Globally Convergent Variable Metric Method for Nonconvex Nondifferentiable Unconstrained Minimization

Lukšan, Ladislav
1999

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization

J. Vlček L. Lukšan

Technical report No. 775

April 1999

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 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

Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization[1]

J. Vlček L. Lukšan

Technical report No. 775
April 1999

Abstract

A special variable metric method is given for finding stationary points of locally Lipschitz continuous functions which are not necessarily convex or differentiable. Time consuming quadratic programming subproblems do not need to be solved. Global convergence of the method is established. Some encouraging numerical experience is reported.

Keywords

Nonsmooth minimization, nonconvex minimization, numerical methods, variable metric methods, global convergence

[1]This work was supported by AS CR Grant A2030706

1 Introduction

This paper is devoted to minimizing locally Lipschitz continuous function $f : \mathcal{R}^N \to \mathcal{R}$. We assume that for each $y \in \mathcal{R}^N$ we can compute the value $f(y)$ and an arbitrary subgradient $g(y)$, i.e. one element of the subdifferential $\partial f(y)$ (called generalized gradient in [3]). Since f is assumed to be locally Lipschitz continuous, f is differentiable at y for all y except in a set of zero (Lebesgue) measure (see [17]).

The most efficient globally convergent methods for nonconvex nonsmooth optimization are various versions of bundle methods (see e.g. [8], [9], [17], [18], [15]). Essentially, instead of the singleton $f_k = f(x_k)$, $g(x_k) \in \partial f(x_k)$, the bundle $\{(f_j^k, g_j) : \; j \in \mathcal{J}_k\}$ is used in the k-th iteration, $k \geq 1$, where $f_j^k = f(y_j) + (x_k - y_j)^T g_j$, $g_j \in \partial f(y_j)$, $\mathcal{J}_k \subset \{1, \dots k\}$, $x_1, \dots, x_k$ are iterates and $y_1, \dots, y_k$ are trial points. The piecewise linear function

$$\check{f}_k(x) = \max_{j \in \mathcal{J}_k} \{ f_k + (x - x_k)^T g_j - \beta_j^k \} \tag{1.1}$$

is constructed, where β_j^k, $\beta_j^k \geq 0$ (to have $f_k \geq f_k - \min_{j \in \mathcal{J}_k} \beta_j^k = \check{f}_k(x_k) \geq \min_x \check{f}_k(x)$) represent some generalization of linearization errors $f_k - f_j^k$, $k \geq 1$, $j \in \mathcal{J}_k$ in the nonconvex case (when it may happen that $f_k < f_j^k$), and the direction vector

$$d_k = \arg\min_{d \in \mathcal{R}^N} \left\{ \check{f}_k(x_k + d) + \tfrac{1}{2} d^T B_k d \right\} \tag{1.2}$$

is determined where matrix B_k is usually positive definite (the additional quadratic term in (1.2) has a similar significance as in the trust region approach). The minimization subproblem (1.2) can be replaced by the quadratic programming subproblem

$$(d_k, \xi_k) = \arg\min_{(d,\xi) \in \mathcal{R}^{N+1}} \left\{ \tfrac{1}{2} d^T B_k d + \xi \right\} \quad \text{subject to} \quad -\beta_j^k + d^T g_j \leq \xi, \; j \in \mathcal{J}_k. \tag{1.3}$$

The presented nonconvex VM method proceeds from the convex method, described in [16] and is based on an observation that standard VM methods are relatively robust and efficient even in the nonsmooth case (see e.g. [12] and also our experiments in [16]). The advantage of standard VM methods consists in the fact that the time consuming quadratic programming subproblem (1.3) does not need to be solved. Although standard VM methods require more function evaluations than bundle methods, the total computational time is frequently shorter. On the other hand, no global convergence has been proved for standard VM methods applied to nonsmooth problems, and possible failures or inaccurate results can sometimes appear in practical computations.

Our main purpose was to obtain a VM method which does not require a solution to the quadratic programming subproblem (1.3) but is globally convergent applied to a locally Lipschitz continuous function. For this purpose, ideas essential for bundle methods were used, especially utilization of null steps which serve for obtaining sufficient information about a minimized nondifferentiable function when a serious descent condition is not satisfied. The VM update still is the most essential part of the method; it is carried out in both descent and null steps whenever conditions for positive definiteness are satisfied.

To prove global convergence, additional features of bundle methods, namely simple aggregation of subgradients and application of subgradient locality measures, have to be utilized. These principles guarantee convergence of aggregate subgradients to zero and allow us to use a suitable termination criterion. To improve the robustness and the efficiency of the method, stepsize selection based on the polyhedral approximation of the objective function and a suitable matrix scaling are finally added.

The paper is organized as follows. Section 2 is devoted to the description of a new method and Section 3 contains the global convergence theory. Section 4 gives more details concerning the implementation of the method, and Section 5 describes numerical experiments confirming the computational efficiency.

2 Derivation of the method

The algorithm given below generates a sequence of basic points $\{x_k\}_{k=1}^{\infty} \subset \mathcal{R}^N$ which should converge to a minimizer of $f : \mathcal{R}^N \to \mathcal{R}$ and a sequence of trial points $\{y_k\}$ satisfying $x_{k+1} = x_k + t_L^k d_k$, $y_{k+1} = x_k + t_R^k d_k$ for $k \geq 1$ with $y_1 = x_1$, where $t_R^k \in (0, t_{max})$, $t_L^k \in [0, t_R^k]$ are appropriately chosen stepsizes, $d_k = -\theta_k H_k \tilde{g}_k$ is a direction vector, $\tilde{g}_k$ is an aggregate subgradient, H_k represents a VM approximation of the aggregate inverse Hessian matrix and the number θ_k guarantees the boundedness of $\{|d_k|\}$.

If the descent condition $f(y_{k+1}) \leq f(x_k) - c_L t_R^k w_k$ is satisfied with suitable t_R^k, where $c_L \in (0, 1/2)$ is fixed and $-w_k < 0$ represents the desirable amount of descent, then $x_{k+1} = y_{k+1}$ (descent step). Otherwise, a null step is taken which keeps the basic points unchanged but accumulates information about the minimized function.

The aggregation is very simple: denoting by m the lowest index j satisfying $x_j = x_k$ (index of the iteration after the last descent step) and having the basic subgradient $g_m \in \partial f(x_k)$, 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 of these subgradients

$$\tilde{g}_{k+1} = \lambda_{k,1} g_m + \lambda_{k,2} g_{k+1} + \lambda_{k,3} \tilde{g}_k,$$

where multipliers $\lambda_{k,i}$, $i \in \{1, 2, 3\}$ can easily be determined by minimization of a simple quadratic function, which depends on these three subgradients and two generalized linearization errors (see Step 6 of Algorithm 1). This approach retains global convergence but eliminates a solution of the rather complicated quadratic programming subproblem (1.3) that appears in standard bundle methods.

Note that the global convergence is also assured in a simpler case when $\lambda_{k,1} = 0$, i.e. $\tilde{g}_{k+1}$ is a convex combination of only two subgradients g_{k+1} and $\tilde{g}_k$. However, this simplification slightly deteriorates the robustness of the method, e.g. it increases the sensitivity to the stepsize determination after the null steps (see Section 4). Moreover, the situation when $d_{k+1}^T g_m \geq 0$ occurred in numerical experiments, was much more frequent in the simplified case.

Matrices H_k are generated by using usual VM updates. After the null steps, symmetric rank one (SR1) update (see [6]) is used, since it preserves the boundedness of the generated matrices as required in the global convergence theory. Because this boundedness is not necessary after descent steps, the standard BFGS update (see [6]) appears to be more suitable.

