}$$

and $\underline{t} > 0$, $0 < \varepsilon_A < \varepsilon_R$ are fixed. Otherwise, if the condition

$$(d^k)^T g^{k+1} \ge \beta^{k+1} - \varepsilon_R w^k \tag{6}$$

is satisfied, where $\varepsilon_L < \varepsilon_R < 1$ is fixed, then $x^{k+1} = x^k$ (null step). In this case, the line-search procedure guarantees that $\|y^{k+1} - z^{k+1}\| \le \Delta$ where Δ is fixed and z^{k+1} is a point such that $F(z^{k+1}) < F(x^k)$.

The aggregation procedure is very simple. Denoting by l the lowest index satisfying $x^l = x^k$ (index of the iteration after the last descent step) and having the basic subgradient $g^l \in \partial F(x^k)$, the trial subgradient $g^{k+1} \in \partial F(y^{k+1})$ and the current aggregate subgradient $\tilde{g}^k$, we define $\tilde{g}^{k+1}$ as a convex combination

$$\tilde{g}^{k+1} = \lambda_1^k g^l + \lambda_2^k g^{k+1} + \lambda_3^k \tilde{g}^k, \tag{7}$$

where multipliers λ_1^k, λ_2^k, λ_3^k can be easily determined by minimization of a simple quadratic function (see Step 7 of Algorithm 1). This approach retains good convergence properties but eliminates the solution of the rather complicated quadratic programming subproblem that appears in standard bundle methods.

Matrices B^k are generated by using partitioned variable metric updates [3]. After the null steps, symmetric rank one (SR1) update is used, since it gives a nondecreasing sequence of matrices as required for proving the global convergence. Because these properties are not necessary after descent steps, the standard BFGS update appears to be more suitable. Note that individual variable metric updates that could violate positive definiteness are skipped.

Efficiency of the algorithm is very sensitive to the initial stepsize selection, though it is not relevant for theoretical investigation. In fact, a bundle containing trial points and corresponding function values and subgradients is required for an efficient stepsize selection. Nevertheless, the initial stepsize selection does not require time-consuming operations (see Section 4 for details).

Now we are in a position to state the basic algorithm.

Algorithm 1

Data: A final accuracy tolerance $\varepsilon \geq 0$, restart parameters $\varepsilon_D \geq 0$, $\overline{H} > 0$, line search parameters $\varepsilon_A \geq 0$, $\varepsilon_L \geq 0$, $\varepsilon_R \geq 0$, stepsize bounds $\underline{t} > 0$, $\Delta > 0$, subgradient locality parameters $\gamma \geq 0$, $\nu \geq 1$ and a correction parameter $\rho \geq 0$.

Step 0: Initiation. Choose the starting point $x^1 \in R^n$ and positive definite reduced matrices B_i^1, $1 \leq i \leq m$ (e.g. $B_i^1 = I$, $1 \leq i \leq m$), set $y^1 = x^1$, $\alpha^1 = 0$ and compute $f_i^1 = f_i(x^1)$, $g_i^1 \in \partial f_i(x^1)$, $1 \leq i \leq m$ and $f^1 = F(x^1)$, $g^1 \in \partial F(x^1)$ (i.e. $g^1 = g_1^1 + \ldots + g_m^1$). Initialize iteration counter $k = 1$.

Step 1: Descent step initiation. Initialize the aggregate subgradient $\tilde{g}^k = g^k$, the aggregate subgradient locality measure $\tilde{\alpha}^k = 0$ and set $l = k$.

Step 2: Direction determination. Determine B_k from B_i^k, $1 \leq i \leq m$, and compute the Choleski decomposition $B^k = L^k D^k (L^k)^T$. Solve $L^k D^k (L^k)^T \tilde{d}^k = -\tilde{g}^k$ and set $d^k = \tilde{d}^k - \rho \tilde{g}^k$ and $w^k = -(\tilde{g}^k)^T d^k + 2\tilde{\alpha}^k$.

Step 3: Restart. If $k = l$ and either $-(d^k)^T g^k < \varepsilon_D \|d^k\| \|g^k\|$ or $\|d^k\| > \overline{H} \|g^k\|$, set $B_i^k = B_i^1$, $1 \leq i \leq m$ and go to Step 2.

Step 4: Stopping criterion. If $w^k \leq \varepsilon$, then stop.

