JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 102, No. 3, pp. 593-613, SEPTEMBER 1999

Globally Convergent Variable Metric Method for Convex Nonsmooth Unconstrained Minimization[1]

L. LUKŠAN[2] AND J. VLČEK[2]

Communicated by L. C. W. Dixon

Abstract. A special variable metric method is given for finding minima of convex functions that are not necessarily 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.

Key Words. Nonsmooth minimization, convex minimization, numerical methods, variable metric methods, global convergence.

1. Introduction

This paper is devoted to seeking a minimum of a convex 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 Clarke (Ref. 1). Since f is assumed to be convex, then for all y except in a set of zero (Lebesgue) measure, f is differentiable at y.

The most sophisticated globally convergent methods for nonsmooth convex optimization are various modifications of the bundle methods; see e.g. Kiwiel (Ref. 2), Mäkelä and Neittaanmäki (Ref. 3), Schramm and Zowe (Ref. 4), Lemaréchal and Sagastizábal (Ref. 5), and Lukšan and Vlček (Ref. 6). Instead of the singleton $f_k = f(x_k)$, $g_k \in \partial f(x_k)$, the bundle $\{(f_j^k, g^j) | j \in \mathcal{I}_k\}$ is used in the kth iteration, $k \geq 1$, where

$$f_j^k = f(y^j) + (x_k - y^j)^T g^j, \; g^j \in \partial f(y^j), \; \mathcal{I}_k \subset \{1, \ldots, k\},$$

[1]This work was supported by Grant 201/96/0918 of the Czech Republic Grant Agency.

[2]Researcher, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, Czech Republic.

0022-3239/99/0900-0593$16.00/0

$x_1, \ldots, x_k$ are iterates and $y^1, \ldots, y^k$ are trial points. The following piecewise-linear function is constructed:

$$\check{f}_k(x) = \max_{j \in \mathcal{J}_k} \{ f(y^j) + (x - y^j)^T g^j \}$$
$$= \max_{j \in \mathcal{J}_k} \{ f(x_k) + (x - x_k)^T g^j - \beta_j^k \}, \tag{1}$$

where

$$\beta_j^k = f(x_k) - f_j^k = f(x_k) - f(y^j) + (y^j - x_k)^T g^j, \qquad j \in \mathcal{J}_k, \tag{2}$$

are nonnegative (since f is convex) linearization errors, and the direction vector

$$d^k = \arg\min_{d \in \mathcal{R}^N} \{ \check{f}_k(x_k + d) + (1/2) d^T B_k d \} \tag{3}$$

is determined; the additional quadratic term in (3) has a similar significance as in the trust region approach. The minimization subproblem (3) can be replaced by the quadratic programming subproblem

$$(d^k, \xi^k) = \arg\min_{(d,\xi) \in \mathcal{R}^{N+1}} \{ (1/2) d^T B_k d + \xi \}, \qquad \text{s.t. } -\beta_j^k + d^T g^j \le \xi, j \in \mathcal{J}_k. \tag{4}$$

The most popular proximal bundle methods are based on the choice $B_k = \mu_k I$, where μ_k, $k \ge 1$, are weighting coefficients. These methods require only $O(N)$ operations for solving a system with matrix B_k, so that they are very efficient from the computational time viewpoint. Another possibility is to use aggregate Hessian matrices. The resulting bundle–Newton method (see Lukšan and Vlček, Ref. 6) reduces significantly the number of iterations and function evaluations, but it requires $O(N^3)$ operations for solving a system with matrix B_k; see Lukšan, Ref. 7. A natural idea is to generate the matrices B_k, $k \ge 1$, by using variable metric (VM) updates; but this leads to methods which do not overcome the efficiency of proximal bundle methods (see e.g., Lemaréchal, Ref. 8). The most promising results are presented in Lemaréchal and Sagastizábal (Ref. 5), where reversal quasi-Newton updates together with a special curvilinear search procedure are used. Nevertheless, time-consuming quadratic programming subproblems have to be solved. Note that the methods with VM updates require $O(N^2)$ operations for solving a system with matrix B_k.

The development of our VM method was motivated by the observation that standard VM methods are relatively robust and efficient even in the nonsmooth case; see e.g. Lemaréchal (Ref. 8) and also our experiments in Table 2. Their advantage consists in the fact that the time-consuming quadratic programming subproblem (4) need not be solved. Although standard

VM methods require more function evaluations than bundle methods, frequently the total computational time is less.

On the other hand, no global convergence is proved for standard VM methods applied to nonsmooth problems, and possible failures or inaccurate results can sometimes appear in practical computations. Our main purpose is to obtain a VM method that does not require the solution to the quadratic programming subproblem (4), but is globally convergent applied to a convex nonsmooth function. For this purpose, ideas which are essential for bundle methods are used. Compared with standard VM methods, the basic difference consists in the utilization of null steps that serve for obtaining sufficient information about a convex nondifferentiable function. In this connection, a line search (typical for standard VM methods) has been replaced by a simple-step selection, which is either accepted (descent step) or not accepted (null step). The VM update is carried out in both cases, whenever positive-definiteness conditions are satisfied. To prove global convergence, additional features of bundle methods, namely simple aggregation of subgradients and application of linearization errors, have to be utilized. These principles guarantee convergence of the aggregate subgradients to zero and allow us to use a suitable termination criterion. To improve the robustness and efficiency of the method, a stepsize selection based on a 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 a description of the new VM method for convex nonsmooth minimization. Section 3 contains the global convergence theory. In Section 4, we give more details concerning the implementation of the method; in Section 5, we describe numerical experiments confirming its computational efficiency.

2. Derivation of the Method

The algorithm given below generates a sequence $\{x_k\}_{k=1}^{\infty} \subset \mathcal{R}^N$ of iterates that should converge to a global minimizer of the convex function $f: \mathcal{R}^N \to \mathcal{R}$. Besides these basic points, the algorithm also calculates trial points

$$y_1^k = x_k, \; y_{j+1}^k = x_k + t_j^k d_j^k, \qquad j \geq 1,$$

in the kth iteration, where $t_j^k \in [t_{\min}, t_{\max}]$ is an appropriately chosen stepsize,

$$d_j^k = -H_j^k \tilde{g}_j^k$$

is a direction vector, $\tilde{g}_j^k$ is an aggregate subgradient, and H_j^k represents a VM approximation of the aggregate inverse Hessian matrix. If the descent condition is satisfied,

$$f(y_{j+1}^k) \leq f(x_k) - c_1 t_j^k w_j^k,$$

where $c_1 \in (0, 1/2)$ is fixed and $-w_j^k < 0$ represents the desirable amount of descent, then $x_{k+1} = y_{j+1}^k$ (descent step). Otherwise, null steps are utilized, which do not have influence on the sequence of basic points, but accumulate information about the function being minimized. The aggregation is very simple: having the basic subgradient $g_k \in \partial f(x_k)$, the trial subgradient $g_{j+1}^k \in \partial f(y_{j+1}^k)$, and the current aggregate subgradient $\tilde{g}_j^k$, we set

