JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 111, No. 2, pp. 407–430, November 2001 (© 2001)

Globally Convergent Variable Metric Method for Nonconvex Nondifferentiable Unconstrained Minimization[1]

J. VLČEK[2] AND L. LUKŠAN[2]

Communicated by L. C. W. Dixon

Abstract. A special variable metric method is given for finding the 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.

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

1. Introduction

This paper is devoted to minimizing a locally Lipschitz continuous function $f: \mathscr{R}^N \to \mathscr{R}$. Such function is differentiable at y for all y except in a set of zero (Lebesgue) measure; see Ref. 1. We assume that, for each $y \in \mathscr{R}^N$, 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 Ref. 2.

The most efficient globally convergent methods for nonconvex nonsmooth optimization are various versions of the bundle methods; see e.g. Refs. 1, 3–6. Essentially, instead of the singleton

$$f_k = f(x_k), \; g(x_k) \in \partial f(x_k),$$

the bundle $\{(f_j^k, g_j): j \in \mathscr{J}_k\}$ is used in the kth iteration, $k \geq 1$, where

$$f_j^k = f(y_j) + (x_k - y_j)^T g_j, \; g_j \in \partial f(y_j), \qquad \mathscr{J}_k \subset \{1, \ldots k\},$$

[1]This work was supported by Grant 201/00/0080 of the Czech Republic Grant Agency.

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

$x_1, \ldots, x_k$ are iterates and $y_1, \ldots, y_k$ are trial points. The piecewise linear function

$$\check{f}_k(x) = \max_{j \in \mathscr{J}_k} \{ f_k + (x - x_k)^T g_j - \beta_j^k \} \tag{1}$$

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

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

is determined, with the matrix B_k usually positive definite [the additional quadratic term in (2) has a similar significance as in the trust region approach]. The minimization subproblem (2) can be replaced by the quadratic programming subproblem

$$(d_k, \xi_k) = \arg\min_{(d,\xi) \in \mathscr{R}^{N+1}} \{ (1/2) d^T B_k d + \xi \}, \qquad \text{s.t.} -\beta_j^k + d^T g_j \leq \xi, j \in \mathscr{J}_k. \tag{3}$$

Unfortunately, a solution of this subproblem is time consuming when the number of variables is large. Therefore, avoiding such extensive computations is desirable. The development of our variable metric (VM) method was motivated by the observation that standard VM methods are relatively robust and efficient even in the nonsmooth case; see e.g. Ref. 7 and also our experiments in Ref. 8. The advantage of standard VM methods consists in the fact that the quadratic programming subproblem (3) need not be solved. Although the 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 appear sometimes in practical computations.

In Ref. 8, we proposed a convex VM method based on ideas essential for bundle methods, especially the utilization of null steps with aggregation of subgradients and application of linearization errors. The null steps serve to obtain sufficient information about a minimized nondifferentiable function when a serious descent condition is not satisfied. This method utilizes VM updates in both the descent and null steps whenever the positive-definiteness conditions (and some others) are satisfied. Since these updates accumulate information about a nondifferentiable function, the quadratic programming dimension can be decreased considerably. In fact, two function values and gradients suffice in the aggregation following the null step.

Our convex VM method performs well in practice even for nonconvex functions, but we are not able to prove its global convergence without assumption of convexity.

In this paper, we are proposing a new nonsmooth VM method which is globally convergent also in the nonconvex case. For this purpose, we involve a special line search procedure which provides stepsizes satisfying the conditions required for global convergence. Fortunately, these conditions are relatively weak so that the initial stepsize is usually taken. Furthermore, linearization errors have been generalized and the notation was modified.

This paper is organized as follows. Section 2 is devoted to the description of the new method. The algorithm presented there has a relatively complicated logical structure, since many subtle effects have to be considered. For keeping some properties of the VM matrices, especially sufficient positive definiteness, we utilize a correction with indicator i_C at Step 2 and a special update control with indicator i_U at Step 7 and Step 9, although these procedures have only marginal significance in practical computations. An extrapolation with indicator i_E at Step 4 and Step 9 and a matrix scaling with indicator i_S at Step 8 increases the efficiency of the method.

Section 3 contains the global convergence theory leading into Theorem 3.2 and implying that the number of steps is finite when the required precision is positive; see Remark 3.1. Section 4 gives more details concerning the implementation of Algorithm 2.1 which are not required for the global convergence theory, but have a significant influence on the computational efficiency. Numerical experiments confirming the algorithm efficiency are described in Section 5.

2. Derivation of the Method

The algorithm given below generates a sequence of basic points $\{x_k\}_{k=1}^{\infty} \subset \mathscr{R}^N$ which should converge to a minimizer of $f: \mathscr{R}^N \to \mathscr{R}$ and a sequence of trial points $\{y_k\}$ satisfying

$$x_{k+1} = x_k + t_L^k d_k, \qquad y_{k+1} = x_k + t_R^k d_k, \qquad \text{for } k \geq 1,$$

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, $d_k = -H_k \tilde{g}_k$ is a direction vector, $\tilde{g}_k$ is an aggregate subgradient, and the VM matrix H_k accumulates information about the previous subgradients and represents an approximation of the inverse Hessian matrix if function f is smooth.

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 the multipliers $\lambda_{k,i}$, $i \in \{1, 2, 3\}$ can be determined easily by minimization of a simple quadratic function, which depends on these three subgradients and two subgradient locality measures; see Step 6 of Algorithm 2.1, where the matrix $W_k = H_k^{1/2}$ is not calculated actually, since it appears only in the inner products, e.g., $|W_k \tilde{g}_k|^2 = \tilde{g}_k^T H_k \tilde{g}_k$. This approach retains global convergence, but eliminates the solution of the rather complicated quadratic programming subproblem (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 deteriorates slightly 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 $d_{k+1}^T g_m \geq 0$ occurs in the numerical experiments is much more frequent in the simplified case.