Efficiency of the algorithm is very sensitive to the initial stepsize selection even if it is not relevant for proving global convergence. 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. Details are discussed in Section 4. To test whether the computed stepsize is too small, the bundle parameter s_k (see Section 4) and the matrix scaling parameter μ are determined and if μ is too large after a descent step, the inverse Hessian matrix is scaled and the BFGS update is not performed, which does not have an influence on the global convergence but improves the efficiency of the method.

Because the proof of global convergence requires boundedness of matrices H_k^{-1}, the correction ϱI, $\varrho > 0$, is added to H_k if needed. In descent steps, if the subgradients are identical in consecutive iterations, we extrapolate doubling the stepsize if possible in order to exit such region quicker.

Now we are in a position to describe the method in detail. We shall state the following basic algorithm.

Algorithm 1

Data: An upper and auxiliary lower bound for descent steps $t_{max} > 2$ and $t_{min} \in (0,1)$, respectively, positive line search parameters c_A, c_L and c_R satisfying $c_L + c_A < c_R < 1/2$, a distance measure parameter $\gamma > 0$, a final accuracy tolerance $\varepsilon \geq 0$, correction parameters $\varrho \in (0,1)$ and $L \geq 1$, a locality measure parameter $\omega \geq 1$, a matrix scaling bound $C > 1$ and an upper bound $D > 0$ for the direction vector length.

Step 0: Initiation. Choose the starting point $x_1 \in \mathcal{R}^N$ and positive definite matrix $\check{H}_1$ (e.g. $\check{H}_1 = I$), set $y_1 = x_1$ and $\alpha_1 = 0$ and compute $f_1 = f(x_1)$ and $g_1 \in \partial f(x_1)$. Initialize the matrix scaling parameter value $\mu = 1$, the correction, extrapolation, matrix scaling and updating indicators $i_C = i_E = i_S = i_U = 0$, the correction counter $n_C = 0$, the function evaluation counter for matrix scaling $n_S = 0$ and the iteration counter $k = 1$.

Step 1: Descent step initialization. Set $\tilde{g}_k = g_k$, $\tilde{\alpha}_k = 0$ and an index variable $m = k$.

Step 2: Correction. Set $\check{w}_k = \tilde{g}_k^T \check{H}_k \tilde{g}_k + 2\tilde{\alpha}_k$. If $\check{w}_k < \varrho|\tilde{g}_k|^2$ or $i_C = i_U = 1$, then set

$$w_k = \check{w}_k + \varrho|\tilde{g}_k|^2, \quad H_k = \check{H}_k + \varrho I \tag{2.1}$$

and $n_C = n_C + 1$, otherwise set $w_k = \check{w}_k$ and $H_k = \check{H}_k$. If $n_C \geq L$, then set $i_C = 1$.

Step 3: Stopping criterion. If $w_k \leq \varepsilon$, then stop.

Step 4: Line search. Set $\theta_k = \min[1, D/(|H_k\tilde{g}_k| + 1)]$, $d_k = -\theta_k H_k \tilde{g}_k$ and $n_S = n_S + 1$. If $i_E = 0$ then determine $t_I^k \in [t_{min}, t_{max})$, otherwise set $t_I^k = 2t_L^{k-1}$ and $i_E = 0$. By a line search procedure as given below find stepsizes t_L^k and t_R^k and the corresponding quantities $x_{k+1} = x_k + t_L^k d_k$, $y_{k+1} = x_k + t_R^k d_k$, $f_{k+1} = f(x_{k+1})$, $g_{k+1} \in \partial f(y_{k+1})$ and

$$\beta_{k+1} = \max[|f_k - f(y_{k+1}) + (y_{k+1} - x_k)^T g_{k+1}|, \gamma |y_{k+1} - x_k|^\omega] \tag{2.2}$$

satisfying $0 \le t_L^k \le t_R^k \le t_I^k$ and the serious descent criterion

$$f_{k+1} \le f_k - c_L t_L^k w_k \tag{2.3}$$

and either a *descent step* is taken: $t_L^k = t_R^k$, $\alpha_{k+1} = 0$ and

$$t_L^k \ge t_{min} \quad \text{or} \quad \beta_{k+1} > c_A w_k, \tag{2.4}$$

or a *null step* occurs: $t_L^k = 0 < t_R^k$, $\alpha_{k+1} = \beta_{k+1}$ and

$$-\alpha_{k+1} + d_k^T g_{k+1} \ge -c_R w_k, \qquad |y_{k+1} - x_{k+1}| < t_{max} D. \tag{2.5}$$

Set $u_k = g_{k+1} - g_m$.

Step 5: Scaling parameter updating. Determine the bundle parameter for matrix scaling $s_k \ge 0$. If $s_k < 10^{30}$, then set $\mu = (2\mu + \min[C, \max[0.1, s_k]])/3$. If $t_L^k > 0$, go to Step 8.

Step 6: Aggregation. Determine multipliers $\lambda_{k,i} \ge 0$, $i \in \{1, 2, 3\}$, $\lambda_{k,1} + \lambda_{k,2} + \lambda_{k,3} = 1$, which minimize the function

$$\varphi(\lambda_1, \lambda_2, \lambda_3) = |\lambda_1 W_k g_m + \lambda_2 W_k g_{k+1} + \lambda_3 W_k \tilde{g}_k|^2 + 2[\lambda_2 \alpha_{k+1} + \lambda_3 \tilde{\alpha}_k], \tag{2.6}$$

where $W_k = H_k^{1/2}$. Set

$$\tilde{g}_{k+1} = \lambda_{k,1} g_m + \lambda_{k,2} g_{k+1} + \lambda_{k,3} \tilde{g}_k, \qquad \tilde{\alpha}_{k+1} = \lambda_{k,2} \alpha_{k+1} + \lambda_{k,3} \tilde{\alpha}_k. \tag{2.7}$$

Step 7: SR1 update. Let $v_k = H_k u_k - t_R^k d_k$. If

$$\tilde{g}_k^T v_k < 0 \tag{2.8}$$

and in case of $i_C = 1$, furthermore

$$\varrho |\tilde{g}_{k+1}|^2 \le (\tilde{g}_{k+1}^T v_k)^2 / u_k^T v_k \quad \text{and} \quad \varrho N \le |v_k|^2 / u_k^T v_k, \tag{2.9}$$

then set $i_U = 1$ and

$$\check{H}_{k+1} = H_k - v_k v_k^T / u_k^T v_k, \tag{2.10}$$

otherwise set $i_U = 0$ and $\check{H}_{k+1} = H_k$. Set $k = k + 1$ and go to Step 2.

Step 8: Matrix scaling. If $\mu > 1$ set $i_S = i_S + 1$. If $\mu > \sqrt{C}$ and $n_S > 3$ and $i_S > 1$, set $n_S = 0$, $i_S = 0$, $H_{k+1} = \mu H_k$, $\mu = \sqrt{\mu}$, $k = k + 1$ and go to Step 1.

Step 9: BFGS update. If $u_k = 0$ and $t_L^k < t_{max}/2$, set $i_E = 1$. If $u_k^T d_k > \varrho$, set $i_U = 1$ and

$$\check{H}_{k+1} = H_k + \left(t_L^k + \frac{u_k^T H_k u_k}{u_k^T d_k} \right) \frac{d_k d_k^T}{u_k^T d_k} - \frac{H_k u_k d_k^T + d_k u_k^T H_k}{u_k^T d_k},$$

otherwise set $i_U = 0$, $\check{H}_{k+1} = H_k$, $k = k + 1$ and go to Step 1.

A few comments on the algorithm are in order.