Step 5: Line search. By the line search procedure given in [14] find stepsizes t_L^k and t_R^k and the corresponding quantities $x^{k+1} = x^k + t_L^k d^k$, $y^{k+1} = x^k + t_R^k d^k$, $f_i^{k+1} = f_i(x^{k+1})$, $g_i^{k+1} \in \partial f_i(y^{k+1})$, $1 \leq i \leq m$ and $f^{k+1} = F(x^{k+1})$, $g^{k+1} \in \partial F(y^{k+1})$ (i.e. $g^{k+1} = g_1^{k+1} + \ldots + g_m^{k+1}$). Compute β^{k+1} by (5). If $t_L^k > 0$, set $\alpha^{k+1} = 0$ (a descent step is taken), otherwise set $\alpha^{k+1} = \beta^{k+1}$ (a null step occurs).

Step 6: Update preparation. For $1 \leq i \leq m$, set $u_i^k = g_i^{k+1} - g_i^l$ and determine s_i^k as a part of $s^k = t_R^k d^k$. If $t_L^k > 0$ (descent step), go to Step 9.

Step 7: Aggregation. Using the Choleski decomposition $B^k = L^k D^k (L^k)^T$, determine multipliers

$$\lambda_i^k \geq 0, \quad i \in \{1,2,3\}, \quad \lambda_1^k + \lambda_2^k + \lambda_3^k = 1,$$

which minimize the function

$$\begin{aligned} \varphi(\lambda_1,\lambda_2,\lambda_3) &= (\lambda_1 g^l + \lambda_2 g^{k+1} + \lambda_3 \tilde{g}^k)^T ((B^k)^{-1} + \rho I)(\lambda_1 g^l + \lambda_2 g^{k+1} + \lambda_3 \tilde{g}^k) \\ &\quad + 2(\lambda_2 \alpha^{k+1} + \lambda_3 \tilde{\alpha}^k), \end{aligned}$$

Set

$$\begin{aligned} \tilde{g}^{k+1} &= \lambda_1^k g^l + \lambda_2^k g^{k+1} + \lambda_3^k \tilde{g}^k, \\ \tilde{\alpha}^{k+1} &= \lambda_2^k \alpha^{k+1} + \lambda_3^k \tilde{\alpha}^k. \end{aligned}$$

Step 8: SR1 update. Let $v_i^k = u_i^k - B_i^k s_i^k$, $1 \le i \le m$. Set

$$\begin{aligned} B_i^{k+1} &= B_i^k + \frac{v_i^k (v_i^k)^T}{(s_i^k)^T v_i^k}, \quad (s_i^k)^T v_i^k > 0, \\ B_i^{k+1} &= B_i^k, \quad (s_i^k)^T v_i^k \le 0. \end{aligned}$$

Set $k = k + 1$ and go to Step 2.

Step 9: BFGS update. Set

$$\begin{aligned} B_i^{k+1} &= B_i^k + \frac{u_i^k (u_i^k)^T}{(s_i^k)^T u_i^k} - \frac{B_i^k s_i^k (B_i^k s_i^k)^T}{(s_i^k)^T B_i^k s_i^k}, \quad (s_i^k)^T u_i^k > 0, \\ B_i^{k+1} &= B_i^k, \quad (s_i^k)^T u_i^k \le 0. \end{aligned}$$

Set $k = k + 1$ and go to Step 1.

Conditions in Step 3 of the algorithm, guaranteing that vectors d^k, $k \ge 1$, are uniformly bounded and eliminating badly conditioned cases, appear rarely and do not have influence on the efficiency of the algorithm. At the same time, corrections with $\rho > 0$ in Step 2 strongly affect efficiency of the method.

3 Properties of the new method

In this section, we prove under mild assumptions that the new method is globally convergent, which means that every cluster point x^* of the sequence $\{x^k\} \subset R^n$ is a stationary point of F, i.e., $0 \in \partial F(x^*)$. For this purpose, we will assume that sequence $\{x^k\}$ is infinite, i.e., $\varepsilon = 0$ in Algorithm 1.

Assumption 1 Points x^k, y^k, $k \ge 1$, lie in a compact region $\mathcal{D}$ and functions $f_i : R^n \to R$, $1 \le i \le m$, are locally Lipschitz on $\mathcal{D}$.

Remark 1 If f_i, $1 \le i \le m$, are locally Lipschitz on $\mathcal{D}$, then also F is locally Lipschitz on $\mathcal{D}$. Thus subgradients $g^k \in \partial F(y^k)$, $k \ge 1$, all their convex combinations and also values $|F(x^k)|$, $k \ge 1$, are uniformly bounded (see [1]). Conditions in Step 2 of Algorithm 1 assure that direction vectors d^k, $k \ge 1$, are uniformly bounded.