$$\tilde{g}_{j+1}^k = \lambda_{j,1}^k g_k + \lambda_{j,2}^k g_{j+1}^k + \lambda_{j,3}^k \tilde{g}_j^k, \tag{5}$$

where

$$\lambda_{j,i}^k \geq 0, \qquad i \in \{1, 2, 3\},$$

are appropriately chosen scalars. These scalars can be determined easily by minimization of a simple quadratic function, which depends on these three subgradients and two modified linearization errors; see below and Step 6 of Algorithm 2.1. This approach retains global convergence, but eliminates the solution of the rather complicated quadratic programming subproblem (4), which appears in standard bundle methods. Note that global convergence is assured also in the simpler case when $\lambda_{j,1}^k = 0$, i.e., $\tilde{g}_{j+1}^k$ is a convex combination of only two subgradients, g_{j+1}^k and $\tilde{g}_j^k$. However, this simplification deteriorates slightly the robustness of the method, e.g., increases sensitivity to the stepsize determination after the null steps; see Section 4. Moreover, the situation when $g_k^T d_{j+1}^k \geq 0$ in numerical experiments is much more frequent in the simplified case.

Furthermore, note that the problem of minimizing the function (10) in Step 6 of Algorithm 2.1 is the dual to the following primal problem:

$$\underset{d \in \mathbb{R}^N}{\text{minimize}} \{(1/2) d^T (H_j^k)^{-1} d + \max[d^T g_k, -\alpha_{j+1}^k + d^T g_{j+1}^k, -\tilde{\alpha}_j^k + d^T \tilde{g}_j^k]\}. \tag{6}$$

In analogy with bundle methods, the value α_{j+1}^k should be the linearization error $f(x_k) - f(y_{j+1}^k) + t_j^k (d_j^k)^T g_{j+1}^k$; see (2). Unfortunately, this leads to theoretical difficulties when the stepsize t_j^k is greater than 1. Therefore, we divide the linearization error by t_j^k; see (9).

The matrices H_j^k are generated by using the usual VM updates. After null steps, the symmetric rank-one update [SR1, see Fletcher (Ref. 9)] is used, since it preserves the boundedness of the matrices generated as required in the global convergence theory. Because this boundedness is not necessary after descent steps, the standard BFGS update [see Fletcher (Ref. 9)] appears to be more suitable.

Even if the stepsize selection is not relevant to proving global convergence, the efficiency of the algorithm is very sensitive to its realization. In fact, a bundle containing trial points and corresponding function values

and subgradients is required for efficient stepsize selection. Nevertheless, the stepsize selection does not require time-consuming operations. We discuss details in Section 4. To test whether the computed stepsize is too small, the bundle parameter s_j^k (see Section 4) and the scaling parameter γ are determined; if γ is too large after a descent step, the inverse Hessian matrix is scaled and the BFGS update is not performed; this does not have an influence on global convergence but improves the efficiency of the method.

Because the proof of global convergence requires the boundedness of the matrices $(H_j^k)^{-1}$, the correction $\varrho_k I$, $\varrho_k > 0$, is added to H_j^k if needed. In descent steps, if the subgradients are identical in consecutive iterations, we extrapolate doubling the stepsize if possible in order to achieve a quick exit from such region.

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

Algorithm 2.1.

Data. Lower and upper bound for descent steps $t_{\min} \in (0, 1)$ and $t_{\max} > 1$, respectively, a descent parameter $c_1 \in (0, 1/2)$, a final accuracy tolerance $\epsilon \geq 0$, correction parameters $\varrho \in (0, 1)$, $\bar{g} > 0$, $L \geq 1$, and a matrix scaling bound $\sigma > 1$.

Step 0. Initiation. Choose the starting point $x_1 \in \mathcal{R}^N$, compute $f(x_1)$, $g_1 \in \partial f(x_1)$, choose the positive-definite matrix H_1 (e.g. $H_1 = I$), and set the scaling parameter value $\gamma = 1$ and the correction value $\varrho_1 = \varrho$. Initialize the extrapolation and matrix scaling indicators $i_E = i_S = 0$, the function evaluation counter for matrix scaling $n_S = 0$, and the iteration counter $k = 1$.

Step 1. Start of Iteration Loop. Set

$$\tilde{g}_1^k = g_1^k = g_k, \qquad \tilde{\alpha}_1^k = \alpha_1^k = 0, \qquad y_1^k = x_k, \qquad \check{H}_1^k = H_k.$$

For this iteration, initialize the corrections and updating indicators $i_C = i_U = 0$, the corrections counter $n_C = 0$, and the index variable for null steps $j = 1$.

Step 2. Corrections. Set

$$\check{w}_j^k = (\tilde{g}_j^k)^T \check{H}_j^k \tilde{g}_j^k + 2\tilde{\alpha}_j^k.$$

If

$$\check{w}_j^k < \varrho_k |\tilde{g}_j^k|^2 \quad \text{or} \quad i_C = i_U = 1,$$

then set

$$w_j^k = \check{w}_j^k + \varrho_k |\tilde{g}_j^k|^2, \qquad H_j^k = \check{H}_j^k + \varrho_k I, \tag{7}$$

and $n_C = n_C + 1$; otherwise, set

$$w_j^k = \check{w}_j^k \quad \text{and} \quad H_j^k = \check{H}_j^k.$$

If $n_C \geq L$, then set $i_C = 1$.

Step 3. Stopping Criterion. If $w_j^k \leq \epsilon$, then stop: x_k is an approximate minimizer.

Step 4. Trial Point Determination. If $i_E = 0$, then set

$$d_j^k = -H_j^k \tilde{g}_j^k$$

and determine $t_j^k \in [t_{\min}, t_{\max}]$ and the bundle parameter for matrix scaling $s_j^k \geq 0$; otherwise, set

$$t_j^k = 2t_j^k \quad \text{and} \quad i_E = 0.$$

Set

$$y_{j+1}^k = x_k + t_j^k d_j^k, \qquad n_S = n_S + 1,$$

and compute $f(y_{j+1}^k)$, $g_{j+1}^k \in \partial f(y_{j+1}^k)$. If $s_j^k < 10^{30}$, then set

$$\gamma = (2\gamma + \min[\sigma, \max[1, s_j^k]])/3.$$

Step 5. Descent Step. If

$$f(y_{j+1}^k) - f(x_k) \leq -c_1 t_j^k w_j^k, \tag{8}$$

then set

$$x_{k+1} = y_{j+1}^k, \quad g_{k+1} = g_{j+1}^k, \quad t_k = t_j^k, \quad d_k = d_j^k,$$
$$\tilde{\alpha}_k = \tilde{\alpha}_j^k, \quad \tilde{g}_k = \tilde{g}_j^k, \quad w_k = w_j^k, \quad H_k = H_j^k,$$
$$\varrho_{k+1} = \varrho \min[1/\min[|\tilde{g}_k|, \bar{g}], |\tilde{g}_k|]/(k+1);$$

otherwise, go to Step 6. If

$$g_{k+1} = g_k \quad \text{and} \quad t_j^k < t_{\max}/2,$$

then set

$$i_E = 1, \qquad H_{k+1} = H_k, \qquad k = k+1,$$

and go to Step 1; otherwise, go to Step 8.