The matrices H_k are generated by using the usual VM updates. After the null steps, a symmetric rank-one update (SR1, Ref. 9) is used, since it gives the possibility of preserving the boundedness and other properties of the generated matrices as required in the proof of global convergence. Because these properties are not necessary after the descent steps, the standard BFGS update (see Ref. 9) appears to be more suitable.

The efficiency of the algorithm is very sensitive to the initial stepsize selection, though 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 (see Section 4 for details). 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 also has no influence on the global convergence but improves the efficiency.

The proof of global convergence requires the boundedness of both the direction vectors length and the matrices H_k^{-1}. Thus, the matrix H_k is scaled appropriately and the correction ϱI, $\varrho > 0$, is added to H_k if needed.

In the descent steps, if the subgradients are identical in consecutive iterations, we extrapolate doubling the stepsize, if possible, in order to exit such region sooner.

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

Algorithm 2.1.

Data. A stepsize bound $t_{\max} > 2$, an auxiliary stepsize $t_{\text{aux}} \in (0, 1)$, a final accuracy tolerance $\epsilon \geq 0$, correction parameters $\varrho \in (0, 1)$ and $L \geq 1$, matrix scaling parameter $C > 1$, and a constant $D > 0$ for the direction vector length control (see Step 2) and for the line search.

Step 0. Initiation. Choose the starting point $x_1 \in \mathscr{R}^N$ and the 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 $\mu = 1$, the correction, extrapolation, scaling, and updating indicators $i_C = i_E = i_S = i_U = 0$, the correction and scaling counters $n_C = n_S = 0$, and the iteration counter $k = 1$.

Step 1. Descent Step Initialization. Initialize the aggregate subgradient $\tilde{g}_k = g_k$ and the aggregate subgradient locality measure $\tilde{\alpha}_k = 0$ and set $m = k$.

Step 2. Correction. If $|\check{H}_k \tilde{g}_k| > D$, multiply the matrix $\check{H}_k$ by factor $D / |\check{H}_k \tilde{g}_k|$. 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

$$w_k = \check{w}_k + \varrho |\tilde{g}_k|^2, \qquad H_k = \check{H}_k + \varrho I, \tag{4}$$

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

Step 3. Stopping Criterion. If $w_k \leq \epsilon$, then stop.

Step 4. Line Search. Set $d_k = -H_k \tilde{g}_k$ and $n_S = n_S + 1$. If $i_E = 0$, then determine the initial stepsize $t_I^k \in [t_{\text{aux}}, t_{\max}]$; otherwise, set $t_I^k = 2t_L^{k-1}$ (extrapolation) and $i_E = 0$. By the line search procedure given below, find the stepsizes t_L^k and t_R^k and the corresponding quantities

$$x_{k+1} = x_k + t_L^k d_k,$$

$$y_{k+1} = x_k + t_R^k d_k,$$

$$f_{k+1} = f(x_{k+1}),$$

$$g_{k+1} \in \partial f(y_{k+1}), \qquad \alpha_{k+1}, \qquad \beta_{k+1}.$$

If $t_L^k > 0$, a descent step is taken; otherwise, a null step occurs; see Lemma 2.1 for the relations holding in these steps.

Step 5. Update Preparation. Set $u_k = g_{k+1} - g_m$. Update the matrix scaling parameter $\mu > 0$ (see Section 4). If $t_L^k > 0$ (descent step), go to Step 8.

Step 6. Aggregation. Determine the multipliers

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

which minimize the function

$$\varphi(\lambda_1, \lambda_2, \lambda_3) = |\lambda_1 W_k g_m + \lambda_2 W_k g_{k+1} + \lambda_3 W_k \tilde{g}_k|^2 + 2[\lambda_2 \alpha_{k+1} + \lambda_3 \tilde{\alpha}_k], \tag{5}$$

where $W_k = H_k^{1/2}$; note that W_k is not computed, see above. Set

$$\tilde{g}_{k+1} = \lambda_{k,1} g_m + \lambda_{k,2} g_{k+1} + \lambda_{k,3} \tilde{g}_k, \tag{6a}$$

$$\tilde{\alpha}_{k+1} = \lambda_{k,2} \alpha_{k+1} + \lambda_{k,3} \tilde{\alpha}_k. \tag{6b}$$

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

$$\tilde{g}_k^T v_k < 0 \tag{7}$$

and either $i_C = 0$ or

$$\varrho |\tilde{g}_{k+1}|^2 \leq (\tilde{g}_{k+1}^T v_k)^2 / u_k^T v_k \quad \text{and} \quad \varrho N \leq |v_k|^2 / u_k^T v_k, \tag{8}$$

then set $i_U = 1$ and

$$\check{H}_{k+1} = H_k - v_k v_k^T / u_k^T v_k; \tag{9}$$

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 $\mu > C, n_S > 3$, and $i_S > 1$, set

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

and go to Step 1.

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

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

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

A few comments on the algorithm are in order.

The problem of minimizing the function (5) in Step 6 is the dual to the following primal problem:

$$\underset{d \in \mathscr{R}^N}{\text{minimize}} \{(1/2) d^T H_k^{-1} d + \max[d^T g_m, -\alpha_{k+1} + d^T g_{k+1}, -\tilde{\alpha}_k + d^T \tilde{g}_k]\}. \tag{10}$$

The minimization of the quadratic function (5) and the determination of the initial stepsize t_I^k at Step 4 will be discussed in Section 4.

Condition (7), 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 3.1, assures the positive definiteness of the matrix obtained by the SR1 update; see e.g. Ref. 9. Similarly, satisfying $u_k^T d_k > 0$ 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 $\check{H}_k$, H_k generated by Algorithm 2.1 are positive definite. The conditions for matrix scaling at Step 8 and the corresponding relations were established empirically.

The constant D is meant to be a maximum reasonable distance in one step. Provided the level set $\{x \in \mathscr{R}^N : f(x) \leq f(x_1)\}$ is bounded, the choice

$$D \approx \sup\{|x - y| : \max[f(x), f(y)] \leq f(x_1)\}$$

seems to be natural.