To generalize linearization errors to the nonconvex case, the subgradient locality measures introduced in [8] have been used. The first absolute value in (2.2) is not necessary but it significantly improves the numerical results.

The problem of minimizing function (2.6) in Step 6 is the dual to the following primal problem

$$\underset{d \in \mathcal{R}^N}{\text{minimize}} \left\{ \tfrac{1}{2} d^T H_k^{-1} d + \max[d^T g_m, -\alpha_{k+1} + d^T g_{k+1}, -\tilde{\alpha}_k + d^T \tilde{g}_k] \right\}. \qquad (2.11)$$

The minimization of the quadratic function (2.6) and the determination of the initial stepsize t_I^k in Step 4 and the bundle parameter for matrix scaling s_k in Step 5 will be discussed in Section 4.

Condition (2.8) (or $u_k^T d_k > t_R^k d_k^T H_k^{-1} d_k$), which implies that $u_k^T v_k > 0$ by Lemma 2, assures positive definiteness of the matrix obtained by the SR1 update (see e.g. [6]). Similarly, satisfying $u_k^T d_k > 0$ assures positive definiteness of the matrix obtained by the BFGS update ($u_k^T d_k \geq 0$ holds whenever f is convex). Therefore all matrices $\check{H}_k$, H_k generated by Algorithm 1 are positive definite. The conditions for matrix scaling in Step 8 and corresponding relations were established empirically.

The constant $D > 0$ is meant to be a maximum reasonable value of $|d_k|$. Provided the level set $\{x \in \mathcal{R}^N : f(x) \leq f(x_1)\}$ is bounded, the choice $D \approx \sup\{|x - y| : \max[f(x), f(y)] \leq f(x_1)\}$ seems to be natural.

The correction (2.1) is used automatically, after every SR1 update, only if the condition $\check{w}_k < \varrho |\tilde{g}_k|^2$ has been satisfied L times at least. Thus we have a possibility to eliminate the use of conditions (2.9) (restricting the use of the SR1 update) at the beginning of the iterative process where the SR1 update may have a significant influence on the rate of convergence.

We shall now present a line search algorithm and subsequent lemma which are based on the ideas contained within [8].

Line Search Procedure

(i) Set $t_A = 0$ and $t = t_U = t_I^k$. Choose $\kappa \in (0, 1/2)$ and $c_T \in (c_L, c_R - c_A)$.

(ii) Calculate $f(x_k + t d_k)$, $g \in \partial f(x_k + t d_k)$ and

$$\beta = \max[|f_k - f(x_k + t d_k) + t d_k^T g|, \gamma (t |d_k|)^\omega]. \qquad (2.12)$$

If $f(x_k + t d_k) \leq f_k - c_T t w_k$, set $t_A = t$, otherwise set $t_U = t$.

(iii) If $f(x_k + t d_k) \leq f_k - c_L t w_k$ and either $t \geq t_{min}$ or $\beta > c_A w_k$, set $t_R^k = t_L^k = t$ and return.

(iv) If $-\beta + d_k^T g \geq -c_R w_k$, set $t_R^k = t$, $t_L^k = 0$ and return.

(v) Choose $t \in [t_A + \kappa(t_U - t_A), t_U - \kappa(t_U - t_A)]$ by some interpolation procedure and go to (ii).

Lemma 1. *Let f satisfy the following "semismoothness" hypothesis (see Remark 3.3.4 in [8]): for any $x \in \mathcal{R}^N$, $d \in \mathcal{R}^N$ and sequences $\{\hat{t}_i\} \subset \mathcal{R}_+$ and $\{\hat{g}_i\} \subset \mathcal{R}^N$ satisfying $\hat{t}_i \downarrow 0$ and $\hat{g}_i \in \partial f(x + \hat{t}_i d)$, one has*

$$\limsup_{i\to\infty} \hat{g}_i^T d \geq \liminf_{i\to\infty} [f(x + \hat{t}_i d) - f(x)]/\hat{t}_i \,.$$

Then the line search procedure terminates in a finite number of iterations, finding stepsizes t_L^k and t_R^k satisfying (2.3) and, in case of $t_L^k = 0$ (null steps), also (2.5).

Proof. If the search terminates then obviously relations mentioned above hold at termination, observing that $t \leq t_I^k < t_{max}$ and $|d_k| < D$. Assume, for contradiction purposes, that the search does not terminate. Let t^i, t_A^i, t_U^i, g^i and β^i denote the values of t, t_A, t_U, g and β, respectively, after the i-th iteration of the procedure, hence $t^i \in \{t_A^i,\ t_U^i\}$ for all i. Since $t_A^i \leq t_A^{i+1} \leq t_U^{i+1} \leq t_U^i$ and $t_U^{i+1} - t_A^{i+1} \leq (1-\kappa)(t_U^i - t_A^i)$ for all i, there exists $t^\star \geq 0$ satisfying $t_A^i \uparrow t^\star$, $t_U^i \downarrow t^\star$, $t^i \to t^\star$. Let $S = \{t \geq 0 : f(x_k + t d_k) \leq f_k - c_T t w_k\}$. Since $\{t_A^i\} \subset S$, $t_A^i \uparrow t^\star$ and f is continuous, we have

$$f(x_k + t^\star d_k) \leq f_k - c_T t^\star w_k \,, \tag{2.13}$$

i.e. $t^\star \in S$. Let $I = \{i :\ t^i \notin S\}$. We prove first that the set I is infinite. If there existed $i_0 \in I$ satisfying $t^i \in S$ for all $i > i_0$, it would be $t_U^{i_0} = t_U^i \downarrow t^\star$ for all $i > i_0$, implying $t^\star = t_U^{i_0} \notin S$, which is a contradiction. Thus I is infinite and we have $f(x_k + t^i d_k) > f_k - c_T t^i w_k$ for all $i \in I$. By (2.13), we obtain

$$\left[f(x_k + t^i d_k) - f(x_k + t^\star d_k)\right] / \left(t^i - t^\star\right) > -c_T w_k$$

for all $i \in I$, hence by assumption

$$-c_T w_k \leq \liminf_{i\to\infty,\, i\in I} \frac{f(x_k + t^\star d_k + (t^i - t^\star) d_k) - f(x_k + t^\star d_k)}{t^i - t^\star} \leq \limsup_{i\to\infty,\, i\in I} d_k^T g^i \tag{2.14}$$

in view of $t_U^i \downarrow t^\star$ and $g^i \in \partial f(x_k + t^i d_k)$. We shall consider the following two cases.

(a) Suppose that $t^\star > 0$. By (2.13), $c_L < c_T$ and $t^i \to t^\star$, it holds $f(x_k + t^i d_k) \leq f_k - c_L t^i w_k$ for large i from the continuity of f. Since the search does not terminate, we must have $\beta^i \leq c_A w_k$ at step (iii) for large i. From step (iv) we get $d_k^T g^i < -c_R w_k + \beta^i \leq (c_A - c_R) w_k < -c_T w_k$ for all large i by $w_k > 0$, which is in contradiction with (2.14).

(b) Suppose $t^\star = 0$. Then $t^i \to 0$, implying $\beta^i \to 0$ by the continuity of f and the locally boundedness of the subgradient mapping ∂f (see [8]). The search does not terminate, thus $-\beta^i + d_k^T g^i < -c_R w_k$ at step (iv) for all i, therefore $\limsup_{i\to\infty,\, i\in I} d_k^T g^i \leq -c_R w_k < -c_T w_k$, which contradicts (2.14). □

3 Global convergence of the method

In this section, we prove global convergence of Algorithm 1 under the assumption that function $f : \mathcal{R}^N \to \mathcal{R}$ is locally Lipschitz continuous, that the level set $\{x \in \mathcal{R}^N : f(x) \le f(x_1)\}$ is bounded and that each execution of the line search procedure is finite. For this purpose we will assume that the final accuracy tolerance ε is set to zero.