Step 6. Null Step. Set

$$\alpha_{j+1}^k = [f(x_k) - f(y_{j+1}^k)]/t_j^k + (d_j^k)^T g_{j+1}^k, \tag{9}$$

and determine multipliers

$$\lambda_{j,i}^k \geq 0, \quad i \in \{1, 2, 3\}, \quad \lambda_{j,1}^k + \lambda_{j,2}^k + \lambda_{j,3}^k = 1,$$

which minimize the function

$$\varphi(\lambda_1, \lambda_2, \lambda_3) = |\lambda_1 W_j^k g_k + \lambda_2 W_j^k g_{j+1}^k + \lambda_3 W_j^k \tilde{g}_j^k|^2 + 2[\lambda_2 \alpha_{j+1}^k + \lambda_3 \tilde{\alpha}_j^k], \tag{10}$$

where $W_j^k = (H_j^k)^{1/2}$. Set

$$\tilde{g}_{j+1}^k = \lambda_{j,1}^k g_k + \lambda_{j,2}^k g_{j+1}^k + \lambda_{j,3}^k \tilde{g}_j^k,$$

$$\tilde{\alpha}_{j+1}^k = \lambda_{j,2}^k \alpha_{j+1}^k + \lambda_{j,3}^k \tilde{\alpha}_j^k. \tag{11}$$

Step 7. SR1 Update. Let

$$u_j^k = g_{j+1}^k - g_k \quad \text{and} \quad v_j^k = H_j^k u_j^k - t_j^k d_j^k.$$

If

$$(\tilde{g}_j^k)^T v_j^k < 0, \tag{12}$$

and in the case $i_C = 1$, if

$$\varrho_k |\tilde{g}_{j+1}^k|^2 \leq [(\tilde{g}_{j+1}^k)^T v_j^k]^2 / (u_j^k)^T v_j^k$$

and

$$N\varrho_k \leq |v_j^k|^2 / (u_j^k)^T v_j^k, \tag{13}$$

then set $i_U = 1$ and

$$\check{H}_{j+1}^k = H_j^k - v_j^k (v_j^k)^T / (u_j^k)^T v_j^k; \tag{14}$$

otherwise, set

$$i_U = 0 \quad \text{and} \quad \check{H}_{j+1}^k = H_j^k.$$

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

Step 8. Matrix Scaling. If $\gamma > 1$, then set $i_S = i_S + 1$. If $\gamma > \sqrt{\sigma}$, $n_S > 3$, $i_S > 1$, then set

$$n_S = 0, \quad i_S = 0, \quad H_{k+1} = \gamma H_k, \quad \gamma = \sqrt{\gamma}, \qquad k = k + 1,$$

and go to Step 1.

Step 9. BFGS Update. If $g_{k+1} = g_k$ and $t_j^k < t_{\max}/2$, then set $i_E = 1$, $k = k + 1$, and go to Step 1; otherwise, set $u_k = g_{k+1} - g_k$. If

$u_k^T d_k > |d_k| 10^{-5}$, then set

$$H_{k+1} = H_k + (t_k + u_k^T H_k u_k / u_k^T d_k) d_k d_k^T / u_k^T d_k - [H_k u_k d_k^T + d_k (H_k u_k)^T] / u_k^T d_k ; \tag{15}$$

otherwise, set $H_{k+1} = H_k$, $k = k+1$, and go to Step 1.

A few comments on the algorithm are in order.

The condition (12), or $(d_j^k)^T u_j^k > t_j^k (d_j^k)^T (H_j^k)^{-1} d_j^k$, which implies that $(u_j^k)^T v_j^k > 0$ by Lemma 3.1, assures the positive definiteness of the matrix obtained by the SR1 update; see e.g. Fletcher, Ref. 9. Similarly, the condition $u_k^T d_k > 0$ assures the positive definiteness of the matrix obtained by the BFGS update ($u_k^T d_k \geq 0$ holds whenever f is convex). Therefore, all the matrices H_k, $\breve{H}_j^k$, H_j^k generated by Algorithm 2.1 are positive definite.

The correction parameters ϱ_{k+1} are chosen in such a way to be small for both small and large values of $\tilde{g}_k$. The corrections (7) are used automatically, after every SR1 update, only if the condition $\breve{w}_j^k < \varrho_k |\tilde{g}_j^k|^2$ has been satisfied at least L times. In this way, we have a possibility to eliminate the use of conditions (13), which restrict 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.

The minimization of the quadratic function (10) and the determination of the stepsize t_j^k and the bundle parameter for matrix scaling s_j^k in Step 4 will be discussed in Section 4.

The conditions for matrix scaling in Step 8 and corresponding relations were established empirically.

3. Global Convergence of the Method

In this section, we prove the global convergence of Algorithm 2.1 under the assumption that the function $f: \mathcal{R}^N \to \mathcal{R}$ is convex and the level set $\{x \in \mathcal{R}^N \mid f(x) \leq f(x_1)\}$ is bounded. For this purpose, we assume that the final accuracy tolerance ϵ is set to zero.

Lemma 3.1. Let the function $f: \mathcal{R}^N \to \mathcal{R}$ be convex. Assume that at least $j \geq 1$ null steps are generated in the kth iteration of Algorithm 2.1. Then,

$$-\alpha_{j+1}^k + (d_j^k)^T g_{j+1}^k > -c_1 w_j^k, \qquad \alpha_{j+1}^k \geq 0, \qquad \tilde{\alpha}_j^k \geq 0. \tag{16}$$

If in addition condition (12) holds, then $(u_j^k)^T v_j^k > 0$.

