Globally Convergent Variable Metric Method for Convex Nonsmooth Unconstrained Minimization

Lukšan, Ladislav
1998

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Globally convergent variable metric method for convex nonsmooth unconstrained minimization

L. Lukšan, J. Vlček

Technical report No. 741

May 1998

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

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Globally convergent variable metric method for convex nonsmooth unconstrained minimization

L. Lukšan, J. Vlček[1]

Technical report No. 741
May 1998

Abstract

A special variable metric method is given for finding minima of convex functions that are not necessarily differentiable. Time consuming quadratic programming subproblems need not be solved. Global convergence of the method is established. Some encouraging numerical experience is reported.

Keywords

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

[1] This work was supported by the grant No. 201/96/0918 given by the Czech Republic Grant Agency

1 Introduction

This paper is devoted to seeking a minimum of a convex continuous function $f : \mathcal{R}^N \to R$. 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 1983). 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 bundle methods (see e.g. Kiwiel 1985, Mäkelä and Neittaanmäki 1992, Schramm and Zowe 1992, Lemaréchal and Sagastizábal 1997, Lukšan and Vlček 1998). Instead of the singleton $f_k = f(x_k)$, $g_k \in \partial f(x_k)$, the bundle $\{(f_j^k, g^j) | j \in \mathcal{J}_k\}$ is used in the k-th iteration, $k \geq 1$, where $f_j^k = f(y^j) + (x_k - y^j)^T g^j$, $g^j \in \partial f(y^j)$, $\mathcal{J}_k \subset \{1, \dots k\}$, $x_1, \dots, x_k$ are iterates and $y^1, \dots, y^k$ are trial points. The piecewise linear function

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

where

$$\beta_j^k = f(x_k) - f_j^k = f(x_k) - f(y^j) + (y^j - x_k)^T g_j^k, \quad j \in \mathcal{J}_k, \tag{1.2}$$

are non-negative (since f is convex) linearization errors, is constructed and the direction vector

$$d^k = \arg\min_{d \in \mathcal{R}^N} \left\{ \check{f}_k(x_k + d) + \tfrac{1}{2} d^T B_k d \right\} \tag{1.3}$$

is determined (the additional quadratic term in (1.3) has a similar significance as in the trust region approach). Minimization subproblem (1.3) can be replaced by the quadratic programming subproblem

$$(d^k, \xi^k) = \arg\min_{(d,\xi) \in \mathcal{R}^{N+1}} \left\{ \tfrac{1}{2} d^T B_k d + \xi \right\} \quad \text{subject to} \quad -\beta_j^k + d^T g^j \leq \xi, \; j \in \mathcal{J}_k. \tag{1.4}$$

The most popular proximal bundle methods are based on the choice $B_k = \mu_k I$ where μ_k, $k \geq 1$ are weighting coefficients. These methods require only $O(N)$ operations for solving a system with the matrix B_k so that they are very efficient measured by the computational time. Another possibility is to use aggregate Hessian matrices. The resulting bundle-Newton method, see Lukšan and Vlček 1998, reduces significantly the number of iterations and function evaluations, but it requires $O(N^3)$ operations for solving a system with the matrix B_k. A natural idea is to generate matrices B_k, $k \geq 1$, by using variable metric (VM) updates, but it leads to methods, which does not overcome efficiency of proximal bundle methods (see e.g. Lemaréchal 1982). The most promising results are presented in Lemaréchal and Sagastizábal 1997, where reversal quasi-Newton updates together with a special curvilinear search procedure are used. The resulting algorithm requires $O(N^2)$ operations for solving a system with the matrix B_k. Nevertheless, time consuming quadratic programming subproblems have to be solved.

The development of our VM method was motivated by an observation that standard VM methods are relatively robust and efficient even in the nonsmooth case (see e.g.

Lemaréchal 1982 and also our experiments in Table 2). Their advantage consists in a fact that the time consuming quadratic programming subproblem (1.4) need not be solved. Although standard VM methods require more function evaluations than bundle methods, the total computational time is frequently 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 was to obtain a VM method that does not require solution to the quadratic programming subproblem (1.4), but is globally convergent applied to a convex nonsmooth function. For this purpose, ideas which are essential for bundle methods were used. The basic difference compared with standard VM methods consists in the utilization of null steps that serve for obtaining a 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 (null step). The VM update is carried out in both cases, whenever conditions for positive definiteness 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 aggregate subgradients to zero and allow us to use a suitable termination criterion. To improve robustness and efficiency of the method, the stepsize selection based on the polyhedral approximation of the objective function and a suitable matrix scaling are finally added.