Remark 2 If the level set $\mathcal{L}(\overline{F}) = \{x \in R^n : F(x) \le \overline{F}\}$ is compact for some $\overline{F} \ge F(x^1)$, then also set $\mathcal{D}(\overline{F}) = \mathcal{L}(\overline{F}) + \Delta\ \mathcal{B}(0,1)$ (where $\mathcal{B}(0,1)$ is the unit ball) is compact and we can assume that $\mathcal{D} = \mathcal{D}(\overline{F})$. Then $x^k \in \mathcal{L}(\overline{F}) \subset \mathcal{D}$ and $y^k \in \mathcal{D}(\overline{F}) = \mathcal{D}$ for $k \ge 1$, since $\|y^k - z^k\| \le \Delta$ for some $z^k \in \mathcal{L}(\overline{F})$ (this is assured by our line search procedure).

First we will investigate null steps. In the null steps, $d^k = -H^k \tilde{g}^k$ and $w^k = (\tilde{g}^k)^T H^k \tilde{g}^k + 2\tilde{\alpha}^k$ hold, where $H^k = (B^k)^{-1} + \rho I$.

Lemma 1 *Let $\rho > 0$. Then matrices H^k, $k \ge 1$, are uniformly positive definite. Moreover matrices H^k are uniformly bounded and diferences $H^k - H^{k+1}$ are positive semidefinite in consecutive null steps (even if $\rho = 0$).*

Proof. Variable metric updates used in Steps 8 and 9 of Algorithm 1 guarantee that all matrices B_i^k, $1 \le i \le m$, $k \ge 1$, are positive definite (see [10]). Since

$$v^T B^k v = \sum_{i=1}^m v^T Z_i B_i^k Z_i^T v = \sum_{i=1}^m v_i^T B_i^k v_i$$

and $v_i = Z_i^T v \neq 0$ for at least one index $1 \leq i \leq m$ if $v \neq 0$, also matrices B^k, $(B^k)^{-1}$, $k \geq 1$, are positive definite and if $\rho > 0$, matrices $H^k = (B^k)^{-1} + \rho I$, $k \geq 1$, are uniformly positive definite. In the null steps, rank 1 updates used in Step 8 of Algorithm 1 assure that diferences $B_i^{k+1} - B_i^k$, $1 \leq 1 \leq m$, are positive semidefinite (since $(s_i^k)^T v_i^k > 0$, if the rank 1 update is not skipped). Thus $B^{k+1} - B^k$ is positive semidefinite and, therefore, $H^k - H^{k+1} = (B^k)^{-1} - (B^{k+1})^{-1}$ is positive semidefinite (the last fact is proved in [9]). Positive semidefiniteness of $H^k - H^{k+1}$ implies that $\|H^{k+1}\| \leq \|H^k\|$. Thus matrices H^k are uniformly bounded in consecutive null steps. □

Lemma 2 *Let $\rho > 0$ and Assumption 1 holds. If the number of descent steps in Algorithm 1 is finite, then $w^k \to 0$.*

Proof. Let x^l be the last point obtained by a descent step and $k \geq l$. Denote

$$\begin{aligned} \tilde{g}^k(\lambda) &= \lambda g^{k+1} + (1-\lambda)\tilde{g}^k, \\ \tilde{\alpha}^k(\lambda) &= \lambda \alpha^{k+1} + (1-\lambda)\tilde{\alpha}^k, \end{aligned}$$

where $0 \leq \lambda \leq 1$. Since matrix $H^k - H^{k+1}$ is positive semidefinite by Lemma 1, we can write

$$\begin{aligned} w^{k+1} &= (\tilde{g}^{k+1})^T H^{k+1} \tilde{g}^{k+1} + 2\tilde{\alpha}^{k+1} \leq (\tilde{g}^{k+1})^T H^k \tilde{g}^{k+1} + 2\tilde{\alpha}^{k+1} \\ &\leq (\tilde{g}^k(\lambda))^T H^k \tilde{g}^k(\lambda) + 2\tilde{\alpha}_k(\lambda) \stackrel{\Delta}{=} w_k(\lambda). \end{aligned}$$