Proof. Since f is convex and $g_{j+1}^k \in \partial f(y_{j+1}^k)$, we can write

$$f(x_k) - f(y_{j+1}^k) + t_j^k (d_j^k)^T g_{j+1}^k \geq 0;$$

thus, $\alpha_{j+1}^k \geq 0$ by (9). The inequality $\tilde{\alpha}_j^k \geq 0$ follows from (11) by induction. Using (9) and the fact that inequality (8) does not hold in a null step, we obtain

$$-\alpha_{j+1}^k + (d_j^k)^T g_{j+1}^k = [f(y_{j+1}^k) - f(x_k)]/t_j^k > -c_1 w_j^k.$$

If $(\tilde{g}_j^k)^T v_j^k < 0$, then $\tilde{g}_j^k \neq 0$ and

$$(d_j^k)^T u_j^k > (d_j^k)^T u_j^k + (\tilde{g}_j^k)^T v_j^k = -t_j^k (d_j^k)^T \tilde{g}_j^k = t_j^k (\tilde{g}_j^k)^T H_j^k \tilde{g}_j^k > 0,$$

by positive definiteness of H_j^k. The last inequality implies that $u_j^k \neq 0$, which yields

$$(u_j^k)^T H_j^k u_j^k > 0.$$

Using the Cauchy inequality, we obtain

$$\begin{aligned} t_j^k [(d_j^k)^T u_j^k]^2 &= t_j^k [(\tilde{g}_j^k)^T H_j^k u_j^k]^2 \\ &\leq t_j^k (\tilde{g}_j^k)^T H_j^k \tilde{g}_j^k (u_j^k)^T H_j^k u_j^k \\ &= (u_j^k)^T H_j^k u_j^k [-t_j^k (d_j^k)^T \tilde{g}_j^k] < (u_j^k)^T H_j^k u_j^k [(d_j^k)^T u_j^k], \end{aligned}$$

which gives

$$(u_j^k)^T v_j^k = (u_j^k)^T H_j^k u_j^k - t_j^k (d_j^k)^T u_j^k > 0. \qquad \square$$

Lemma 3.2. Let at least $j-1 \geq 0$ null steps be generated in the kth iteration of Algorithm 2.1. Then, numbers $\lambda_i^{k,j} \geq 0$, $i = 1, \ldots, j$, exist satisfying

$$(\tilde{g}_j^k, \tilde{\alpha}_j^k) = \sum_{i=1}^{j} \lambda_i^{k,j} (g_i^k, \alpha_i^k), \qquad \sum_{i=1}^{j} \lambda_i^{k,j} = 1. \tag{17}$$

Proof. The proof will proceed by induction. If $j = 1$, then we set $\lambda_1^{k,1} = 1$. Let $n \in \{1, \ldots, j-1\}$, and let (17) hold for j replaced by n. Define

$$\begin{aligned} &\lambda_1^{k,n+1} = \lambda_{m,1}^k + \lambda_{m,3}^k \lambda_1^{k,n}, \\ &\lambda_i^{k,n+1} = \lambda_{m,3}^k \lambda_i^{k,n}, \qquad 2 \leq i \leq n, \\ &\lambda_{n+1}^{k,n+1} = \lambda_{m,2}^k. \end{aligned}$$

It is clear that

$$\lambda_i^{k,n+1} \geq 0, \qquad \text{for all } i \leq n+1,$$

and

$$\sum_{i=1}^{n+1} \lambda_i^{k,n+1} = \lambda_{n,1}^k + \lambda_{n,3}^k \left(\lambda_1^{k,n} + \sum_{i=2}^{n} \lambda_i^{k,n} \right) + \lambda_{n,2}^k = 1.$$

Using the relations (11), in view of $\alpha_1^k = 0$, $g_1^k = g_k$, we obtain

$$\begin{aligned}(\tilde{g}_{n+1}^k, \tilde{\alpha}_{n+1}^k) &= \lambda_{n,1}^k (g_1^k, \alpha_1^k) + \lambda_{n,2}^k (g_{n+1}^k, \alpha_{n+1}^k) + \sum_{i=1}^{n} \lambda_{n,3}^k \lambda_i^{k,n} (g_i^k, \alpha_i^k) \\ &= \sum_{i=1}^{n+1} \lambda_i^{k,n+1} (g_i^k, \alpha_i^k). \end{aligned}$$

□

Lemma 3.3. Let the function f be convex and let the quantities $\tilde{g}_j^k$, $\tilde{\alpha}_j^k$, $j \geq 1$, be generated in the kth iteration of Algorithm 2.1. Then,

$$f(z) \geq f(x_k) + (z - x_k)^T \tilde{g}_j^k - t_{\max} \tilde{\alpha}_j^k, \qquad \text{for all } z \in \mathcal{R}^N. \tag{18}$$

Proof. Lemma 3.2 implies that numbers $\lambda_i^{k,j} \geq 0$, $i = 1, \ldots, j$, exist such that (17) holds. Since $g_i^k \in \partial f(y_i^k)$, we can write

$$f(z) \geq f(y_i^k) + (g_i^k)^T (z - y_i^k) = f(x_k) + (z - x_k)^T g_i^k - t_{i-1}^k \alpha_i^k,$$

for an arbitrary $z \in \mathcal{R}^N$ and all $i = 1, \ldots, j$; t_0^k can be chosen arbitrarily, since $\alpha_1^k = 0$. Using Lemma 3.2, we obtain

$$\begin{aligned} f(z) &= \sum_{i=1}^{j} \lambda_i^{k,j} f(z) \\ &\geq \sum_{i=1}^{j} \lambda_i^{k,j} f(x_k) + (z - x_k)^T \sum_{i=1}^{j} \lambda_i^{k,j} g_i^k - t_{\max} \sum_{i=1}^{j} \lambda_i^{k,j} \alpha_i^k \\ &= f(x^k) + (z - x_k)^T \tilde{g}_j^k - t_{\max} \tilde{\alpha}_j^k. \end{aligned}$$

□

Lemma 3.4. Let the function f be convex. If Algorithm 2.1 terminates due to $w_j^k = 0$, then the point x_k is a global minimizer of f.

Proof. It follows from $w_j^k = 0$ that $\tilde{g}_j^k = 0$, $\tilde{\alpha}_j^k = 0$, and Lemma 3.3 implies that

$$f(z) \geq f(x_k), \qquad \text{for all } z \in \mathcal{R}^N.$$

□

From now on, we assume that Algorithm 2.1 does not terminate, i.e., that $w_j^k > 0$ for all generated indices k and j.

Lemma 3.5. Let the vectors p, q and the numbers $w \geq 0$, $\alpha \geq 0$, $\beta \geq 0$, $M \geq 0$, $c \in (0, 1/2)$ satisfy the conditions

$$w = |p|^2 + 2\alpha, \ \beta + p^T q \leq cw, \qquad \max[|p|, |q|, \sqrt{\alpha}] \leq M.$$

Let

$$Q(\lambda) = |\lambda q + (1-\lambda)p|^2 + 2[\lambda\beta + (1-\lambda)\alpha], \qquad b = (1-2c)/4M.$$

Then,

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

Proof. After straightforward manipulations, we obtain

$$\begin{aligned} Q(\lambda) &= |p|^2 + 2\alpha + 2\lambda[p^T q - |p|^2 + \beta - \alpha] + \lambda^2 |p-q|^2 \\ &\leq w + 2\lambda[cw - \beta - |p|^2/2 + \beta - \alpha] + \lambda^2 (|p| + |q|)^2 \\ &\leq w + 2\lambda (c - 1/2)w + 4\lambda^2 M^2 \\ &= w - \lambda (1-2c)w + 4\lambda^2 M^2. \end{aligned}$$