The correction (4) 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. This gives a possibility to eliminate the use of conditions (8), 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.

To generalize the linearization errors to the nonconvex case, the subgradient locality measures α_k introduced in Ref. 3 and the auxiliary quantities β_k have been used. They are defined in Lemma 2.1. The first absolute value in (15) is not necessary, but it improves significantly the numerical results.

We will now present a line search algorithm and subsequent two lemmas based on the ideas contained in Ref. 3. The value of γ should correspond to the measure of nonconvexity of f. The bigger value of γ can slow down the convergence for convex function, but accelerate it in a nonconvex case. The range for c_T[i.e., $c_T \in (c_L, c_R - c_A)$] and the corresponding condition at step (ii) are important for the finiteness of the line search procedure; see the proof of Lemma 2.2. The motivation of the other conditions is also mainly theoretical, to prove global convergence. The auxiliary stepsize t_{aux} is not necessary for global convergence. It helps to exit the line search procedure sooner when the descent condition at step (iii) is satisfied. The choice of the interpolation procedure mentioned at step (v) is not essential for our theoretical results. We combine quadratic interpolation with the bisection.

Line Search Procedure.

Data. Positive parameters c_A, c_L, c_R, c_T satisfying $c_T + c_A < c_R < 1/2$ and $c_L < c_T$, a distance measure parameter $\gamma > 0$, an interpolation parameter $\kappa \in (0, 1/2)$, and a locality measure parameter $\omega \geq 1$. All of these parameters are constant.

Step (i) Set $t_A = 0$ and $t = t_U = t_I^k$.

Step (ii) Calculate $f(x_k + td_k)$, $g \in \partial f(x_k + td_k)$ and

$$\beta = \max[|f_k - f(x_k + td_k) + td_k^T g|, \gamma(t|d_k|)^{\omega}]. \tag{11}$$

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

Step (iii) If $f(x_k + td_k) \leq f_k - c_L t w_k$ and either $t \geq t_{\text{aux}}$ or $\beta > c_A w_k$, set $t_R = t_L = t$, $\alpha = 0$, and return.

Step (iv) If $-\beta + d_k^T g \geq -c_R w_k$ and $(t - t_A)|d_k| \leq D$, set $t_R = t$, $t_L = 0$, $\alpha = \beta$, and return.

Step (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 Step (ii).

Lemma 2.1. Let the line search procedure terminate. Then, on its output, at Step 4 of Algorithm 2.1, the serious descent criterion

$$f_{k+1} \leq f_k - c_L t_L^k w_k \tag{12}$$

is satisfied and $f(z_{k+1}) \leq f_k$ holds for $z_{k+1} = x_k + t_A^k d_k$. Moreover, one has $t_R^k \leq t_1^k$ and either (a) or (b) below:

(a) $t_R^k = t_L^k > 0$, $\alpha_{k+1} = 0$, and

$$t_L^k \geq t_{\text{aux}} \quad \text{or} \quad \beta_{k+1} > c_A w_k \tag{13}$$

in the descent step;

(b) $t_R^k > t_L^k = 0$, $\alpha_{k+1} = \beta_{k+1}$, and

$$-\alpha_{k+1} + d_k^T g_{k+1} \geq -c_R w_k, \qquad |y_{k+1} - z_{k+1}| \leq D \tag{14}$$

in the null step, where

$$\beta_{k+1} = \max[|f_k - f(y_{k+1}) + (y_{k+1} - x_k)^T g_{k+1}|, \gamma|y_{k+1} - x_k|^{\omega}]. \tag{15}$$

Proof. The relations mentioned above hold obviously at the termination of the line search procedure. □

Lemma 2.2. Let f satisfy the following semismoothness hypothesis (see Remark 3.3.4 in Ref. 3): for any $x \in \mathscr{R}^N$, $d \in \mathscr{R}^N$, and sequences $\{\hat{t}_i\} \subset \mathscr{R}_+$ and $\{\hat{g}_i\} \subset \mathscr{R}^N$ 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.

Proof. For contradiction purposes, assume that the search does not terminate. Let $t^i, t_A^i, t_U^i, g^i, \beta^i$ denote the values of t, t_A, t_U, g, β after the ith iteration of the procedure; hence,

$$t^i \in \{t_A^i, t_U^i\}, \qquad \text{for all } i.$$

Since

$$t_A^i \leq t_A^{i+1} \leq t_U^{i+1} \leq t_U^i,\ t_U^{i+1} - t_A^{i+1} \leq (1-\kappa)(t_U^i - t_A^i), \qquad \text{for all } i,$$

there exists $t^* \geq 0$ satisfying

$$t_A^i \uparrow t^*, \qquad t_U^i \downarrow t^*, \qquad t^i \to t^*.$$

Let

$$S = \{t \geq 0 : f(x_k + td_k) \leq f_k - c_T t w_k\}.$$

Since $\{t_A^i\} \subset S$, $t_A^i \uparrow t^*$, and f is continuous, we have

$$f(x_k + t^* d_k) \leq f_k - c_T t^* w_k, \tag{16}$$

i.e., $t^* \in S$. Let

$$I = \{i : t^i \notin S\}.$$

We prove first that set I is infinite. If there existed $i_0 \in I$ satisfying $t^i \in S$ for all $i > i_0$, it would be

$$t_U^{i_0} = t_U^i \downarrow t^*, \qquad \text{for all } i > i_0,$$

implying $t^* = t_U^{i_0} \notin S$, which is a contradiction. Thus, I is infinite and we have

$$f(x_k + t^i d_k) > f_k - c_T t^i w_k, \qquad \text{for all } i \in I.$$

By (16), we obtain

$$[f(x_k + t^i d_k) - f(x_k + t^* d_k)]/(t^i - t^*) > -c_T w_k,$$

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

$$\begin{aligned} -c_T w_k &\leq \liminf_{i\to\infty,\, i\in I} [f(x_k + t^* d_k + (t^i - t^*) d_k) - f(x_k + t^* d_k)]/(t^i - t^*) \\ &\leq \limsup_{i\to\infty,\, i\in I} d_k^T g^i, \end{aligned} \tag{17}$$

in view of $t_U^i \downarrow t^*$ and $g^i \in \partial f(x_k + t^i d_k)$. For sufficiently large i, we have $(t^i - t_A^i)|d_k| \leq D$. We will consider the following two cases:

(a) Suppose that $t^* > 0$. By (16), $c_L < c_T$, and $t^i \to t^*$, it holds that

$$f(x_k + t^i d_k) \leq f_k - c_L t^i w_k,$$

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 (17).