The paper is organized as follows. Section 2 is devoted to a description of a new VM method for convex nonsmooth minimization. Section 3 contains the global convergence theory. In Section 4, we give more details concerning implementation of the method and 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 $y_1^k = x_k$, $y_{j+1}^k = x_k + t_j^k d_j^k$, $j \geq 1$ in the k-th 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 $f(y_{j+1}^k) \leq f(x_k) - c_1 t_j^k w_j^k$ is satisfied, 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 an information about the minimized function. 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

$$\tilde{g}_{j+1}^k = \lambda_{j,1}^k g_k + \lambda_{j,2}^k g_{j+1}^k + \lambda_{j,3}^k \tilde{g}_j^k, \tag{2.1}$$

where $\lambda_{j,i}^k \geq 0$, $i \in \{1, 2, 3\}$ are appropriately chosen scalars. These scalars can be easily determined by minimization of a simple quadratic function, which depends on a convex combination of this three subgradients and two modified linearization errors (see below and Step 6 of Algorithm 1). This approach retains global convergence, but eliminates solution of the rather complicated quadratic programming subproblem (1.4), which appears in standard bundle methods. Note that the 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 slightly deteriorates robustness of the method, e.g. increases sensitivity to the stepsize determination after the null steps (see Section 4). Moreover, the situation when $g_k^T d_{j+1}^k \geq 0$ occurred in numerical experiments, was much more frequent in the simplified case.

Note furthermore that the problem to minimize the function (2.6) in the Step 6 of Algorithm 1 is the dual to the following primal problem

$$\underset{d \in \mathcal{R}^N}{\text{minimize}} \left\{ \tfrac{1}{2} d^T (H_j^k)^{-1} d + \max[d^T g_k, -\alpha_{j+1}^k + d^T g_{j+1}^k, -\tilde{\alpha}_j^k + d^T \tilde{g}_j^k] \right\}. \qquad (2.2)$$

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 (1.2)). Unfortunately, it 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 (2.5)).

The matrices H_j^k are generated using usual VM updates. After null steps, the symmetric rank one (SR1) update (see Fletcher 1987) is used, since it preserves boundedness of generated matrices as required in the global convergence theory. Because this boundedness is not necessary after descent steps, the standard BFGS update (see Fletcher 1987) appears to be more suitable.

Even if the stepsize selection is not relevant for proving global convergence, efficiency of the algorithm is very sensitive on 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 is determined and if γ is too large after descent steps, the inverse Hessian matrix is scaled and the BFGS update is not performed, which does not have an influence on the global convergence but improves efficiency of the method.

Because the proof of global convergence requires 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 subgradients are identical in consecutive iterations, we extrapolate doubling the stepsize if possible to quicker exit such region.

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

Algorithm 1.

Data: A 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 $\varepsilon \geq 0$, correction parameters $\varrho \in (0,1)$ and $L \geq 1$ 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 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 evaluations counter for matrix scaling $n_S = 0$ and the iteration counter $k = 1$.

Step 1: Start of iteration loop. Set $\tilde{g}_1^k = g_1^k = g_k$, $\tilde{\alpha}_1^k = \alpha_1^k = 0$, $y_1^k = x_k$ and $\check{H}_1^k = H_k$. 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 $\check{w}_j^k < \varrho_k |\tilde{g}_j^k|^2$ or $i_C = i_U = 1$, then set

$$w_j^k = \check{w}_j^k + \varrho_k |\tilde{g}_j^k|^2, \quad H_j^k = \check{H}_j^k + \varrho_k I \tag{2.3}$$

and $n_C = n_C + 1$, otherwise set $w_j^k = \check{w}_j^k$ and $H_j^k = \check{H}_j^k$. If $n_C \geq L$ then set $i_C = 1$.

Step 3: Stopping criterion. If $w_j^k \leq \varepsilon$, 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 $t_j^k = 2t_j^k$ and $i_E = 0$. Set $y_{j+1}^k = x_k + t_j^k d_j^k$, $n_S = n_S + 1$ and compute $f(y_{j+1}^k)$, $g_{j+1}^k \in \partial f(y_{j+1}^k)$. If $s_j^k < \sigma$ then set $\gamma = (2\gamma + \min[\sigma, \max[1, s_j^k]])/3$.