The last expression reaches its minimum for

$$\begin{aligned} \bar{\lambda} = w(1-2c)/8M^2 &\leq (|p|^2 + 2\alpha)/8M^2 \\ &\leq (M^2 + 2M^2)/8M^2 < 1; \end{aligned}$$

thus, we have

$$\min\{Q(\lambda) \mid \lambda \in [0, 1]\} \leq Q(\bar{\lambda}) \leq w - 2w^2 b^2 + w^2 b^2 = w - w^2 b^2. \qquad \square$$

Lemma 3.6. Let the number of null steps be infinite in the kth iteration of Algorithm 2.1. Then, an index $j_0 \geq 1$ exists such that

$$w_{j+1}^k \leq (\tilde{g}_{j+1}^k)^T H_j^k \tilde{g}_{j+1}^k + 2\tilde{a}_{j+1}^k, \qquad \mathrm{Tr}(H_{j+1}^k) \leq \mathrm{Tr}(H_j^k), \quad \text{for all } j \geq j_0. \tag{19}$$

Proof. If $n_C < L$ for all $j \geq 1$, we can take the index of the null step in which n_C changed last as j_0 (or $j_0 = 1$ if $n_C = 0$ for all $j \geq 1$). To see this, let $j \geq j_0$. Then,

$$w_{j+1}^k = \check{w}_{j+1}^k \quad \text{and} \quad H_{j+1}^k = \check{H}_{j+1}^k.$$

If the SR1 update is not used, then (19) holds with equalities; otherwise, Lemma 3.1 implies that

$$(u_j^k)^T v_j^k > 0,$$

which together with (14) gives (19).

If $n_C < L$ does not hold for all $j \geq 1$, then we set j_0 equal to the index of the null step in which $i_C = 1$ occurred first. Then, the matrix $H_{j_0}^k - \varrho_k I$ is positive definite, since $\check{H}_{j_0}^k$ is positive definite and

$$H_{j_0}^k = \check{H}_{j_0} + \varrho_k I,$$

by the definition of $j_0 \geq 1$. We can prove easily by induction that all matrices $H_j^k - \varrho_k I, j \geq j_0$, are positive definite. If the SR1 update is used, then $i_C = i_U = 1$ and therefore $H_{j+1}^k = \check{H}_{j+1}^k + \varrho_k I$; otherwise, the matrix $\check{H}_{j+1}^k - \varrho_k I = H_j^k - \varrho_k I$ is positive definite, and the more so is the matrix $H_{j+1}^k - \varrho_k I$.

Assume that $j \geq j_0$. If the SR1 update is not used, then $i_U = 0$ and $\check{H}_{j+1}^k = H_j^k$. Thus $\check{w}_{j+1}^k \geq \varrho_k |\tilde{g}_{j+1}^k|^2$, since the matrix $H_j^k - \varrho_k I$ is positive definite. Therefore,

$$w_{j+1}^k = \check{w}_{j+1}^k, \qquad H_{j+1}^k = \check{H}_{j+1}^k = H_j^k,$$

and (19) holds with equalities. If the SR1 update is used, then all conditions (12) and (13) are satisfied and $i_C = i_U = 1$; therefore, the corrections (7), with j replaced by $j+1$, are realized. Using (14), we can write

$$\begin{aligned} w_{j+1}^k = {} & (\tilde{g}_{j+1}^k)^T H_j^k \tilde{g}_{j+1}^k + 2\tilde{\alpha}_{j+1}^k + \varrho_k |\tilde{g}_{j+1}^k|^2 \\ & - [(\tilde{g}_{j+1}^k)^T v_j^k]^2 / (u_j^k)^T v_j^k, \end{aligned}$$

and the first part of (19) follows from the first part of (13). Furthermore, (14) implies

$$\mathrm{Tr}(H_{j+1}^k) = \mathrm{Tr}(H_j^k) + \varrho_k N - |v_j^k|^2 / (u_j^k)^T v_j^k,$$

and the second part of (19) follows from the second part of (13). □

Lemma 3.7. Let the function f be convex and let the number of null steps be infinite in the kth iteration of Algorithm 2.1. Then, the point x_k is a global minimizer of f.

Proof. Since

$$(\tilde{g}_{j+1}^k)^T H_j^k \tilde{g}_{j+1}^k + 2\tilde{\alpha}_{j+1}^k = \varphi(\lambda_{j,1}^k, \lambda_{j,2}^k, \lambda_{j,3}^k) \leq \varphi(0, 0, 1) = w_j^k$$

by (10), Lemma 3.6 implies that

$$w_{j+1}^k \leq w_j^k, \qquad \text{for } j \geq j_0;$$

therefore, the sequences $\{w_j^k\}_{j=1}^\infty$, $\{W_j^k \tilde{g}_j^k\}_{j=1}^\infty$, $\{\tilde{\alpha}_j^k\}_{j=1}^\infty$ are bounded. Moreover, Lemma 3.6 assures the boundedness of the sequences $\{H_j^k\}_{j=1}^\infty$, $\{W_j^k\}_{j=1}^\infty$, which together with

$$|d_j^k| = |H_j^k \tilde{g}_j^k| \leq \|W_j^k\| |W_j^k \tilde{g}_j^k|, \qquad j \geq 1,$$

yields the boundedness of $\{d_j^k\}_{j=1}^{\infty}$. Since $t_j^k \le t_{\max}, j \ge 1$, the sequence $\{y_j^k\}_{j=1}^{\infty}$ is also bounded and the local boundedness of ∂f (see Kiwiel, Ref. 2) implies the boundedness of $\{g_j^k\}_{j=1}^{\infty}$ and $\{W_j^k g_{j+1}^k\}_{j=1}^{\infty}$. Denote

$$M = \sup\{|W_j^k g_{j+1}^k|, |W_j^k \tilde{g}_j^k|, \sqrt{\tilde{\alpha}_j^k} | j \ge j_0\}, \tag{20a}$$

$$b = (1 - 2c_1)/4M, \tag{20b}$$

and assume first that

$$w_j^k > \delta > 0, \qquad \text{for all } j \ge j_0.$$

Since

$$\min\left\{\varphi(\lambda_1, \lambda_2, \lambda_3) | \lambda_i \ge 0, i = 1, 2, 3, \sum_{i=1}^{3} \lambda_i = 1\right\}$$

$$\le \min_{\lambda \in [0,1]} \varphi(0, \lambda, 1 - \lambda),$$

we can use (19), Lemma 3.1, and Lemma 3.5, with

$$p = W_j^k \tilde{g}_j^k, \qquad q = W_j^k g_{j+1}^k,$$

$$w = w_j^k, \qquad \alpha = \tilde{\alpha}_j^k, \qquad \beta = \alpha_{j+1}^k, \qquad c = c_1,$$