The last inequality follows from the fact that pair $(\tilde{g}^{k+1}, \tilde{\alpha}^{k+1})$ minimizes $\varphi(\lambda_1, \lambda_2, \lambda_3)$ (in Step 7 of Algorithm 1) over all convex combinations of pairs (g^l, α^l), (g^{k+1}, α^{k+1}), $(\tilde{g}^k, \tilde{\alpha}^k)$. Furthermore, inequality

$$\alpha_{k+1} + (g^{k+1})^T H^k \tilde{g}^k \leq \varepsilon_R w_k$$

holds for $k \geq l$, since $\alpha^{k+1} = \beta^{k+1}$ in null steps. By successive arrangements, we obtain

$$\begin{aligned} w^k(\lambda) &= (\tilde{g}^k(\lambda))^T H^k \tilde{g}^k(\lambda) + 2\tilde{\alpha}_k(\lambda) \\ &= (\tilde{g}^k)^T H^k \tilde{g}^k + 2\tilde{\alpha}^k + 2\lambda \left((g^{k+1})^T H^k \tilde{g}^k - (\tilde{g}^k)^T H^k \tilde{g}^k + \alpha^{k+1} - \tilde{\alpha}^k \right) \\ &\quad + 2\lambda^2 \left(g^{k+1} - \tilde{g}^k \right)^T H^k (g^{k+1} - \tilde{g}^k) \\ &\leq w_k + 2\lambda \varepsilon_R w_k - 2\lambda w_k + 2\lambda^2 (g^{k+1} - \tilde{g}^k)^T H^k (g^{k+1} - \tilde{g}^k) \\ &\leq w_k + 2\lambda(\varepsilon_R w_k - w_k) + \lambda^2 M, \end{aligned}$$

where the existence of constant M follows from the boundedness of vectors g^{k+1}, $\tilde{g}^k$ and matrices H_k (see Remark 1 and Lemma 1). The expression on the right hand side acquires the minimum for $\lambda = (1-\varepsilon_R) w_k / M$ and its minimum value is equal to $w_k - (1-\varepsilon_R)^2 w_k^2 / M$. Therefore, one has

$$w_{k+1} \leq w_k - \frac{(1-\varepsilon_R)^2 w_k^2}{M}. \tag{8}$$

Now we can easily finish the proof. We show that $w_k \to 0$. If it were not true, constant $\delta > 0$ would have to exist such that $w_k \geq \delta$, $\forall k \geq l$ (since sequence $\{w_k\}$ is nonincreasing for $k \geq l$). Then we would obtain $w_{k+1} \leq w_k - (1-\varepsilon_R)^2 \delta^2 / M$ $\forall k \geq l$ from (8) so that $w_k < \delta$ would hold for sufficiently large indices, which is a contradiction. □

Theorem 1 *Let the number of descent steps in Algorithm 1 be finite and x^l be the last point obtained by a descent step. If $\rho > 0$ and Assumption 1 holds, then x^l is a stationary point of F, i.e., $0 \in \partial F(x^l)$.*

Proof. If $k = l$, then $\tilde{g}^k = g^k$ and $\tilde{\alpha}^k = 0$ (Step 1 of Algorithm 1). If $k > l$, then pair $(\tilde{g}^k, \tilde{\alpha}^k)$ is a convex combination of pairs (g^l, α^l), (g^k, α^k), $(\tilde{g}^{k-1}, \tilde{\alpha}^{k-1})$ (Step 7 of Algorithm 1) and, therefore, it is a convex combination of pairs (g^i, α^i), $l \le i \le k$, where

$$\alpha^i = \max\left(|F(y^i) - F(x^l) - (y^i - x^l)^T g^i|, \gamma|y^i - x^l|^\nu\right) \ge \gamma|y^i - x^l|^\nu \tag{9}$$

(with $\alpha^l = 0$, since $y^l = x^l$). By the Caratheodory theorem, there exist at most $n+2$ pairs $(g^{k,i}, \alpha^{k,i}), g^{k,i} \in \partial f(y^{k,i})$, $(y^{k,i}, g^{k,i}, \alpha^{k,i}) \in \{(y^i, g^i, \alpha^i) : l \le i \le k\}$ such that

$$(\tilde{g}^k, \tilde{\alpha}^k) = \sum_{i=1}^{n+2} \lambda^{k,i} \left(g^{k,i}, \alpha^{k,i}\right), \tag{10}$$