Lemma 2. *At the k-th iteration of Algorithm 1, one has $w_k = \tilde{g}_k^T H_k \tilde{g}_k + 2\tilde{\alpha}_k$, $w_k \ge \varrho|\tilde{g}_k|^2$, $w_k \ge 2\tilde{\alpha}_k \ge 0$ and $\alpha_{k+1} \ge \gamma|y_{k+1} - x_{k+1}|^\omega$. If in addition the condition (2.8) in Step 7 holds, then $u_k^T v_k > 0$.*

Proof. Considering that $\tilde{\alpha} \ge 0$ by (2.2) and (2.7), relations $w_k = \tilde{g}_k^T H_k \tilde{g}_k + 2\tilde{\alpha}_k$, $w_k \ge \varrho|\tilde{g}_k|^2$, $w_k \ge 2\tilde{\alpha}_k$ follow immediately from (2.1). Since $\alpha_{k+1} = \beta_{k+1}$ and $x_k = x_{k+1}$ for null steps, $\alpha_{k+1} = 0$ and $|y_{k+1} - x_{k+1}| = 0$ for descent steps, we always have $\alpha_{k+1} \ge \gamma|y_{k+1} - x_{k+1}|^\omega$ from (2.2).

If $\tilde{g}_k^T v_k < 0$, then $\tilde{g}_k \ne 0$, $\theta_k > 0$ and, since $v_k = H_k u_k - t_R^k d_k$, we get

$$d_k^T u_k > d_k^T u_k + \theta_k \tilde{g}_k^T v_k = -\theta_k t_R^k d_k^T \tilde{g}_k = \theta_k^2 t_R^k \tilde{g}_k^T H_k \tilde{g}_k > 0$$

by positive definiteness of H_k. The last inequality implies $u_k \ne 0$, which yields $u_k^T H_k u_k > 0$. Using the Cauchy's inequality, we obtain

$$(d_k^T u_k)^2 = (\theta_k \tilde{g}_k^T H_k u_k)^2 \le \theta_k^2 \tilde{g}_k^T H_k \tilde{g}_k u_k^T H_k u_k = u_k^T H_k u_k (-\theta_k d_k^T \tilde{g}_k) < \frac{u_k^T H_k u_k d_k^T u_k}{t_R^k},$$

which gives $0 < u_k^T H_k u_k - t_R^k d_k^T u_k = u_k^T v_k$. □

Lemma 3. *Suppose Algorithm 1 did not stop before the k-th iteration. Then the numbers $\lambda_j^k \ge 0$, $j = 1, \dots, k$ and $\tilde{\sigma}_k$ exist satisfying*

$$(\tilde{g}_k, \tilde{\sigma}_k) = \sum_{j=1}^k \lambda_j^k (g_j, |y_j - x_k|), \quad \sum_{j=1}^k \lambda_j^k = 1, \quad \tilde{\alpha}_k \ge \gamma \tilde{\sigma}_k^\omega. \tag{3.1}$$

Proof. We shall first establish the existence of numbers $\lambda_j^k \ge 0$, $j = 1, \dots, k$ satisfying

$$(\tilde{g}_k, \tilde{\alpha}_k) = \sum_{j=1}^k \lambda_j^k (g_j, \alpha_j), \quad \sum_{j=1}^k \lambda_j^k = 1, \quad \lambda_j^k (x_k - x_j) = 0, \ j = 1, \dots, k. \tag{3.2}$$

The proof will proceed by induction. If $k = 1$, then we set $\lambda_1^1 = 1$. Let $i \in \{1, \dots, k-1\}$ and let (3.2) holds for k replaced by i. If the line search procedure results in a descent step in the i-th iteration, we set $\lambda_j^{i+1} = 0$, $j = 1, \dots, i$, $\lambda_{i+1}^{i+1} = 1$. Since $\tilde{g}_{i+1} = g_{i+1}$, $\tilde{\alpha}_{i+1} = \alpha_{i+1} = 0$ at Step 1, (3.2) holds for $i + 1$. In case of a null step, we denote by n the value of the index variable m (defined in Step 1) at the i-th iteration (index of the iteration after the last descent step, i.e. it holds $x_j = x_{i+1}$ for $j = n, \dots, i + 1$) and define $\lambda_n^{i+1} = \lambda_{i,1} + \lambda_{i,3}\lambda_n^i$, $\lambda_j^{i+1} = \lambda_{i,3}\lambda_j^i$ for $1 \le j \le i$, $j \ne n$ and $\lambda_{i+1}^{i+1} = \lambda_{i,2}$. It is clear that $\lambda_j^{i+1} \ge 0$ for all $j \le i+1$ and

$$\sum_{j=1}^{i+1} \lambda_j^{i+1} = \lambda_{i,1} + \lambda_{i,3}\Big(\lambda_n^i + \sum_{j=1}^{n-1} \lambda_j^i + \sum_{j=n+1}^{i} \lambda_j^i\Big) + \lambda_{i,2} = 1.$$

Using relations (2.7), we obtain

$$(\tilde{g}_{i+1}, \tilde{\alpha}_{i+1}) = \lambda_{i,1}(g_n, 0) + \lambda_{i,2}(g_{i+1}, \alpha_{i+1}) + \sum_{j=1}^{i} \lambda_{i,3}\lambda_j^i(g_j, \alpha_j) = \sum_{j=1}^{i+1} \lambda_j^{i+1}(g_j, \alpha_j)$$

due to $\alpha_n = 0$. Finally, we have $\lambda_j^{i+1}(x_{i+1} - x_j) = \lambda_{i,3}\lambda_j^i(x_i - x_j) = 0$ for $j < n$, which together with $x_j = x_{i+1}$, $j = n, \ldots, i+1$ completes the induction.

Setting $\tilde{\sigma}_k = \sum_{j=1}^k \lambda_j^k |y_j - x_k|$, we get

$$\gamma\tilde{\sigma}_k^\omega = \gamma\Big(\sum_{j=1}^k \lambda_j^k |y_j - x_j|\Big)^\omega \le \sum_{j=1}^k \lambda_j^k \gamma |y_j - x_j|^\omega \le \sum_{j=1}^k \lambda_j^k \alpha_j = \tilde{\alpha}_j$$

from (3.2), which implies $\tilde{\sigma}_k = \sum_{j=1}^k \lambda_j^k |y_j - x_j|$, from Lemma 2 and convexity of the function $\xi \to \gamma\xi^\omega$ on $\mathcal{R}_+$ for $\gamma > 0$ and $\omega \ge 1$. □

Lemma 4. *Let $\bar{x} \in R^N$ be given and suppose that there exist vectors $\bar{q}$, $\bar{g}_i$, $\bar{y}_i$ and numbers $\bar{\lambda}_i \ge 0$ for $i = 1, \ldots, l$, $l \ge 1$, satisfying*

$$(\bar{q}, 0) = \sum_{i=1}^{l} \bar{\lambda}_i(\bar{g}_i, |\bar{y}_i - \bar{x}|), \quad \bar{g}_i \in \partial f(\bar{y}_i), \quad i = 1, \ldots, l, \quad \sum_{i=1}^{l} \bar{\lambda}_i = 1. \tag{3.3}$$

Then $\bar{q} \in \partial f(\bar{x})$.