to obtain

$$w_{j+1}^k \le (\tilde{g}_{j+1}^k)^T H_j^k \tilde{g}_{j+1}^k + 2\tilde{\alpha}_{j+1}^k$$

$$\le w_j^k - (w_j^k b)^2 < w_j^k - (\delta b)^2, \qquad \text{for } j \ge j_0;$$

thus, for sufficiently large j, we have a contradiction with the assumption $w_j^k > \delta$. Therefore, $w_j^k \to 0$ as $j \to \infty$, which together with

$$w_j^k \ge (\tilde{g}_j^k)^T H_j^k \tilde{g}_j^k + 2\tilde{\alpha}_j^k$$

and the positive definiteness of all matrices H_j^k gives $\tilde{\alpha}_j^k \to 0$ as $j \to \infty$. Since $0 \le \varrho_k |\tilde{g}_j^k|^2 \le w_j^k$ [see (7) and correction conditions in Step 2] and $\varrho_k > 0$, we can write $\tilde{g}_j^k \to 0$ as $j \to \infty$. Using Lemma 3.3 and letting $j \to \infty$, we obtain

$$f(z) \ge f(x_k), \qquad \text{for all } z \in \mathcal{R}^N. \qquad \square$$

Theorem 3.1. Let the function $f: \mathcal{R}^N \to \mathcal{R}$ be convex and suppose that the sequence $\{x_k\}$ is bounded [e.g., when the level set $\{x \in \mathcal{R}^N | f(x) \le f(x_1)\}$ is bounded]. If Algorithm 2.1 terminates in the kth iteration or if the number of null steps is infinite in the kth iteration, then the point x_k is a global minimizer of f. Otherwise, i.e., when the number of iterations is infinite, then every cluster point of $\{x_k\}$ is a global minimizer of f.

Proof. The first assertion follows immediately from Lemma 3.4 and Lemma 3.7. Thus, we can restrict the proof to the case when the number of iterations is infinite. Let $\bar{x}$ be a cluster point of $\{x_k\}$ and let $K \subset \{1, 2, \ldots\}$ be an infinite set such that $x_k \xrightarrow{K} \bar{x}$. The continuity of f implies that $f(x_k) \xrightarrow{K} f(\bar{x})$, and therefore $f(x_k) \downarrow f(\bar{x})$ by monotonicity of $\{f(x_k)\}$, which follows from the descent condition (8). Using the positive definiteness of H_j^k and condition (8), we obtain

$$0 \leq 2c_1 \tilde{\alpha}_k \leq c_1 w_k \leq [f(x_k) - f(x_{k+1})]/t_{\min} \to 0, \qquad k \geq 1; \tag{21}$$

thus, $\tilde{\alpha}_k \to 0$. Furthermore, the correction conditions in Step 2 and relations (7), (21) imply

$$0 \leq c_1 t_{\min} \varrho_k |\tilde{g}_k|^2 \leq c_1 t_{\min} w_k \leq f(x_k) - f(x_{k+1}), \qquad k \geq 1;$$

therefore,

$$c_1 t_{\min} \sum_{k=1}^{\infty} \varrho_k |\tilde{g}_k|^2 \leq f(x_1) - f(\bar{x}) < +\infty. \tag{22}$$

Assume first that

$$|\tilde{g}_k| > \delta > 0, \qquad \text{for all } k \geq 1.$$

Then, it follows from Step 5 of Algorithm 2.1 that

$$\varrho_k \geq \varrho \min[1/\bar{g}, \delta]/k, \qquad k \geq 1.$$

Using (22), we have

$$+\infty = \varrho \delta^2 \min[1/\bar{g}, \delta] \sum_{k=1}^{\infty} (1/k) \leq \sum_{k=1}^{\infty} \varrho_k |g_k|^2 < +\infty,$$

which is a contradiction. Therefore, the infinite set $\bar{K} \subset \{1, 2, \ldots\}$ such that $|\tilde{g}_k| \xrightarrow{\bar{K}} 0$ exists. Since

$$f(z) \geq f(x_k) + (z - x_k)^T \tilde{g}_k - t_{\max} \tilde{\alpha}_k,$$

for all $z \in \mathscr{R}^N$ and $k \geq 1$ by Lemma 3.3, and since the sequence $\{x_k\}$ is bounded, we obtain

$$f(z) \geq f(\bar{x}), \qquad \text{for all } z \in \mathscr{R}^N,$$

by letting $k \xrightarrow{\bar{K}} \infty$. □

Remark 3.1. If we choose $\epsilon > 0$, then Algorithm 2.1 terminates always in a finite number of steps, since $w_j^k \to 0$ as $j \to \infty$ in case the number of null steps is infinite in the kth iteration (see proof of Lemma 3.7) and since $w_k \to 0$ in case the number of iterations is infinite [see (21)].

4. Implementation

In this section, we discuss some details concerning our implementation of the algorithm. Assume that we have the current iteration x_k, $f_k=f(x_k)$, $g_k\in\partial f(x_k)$, $k\geq 1$, 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 $y^j\neq x_k$, $j\in\mathcal{J}_k$, are some trial points. Furthermore, we denote the current aggregate subgradient by $\hat{g}_k$, the stepsize by t_k, and the bundle parameter for scaling by s_k.

After the descent step, we have $\hat{g}_k=g_k$, the positive-definite VM approximation of the inverse Hessian matrix is H_k, and the search direction is $d_k=-H_k g_k$. We search for a suitable stepsize t_k. The significant descent in the last step encourages us to construct the following quadratic approximation of $f(x_k+td_k)$:

$$\psi_Q^k(t)=f_k+td_k^Tg_k+(1/2)t^2d_k^T(H_k)^{-1}d_k=f_k+[t-(1/2)t^2]d_k^Tg_k.$$

The bundle represents the polyhedral function (1) with linearization errors $\beta_j^k\geq 0$ given by (2). For $x=x_k+td_k$, we have the following piecewise linear approximation of $f(x_k+td_k)$:

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

To calculate t_k, we minimize the convex function

$$\psi_k(t)=\max[\psi_Q^k(t),\ \psi_P^k(t)]$$

within $[0, 2]$, since obviously

$$\psi_k(0)=f_k,$$

$$\psi_k(t)\geq\psi_Q^k(t)>f_k,\qquad \text{for } t\notin[0,2],\qquad \text{and } g_k\neq 0.$$

Thus, we set

$$t_k=\arg\min\{\psi_k(t)\,|\,t\in[t_{\min},\min[t_{\max},2,D/|d_k|]]\},$$

where D 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 information about the function f, included in the matrix H_k, is not sufficient for the proper stepsize determination in the nonsmooth case.