where $\lambda^{k,i} \ge 0$, $1 \le i \le n+2$, $\lambda^{k,1} + \ldots + \lambda^{k,n+2} = 1$. Since vectors $y^{k,i}$, $g^{k,i}$, $1 \le i \le n+2$, are uniformly bounded, there is a subset K such that $y^{k,i} \overset{K}{\to} y^{*,i}$, $g^{k,i} \overset{K}{\to} g^{*,i}$, $\lambda^{k,i} \overset{K}{\to} \lambda^{*,i}$, $1 \le i \le n+2$. But $g^{*,i} \in \partial f(y^{*,i})$, $1 \le i \le n+2$ (see [1]) and (10) implies that $(\tilde{g}^k, \tilde{\alpha}^k) \to (\tilde{g}^*, \tilde{\alpha}^*)$, where

$$(\tilde{g}^*, \tilde{\alpha}^*) = \sum_{i=1}^{n+2} \lambda^{*,i}(g^{*,i}, s^{*,i})$$

and $\lambda^{*,i} \ge 0$, $1 \le i \le n+2$, $\lambda^{*,1} + \ldots + \lambda^{*,n+2} = 1$. Since $w^k \to 0$ by Lemma 2 and matrices H^k are uniformly positive definite, one has $\tilde{g}^k \overset{K}{\to} 0$, $\tilde{\alpha}^k \overset{K}{\to} 0$, which implies

$$(0, 0) = \sum_{i=1}^{n+2} \lambda^{*,i}(g^{*,i}, \alpha^{*,i}). \tag{11}$$

Assume without loss of generality that $\lambda^{*,i} > 0$, $1 \le i \le n+2$ (zero multipliers can be omitted). Since $\alpha^{*,i} \ge \gamma|y^{*,i} - x^l|^\nu \ge 0$ by (9), we obtain $\alpha^{*,i} = 0$ and $y^{*,i} = x^l$ for $1 \le i \le n+2$. Thus $g_i^* \in \partial F(y_i^*) = \partial F(x^l)$ and $0 = \lambda_1^* g_1^* + \ldots + \lambda_{n+2}^* g_{n+2}^* \in \partial F(x^l)$. □

Theorem 2 *Let $\rho > 0$ and Assumption 1 holds. Then every cluster point of the sequence $\{x^k\} \subset R^n$ obtained by Algorithm 1 is a stationary point of F.*

Proof. If the number of descent steps in Algorithm 1 is finite, then there is a unique cluster point x^l of the sequence $\{x^k\}$, which is a stationary point of F by Theorem 1. Assume that the number of descent steps is infinite and $x^k \overset{K}{\to} x^*$. Since the current point is unchanged in null steps, we can assume without loss of generality that points $x^k \in K$ were chosen in such a way that the step $x^{k+1} = x^k + t_L^k d^k$ is descent (note that point x^{k+1} can lie outside K). Since sequence $\{F(x^k)\}$ is non-increasing and bounded from below by Assumption 1, one has $F(x^k) - F(x^{k+1}) \overset{K}{\to} 0$, which together with

$$0 \le \varepsilon_L t_L^k w^k \le F(x^k) - F(x^{k+1})$$

(see (4)) gives $t_L^k w^k \overset{K}{\to} 0$. Let $K = K_1 \cup K_2$ where $K_1 = \{k \in K : t_L^k \ge \underline{t}\}$ and $K_2 = \{k \in K : \beta^{k+1} > \varepsilon_A w^k\}$ (this partitioning is guaranteed by our line search procedure, see Section 2). If K_1 is infinite, then $t_L^k w^k \overset{K_1}{\to} 0$ implies $w^k \overset{K_1}{\to} 0$, which together with $w^k = (g_k)^T H_k g^k$ (since $\tilde{g}^k = g^k$ and $\tilde{\alpha}^k = 0$ in descent steps) and uniform positive definiteness of matrices H^k (Lemma 1) gives $g^k \overset{K_1}{\to} 0$. Thus $0 \in \partial F(x^*)$ (see [1]). If K_1 is finite, then K_2 is infinite.

Assume first that subset $K_3 = \{k \in K_2 : w^k \geq \delta\}$ is infinite for some $\delta > 0$. Then $t_L^k w^k \overset{K_3}{\to} 0$ implies $t_L^k \overset{K_3}{\to} 0$ and since vectors d^k are uniformly bounded (Remark 1), one has

$$\|x^{k+1} - x^k\| = t_L^k \|d^k\| \overset{K_3}{\to} 0.$$

Since $y^{k+1} = x^{k+1}$ in descent steps, we obtain $\|y^{k+1} - x^k\| \overset{K_3}{\to} 0$, which together with (5) and continuity of F gives $\beta^{k+1} \overset{K_3}{\to} 0$. Since $K_3 \subset K_2$ one has $0 \leq \varepsilon_A w^k < \beta^{k+1}$ for all $k \in K_3$. Thus $w_k \overset{K_3}{\to} 0$, which is the contradiction with the definition of K_3 implying that K_3 is finite. In this way, we have proved that $w^k \overset{K_2}{\to} 0$. Thus $g^k \overset{K_2}{\to} 0$ and $0 \in \partial F(x^*)$ as in the case when K_1 is infinite. □