Step 5: Descent step. If

$$f(y_{j+1}^k) - f(x_k) \leq -c_1 t_j^k w_j^k, \tag{2.4}$$

then set $x_{k+1} = y_{j+1}^k$, $g_{k+1} = g_{j+1}^k$, $t_k = t_j^k$, $d_k = d_j^k$, $\tilde{\alpha}_k = \tilde{\alpha}_j^k$, $\tilde{g}_k = \tilde{g}_j^k$, $w_k = w_j^k$, $H_k = H_j^k$, $\varrho_{k+1} = \varrho \min[1/|\tilde{g}_k|, |\tilde{g}_k|]/(k+1)$, otherwise go to Step 6. 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 go to Step 8.

Step 6: Null step. Set

$$\alpha_{j+1}^k = (f(x_k) - f(y_{j+1}^k))/t_j^k + (d_j^k)^T g_{j+1}^k \tag{2.5}$$

and determine multipliers $\lambda_{j,i}^k \geq 0$, $i \in \{1, 2, 3\}$, $\lambda_{j,1}^k + \lambda_{j,2}^k + \lambda_{j,3}^k = 1$, which minimize the function

$$\varphi(\lambda_1, \lambda_2, \lambda_3) = |\lambda_1 W_j^k g_k + \lambda_2 W_j^k g_{j+1}^k + \lambda_3 W_j^k \tilde{g}_j^k|^2 + 2[\lambda_2 \alpha_{j+1}^k + \lambda_3 \tilde{\alpha}_j^k], \tag{2.6}$$

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

$$\tilde{g}_{j+1}^k = \lambda_{j,1}^k g_k + \lambda_{j,2}^k g_{j+1}^k + \lambda_{j,3}^k \tilde{g}_j^k, \qquad \tilde{\alpha}_{j+1}^k = \lambda_{j,2}^k \alpha_{j+1}^k + \lambda_{j,3}^k \tilde{\alpha}_j^k. \tag{2.7}$$

Step 7: SR1 update. Let $u_j^k = g_{j+1}^k - g_k$ and $v_j^k = H_j^k u_j^k - t_j^k d_j^k$. If

$$(\tilde{g}_j^k)^T v_j^k < 0 \tag{2.8}$$

and, in case of $i_C = 1$, furthermore

$$\varrho_k |\tilde{g}_{j+1}^k|^2 \leq [(\tilde{g}_{j+1}^k)^T v_j^k]^2 / (u_j^k)^T v_j^k \quad \text{and} \quad N \varrho_k \leq |v_j^k|^2 / (u_j^k)^T v_j^k, \tag{2.9}$$

then set $i_U = 1$ and

$$\check{H}^k_{j+1} = H^k_j - v^k_j (v^k_j)^T / (u^k_j)^T v^k_j, \tag{2.10}$$

otherwise set $i_U = 0$ and $\check{H}^k_{j+1} = H^k_j$. Set $j = j + 1$ and go to Step 2.

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

Step 9: BFGS update. If $g_{k+1} = g_k$ and $t^k_j < t_{max}/2$, then set $i_E = 1$, $k = k + 1$ and go to Step 1, otherwise set $u_k = g_{k+1} - g_k$. If $d_k^T u_k > |d_k| 10^{-5}$, then set

$$H_{k+1} = H_k + \left(t_k + \frac{u_k^T H_k u_k}{u_k^T d_k} \right) \frac{d_k d_k^T}{u_k^T d_k} - \frac{H_k u_k d_k^T + d_k (H_k u_k)^T}{u_k^T d_k}, \tag{2.11}$$

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 (2.8) (or $(d^k_j)^T u^k_j > t^k_j (d^k_j)^T (H^k_j)^{-1} d^k_j$), which implies that $(u^k_j)^T v^k_j > 0$ by Lemma 1, assures positive definiteness of the matrix obtained by the SR1 update (see e.g. Fletcher 1987). Similarly, the condition $d_k^T u_k > 0$ assures positive definiteness of the matrix obtained by the BFGS update ($d_k^T u_k \geq 0$ holds whenever f is convex). Therefore, all matrices H_k, $\check{H}^k_j$, H^k_j generated by Algorithm 1 are positive definite.