Proof. Let $I = \{i : \ 1 \le i \le l, \bar{\lambda}_i > 0\}$. By (3.3), $\bar{y}_i = \bar{x}$ and $\bar{g}_i \in \partial f(\bar{x})$ for all $i \in I$. Thus we have $\bar{q} = \sum_{i \in I} \bar{\lambda}_i \bar{g}_i$, $\bar{\lambda}_i > 0$ for $i \in I$, $\sum_{i \in I} \bar{\lambda}_i = 1$, so $\bar{q} \in \partial f(\bar{x})$ by the convexity of $\partial f(\bar{x})$ (see [8]). □

Theorem 1. *If Algorithm 1 terminates at the k-th iteration, the point x_k is stationary for f.*

Proof. If the algorithm terminates at Step 3, then $\varepsilon = 0$ implies $w_k = 0$ and $\tilde{g}_k = 0$, $\tilde{\alpha}_k = \tilde{\sigma}_k = 0$ by Lemma 2 and Lemma 3. By Lemma 3 and using Lemma 4 with $\bar{x} = x_k$, $l = k$, $\bar{q} = \tilde{g}_k$, $\bar{g}_i = g_i$, $\bar{y}_i = y_i$, $\bar{\lambda}_i = \lambda_i^k$ for $i \le k$ we have $0 = \bar{q} \in \partial f(\bar{x})$. □

From now on we will assume that Algorithm 1 does not terminate, i.e. that $w_k > 0$ for all $k \ge 1$.

Lemma 5. *Suppose that $\{x_k\}$ is bounded (e.g. when the level set $\{x \in R^N : \ f(x) \le f(x_1)\}$ is bounded) and that there exist a point $\bar{x} \in \mathcal{R}^N$ and an infinite set $K \subset \{1, 2, \ldots\}$ satisfying $x_k \xrightarrow{K} \bar{x}$, $w_k \xrightarrow{K} 0$. Then $0 \in \partial f(\bar{x})$.*

Proof. Let $I = \{1, \ldots, N+2\}$. From $g_k \in \partial f(y_k)$, $k \geq 1$, Lemma 3 and Caratheodory's Theorem (see [7]) we deduce the existence of vectors $y^{k,i}$, $g^{k,i}$ and numbers $\lambda^{k,i} \geq 0$ and $\tilde{\sigma}_k$ for $i \in I$, $k \geq 1$, satisfying

$$(\tilde{g}_k, \tilde{\sigma}_k) = \sum_{i \in I} \lambda^{k,i}(g^{k,i}, |y^{k,i} - x_k|), \ \sum_{i \in I} \lambda^{k,i} = 1, \ g^{k,i} \in \partial f(y^{k,i}), \ i \in I, \ k \geq 1 \tag{3.4}$$

with $(y^{k,i}, g^{k,i}) \in \{(y_j, g_j): \ j = 1, \ldots, k\}$, $i \in I$, $k \geq 1$. By (2.5) and the fact that $x_{k+1} = y_{k+1}$ for descent steps, we always have $|x_k - y_k| \leq t_{max} D$, $k \geq 1$. By assumption this gives boundedness of $\{y_k\}$ and existence of points $y_i^\star$, $i \in I$ and an infinite set $K_0 \subset K$ satisfying $y^{k,i} \xrightarrow{K_0} y_i^\star$ for $i \in I$. By the local boundedness and the upper semicontinuity of ∂f (see [8]) and the boundedness $\{\lambda^{k,i}\}$, we obtain boundedness of $\{g_k\}$ and existence of vectors $g_i^\star \in \partial f(y_i^\star)$ and numbers $\lambda_i^\star$ for $i \in I$ and an infinite set $\overline{K} \subset K_0$ satisfying $g^{k,i} \xrightarrow{\overline{K}} g_i^\star$ and $\lambda^{k,i} \xrightarrow{\overline{K}} \lambda_i^\star$ for $i \in I$. Obviously $\lambda_i^\star \geq 0$, $i \in I$, $\sum_{i \in I} \lambda_i^\star = 1$ by (3.4).

From $w_k \xrightarrow{K} 0$, Lemma 2 and Lemma 3, we obtain $\tilde{g}_k \xrightarrow{K} 0$, $\tilde{\alpha}_k \xrightarrow{K} 0$, $\tilde{\sigma}_k \xrightarrow{K} 0$. Letting $k \in \overline{K}$ approach infinity in (3.4) and using Lemma 4 with $l = N+2$, $\bar{q} = 0$, $\bar{g}_i = g_i^\star$, $\bar{y}_i = y_i^\star$, $\bar{\lambda}_i = \lambda_i^\star$, we get $0 \in \partial f(\bar{x})$. □

Lemma 6. *Let vectors p, q and numbers $w \geq 0$, $\alpha \geq 0$, $\beta \geq 0$, $M \geq 0$, $c \in (0, 1/2)$ satisfy conditions $w = |p|^2 + 2\alpha$, $\beta + p^T q \leq cw$ and $\max[|p|, |q|, \sqrt{\alpha}] \leq M$. Let $Q(\lambda) = |\lambda q + (1-\lambda)p|^2 + 2[\lambda\beta + (1-\lambda)\alpha]$, $b = (1-2c)/(4M)$. Then*

$$\min\{Q(\lambda): \ \lambda \in [0, 1]\} \leq w - w^2 b^2.$$

Proof. See the proof of Lemma 3.5 in [16].

Lemma 7. *Let the number of descent steps be finite and let the last descent step occurs at the $\hat{k}$-th iteration. Then the point $x_{\hat{k}+1}$ is stationary for f.*

Proof. (i) At first we establish the existence of a number $k^\star$, $k^\star > \hat{k}$ (to have solely null steps), such that

$$w_{k+1} \leq \tilde{g}_{k+1}^T H_k \tilde{g}_{k+1} + 2\tilde{\alpha}_{k+1}, \quad Tr(H_{k+1}) \leq Tr(H_k), \quad k \geq k^\star. \tag{3.5}$$

If $n_C < L$ for all $k \geq 1$, we can set $k^\star = \max[\bar{k}, \hat{k} + 1]$, where $\bar{k}$ is the index k, in which n_C changed last (or $\bar{k} = 1$ if $n_C = 0$ for all $k \geq 1$). To see this, let $k \geq k^\star$. Then $w_{k+1} = \check{w}_{k+1}$ and $H_{k+1} = \check{H}_{k+1}$. If the SR1 update is not used, then (3.5) holds with equalities, otherwise Lemma 2 implies $u_k^T v_k > 0$, which together with (2.10) gives (3.5).

If $n_C < L$ does not hold for all $k \geq 1$, then we set $\bar{k}$ equal to the index k in which $i_C = 1$ occurs first and again set $k^\star = \max[\bar{k}, \hat{k} + 1]$. Then matrix $H_{\bar{k}} - \varrho I$ is positive definite, since $\check{H}_{\bar{k}}$ is positive definite and $H_{\bar{k}} = \check{H}_{\bar{k}} + \varrho I$ by the definition of $\bar{k}$. We can easily prove by induction that all matrices $H_k - \varrho I$, $k \geq \bar{k}$ are positive definite. (If the

SR1 or BFGS update is used, $i_C = i_U = 1$ and therefore $H_{k+1} = \check{H}_{k+1} + \varrho I$, otherwise matrix $\check{H}_{k+1} - \varrho I = H_k - \varrho I$ is positive definite and the more so is matrix $H_{k+1} - \varrho I$).