After the null step, the unit stepsize is mostly satisfactory, as we have found from numerical experiments. To utilize the bundle and improve the robustness and efficiency of the method, we use the aggregate subgradient $\hat{g}_k$ to construct the linear approximation

$$\psi_L^k(t)=f_k+td_k^T\hat{g}_k$$

of $f(x_k+td_k)$ and set

$$t_k=\arg\min\{\max[\psi_L^k(t),\ \psi_P^k(t)]|t\in[t_{\min},\ \min[1,\ D/|d_k|]]\}.$$

The function $\psi_P^k(t)$ has sometimes no influence on the stepsize determination; then, obviously $t_k=1$. It can mean that the stepsize is too small. Thus, we have introduced the bundle parameter for scaling s_k; we define it as the minimum abscissa of an intersection of the lines which create $\psi_P^k(t)$ and have $d_k^Tg^j>0$ with $\psi_L^k(t)$ and set

$$s_k=\min\{10^{30},\ \beta_j^k/d_k^T(g^j-\hat{g}_k)|d_k^Tg^j>0, j\in\mathcal{J}_k\}.$$

From now on, we use the same notation as in Algorithm 2.1. The minimization of the quadratic function (10) 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 an intersection of the straight lines

$$\partial\tilde{\varphi}/\partial\lambda_1=0,\qquad \partial\tilde{\varphi}/\partial\lambda_2=0,$$

the convexity of $\tilde{\varphi}$ implies that we can restrict the analysis to the lines

$$\lambda_1=0,\qquad \lambda_2=0,\qquad \lambda_1+\lambda_2=1.$$

As an example, we give a formula for minimization within the line $\lambda_1=0$, which we apply regularly in the first null step after any descent step due to $\tilde{g}_1^k=g_k$ and $\tilde{\alpha}_1^k=0$. If $g_{j+1}^k\neq\tilde{g}_j^k$, then set

$$\lambda_{j,2}^k=\min\{1,\max[0,[(d_j^k)^T(g_{j+1}^k-\tilde{g}_j^k)+\tilde{\alpha}_j^k-\alpha_{j+1}^k]/[(g_{j+1}^k-\tilde{g}_j^k)^TH_j^k(g_{j+1}^k-\tilde{g}_j^k)]]\};$$

otherwise, set

$$\lambda_{j,2}^k=0,\qquad \text{for } \tilde{\alpha}_j^k<\alpha_{j+1}^k,$$

$$\lambda_{j,2}^k=1,\qquad \text{for } \tilde{\alpha}_j^k\geq\alpha_{j+1}^k.$$

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

Step 3′. If $w_j^k\leq\epsilon$ and if either $\Delta_k/\max[1,\ f(x_k)]<2\epsilon_f$, for $j=1$, or $w_{j-1}^k\leq\epsilon$, for $j>1$, then stop.

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

Step 5′. If $|f(y_{j+1}^k)-f(x_k)| \geq 10^{-5}\Delta_k$, then set

$$\Delta = |f(y_{j+1}^k)-f(x_k)|;$$

otherwise, set $\Delta=\Delta_k$. If

$$\Delta/\max[1, f(y_{j+1}^k)] \leq \epsilon_f \quad \text{or} \quad f(y_{j+1}^k) \approx f(x_k),$$

then set $n_f=n_f+1$; otherwise, set $n_f=0$. If $n_f \geq m_f$, then stop. If (8) holds, then set

$$x_{k+1}=y_{j+1}^k, \quad g_{k+1}=g_{j+1}^k, \quad t_k=t_j^k, \quad d_k=d_j^k,$$

$$\tilde{\alpha}_k=\tilde{\alpha}_j^k, \quad \tilde{g}_k=\tilde{g}_j^k, \quad w_k=w_j^k, \quad H_k=H_j^k,$$

$$\varrho_{k+1}=\varrho \min[1/\min[|\tilde{g}_k|, \bar{g}], |\tilde{g}_k|]/(k+1), \quad \Delta_{k+1}=\Delta;$$

otherwise, set $\Delta_{k+1}=\Delta_k$ and go to Step 6. If $g_{k+1}=g_k$ and $t_j^k < t_{\max}/2$, then set $i_E=1$, $H_{k+1}=H_k$, $k \approx k+1$, and go to Step 1; otherwise go to Step 8.

Finally, if we use the algorithm for a function f which is not convex, it can happen that $\alpha_{j+1}^k < 0$ and cause so many difficulties. Thus, we define α_{j+1}^k as the absolute value of the quantity in (9). Note that a nonconvex version of the method is being prepared.

5. Numerical Examples

The above concept was implemented in FORTRAN 77 as VMC. In this section, we compare our results for 28 standard test problems from literature (Problem 1 is smooth, all the others are nonsmooth) with those obtained by a smooth VM method (SVM) given in Lukšan (Ref. 10) with the update U8, controlled scaling, and backward Taylor stabilization) and by the proximal bundle method PBL mentioned in Lukšan and Vlček (Ref. 6). Note that we tested a number of other standard VM codes (e.g., Harwell routine VA13D), and the results were similar; nevertheless, we had slight difficulties to reach the desired accuracy. Problems 1–16 are described in Mäkelä and Neittaanmäki (Ref. 3); Problems 17 and 18 in Zowe (Ref. 11); Problems 19–21 in Kiwiel (Ref. 12); Problem 22 in Bihain (Ref. 13); Problem 23 in Facchinei and Lucidi (Ref. 14); Problems 24–28 in Lukšan (Ref. 15); and Problem 25 also in Bandler, Srinivasan, and Charalambous (Ref. 16).

Table 1. Test problems.

No.	N	Problem	Minimum
1	2	Rosenbrock	0
2	2	Crescent	0
3	2	CB2	1.9522245
4	2	CB3	2.0
5	2	DEM	−3.0
6	2	QL	7.20
7	2	LQ	−1.4142136
8	2	Mifflin1	−1.0
9	2	Mifflin2	−1.0
10	4	Rosen	−44.0
11	5	Shor	22.600162
12	10	Maxquad1	−0.8414083
13	20	Maxq	0
14	20	Maxl	0
15	48	TR48	−638565.0
16	50	Goffin	0
17	6	El Attar	0.5598131
18	2	Wolfe	−8.0
19	50	MXHILB	0
20	50	L1HILB	0
21	5	Colvillel	−32.348679
22	10	Gill	9.7857721
23	12	Steiner2	16.703838
24	5	EXP	0.0001224
25	6	TRANSF	0.1972906
26	7	Wong1	680.63006
27	10	Wong2	24.306209
28	20	Wong3	133.72828

In Table 1, we give the optimal values of the tested functions. The parameters of the algorithm had the values

$$t_{\min}=10^{-10}, \quad t_{\max}=10^{3}, \quad c_1=10^{-4},$$
$$\epsilon=5\cdot 10^{-7}, \quad \epsilon_f=10^{-7}, \quad \varrho=10^{-6},$$
$$\bar{g}=10^{3}, \quad L=1, \quad \sigma=100,$$
$$\mathcal{J}_k=\{\max[1,k-N-2],\ldots,k\}, \quad k\geq 1;$$

also, $m_f=2$ for Problems 1–23 and 26–28, $m_f=5$ for Problem 24, and $m_f=3$ for Problem 25.