4 Implementation details

Algorithm 1 contains all important features of the partitioned nonsmooth variable metric method, which are necessary for the theoretical investigation. In this section we discuss some details concerning our implementation of the algorithm. These details are in fact the same as described in [14], i.e., Algorithm 1 is implemented in a similar way as Algorithm 2.1 in [14]. Therefore, we only mention main ideas, details can be found in [14].

Since quadratic programming subproblem used in Step 7 of Algorithm 1 is very simple, initial stepsize $t = 1$ need not be a good choice in connection with its solution. Therefore we store and use a bundle of values $F^j = F(y^j)$, $g^j \in \partial F(y^j)$ obtained at trial points y^j, $j \in \mathcal{J}_k = \{k - n_B + 1, \ldots, k\}$. These values serve for the construction of the piecewise linear function

$$\varphi_P^k(t) = \max_{j \in \mathcal{J}_k} \{F(x^k + td^k) + t(d^k)^T g^j - \beta_j^k\},$$

where

$$\beta_j^k = \max(|F^k - F(y_j) - (x^k - y^j)^T g^j|, \gamma |x_k - y_j|^\nu).$$

After a descent step, we use quadratic approximation

$$\varphi_Q^k(t) = F^k + t(d^k)^T g^k + \frac{1}{2} t^2 (d^k)^T (H^k)^{-1} d^k = F^k + (t - \frac{1}{2} t^2)(d^k)^T g^k$$

and compute the initial stepsize by minimizing the function $\varphi^k(t) = \max(\varphi_P^k(t), \varphi_Q^k(t))$ in the interval $0 \leq t \leq 2$. After a null step, the unit stepsize is mostly satisfactory. To utilize the bundle and improve the robustness and the efficiency of the method, we use the aggregate subgradient $\tilde{g}_k$ to construct the linear approximation $\varphi_L^k(t) = F^k + t(d^k)^T \tilde{g}^k$ of $F(x^k + td^k)$ and compute the initial stepsize by minimizing the function

$$\tilde{\varphi}^k(t) = \max(\psi_L^k(t), \psi_P^k(t)) + \frac{1}{2} t^2 (d^k)^T (H^k)^{-1} d^k = \max(\psi_L^k(t), \psi_P^k(t)) - \frac{1}{2} t^2 (d^k)^T g^k$$

in the interval $0 \leq t \leq 1$.

The second comment is related to the solution of the simple quadratic programming subproblem in Step 7 of Algorithm 1. This computation is not time consuming, but corresponding formulas are not simple because of the possible influence of round-off errors. More details are given in [14].

What concerns the termination criterion, the simple test $w^k \leq \varepsilon$ is not always suitable, since it can lead to premature termination. Therefore additional conditions should be satisfied simultaneously. These conditions are discussed in [14] where a suitable termination criterion is introduced.

Restarts in Step 3 are necessary for proving the global convergence of Algorithm 1. However, these restarts never appeared in our computational experiments.

5 Computational experiments

Our partitioned nonsmooth variable metric method was tested by using the collection of relatively difficult problems with optional dimension chosen from [11], which can be downloaded (together with the above report) from `www.cs.cas.cz~luksan/test.html` as Test 15. In [11], functions $f_i(x)$, $1 \leq i \leq m$, are given, which serve for defining the objective function

$$F(x) = \sum_{i=1}^{m} |f_i(x)|.$$

We have used parameters $\varepsilon = 10^{-8}$, $\varepsilon_D = 10^{-6}$, $\varepsilon_A = 10^{-4}$, $\varepsilon_L = 10^{-4}$, $\varepsilon_R = 0.25$, $\overline{H} = 10^{10}$, $\underline{t} = 10^{-10}$, $\nu = 2$, $\rho = 10^{-12}$ and $n_B = 20$ (size of the bundle for the initial stepsize selection) in our tests. Parameters Δ (the maximum stepsize) and γ (subgradient locality parameter) were carefully tuned for every method (including VBM and PBM).