Assume that $k \geq k^\star$. If the SR1 update is not used, then $i_U = 0$ and $\check{H}_{k+1} = H_k$. Thus $\check{w}_{k+1} \geq \varrho|\tilde{g}_{k+1}|^2$ since the matrix $H_k - \varrho I$ is positive definite. Therefore $w_{k+1} = \check{w}_{k+1}$, $H_{k+1} = \check{H}_{k+1} = H_k$ and (3.5) holds with equalities. If the SR1 update is used, all conditions (2.8)-(2.9) are satisfied and $i_C = i_U = 1$, therefore correction (2.1) (with k replaced by $k+1$) is realized. Using (2.10), we can write

$$w_{k+1} = \tilde{g}_{k+1}^T H_k \tilde{g}_{k+1} + 2\tilde{\alpha}_{k+1} + \varrho|\tilde{g}_{k+1}|^2 - (\tilde{g}_{k+1}^T v_k)^2 / u_k^T v_k$$

and the first part of (3.5) follows from the first part of (2.9). Furthermore, (2.10) implies

$$Tr(H_{k+1}) = Tr(H_k) + \varrho N - |v_k|^2 / u_k^T v_k$$

and the second part of (3.5) follows from the second part of (2.9).

(ii) Combining (3.5) with (2.6) and Lemma 2, we obtain

$$w_{k+1} \leq \tilde{g}_{k+1}^T H_k \tilde{g}_{k+1} + 2\tilde{\alpha}_{k+1} = \varphi(\lambda_{k,1}, \lambda_{k,2}, \lambda_{k,3}) \leq \varphi(0,0,1) = w_k \tag{3.6}$$

for $k \geq k^\star$ and therefore the sequences $\{w_k\}$, $\{W_k \tilde{g}_k\}$, $\{\tilde{\alpha}_k\}$ are bounded. Moreover, (3.5) assures boundedness of sequences $\{H_k\}$ and $\{W_k\}$. By (2.5) we have $|x_{k+1} - y_{k+1}| \leq t_{max} D$, $k \geq k^\star$, which gives boundedness of $\{y_k\}$ and by the local boundedness of ∂f (see [8]) also boundedness of $\{g_k\}$ and $\{W_k g_{k+1}\}$. Denote

$$M = \sup\{|W_k g_{k+1}|, |W_k \tilde{g}_k|, \sqrt{\tilde{\alpha}_k} : \ k \geq k^\star\}, \quad b = (1 - 2c_R)/(4M) \tag{3.7}$$

and assume first that $w_k > \delta > 0$ for all $k \geq k^\star$. Since

$$\min\left\{\varphi(\lambda_1, \lambda_2, \lambda_3) : \ \lambda_i \geq 0, i = 1,2,3, \ \sum_{i=1}^{3} \lambda_i = 1\right\} \leq \min_{\lambda \in [0,1]} \varphi(0, \lambda, 1-\lambda),$$

we can use (3.5), (2.5) and Lemma 6 with $p = W_k \tilde{g}_k$, $q = W_k g_{k+1}$, $w = w_k$, $\alpha = \tilde{\alpha}_k$, $\beta = \alpha_{k+1}$, $c = c_R$ to obtain

$$w_{k+1} \leq \tilde{g}_{k+1}^T H_k \tilde{g}_{k+1} + 2\tilde{\alpha}_{k+1} \leq w_k - (w_k b)^2 < w_k - (\delta b)^2$$

for $k \geq k^\star$ and thus, for sufficiently large k, we have a contradiction with the assumption $w_k > \delta$. Therefore $w_k \to 0$ due to the monotonicity of w_k for $k \geq k^\star$, $x_k \to x_{\hat{k}+1}$ and Lemma 5 gives $0 \in \partial f(x_{\hat{k}+1})$. □

Theorem 2. *Suppose sequence $\{x_k\}$ is bounded. Then every cluster point of $\{x_k\}$ is stationary for f.*

Proof. Let $\bar{x}$ be a cluster point of $\{x_k\}$ and $K \subset \{1, 2, \ldots\}$ be an infinite set such that $x_k \xrightarrow{K} \bar{x}$. In view of Lemma 7, we can restrict to the case when the number of descent steps (with $t_L^k > 0$) is infinite. We denote $K' = \{k : \ t_L^k > 0, \ \exists i \in K, \ i \leq k, \ x_i = x_k\}$. Obviously K' is infinite and $x_k \xrightarrow{K'} \bar{x}$. Continuity of f implies that $f_k \xrightarrow{K'} f(\bar{x})$ and therefore $f_k \downarrow f(\bar{x})$ by monotonicity of $\{f_k\}$, which follows from the descent condition (2.3). Using nonnegativity of t_L^k for $k \geq 1$ and the condition (2.3), we obtain

$$0 \leq c_L t_L^k w_k \leq f_k - f_{k+1} \to 0, \quad k \geq 1. \tag{3.8}$$

If the set $K_1 = \{k \in K' : \ t_L^k \geq t_{min}\}$ is infinite then $w_k \xrightarrow{K_1} 0$, $x_k \xrightarrow{K_1} \bar{x}$ by (3.8) and $0 \in \partial f(\bar{x})$ by Lemma 5.

If K_1 is finite, the set $K_2 = \{k \in K' : \ \beta_{k+1} > c_A w_k\}$ must be infinite by (2.4). For contradiction purposes, we assume that $w_k \geq \delta > 0$ for all $k \in K_2$. From (3.8) we have $t_L^k \xrightarrow{K_2} 0$ and Step 4 implies $|x_{k+1} - x_k| = t_L^k |d_k| \leq t_L^k D$ for $k \geq 1$, thus $x_{k+1} - x_k \xrightarrow{K_2} 0$. Since $\{x_k\}$ is bounded and $y_{k+1} = x_{k+1}$ for descent steps, the local boundedness of ∂f (see [8]) yields also boundedness of $\{g_{k+1}\}_{k \in K'}$. By (2.2) and (3.8) we obtain $\beta_{k+1} \xrightarrow{K_2} 0$, which is in contradiction with $c_A \delta \leq c_A w_k < \beta_{k+1}$, $k \in K_2$. Therefore there exists an infinite set $K_3 \subset K_2$ satisfying $w_k \xrightarrow{K_3} 0$, $x_k \xrightarrow{K_3} \bar{x}$ and $0 \in \partial f(\bar{x})$ by Lemma 5. □

Remark 1. If we choose $\varepsilon > 0$, Algorithm 1 always terminates in a finite number of steps, since $w_k \to 0$ in case the number of descent steps is finite (see the proof of Lemma 7) and since $w_k \xrightarrow{K_1} 0$ or $w_k \xrightarrow{K_3} 0$ in case the number of descent steps is infinite (see the proof of Theorem 2).

4 Implementation

In this section we discuss some details concerning our implementation of the algorithm. Assume that we have the current iteration x_k, $f_k = f(x_k)$, $g(x_k) \in \partial f(x_k)$, $k \geq 1$ and a bundle y_j, $f(y_j)$, $g_j \in \partial f(y_j)$, $j \in \mathcal{J}_k \subset \{1, \ldots, k\}$, where y_j, $j \in \mathcal{J}_k$ are some of the trial points. Furthermore, we have the current aggregate subgradient $\tilde{g}_k$, the positive definite VM approximation H_k of the inverse Hessian matrix, the search direction $d_k = -H_k \tilde{g}_k$ and the bundle parameter for matrix scaling s_k and define generalized linearization errors $\beta_j^k = \max[|f_k - f(y_j) - (x_k - y_j)^T g_j|, \gamma |x_k - y_j|^\omega]$.