Our results are summarized in Table 2, in which the following notation is used. N_i is the number of iterations (number of descent steps + null steps such that $i_E=0$ for VMC); N_f is the number of objective function (and also

Table 2. Test results.

	SVM			VMC				PBL		
No.	N_i	N_f	F	N_i	N_f	F	D	N_i	N_f	F
1	37	54	0.198E−20	36	36	0.416E−10	1	42	45	0.381E−06
2	29	61	0.174E−07	53	54	0.189E−05	1	18	20	0.462E−08
3	21	57	1.9522245	17	17	1.9522246	1	31	33	1.9522245
4	22	69	2.0000001	17	17	2.0000000	10^3	14	16	2.0000000
5	36	78	−3.0000000	20	21	−3.0000000	10^3	17	19	−3.0000000
6	25	71	7.2000000	21	22	7.2000001	10^3	13	15	7.2000015
7	17	52	−1.4142136	7	8	−1.4142136	10^3	11	12	−1.4142136
8	1	22	−0.8000000	143	186	−0.9999980	10	66	68	−0.9999994
9	24	89	−1.0000000	28	28	−1.0000000	1	13	15	−1.0000000
10	34	89	−44.000000	37	38	−43.999991	1	43	45	−43.999999
11	46	128	22.600162	37	38	−22.600163	10^3	27	29	22.600162
12	126	303	−0.8414079	87	87	−0.8413999	1	74	75	−0.8414083
13	94	173	0.460E−07	135	135	0.775E−06	10	150	151	0.167E−06
14	136	420	0.408E−07	22	23	0	10^3	39	40	0.124E−12
15	380	992	−638564.54	305	306	−638561.57	10^3	245	251	−638530.48
16	386	984	0.294E−05	240	242	0.258E−05	10^3	52	53	0.117E−11
17	74	180	0.5598152	114	115	0.5598147	1	92	93	0.5598157
18	18	36	−8.0000000	18	18	−7.9999995	1	43	46	−8.0000000
19	59	147	0.978E−06	67	75	0.134E−05	10^3	19	20	0.513E−08
20	78	123	0.702E−06	68	68	0.122E−05	10	27	28	0.234E−07
21	58	191	−32.348661	64	64	−32.348595	0.1	60	62	−32.348679
22	873	1711	9.7858074	124	124	9.7858075	10	160	162	9.7857723
23	79	186	16.703839	78	79	16.703848	1	128	143	16.703862
24	86	204	0.0001225	81	82	0.0001295	0.1	95	102	0.0001224
25	67	161	0.1972907	73	73	0.1972932	0.05	153	157	0.1972973
26	121	301	680.63043	51	52	680.63026	1	100	102	680.63007
27	79	296	24.306209	96	97	24.306219	10	102	104	24.306213
28	140	523	133.72838	238	239	133.72841	10	184	192	133.72832
$\sum$	3146	7701		2277	2344			2018	2098	
	Time = 17.30 (3.29) sec			Time = 3.46 (2.55) sec				Time = 8.07 (6.40) sec		

subgradient) evaluations; F is the objective function value at termination; and D is the maximum allowable distance in one step (see Section 4); the values of D were chosen experimentally. For better comparison, we give in the last line two time data: the first time concerns all 28 problems; the second time (in parentheses) concerns only 27 problems, with Problem 22 removed.

As a conclusion, from our limited numerical experiments, we state the following facts:

(i) the standard VM method is able to find a solution to almost all problems (here, it failed once) and the computational time can be essentially less than for proximal bundle methods;

(ii) our method is comparable with proximal bundle methods in the number of function and subgradient evaluations, but the computational time can be less significantly;

(iii) although our method is designed for convex functions, it can be applied also to some nonconvex problems.

References

1. CLARKE, F. H., *Optimization and Nonsmooth Analysis*, Wiley (Interscience), New York, New York, 1983.
2. KIWIEL, K. C., *Methods of Descent for Nondifferentiable Optimization*, Lecture Notes in Mathematics, Springer Verlag, Berlin, Germany, Vol. 1133, 1985.
3. MÄKELÄ, M. M., and NEITTAANMÄKI, P., *Nonsmooth Optimization*, World Scientific Publishing Company, London, England, 1992.
4. SCHRAMM, H., and ZOWE, J., *A Version of the Bundle Idea for Minimizing a Nonsmooth Function: Conceptual Idea, Convergence Analysis, Numerical Results*, SIAM Journal on Optimization, Vol. 2, pp. 121–152, 1992.
5. LEMARÉCHAL, C., and SAGASTIZÁBAL, C., *Variable Metric Bundle Methods: From Conceptual to Implementable Forms*, Mathematical Programming, Vol. 76, pp. 393–410, 1997.
6. LUKŠAN, L., and VLČEK, *A Bundle-Newton Method for Nonsmooth Unconstrained Minimization*, Mathematical Programming, Vol. 83, pp. 373–391, 1998.
7. LUKŠAN, L., *Dual Method for Solving a Special Problem of Quadratic Programming as a Subproblem of Linearly Constrained Nonlinear Minimax Approximation*, Kybernetika, Vol. 20, pp. 445–457, 1984.
8. LEMARÉCHAL, C., *Numerical Experiments in Nonsmooth Optimization*, Progress in Nonsmooth Optimization, Edited by E. A. Nurminski, IIASA, Laxenburg, Austria, pp. 61–84, 1982.
9. FLETCHER, R., *Practical Methods of Optimization*, John Wiley and Sons, Chichester, England, 1987.
10. LUKŠAN, L., *Computational Experience with Known Variable Metric Updates*, Journal of Optimization Theory and Applications, Vol. 83, pp. 27–47, 1994.

11. ZOWE, J., *Nondifferentiable Optimization*, Computational Mathematical Programming, Edited by K. Schittkowski, Springer Verlag, Berlin, Germany, pp. 323–356, 1985.
12. KIWIEL, K. C., *An Ellipsoid Trust Region Bundle Method for Nonsmooth Convex Minimization*, SIAM Journal on Control and Optimization, Vol. 27, pp. 737–757, 1989.
13. BIHAIN, A., *Optimization of Upper Semidifferentiable Functions*, Journal of Optimization Theory and Applications, Vol. 4, pp. 545–568, 1984.
14. 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.
15. LUKŠAN, L., *A Compact Variable Metric Algorithm for Linear Minimax Approximation*, Computing, Vol. 36, pp. 355–373, 1986.
16. BANDLER, J. W., SRINIVASAN, T. V., and CHARALAMBOUS, C., *Minimax Optimization of Networks by Grazor Search*, IEEE Transactions on Microwave Theory and Techniques, Vol. 20, pp. 596–604, 1972.