(b) Suppose that $t^* = 0$. 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 Ref. 3. The search does not terminate; thus,

$$-\beta^i + d_k^T g^i < -c_R w_k$$

at Step (iv) for all large 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 (17). $\square$

3. Global Convergence of the Method

In this section, we prove the global convergence of Algorithm 2.1 under the assumptions that the function $f: \mathscr{R}^N \to \mathscr{R}$ is locally Lipschitz continuous, that the level set $\{x \in \mathscr{R}^N : f(x) \leq f(x_1)\}$ is bounded, and that each execution of the line search procedure is finite. After three technical results (Lemma 3.1 to Lemma 3.3), we will prove (Theorem 3.1) that $w_k = 0$ implies $0 \in \partial f(x_k)$; i.e., x_k is a stationary point for the function f. For infinite $\{x_k\}$, we will show first (Lemma 3.4) that $x_k \xrightarrow{K} \bar{x}$ and $w_k \xrightarrow{K} 0$ [i.e., $\tilde{g}_k \xrightarrow{K} 0$ and $\tilde{\alpha}_k \xrightarrow{K} 0$] for some subset $K \subset \{1, 2, \ldots\}$ imply $0 \in \partial f(\bar{x})$; this assertion requires the uniform positive definiteness of the matrices $\{H_k\}$, which is ensured by the correction (4). Furthermore, using the technical Lemma 3.5 and the fact that the sequence w_k is nonincreasing in the null steps [see (23)] thanks to using the SR1 update and a special update control at Step 7, we will prove that the infinite sequence of subsequent null steps with $x_k = x^*$ implies $0 \in \partial f(x^*)$ (Lemma 3.6). Finally, Theorem 3.2 connects all the results, showing that every cluster point of $\{x_k\}$ is stationary for f. Moreover, we will state that the number of steps cannot be infinite for the positive final accuracy tolerance ϵ (Remark 3.1).

Lemma 3.1. At the kth iteration of Algorithm 2.1, one has

$$w_k = \tilde{g}_k^T H_k \tilde{g}_k + 2\tilde{\alpha}_k,$$

$$w_k \geq \varrho |\tilde{g}_k|^2,$$

$$w_k \geq 2\tilde{\alpha}_k \geq 0,$$

$$\alpha_{k+1} \geq \gamma |y_{k+1} - x_{k+1}|^{\omega}.$$

In addition, if condition (7) in Step 7 holds, then $u_k^T v_k > 0$.

Proof. We recall first that $\alpha_{k+1} = \beta_{k+1}$ for null steps and $\alpha_{k+1} = 0$ for descent steps (see Lemma 2.1). Considering that $\tilde{\alpha}_k \geq 0$ by (6) and (15), the relations

$$w_k = \tilde{g}_k^T H_k \tilde{g}_k + 2\tilde{\alpha}_k, \qquad w_k \geq \varrho |\tilde{g}_k|^2, \qquad w_k \geq 2\tilde{\alpha}_k$$

follow immediately from (4). Since $x_k = x_{k+1}$ for null steps and $|y_{k+1} - x_{k+1}| = 0$ for descent steps, by Lemma 2.1 and Step 4, we have always

$$\alpha_{k+1} \geq \gamma |y_{k+1} - x_{k+1}|^{\omega}$$

from (15).

If $\tilde{g}_k^T v_k < 0$, then $\tilde{g}_k \neq 0$; and since

$$v_k = H_k u_k - t_R^k d_k,$$

we get

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

by the positive definiteness of H_k. The last inequality implies $u_k \neq 0$, which yields

$$u_k^T H_k u_k > 0.$$

Using the Cauchy inequality, we obtain

$$\begin{aligned}(d_k^T u_k)^2 &= (\tilde{g}_k^T H_k u_k)^2 \\ &\leq \tilde{g}_k^T H_k \tilde{g}_k u_k^T H_k u_k \\ &= u_k^T H_k u_k(-d_k^T \tilde{g}_k) \\ &< u_k^T H_k u_k d_k^T u_k / t_R^k,\end{aligned}$$

which gives

$$0 < u_k^T H_k u_k - t_R^k d_k^T u_k = u_k^T v_k. \qquad \square$$

Lemma 3.2. Suppose that Algorithm 2.1 did not stop before the kth iteration. Then, numbers $\lambda_j^k \geq 0, j = 1, \ldots, k$, and $\tilde{\sigma}_k$ exist satisfying

$$(\tilde{g}_k, \tilde{\sigma}_k) = \sum_{j=1}^{k} \lambda_j^k (g_j, |y_j - x_k|), \tag{18a}$$

$$\sum_{j=1}^{k} \lambda_j^k = 1, \qquad \tilde{\alpha}_k \geq \gamma \tilde{\sigma}_k^{\omega}. \tag{18b}$$

Proof. We assign to k an index m defined at Step 1 (index of the iteration after the last descent step preceding iteration k, with the property $x_j = x_m$ for $j = m, \ldots, k$). First, we establish the existence of numbers $\lambda_j^k \geq 0, j = m, \ldots, k$, satisfying

$$(\tilde{g}_k, \tilde{\alpha}_k) = \sum_{j=m}^{k} \lambda_j^k (g_j, \alpha_j), \qquad \sum_{j=m}^{k} \lambda_j^k = 1. \tag{19}$$