Results of computational experiments are given in three tables, where P is the problem number, NEV is the number of function and also gradient evaluations and F is the function value reached. Note that problems used are relatively difficult, usually having more local solutions (values 8.000000, 64.000000 in Table 1 and 2.000000, 390.000000, 264.000000 in Table 2 seems to be local solutions different from the global ones). The last row of every table contains summary results: the total number of function evaluations and the total computational time. Tables 1 and 2 contain comparison of the new method PSVBM (Algorithm 1) with the nonsmooth variable metric method VBM described in [14] and the proximal bundle method PBM (see [7]) on small (n=50) and medium (n=200) size problems. Table 3 compares three versions ($\rho = 0$, $\rho = 10^{-12}$ and $\rho = 10^{-10}$) of our PSVBM method on large scale partially separable problems with 1000 variables.

	PSVBM		VBM		PBM	
P	NEV	F	NEV	F	NEV	F
1	605	.102544E-08	5096	.545119E-13	6258	.387512E-07
2	235	.275184E-08	998	8.000000000	1242	.156714E-07
3	70	.758451E-10	440	.532703E-08	2631	.211869E-07
4	72	29.10695109	338	29.10695109	169	34.10626848
5	242	.663320E-09	353	.473741E-08	133	.180701E-08
6	248	.323124E-08	447	.445267E-08	197	.887983E-08
7	296	583.9000724	1000	566.7522096	1175	583.9002160
8	356	.765773E-08	1320	.598195E-08	5476	.150995E-05
9	1821	134.8481786	688	132.6493773	3027	137.8278133
10	748	33.23330761	1051	32.36926315	3274	32.36926680
11	25887	.375443E-09	11198	.505268E-08	15915	.678975E-08
12	5310	150.3277379	3886	171.9263661	8489	150.3276924
13	440	709.6182298	806	709.6182298	10865	709.6182308
14	303	27.22786762	440	27.22786762	298	27.22786763
15	267	8.749955784	404	8.749955780	214	8.749955787
16	506	3.200000006	1214	3.200000000	1504	3.200000136
17	486	.283228E-08	571	.435820E-01	2704	.485198E-06
18	10119	.272617E-08	1565	18.55686116	3257	.368177E-07
19	462	.486099E-08	520	.438375E-08	1083	.208525E-05
20	443	.483206E-08	634	.423488E-08	9669	.330697E-07
21	665	64.00000000	316	63.98856687	562	64.00000001
22	385	143.3417786	4084	143.3778676	17004	143.3677713
Σ	49920	TIME = 5.83	37369	TIME = 5.20	95146	TIME = 48.82

Table 1: 22 problems with 50 variables

	PSVBM		VBM		PBM	
P	NEV	F	NEV	F	NEV	F
1	3134	.287703E-08	12782	2.000000022	5337	.309460E-04
2	305	.379422E-08	1471	390.0000000	7527	388.0000385
3	74	.226193E-09	1259	.807210E-07	20000	.641472E-03
4	51	126.8635489	89	126.8635489	68	126.8635555
5	282	.732927E-07	699	.883032E-07	269	.335841E-06
6	344	.836329E-08	1269	.895936E-07	305	.499088E-04
7	289	2391.169985	1891	2351.383667	5294	2391.170012
8	616	.317243E-05	2571	.971029E-06	4803	.123909E-03
9	2516	552.3805506	4061	550.4463151	20000	550.4604738
10	907	131.8884763	1581	131.0242899	6475	131.0243091
11	17908	.654127E-02	20000	310.9025527	20000	325.6756651
12	2043	621.1289465	20000	635.0621402	20000	621.4310634
13	718	2940.509429	1352	2940.509413	20001	2940.561237
14	348	112.3149539	1018	112.3149541	473	112.3149622
15	364	36.09356760	1896	36.09356759	282	36.09358633
16	1070	13.20000001	1573	13.20000002	9671	13.20004084
17	380	.268534E-01	1314	.929480E-07	5722	.269042E-01
18	5225	.8002196153	2219	.9441431850	2513	.7756495183
19	4056	.565862E-08	1679	.963094E-07	20000	1.052512009
20	701	.404661E-08	1845	.104635E-06	531	187.4357267
21	253	264.0000000	1122	264.0000001	1309	263.9885705
22	1425	593.3607049	8914	593.3687578	15981	593.3677799
Σ	48021	TIME = 28.78	90605	TIME = 88.83	186561	TIME = 1204.50