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 (2.3) are used automatically, after every SR1 update, only if the condition $\check{w}^k_j < \varrho_k |\tilde{g}^k_j|^2$ has been satisfied at least L times. In this way we have a possibility to eliminate the use of conditions (2.9) (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 (2.6) and the determination of the stepsize t^k_j and the bundle parameter for matrix scaling s^k_j 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 global convergence of Algorithm 1 under the assumption that the function $f : \mathcal{R}^N \to \mathcal{R}$ is convex and the level set $\{x \in \mathcal{R}^N | f(x) \leq f(x_1)\}$ is bounded. For this purpose, we will assume that the final accuracy tolerance ε is set to zero.

Lemma 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 k-th iteration of Algorithm 1. Then*

$$-\alpha^k_{j+1} + (d^k_j)^T g^k_{j+1} > -c_1 w^k_j, \quad \alpha^k_{j+1} \geq 0, \quad \tilde{\alpha}^k_j \geq 0. \tag{3.1}$$

If in addition the condition (2.8) holds, then $(u^k_j)^T v^k_j > 0$.

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 (2.5). The inequality $\tilde{\alpha}_j^k \geq 0$ follows from (2.7) by induction. Using (2.5) and the fact that inequality (2.4) 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 $(u_j^k)^T v_j^k = (u_j^k) H_j^k u_j^k - t_j^k (d_j^k)^T u_j^k > 0$. □

Lemma 2. *Let at least $j - 1 \geq 0$ null steps be generated in the k-th iteration of Algorithm 1. Then the numbers $\lambda_i^{k,j} \geq 0$, $i = 1, \ldots, j$, exist satisfying*

$$(\tilde{g}_j^k, \tilde{\alpha}_j^k) = \sum_{i=1}^{j} \lambda_i^{k,j} (g_i^k, \alpha_i^k), \quad \sum_{i=1}^{j} \lambda_i^{k,j} = 1. \tag{3.2}$$

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 (3.2) hold for j replaced by n. Define $\lambda_1^{k,n+1} = \lambda_{n,1}^k + \lambda_{n,3}^k \lambda_1^{k,n}$, $\lambda_i^{k,n+1} = \lambda_{n,3}^k \lambda_i^{k,n}$ for $2 \leq i \leq n$, $\lambda_{n+1}^{k,n+1} = \lambda_{n,2}^k$. It is clear that $\lambda_i^{k,n+1} \geq 0$ 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 (2.7), we obtain in view of $\alpha_1^k = 0$, $g_1^k = g_k$

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

□

Lemma 3. *Let the function f be convex and the quantities $\tilde{g}_j^k$, $\tilde{\alpha}_j^k$, $j \geq 1$ be generated in the k-th iteration of Algorithm 1. Then*

$$f(z) \geq f(x_k) + (z - x_k)^T \tilde{g}_j^k - t_{max} \tilde{\alpha}_j^k \tag{3.3}$$

for all $z \in \mathcal{R}^N$.

Proof. Lemma 2 implies that the numbers $\lambda_i^{k,j} \geq 0$, $i = 1, \ldots, j$, exist such that (3.2) 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 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 4. *Let the function f be convex. If Algorithm 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 implies that $f(z) \geq f(x_k)$ for all $z \in \mathcal{R}^N$. □

From now on we assume that Algorithm 1 does not terminate, i.e. that $w_j^k > 0$ for all generated indices k and j.

Lemma 5. *Let 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 $w = |p|^2 + 2\alpha$, $\beta + p^T q \leq cw$ and $\max[|p|, |q|, \sqrt{\alpha}] \leq M$. Let $Q(\lambda) = |\lambda q + (1 - \lambda)p|^2 + 2[\lambda\beta + (1 - \lambda)\alpha]$, $b = (1 - 2c)/(4M)$. Then*

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

Proof. We obtain after straightforward manipulations

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

The last expression reaches its minimum for $\bar{\lambda} = w(1 - 2c)/(8M^2) \leq (|p|^2 + 2\alpha)/(8M^2) \leq (M^2 + 2M^2)/(8M^2) = 3/8 < 1$, thus we have

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

Lemma 6. *Let the number of null steps be infinite in the k-th iteration of Algorithm 1. Then an index $j_0 \geq 1$ exists such that*

$$w_{j+1}^k \leq (\tilde{g}_{j+1}^k)^T H_j^k \tilde{g}_{j+1}^k + 2\tilde{\alpha}_{j+1}^k, \quad Tr(H_{j+1}^k) \leq Tr(H_j^k) \tag{3.4}$$

for all $j \geq j_0$.