The proof will proceed by induction. If $k = m$, then we can set $\lambda_m^m = 1$, since $\tilde{g}_m = g_m$ and $\tilde{\alpha}_m = 0$ at Step 1 and $\alpha_m = 0$ by (13). Let $k > m, i \in \{m, \ldots, k-1\}$, and let (19) hold for k replaced by i. We define

$$\begin{aligned}\lambda_m^{i+1} &= \lambda_{i,1} + \lambda_{i,3} \lambda_m^i, \\ \lambda_j^{i+1} &= \lambda_{i,3} \lambda_j^i, \qquad \text{for } j = m+1, \ldots, i, \\ \lambda_{i+1}^{i+1} &= \lambda_{i,2}.\end{aligned}$$

It is clear that

$$\lambda_j^{i+1} \geq 0, \qquad \text{for all } j = m, \ldots, i+1,$$

and

$$\sum_{j=m}^{i+1} \lambda_j^{i+1} = \lambda_{i,1} + \lambda_{i,3} \left(\lambda_m^i + \sum_{j=m+1}^{i} \lambda_j^i \right) + \lambda_{i,2} = 1.$$

Using the relations (6), we obtain

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

due to $\alpha_m = 0$ which completes the induction.

Now, we define

$$\lambda_j^k = 0, \qquad \text{for } 1 \leq j < m,$$

and

$$\tilde{\sigma}_k = \sum_{j=1}^{k} \lambda_j^k |y_j - x_k|.$$

Since $x_j = x_k$, for $j = m, \ldots, k$, we have

$$\tilde{\sigma}_k = \sum_{j=m}^{k} \lambda_j^k |y_j - x_j|,$$

and thus,

$$\gamma \tilde{\sigma}_k^\omega = \gamma \left(\sum_{j=m}^{k} \lambda_j^k |y_j - x_j| \right)^\omega$$
$$\leq \sum_{j=m}^{k} \lambda_j^k \gamma |y_j - x_j|^\omega$$
$$\leq \sum_{j=m}^{k} \lambda_j^k \alpha_j$$
$$= \tilde{\alpha}_k,$$

from (19), Lemma 3.1, and the convexity of the function $\xi \to \gamma \xi^\omega$ on $\mathscr{R}_+$ for $\gamma > 0$ and $\omega \geq 1$. □

Lemma 3.3. Let $\bar{x} \in R^N$ be given and suppose that there exist vectors $\bar{q}$, $\bar{g}_i$, $\bar{y}_i$ and numbers $\bar{\lambda}_i \geq 0$, for $i = 1, \ldots, l$, $l \geq 1$, satisfying

$$(\bar{q}, 0) = \sum_{i=1}^{l} \bar{\lambda}_i (\bar{g}_i, |\bar{y}_i - \bar{x}|), \tag{20a}$$

$$\bar{g}_i \in \partial f(\bar{y}_i), \qquad i = 1, \ldots, l \tag{20b}$$

$$\sum_{i=1}^{l} \bar{\lambda}_i = 1. \tag{20c}$$

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

Proof. Let

$$I = \{i: 1 \le i \le l, \bar{\lambda}_i > 0\}.$$

By (20),

$$\bar{y}_i = \bar{x} \quad \text{and} \quad \bar{g}_i \in \partial f(\bar{x}),$$

for all $i \in I$. Thus, we have

$$\bar{q} = \sum_{i \in I} \bar{\lambda}_i \bar{g}_i, \ \bar{\lambda}_i > 0, \qquad \text{for } i \in I, \qquad \sum_{i \in I} \bar{\lambda}_i = 1,$$

so $\bar{q} \in \partial f(\bar{x})$ by the convexity of $\partial f(\bar{x})$; see Ref. 3. □

Theorem 3.1. If Algorithm 2.1 terminates at the kth iteration with $w_k = 0$, then the point x_k is stationary for f.

Proof. If the algorithm terminates at Step 3, then $w_k = 0$ implies $\tilde{g}_k = 0$ and $\tilde{\alpha}_k = \tilde{\sigma}_k = 0$ by Lemma 3.1 and Lemma 3.2. By Lemma 3.2 and using Lemma 3.3 with

$$\bar{x} = x_k, \qquad l = k, \qquad \bar{q} = \tilde{g}_k,$$

$$\bar{g}_i = g_i, \qquad \bar{y}_i = y_i, \qquad \bar{\lambda}_i = \lambda_i^k, \qquad \text{for } i \le k,$$

we have $0 = \bar{q} \in \partial f(\bar{x})$. □

From now on, we will assume that Algorithm 2.1 does not terminate with $w_k = 0$ and that the sequence $\{x_k\}$ is infinite (e.g., when $\epsilon = 0$), i.e., that $w_k > 0$ for all $k \ge 1$.

Lemma 3.4. Suppose that the level set $\{x \in \mathcal{R}^N: f(x) \le f(x_1)\}$ is bounded. Then, the sequences $\{y_k\}$ and $\{g_k\}$ are also bounded. In addition, if 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. The sequence $\{x_k\}$ is bounded by assumption and the monotonicity of $\{f_k\}$. Lemma 2.1 gives

$$|y_{k+1} - z_{k+1}| \le D,$$

with

$$z_{k+1} \in \{x \in \mathcal{R}^N: f(x) \le f(x_1)\},$$

for null steps; thus, also $\{y_k\}$ is bounded. By the local boundedness and the upper semicontinuity of ∂f (see Ref. 3), we obtain the boundedness of $\{g_k\}$.