Table 2: 22 problems with 200 variables

	$\rho = 0$		$\rho = 10^{-12}$		$\rho = 10^{-10}$	
P	NEV	F	NEV	F	NEV	F
1	545	998.5562640	540	.815757E-08	1312	.394281E-06
2	295	89.80742693	473	.153343E-07	2388	.797028E-06
3	78	.364876E-13	114	.374913E-08	274	.781997E-07
4	55	648.2320706	54	648.2320706	54	648.2320706
5	166	.147116E-03	285	.422724E-05	2896	.407612E-07
6	400	.481325E-05	560	.649530E-08	966	.423583E-08
7	582	12029.94285	650	12029.94285	375	12029.94286
8	337	.296523E-01	1032	.680061E-04	988	.191923E-04
9	1428	4020.325073	4429	2780.112235	6830	2777.336534
10	555	658.0486676	704	658.0486567	2612	658.0486551
11	209	1017.127487	3754	992.9332153	4650	993.7932861
12	7066	3135.352992	2183	3125.893056	2584	3125.777919
13	396	14808.85245	728	14808.85239	1249	14808.85054
14	277	566.1127477	514	566.1127477	1056	566.1127477
15	185	181.9261656	654	181.9261639	766	181.9261639
16	1223	66.53758673	1376	66.53333334	11108	66.53333334
17	512	.3971713814	9092	.337978E-08	1568	.547473E-08
18	1258	.8020794852	173	.8022190262	966	.8012616805
19	2517	1.389267650	15944	.239244E-08	6688	.241375E-07
20	6428	.420664E-02	2311	.145626E-03	1337	.931539E-07
21	223	1328.000003	1545	1327.988568	1297	1327.950162
22	325	2993.587989	9875	2993.375706	8790	2993.372248
Σ	25060	TIME = 110.67	56990	TIME = 296.86	60754	TIME = 325.88

Table 3: 22 problems with 1000 variables

Results presented in the above tables imply several conclusions:

- Our partitioned nonsmooth variable metric method is competitive with standard nonsmooth methods VBM and PBM in small-size problems. It gives the best results for medium-size problems with 200 variables for which the proximal bundle method (that uses quadratic programming subproblems with large numbers of constraints) is unsuitable.
- Partitioned nonsmooth variable metric method succesfully solves large-scale problems, which cannot be solved by standard nonsmooth methods utilizing dense matrices.
- A nonzero value of parameter ρ has not only a theoretical significance, but it really improves efficiency and robustness of the method. Usually very small values $\rho = 10^{-12}$ or $\rho = 10^{-10}$ are sufficient. Greater values, e.g. $\rho = 10^{-8}$, decrease the efficiency of the method.

References

[1] F.H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley & Sons, New York, 1983.

[2] A. Griewank, The global convergence of partitioned BFGS method on problems with convex decompositions and Lipschitzian gradients, *Math. Program.* 50 (1991) 141–175.

[3] A. Griewank and P.L. Toint, Partitioned variable metric updates for large-scale structured optimization problems, *Numer. Math.* 39 (1982) 119–137.

[4] A. Griewank and P.L. Toint, Local convergence analysis for partitioned quasi-Newton updates, *Numer. Math.* 39 (1982) 429–448.

[5] M .Haarala, K.Mietinen and M.M. Mäkelä, New limited memory bundle method for large-scale nonsmooth optimization, *Optimizaton Methods and Software* 19 (2004) 673–692.

[6] M. Haarala, K. Mietinen and M.M. Mäkelä, Limited memory bundle method for large-scale nonsmooth optimization: Convergence analysis. Report B 12/2003, University of Jyvaskylä, Jyvaskylä 2003.

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

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

[9] L. Lukšan, C. Matonoha and J. Vlček, A shifted Steihaug-Toint method for computing a trust-region step Report V-914, Prague, ICS AS CR, 2004.

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

[11] L. Lukšan and J. Vlček, Sparse and partially separable test problems for unconstrained and equality constrained optimization, Report V-767, Prague, ICS AS CR, 1998.

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

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

[14] J. Vlček and L. Lukšan, Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization, *Journal of Optimization Theory and Applications* 111 (2001) 407–430.

Manuscript received 23 March 2005
revised 15 November 2005
accepted for publication 15 November 2005

Ladislav Lukšan
Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 182 07 Praha 8, Czech Republic,
Technical University of Liberec, Hálkova 6, 46117 Liberec, Czech Republic
E-mail address: luksan@cs.cas.cz

Jan Vlček
Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 182 07 Praha 8, Czech Republic
E-mail address: vlcek@cs.cas.cz