Proof. If $n_C < L$ for all $j \geq 1$, then 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 $w_{j+1}^k = \check{w}_{j+1}^k$ and $H_{j+1}^k = \check{H}_{j+1}^k$. If the SR1 update is not used then (3.4) holds with equalities, otherwise Lemma 1 implies that $(u_j^k)^T v_j^k > 0$, which together with (2.10) gives (3.4).

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 easily prove 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 $w_{j+1}^k = \check{w}_{j+1}^k$, $H_{j+1}^k = \check{H}_{j+1}^k = H_j^k$ and (3.4) holds with equalities. If the SR1 update is used, then all conditions (2.8)-(2.9) are satisfied and $i_C = i_U = 1$, therefore the corrections (2.3) (with j replaced by $j+1$) are realized. Using (2.10), we can write

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

and the first part of (3.4) follows from the first part of (2.9). Furthermore, (2.10) implies

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

and the second part of (3.4) follows from the second part of (2.9). □

Lemma 7. *Let the function f be convex and the number of null steps be infinite in the k-th iteration of Algorithm 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 (2.6), the Lemma 6 implies that $w_{j+1}^k \leq w_j^k$ for $j \geq j_0$ and 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 6 assures boundedness of the sequences $\{H_j^k\}_{j=1}^{\infty}$, $\{W_j^k\}_{j=1}^{\infty}$, which together with $|d_j^k| = |H_j^k \tilde{g}_j^k| \leq \|W_j^k\| |W_j^k \tilde{g}_j^k|$, $j \geq 1$, yields boundedness of $\{d_j^k\}_{j=1}^{\infty}$. Since $t_j^k \leq t_{max}$, $j \geq 1$, the sequence $\{y_j^k\}_{j=1}^{\infty}$ is also bounded and the local boundedness of ∂f (see Kiwiel 1985) implies boundedness of $\{g_j^k\}_{j=1}^{\infty}$ and $\{W_j^k g_{j+1}^k\}_{j=1}^{\infty}$. Denote

$$M = \sup\{|W_j^k g_{j+1}^k|, |W_j^k \tilde{g}_j^k|, \sqrt{\tilde{\alpha}_j^k}\ | j \geq j_0\}, \quad b = (1 - 2c_1)/(4M) \tag{3.5}$$

and assume first that $w_j^k > \delta > 0$ for all $j \geq j_0$. Since

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

we can use (3.4), Lemma 1 and Lemma 5 with $p = W_j^k \tilde{g}_j^k$, $q = W_j^k g_{j+1}^k$, $w = w_j^k$, $\alpha = \tilde{\alpha}_j^k$, $\beta = \alpha_{j+1}^k$, $c = c_1$ to obtain

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

for $j \geq j_0$ and 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 $w_j^k = (\tilde{g}_j^k)^T H_j^k \tilde{g}_j^k + 2\tilde{\alpha}_j^k$ and positive definiteness of all matrices H_j^k gives $\tilde{\alpha}_j^k \to 0$ as $j \to \infty$. Since $0 \leq \varrho_k |\tilde{g}_j^k|^2 \leq w_j^k$ (see (2.3) 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 and letting $j \to \infty$, we obtain $f(z) \geq f(x_k)$ for all $z \in \mathcal{R}^N$.
□

Theorem 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 $\{x \in \mathcal{R}^N | f(x) \leq f(x_1)\}$ is bounded). If Algorithm 1 terminates in the k-th 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 4 and Lemma 7. Thus we can restrict to the case when the number of iterations is infinite. Let $\bar{x}$ be a cluster point of $\{x_k\}$ and $K \subset \{1, 2, \ldots\}$ be an infinite set such that $x_k \xrightarrow{K} \bar{x}$. 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 (2.4). Using positive definiteness of H_j^k and the condition (2.4), we obtain

$$0 \leq 2c_1\tilde{\alpha}_k \leq c_1 w_k \leq (f(x_k) - f(x_{k+1}))/t_{min} \to 0, \quad k \geq 1, \tag{3.6}$$

thus $\tilde{\alpha}_k \to 0$. Furthermore, correction conditions in Step 2 and relations (2.3), (3.6) imply that $0 \leq c_1 t_{min} \varrho_k |\tilde{g}_k|^2 \leq c_1 t_{min} w_k \leq f(x_k) - f(x_{k+1})$, $k \geq 1$ and therefore

$$c_1 t_{min} \sum_{k=1}^{\infty} \varrho_k |\tilde{g}_k|^2 \leq f(x_1) - f(\bar{x}) < +\infty. \tag{3.7}$$