Let

$$I = \{1, \ldots, N+2\}.$$

From $g_k \in \partial f(y_k)$, $k \geq 1$, Lemma 3.2, and the Caratheodory theorem (see Ref. 10), we deduce the existence of vectors $y^{k,i}$, $g^{k,i}$, and numbers $\lambda^{k,i} \geq 0$ and $\tilde{\sigma}_k$, for $i \in I$ and $k \geq 1$, satisfying

$$(\tilde{g}_k, \tilde{\sigma}_k) = \sum_{i \in I} \lambda^{k,i}(g^{k,i}, |y^{k,i} - x_k|), \tag{21a}$$

$$\sum_{i \in I} \lambda^{k,i} = 1, \qquad g^{k,i} \in \partial f(y^{k,i}), \tag{21b}$$

with

$$(y^{k,i}, g^{k,i}) \in \{(y_j, g_j): j = 1, \ldots, k\}.$$

From the boundedness of $\{y_k\}$, we get the existence of points y_i^*, $i \in I$, and an infinite set $K_0 \subset K$ satisfying $y^{k,i} \xrightarrow{K_0} y_i^*$, for $i \in I$. The boundedness $\{g_k\}$ and $\{\lambda^{k,i}\}$ gives the existence of vectors $g_i^* \in \partial f(y_i^*)$, numbers λ_i^*, for $i \in I$, and an infinite set $\bar{K} \subset K_0$ satisfying $g^{k,i} \xrightarrow{\bar{K}} g_i^*$ and $\lambda^{k,i} \xrightarrow{\bar{K}} \lambda_i^*$ for $i \in I$. Obviously

$$\lambda_i^* \geq 0, \qquad i \in I, \qquad \sum_{i \in I} \lambda_i^* = 1,$$

by (21).

From $w_k \xrightarrow{K} 0$, Lemma 3.1, and Lemma 3.2, we obtain

$$\tilde{g}_k \xrightarrow{K} 0, \qquad \tilde{\alpha}_k \xrightarrow{K} 0, \qquad \tilde{\sigma}_k \xrightarrow{K} 0.$$

Letting $k \in \bar{K}$ approach infinity in (21), and using Lemma 3.3 with

$$l = N+2, \qquad \bar{q} = 0, \qquad \bar{g}_i = g_i^*, \qquad \bar{y}_i = y_i^*, \qquad \bar{\lambda}_i = \lambda_i^*,$$

we get

$$0 \in \partial f(\bar{x}). \qquad \square$$

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

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

Let

$$Q(\lambda) = |\lambda q + (1-\lambda)p|^2 + 2[\lambda\beta + (1-\lambda)\alpha],$$

$$b = (1-2c)/4M.$$

Then,

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

Proof. See the proof of Lemma 3.5 in Ref. 8. □

Lemma 3.6. Suppose that the level set $\{x \in \mathcal{R}^N: f(x) \leq f(x_1)\}$ is bounded, the number of descent steps is finite, and the last descent step occurs at the $\hat{k}$th iteration. Then, the point $x_{\hat{k}+1}$ is stationary for f.

Proof.

(i) First we establish the existence of a number $k^*, k^* > \hat{k}$ (to have solely null steps), such that

$$w_{k+1} \leq \tilde{g}_{k+1}^T H_k \tilde{g}_{k+1} + 2\tilde{\alpha}_{k+1}, \tag{22a}$$

$$\mathrm{Tr}\,(H_{k+1}) \leq \mathrm{Tr}\,(H_k), \qquad k \geq k^*. \tag{22b}$$

If $n_C < L$ for all $k \geq 1$, we can set

$$k^* = \max[\bar{k}, \hat{k} + 1],$$

where $\bar{k}$ is the index k when n_C changed last (or $\bar{k} = 1$) if $n_C = 0$ for all $k \geq 1$). To see this, let $k \geq k^*$. Then, n_C does not change; thus, $w_{k+1} = \check{w}_{k+1}$ and $H_{k+1} = \check{H}_{k+1}$ (see Step 2). If the SR1 update is not used, then (22) holds with equalities; otherwise, Lemma 3.1 implies $u_k^T v_k > 0$, which together with (9) gives (22).

If $n_C < L$ does not hold for all $k \geq 1$, then we set $\bar{k}$ equal to the index k when $i_C = 1$ occurred first and again set

$$k^* = \max[\bar{k}, \hat{k} + 1].$$

Then, the 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 prove easily by induction that all the 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

$$H_{k+1} = \check{H}_{k+1} + \rho I,$$

otherwise the 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 $k \geq k^*$. 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 $\check{H}_{k+1} - \varrho I = H_k - \varrho I$ is positive definite. Therefore,

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

and (22) holds with equalities. If the SR1 update is used, all conditions (7)–(8) are satisfied and $i_C = i_U = 1$; therefore, the correction (4), with k replaced by $k+1$, is realized. Using (9), 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 (22) follows from the first part of (8). Moreover, (9) implies

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

and the second part of (22) follows from the second part of (8).

(ii) Combining (22) with (5), (6), and Lemma 3.1, we obtain

$$\begin{aligned} w_{k+1} &\le \tilde{g}_{k+1}^T H_k \tilde{g}_{k+1} + 2\tilde{\alpha}_{k+1} \\ &= \varphi(\lambda_{k,1}, \lambda_{k,2}, \lambda_{k,3}) \\ &\le \varphi(0, 0, 1) \\ &= w_k, \end{aligned} \tag{23}$$

for $k \ge k^*$; therefore, the sequences $\{w_k\}$, $\{W_k \tilde{g}_k\}$, $\{\tilde{\alpha}_k\}$ are bounded. Moreover, (22) assures the boundedness of the sequences $\{H_k\}$ and $\{W_k\}$. By Lemma 3.4, we obtain the boundedness of $\{y_k\}$, $\{g_k\}$, $\{W_k g_{k+1}\}$. Denote

$$M = \sup\{|W_k g_{k+1}|, |W_k \tilde{g}_k|, \sqrt{\tilde{\alpha}_k} : k \ge k^*\}, \qquad b = (1 - 2c_R)/4M, \tag{24}$$

and assume first that $w_k > \delta > 0$ for all $k \ge k^*$. It follows from

$$\begin{aligned} &\min\left\{\varphi(\lambda_1, \lambda_2, \lambda_3) : \lambda_i \ge 0,\ i = 1, 2, 3,\ \sum_{i=1}^{3} \lambda_i = 1\right\} \\ &\le \min_{\lambda \in [0,1]} \varphi(0, \lambda, 1 - \lambda) \end{aligned}$$

and (23) that

$$w_{k+1} \le \min\{|\lambda W_k g_{k+1} + (1-\lambda) W_k \tilde{g}_k|^2 + 2[\lambda \alpha_{k+1} + (1-\lambda)\tilde{\alpha}_k], \lambda \in [0, 1]\}.$$