After the descent step we have $\tilde{g}_k = g_k = g(x_k)$ and we search for a suitable initial stepsize t_I^k for the line search procedure. The significant descent in the last step encourages us to construct the following quadratic approximation of $f(x_k + t d_k)$:

$$\psi_Q^k(t) = f_k + t d_k^T g_k + \tfrac{1}{2} t^2 d_k^T H_k^{-1} d_k = f_k + (t - \tfrac{1}{2} t^2) d_k^T g_k.$$

The bundle represents the polyhedral function (1.1). For $x = x_k + t d_k$ we have the following piecewise linear approximation of $f(x_k + t d_k)$

$$\psi_P^k(t) = \check{f}_k(x_k + t d_k) = \max_{j \in \mathcal{J}_k} \{f_k - \beta_j^k + t d_k^T g_j\}.$$

To calculate t_I^k we will minimize the convex function $\psi_k(t) = \max[\psi_Q^k(t), \psi_P^k(t)]$ within $[0, 2]$, since obviously $\psi_k(0) = f_k$ and $\psi_k(t) \geq \psi_Q^k(t) > f_k$ for $t \notin [0, 2]$ and $g_k \neq 0$. Thus we set

$$t_I^k = \arg\min \{\psi_k(t) : \ t \in [t_{min}, \min[t_{max}, 2, B/|d_k|]]\},$$

where B is a given upper bound for the distance from point x_k in one step. Note that the possibility of stepsizes greater than 1 is useful here, because the information about function f, included in matrix H_k, is not sufficient for a proper stepsize determination in the nonsmooth case.

After the null step, the unit stepsize is mostly satisfactory, as has been found from numerical experiments. 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 $\psi_L^k(t) = f_k + t d_k^T \tilde{g}_k$ of $f(x_k + t d_k)$ and set

$$t_I^k = \arg\min\left\{\max[\psi_L^k(t), \psi_P^k(t)] + \tfrac{1}{2} t^2 d_k^T H_k^{-1} d_k :\ t \in [t_{min}, \min[1, B/|d_k|]]\right\}.$$

The function $\psi_P^k(t)$ has sometimes no influence on the stepsize determination (then obviously $t_I^k = 1$). It can mean that the initial stepsize is too small. Thus we have introduced the bundle parameter for matrix scaling s_k; in view of (2.11), (1.2) and since function (2.6) is not minimized for descent steps, we could define s_k by

$$\arg\min_{s\in\mathcal{R}}\left\{\max[\psi_L^k(s), \psi_P^k(s)] + \tfrac{1}{2}\nu_k s \tilde{g}_k^T H_k \tilde{g}_k\right\}, \tag{4.1}$$

where $\nu_k = 1$ for null steps, $\nu_k = 0$ for descent steps. For simplification, we omit in (4.1) the lines of ψ_P^k with $d_k^T g_j \le \frac{1}{2}\nu_k d_k^T \tilde{g}_k$ and set

$$s_k = \min\left\{10^{30}, \beta_j^k / d_k^T(g_j - \tilde{g}_k) :\ d_k^T g_j > \tfrac{1}{2}\nu_k d_k^T \tilde{g}_k, j \in \mathcal{J}_k\right\}$$

(minimum abscissa of an intersection of the lines, which create $\psi_P^k(t)$ and have $d_k^T g_j > \frac{1}{2}\nu_k d_k^T \tilde{g}_k$, with $\psi_L^k(t)$).

From now on we will use the same notation as in Algorithm 1. The minimization of the quadratic function (2.6) in Step 6, or $\tilde{\varphi}(\lambda_1, \lambda_2) = \varphi(\lambda_1, \lambda_2, 1 - \lambda_1 - \lambda_2)$, is not complicated. If it is not possible to compute the intersection of straight lines $\partial\tilde{\varphi}/\partial\lambda_1 = 0$, $\partial\tilde{\varphi}/\partial\lambda_2 = 0$, the convexity of $\tilde{\varphi}$ implies that we can restrict our attention to the lines $\lambda_1 = 0$, $\lambda_2 = 0$ and $\lambda_1 + \lambda_2 = 1$. As an example we give a formula for minimization within the line $\lambda_1 = 0$, which we regularly apply in the first null step after any descent step due to $\tilde{g}_k = g_k = g_m$ and $\tilde{\alpha}_k = 0$. If $g_{k+1} \ne \tilde{g}_k$, then set

$$\lambda_{k,2} = \min\left[1, \max\left[0, \frac{d_k^T(g_{k+1} - \tilde{g}_k) + \tilde{\alpha}_k - \alpha_{k+1}}{(g_{k+1} - \tilde{g}_k)^T H_k (g_{k+1} - \tilde{g}_k)}\right]\right],$$

otherwise set $\lambda_{k,2} = 0$ for $\tilde{\alpha}_k < \alpha_{k+1}$ or $\lambda_{k,2} = 1$ for $\tilde{\alpha}_k \ge \alpha_{k+1}$.

Finally we mention the stopping criterion. We define the descent tolerance $\varepsilon_f > 0$ and the maximum number $m_f \ge 1$ of consecutive too small function value variations and add to Step 0 the initialization of auxiliary variables $n_f = 0$ and $\Delta_1 = |f_1| + 1$. To prevent accidental termination, we modify Step 3 in the following way:

Step 3': If $w_k \le \varepsilon$ and either $\Delta_k / \max[1, f_k] < 100\varepsilon_f$ after a descent step, or $w_{k-1} \le \varepsilon$ after two consecutive null steps, then stop.

To cut off useless iterations and update Δ_k, we modify Step 5 in the following way:

Step 5': If $|f(y_{k+1}) - f_k| \geq 10^{-5}\Delta_k$, set $\Delta = |f(y_{k+1}) - f_k|$, otherwise set $\Delta = \Delta_k$. If $\Delta / \max[1, f(y_{k+1})] \leq \varepsilon_f$ or $f(y_{k+1}) = f_k$, then set $n_f = n_f+1$, otherwise set $n_f = 0$. If $n_f \geq m_f$, then stop. Determine the bundle parameter for matrix scaling $s_k \geq 0$ and set $\Delta_{k+1} = \Delta_k$. If $s_k < 10^{30}$, set $\mu = (2\mu + \min[C, \max[0.1, s_k]])/3$. If $t_L^k > 0$, set $\Delta_{k+1} = \Delta$ and go to Step 8.

5 Numerical examples

The above concept was implemented in FORTRAN 77 as VMNC. In this section we compare our results for 30 standard test problems from literature (problem 1 is smooth, all the others are nonsmooth) with those obtained by our convex VM method [16] (VMC) and by our proximal bundle method PBL mentioned in [15]. A comparison with the BT algorithm [18] and the ellipsoid bundle method [10] for some problems can be found in [15], a comparison with a smooth VM method from [14] in [16]. Problems 1-16 are described in [17], problems 17-18 in [19], problems 19-22 in [10], problem 23 in [2], problem 24 in [5], problems 25-29 in [13], problem 30 in [4]; details to problems 15 and 22 can be found in [11] and to problem 26 in [1].

In Table 1 we give optimal values of the functions tested.

Nr.	N	Problem	Minimum	Nr.	N	Problem	Minimum
1	2	Rosenbrock	0	16	50	Goffin	0
2	2	Crescent	0	17	6	El Attar	0.5598131
3	2	CB2	1.9522245	18	2	Wolfe	-8.0
4	2	CB3	2.0	19	50	MXHILB	0
5	2	DEM	-3.0	20	50	L1HILB	0
6	2	QL	7.20	21	5	Colville1	-32.348679
7	2	LQ	-1.4142136	22	15	SHELL DUAL	32.348679
8	2	Mifflin1	-1.0	23	10	Gill	9.7857721
9	2	Mifflin2	-1.0	24	12	Steiner2	16.703838
10	4	Rosen	-44.0	25	5	EXP	0.0001224
11	5	Shor	22.600162	26	6	TRANSF	0.1972906
12	10	Maxquad1	-0.8414083	27	7	Wong1	680.63006
13	20	Maxq	0	28	10	Wong2	24.306209
14	20	Maxl	0	29	20	Wong3	133.72828
15	48	TR48	-638565.0	30	9	Filter	0.0061853

Table 1. Test problems

The parameters of the algorithm had the values $t_{\min} = 10^{-10}$, $t_{\max} = 10^3$, $c_A = c_L = 10^{-4}$, $c_R = 0.25$, $c_T = 2 \cdot 10^{-4}$, $\varepsilon = 10^{-6}$, $\varepsilon_f = 5 \cdot 10^{-7}$, $\varrho = 10^{-12}$, $L = 1$, $\omega = 2$, $C = 100$, $D = 10^{50}$, $\mathcal{J}_k = \{\max[1, k-N-2], \ldots, k\}$, $k \geq 1$ and $m_f = 2$ for problems 1-14, 17-21, 23-24 and 26-29, $m_f = 3$ for problem 15, $m_f = 4$ for problem 16 and $m_f = 5$ for problems 22, 25 and 30.