Since

$$w_k = \tilde{g}_k^T H_k \tilde{g}_k + 2\tilde{\alpha}_k$$

by Lemma 3.1, $d_k = -H_k \tilde{g}_k$ (see Step 4 of Algorithm 2.1) and

$$-\alpha_{k+1} + d_k^T g_{k+1} \ge -c_R w_k$$

by (14), we can use Lemma 3.5, with

$$p = W_k \tilde{g}_k, \qquad q = W_k g_{k+1}, \qquad w = w_k,$$
$$\alpha = \tilde{\alpha}_k, \qquad \beta = \alpha_{k+1}, \qquad c = c_R,$$

to obtain

$$w_{k+1} \leq w_k - (w_k b)^2 < w_k - (\delta b)^2,$$

for $k \geq k^*$; 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 $k \geq k^*$, $x_k \to x_{\hat{k}+1}$, and Lemma 3.4 gives $0 \in \partial f(x_{\hat{k}+1})$. □

Theorem 3.2. Suppose that the level set $\{x \in \mathscr{R}^N : f(x) \leq f(x_1)\}$ 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 let $K \subset \{1, 2, \ldots\}$ be an infinite set such that $x_k \xrightarrow{K} \bar{x}$. In view of Lemma 3.6, we can restrict the analysis 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 $x_k \xrightarrow{K'} \bar{x}$. The continuity of f implies that $f_k \xrightarrow{K'} f(\bar{x})$ and therefore $f_k \downarrow f(\bar{x})$ by the monotonicity of $\{f_k\}$, which follows from the descent condition (12). Using the nonnegativity of t_L^K for $k \geq 1$ and condition (12), we obtain

$$0 \leq c_L t_L^k w_k \leq f_k - f_{k+1} \to 0, \qquad k \geq 1. \tag{25}$$

If the set

$$K_1 = \{k \in K' : t_L^k \geq t_{\text{aux}}\}$$

is infinite, then $w_k \xrightarrow{K_1} 0$, $x_k \xrightarrow{K_1} \bar{x}$ by (25) and $0 \in \partial f(\bar{x})$ by Lemma 3.4.

If K_1 is finite, the set

$$K_2 = \{k \in K' : \beta_{k+1} > c_A w_k\}$$

must be infinite by (13). For contradiction purposes, we assume that

$$w_k \geq \delta > 0, \qquad \text{for all } k \in K_2.$$

The sequence $\{g_k\}$ is bounded by Lemma 3.4 and thus also $\{\tilde{g}_k\}$ is bounded by (6). From (25), we have $t_L^k \xrightarrow{K_2} 0$ and Step 2 with $d_k = -H_k \tilde{g}_k$ implies

$$|x_{k+1} - x_k| = t_L^k |d_k| \leq t_L^k (|\check{H}_k \tilde{g}_k| + \rho |\tilde{g}_k|) \leq t_L^k (D + \varrho \sup\{\tilde{g}_k : k \geq 1\}),$$

for $k \geq 1$; thus, $x_{k+1} - x_k \xrightarrow{K_2} 0$. By (15), (25), and the boundedness of $\{g_k\}$, and since $y_{k+1} = x_{k+1}$ for descent steps, we obtain $\beta_{k+1} \xrightarrow{K_2} 0$, which is in contradiction with

$$c_A \delta \leq c_A w_k < \beta_{k+1}, \qquad k \in K_2.$$

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 3.4. □

Remark 3.1. If we choose $\epsilon > 0$, Algorithm 2.1 terminates always in a finite number of steps, since $w_k \rightarrow 0$ in case the number of descent steps is finite (see the proof of Lemma 3.6) 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 3.2).

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, suppose that we have the current aggregate subgradient $\tilde{g}_k$, the positive definite VM approximation H_k of the inverse Hessian matrix, and the search direction $d_k = -H_k \tilde{g}_k$, and define the subgradient locality measures (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 $f(x_k + td_k)$:

$$\begin{aligned}\psi_Q^k(t) &= f_k + t d_k^T g_k + (1/2) t^2 d_k^T H_k^{-1} d_k \\ &= f_k + (t - (1/2) t^2) d_k^T g_k.\end{aligned}$$

The bundle represents the polyhedral function (1). For $x = x_k + td_k$, we have the following piecewise linear approximation of $f(x_k + td_k)$:

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

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

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

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 $f(x_k + td_k)$ and set

$$\hat{\psi}_k(t) = \max[\psi_L^k(t), \psi_P^k(t)] + (1/2)t^2 d_k^T H_k^{-1} d_k,$$

$$t_I^k = \arg\min\{\hat{\psi}_k(t): t \in [t_{\text{aux}}, \min[t_{\text{aux}}, \max[1, B/|d_k|]]]\}.$$

The function $\psi_P^k(t)$ has sometimes no influence on the stepsize determination. It can mean that the initial stepsize is too small. Thus, we have introduced the bundle parameter for the matrix scaling s_k; in view of (10), (2), and since function (5) is not minimized for descent steps, we could define s_k by

$$\arg\min_{s \in \mathscr{S}}\{\max[\psi_L^k(s), \psi_P^k(s)] + (1/2)\nu_k s \tilde{g}_k^T H_k \tilde{g}_k\}, \tag{26}$$

where $\nu_k = 1$ for null steps and $\nu_k = 0$ for descent steps. For simplification, we omit in (26) the lines of ψ_P^k with $d_k^T g_j \leq (1/2)\nu_k d_k^T \tilde{g}_k$ and set

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

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

From now on, we use the same notation as in Algorithm 2.1. The minimization of the quadratic function (5) at 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 the analysis to the lines $\lambda_1 = 0, \lambda_2 = 0, \lambda_1 + \lambda_2 = 1$. 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 $\tilde{g}_k = g_k = g_m$ and $\tilde{\alpha}_k = 0$. If $g_{k+1} \neq \tilde{g}_k$, then set

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

otherwise, set

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

or

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

Further, we mention the stopping criterion. We define the descent tolerance $\epsilon_f > 0$ and the maximum number $m_f \geq 1$ of consecutive too small function value variations and add to Step 0 the initialization of the 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 $w_k \leq \epsilon$ and either $\Delta_k / \max[1, f_k] < 100\epsilon_f$ after a descent step, or $w_{k-1} \leq \epsilon$ after two consecutive null steps, then stop.

To cut off useless iterations and update Δ_k, we finally modify Step 5; the full wording (including the matrix scaling parameter μ updating) is as follows:

Step 5′. Set

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

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 $\Delta / \max[1, f(y_{k+1})] \leq \epsilon_f$ 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 the matrix scaling $s_k \geq 0$ and set $\Delta_{k+1} = \Delta_k$. If $s_k < 10^{30}$, set

$$\mu = (2\mu + \min[C^2, \max[0.1, s_k]])/3.$$

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 test problems from the literature (Problem 1 is smooth, all the others are nonsmooth) with those obtained by our convex VM method (VMC, Ref. 8) and by our proximal bundle method

(PBL, Ref. 6). A comparison with the BT algorithm of Ref. 5 and the ellipsoid bundle method of Ref. 11 for some problems can be found in Ref. 6; a comparison with a smooth VM method from Ref. 12 can be found in Ref. 8.

All problems are described in Ref. 13; this report and corresponding FORTRAN codes TEST06 (25 test problems for minimax), regarding Problems 25–30, and TEST19 (25 test problems for nonsmooth unconstrained optimization), regarding Problems 1–24, can be downloaded from the web page http://www.cs.cas.cz/˜luksan/test.html. Problems 1–16 are also described in Ref. 1 and Problems 19–22 in Ref. 11.

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

$$t_{\text{aux}} = 10^{-10}, \qquad t_{\max} = 10^{3}, \qquad c_A = c_L = 10^{-4}, \qquad c_R = 0.25,$$

$$c_T = 2 \times 10^{-4}, \qquad \epsilon = 10^{-6}, \qquad \epsilon_f = 5 \times 10^{-7}, \qquad \varrho = 10^{-12},$$

$$L = 1, \qquad \omega = 2, \qquad C = 10, \qquad D = 10^{50},$$

$$\mathscr{J}_k = \{\max[1, k - N - 2], \ldots, k\}, \qquad k \geq 1,$$

and

$$m_f = 2, \qquad \text{for Problems 1–14, 17–21, 23–24, 26–29,}$$

$$m_f = 3, \qquad \text{for Problem 15,}$$

$$m_f = 4, \qquad \text{for Problem 16,}$$

$$m_f = 5, \qquad \text{for Problems 22, 25, 30.}$$

Table 1. Test problems.

#	N	Problem	min f	#	N	Problem	min f
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–Suzuki	−44.0	25	5	EXP	0.0001224
11	5	Shor	22.600162	26	6	Transformer	0.1972906
12	10	Maxquad	−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 2. Test results.

	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
	2441	2474				2635		2727	
Σ	Time = 9.34 sec					Time = 8.29 sec		Time = 23.17 sec	

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; the values of B and γ, which depends on the function f, were chosen experimentally: a too big value of B can cause the information about f, which is accumulated in H_k, to become irrelevant; a too small value of B can slow down convergence; for the values of γ, see the comments to Algorithm 2.1. Note that a similar choice of parameters (to optimize N_f) was also performed for VMC and PBL; we refer to Ref. 8 for the 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. The achieved improving, as compared with the convex VMC method, can be seen in the results for substantially nonconvex problems, namely, Problems, 2, 17–18, 22–23, 25–26, and 30.

References

1. Mäkelä, M. M., and Neittaanmäki, P., *Nonsmooth Optimization*, World Scientific Publishing Company, London, England, 1992.
2. Clarke, F. H., *Optimization and Nonsmooth Analysis*, Wiley-Interscience, New York, NY, 1983.
3. Kiwiel, K. C., *Methods of Descent for Nondifferentiable Optimization*, Lecture Notes in Mathematics, Springer Verlag, Berlin, Germany, Vol. 1133, 1985.
4. Kiwiel, K. C., *A Method of Linearizations for Linearly Constrained Nonconvex Nonsmooth Minimization*, Mathematical Programming, Vol. 34, pp. 175–187, 1986.
5. 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.
6. Lukšan, L., and Vlček, J., *A Bundle-Newton Method for Nonsmooth Unconstrained Minimization*, Mathematical Programming, Vol. 83, pp. 373–391, 1998.
7. Lemaréchal, C., *Numerical Experiments in Nonsmooth Optimization*, Progress in Nonsmooth Optimization, Edited by E. A. Nurminski, IIASA, Laxenburg, Austria, pp. 61–84, 1982.
8. Lukšan, L., and Vlček, J., *Globally Convergent Variable Metric Method for Convex Nonsmooth Unconstrained Minimization*, Journal of Optimization Theory and Applications, Vol. 102, pp. 593–613, 1999.
9. Fletcher, R., *Practical Methods of Optimization*, John Wiley and Sons, Chichester, England, 1987.
10. Hiriart-Urruty, J. B., and Lemarechal, C., *Convex Analysis and Minimization Algorithms, I, II*, Springer Verlag, Berlin, Germany, 1993.
11. 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.
12. Lukšan, L., *Computational Experience with Known Variable Metric Updates*, Journal of Optimization Theory and Applications, Vol. 83, pp. 27–47, 1994.
13. Lukšan, L., and Vlček, J., *Test Problems for Nonsmooth Unconstrained and Linearly Constrained Optimization*, Report 798, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, Czech Republic, 2000.