Our results are summarized in Table 2, in which the following notation is used. N_i is the number of iterations, N_f is the number of objective function - and also subgradient - evaluations, F is the objective function value at termination, B is the maximum allowable distance in one step (see Section 4) and γ is the distance measure parameter; values of B and γ were chosen experimentally. Note that a similar choice of parameters (to optimize N_f) was also performed for VMC and PBL; we refer to [16] for values of B in the case of VMC.

Our limited numerical experiments indicate that the adapted VM methods can compete with the well-known proximal bundle methods in the number of function and subgradient evaluations, applied to nonconvex nonsmooth problems. Moreover, we can expect that the computational time will be mostly significantly shorter.

Nr.	VMNC					VMC		PBL	
	N_i	N_f	F	B	γ	N_f	F	N_f	F
1	33	33	0.320E-07	1	1	36	0.416E-10	45	0.381E-06
2	13	15	0.949E-10	10^3	2	54	0.189E-05	20	0.462E-08
3	15	16	1.9522250	1	2	17	1.9522246	33	1.9522245
4	17	17	2.0000000	10^3	10^{-9}	17	2.0000000	16	2.0000000
5	19	20	-2.9999997	10^3	1	22	-3.0000000	19	-3.0000000
6	17	18	7.2000023	1	10^{-9}	22	7.2000001	15	7.2000015
7	10	10	-1.4142133	1	2	8	-1.4142136	12	-1.4142136
8	55	59	-0.9999925	0.2	0.01	179	-0.9999979	68	-0.9999994
9	35	35	-0.9999998	1	10^{-9}	28	-1.0000000	15	-1.0000000
10	31	32	-43.999975	1	10^{-9}	38	-43.999991	45	-43.999999
11	29	30	22.600186	1	10^{-9}	38	22.600163	29	22.600162
12	89	89	-0.8414057	20	10^{-3}	87	-0.8413999	75	-0.8414083
13	110	111	0.898E-05	10	0.1	135	0.775E-06	151	0.167E-06
14	23	23	0	10^3	10^{-9}	23	0	40	0.124E-12
15	293	295	-638562.27	10^3	0.1	285	-638559.63	251	-638530.48
16	368	368	0.332E-05	10^3	10^{-9}	225	0.164E-05	53	0.117E-11
17	74	76	0.5598184	1	1	115	0.5598147	93	0.5598157
18	14	14	-7.9999998	1	1	18	-7.9999995	46	-8.0000000
19	66	67	0.201E-05	1	10^{-5}	74	0.175E-05	20	0.513E-08
20	63	64	0.153E-05	5	0.1	68	0.122E-05	28	0.234E-07
21	46	47	-32.348675	0.5	0.25	64	-32.348595	62	-32.348679
22	286	289	32.349018	10	0.1	165	32.470010	598	32.348768
23	107	108	9.7862324	10	0.25	124	9.7858075	162	9.7857723
24	61	62	16.703937	1	2	79	16.703848	143	16.703862
25	68	70	0.0001224	0.1	0.25	82	0.0001295	92	0.0001224
26	70	71	0.1972947	1	10^{-9}	73	0.1972932	135	0.1972923
27	46	47	680.63011	1	10^{-9}	52	680.63026	96	680.63011
28	75	76	24.306706	2	10^{-9}	97	24.306219	90	24.306224
29	220	221	133.73418	10^2	0.1	239	133.72841	156	133.72864
30	90	91	0.0061862	1	0.5	171	0.0061855	119	0.0061853
$\sum$	2441	2474				2635		2727	
	Time = 9.34 sec					Time = 8.29 sec		Time = 23.17 sec	

Table 2. Our test results

Bibliography

[1] Bandler, J.W., Srinivasan, T.V., and Charalambous, C., *Minimax Optimization of Networks by Grazor Search*, IEEE Transactions on Microwawe Theory and Techniques, Vol. MTT-20, pp. 596-604, 1972.

[2] Bihain, A., *Optimization of Upper Semidifferentiable Functions*, Journal of Optimization Theory and Applications, Vol.4, pp. 545-568, 1984.

[3] Clarke, F.H., *Optimization and Nonsmooth Analysis*, Wiley-Interscience, New York, 1983.

[4] Charalambous, C., *Acceleration of the Least pth Algorithm for Minimax Optimization with Engineering Applications*, Mathematical Programming 17, pp. 270-297, 1979.

[5] Facchinei, F., and Lucidi, S., *Nonmonotone Bundle-Type Scheme for Convex Nonsmooth Minimization*, Journal of Optimization Theory and Applications, Vol. 76, pp. 241-257, 1993.

[6] Fletcher, R., *Practical Methods of Optimization*, John Wiley & Sons, Chichester, 1987.

[7] Hiriart-Urruty, J.B., and Lemarechal, C., *Convex Analysis and Minimization Algorithms I, II*, Springer, Berlin, Heidelberg, New York, 1993.

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

[9] Kiwiel, K.C., *A Method of Linearizations for Linearly Constrained Nonconvex Nonsmooth Minimization*, Mathematical Programming 34, pp. 175-187, 1986.

[10] Kiwiel, K.C., *An Ellipsoid Trust Region Bundle Method for Nonsmooth Convex Minimization*, SIAM J. Control and Optimization, Vol. 27, pp. 737-757, 1989.

[11] Lemaréchal, C., and Mifflin, R., eds., *Nonsmooth Optimization*, Pergamon Press, Oxford, 1978.

[12] Lemaréchal, C., *Numerical Experiments in Nonsmooth Optimization*, in: *Progress in Nonsmooth Optimization* (E.A. Nurminski, ed.), pp. 61-84, IIASA, Laxenburg, Austria, 1982.

[13] Lukšan, L., *A Compact Variable Metric Algorithm for Linear Minimax Approximation*, Computing, Vol. 36, pp. 355-373, 1986.

[14] Lukšan, L., *Computational Experience with Known Variable Metric Updates*, Journal of Optimization Theory and Applications, Vol. 83, pp. 27-47, 1994.

[15] Lukšan, L., and Vlček, J., *A Bundle-Newton Method for Nonsmooth Unconstrained Minimization*, Mathematical Programming 83, pp. 373-391, 1998.

[16] Lukšan, L., and Vlček, J., *Globally Convergent Variable Metric Method for Convex Nonsmooth Unconstrained Minimization*, accepted in Journal of Optimization Theory and Applications.

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

[18] Schramm, H., and Zowe, J., *A Version of the Bundle Idea for Minimizing a Nonsmooth Function: Conceptual Idea, Convergence Analysis, Numerical Results*, SIAM J. Optimization, Vol. 2, pp. 121-152, 1992.

[19] Zowe, J., *Nondifferentiable Optimization*, in: *Computational Mathematical Programming* (K. Schittkowski, ed.), pp. 323-356, Springer Verlag, Berlin, 1985.