Assume first that $|\tilde{g}_k| > \delta > 0$ for all $k \geq 1$. Then it follows from Step 5 of Algorithm 1 that $\varrho_k \geq \varrho \min[1/\delta, \delta]/k$, $k \geq 1$. Using (3.7), we have

$$+\infty = \varrho\delta \min[1, \delta^2] \sum_{k=1}^{\infty} \frac{1}{k} \leq \sum_{k=1}^{\infty} \varrho_k |\tilde{g}_k|^2 < +\infty,$$

which is the contradiction. Therefore, 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 $z \in \mathcal{R}^N$ and $k \geq 1$ by Lemma 3 and the sequence $\{x_k\}$ is bounded, we obtain $f(z) \geq f(\bar{x})$ for all $z \in \mathcal{R}^N$ by letting $k \xrightarrow{\bar{K}} \infty$. □

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 of trial points. Furthermore, we denote here 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 the suitable stepsize t_k. The significant descent in the last step encourages us to construct the following quadratic approximation of $f(x_k + td_k)$

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

The bundle represents the polyhedral function (1.1) with the linearization errors $\beta_j^k \geq 0$ given by (1.2). For $x = x_k + td_k$ we have the following piecewise linear approximation of $f(x_k + td_k)$

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

To calculate t_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

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

where D is a given upper bound for distance from the point x_k in one step. Note that the possibility of stepsizes greater than 1 is useful here, because an 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 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

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

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 a minimum abscissa of an intersection of the lines, which create $\psi_P^k(t)$ and have $d_k^T g^j > 0$, with $\psi_L^k(t)$ and set

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

From now on we let the notation the same as in Algorithm 1. The minimization of the quadratic function (2.6) in Step 6, or $\tilde{\varphi}(\lambda_1, \lambda_2) = \varphi(\lambda_1, \lambda_2, 1 - \lambda_1 - \lambda_2)$, is not complicated. If it is not possible to compute an intersection of straight lines $\partial\tilde{\varphi}/\partial\lambda_1 = 0$, $\partial\tilde{\varphi}/\partial\lambda_2 = 0$, convexity of $\tilde{\varphi}$ implies that we can restrict to the lines $\lambda_1 = 0$, $\lambda_2 = 0$ and $\lambda_1 + \lambda_2 = 1$. As an example we give a formula for minimization within the line $\lambda_1 = 0$, which we regularly apply in the first null step after any descent step due to $\tilde{g}_1^k = g_k$. If $g_{j+1}^k \neq \tilde{g}_j^k$, then set

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

otherwise set $\lambda_{j,2}^k = 0$ for $\tilde{\alpha}_j^k < \alpha_{j+1}^k$ or $\lambda_{j,2}^k = 1$ for $\tilde{\alpha}_j^k \geq \alpha_{j+1}^k$.

Further we mention the stopping criterion. We define a descent tolerance $\varepsilon_f > 0$ and a maximum number $m_f \geq 1$ of consecutive too small function value variations 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 $w_j^k \leq \varepsilon$ and either $\Delta_k / \max[1, f(x_k)] < 2\varepsilon_f$ for $j = 1$, or $w_{j-1}^k \leq \varepsilon$ 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 $\Delta/\max[1, f(y_{j+1}^k)] \leq \varepsilon_f$ or $f(y_{j+1}^k) = f(x_k)$, then set $n_f = n_f + 1$, otherwise set $n_f = 0$. If $n_f \geq m_f$, then stop. If (2.4) holds, then set $x_{k+1} = y_{j+1}^k$, $g_{k+1} = g_{j+1}^k$, $t_k = t_j^k$, $d_k = d_j^k$, $\tilde{\alpha}_k = \tilde{\alpha}_j^k$, $\tilde{g}_k = \tilde{g}_j^k$, $w_k = w_j^k$, $H_k = H_j^k$, $\varrho_{k+1} = \varrho \min[1/|\tilde{g}_k|, |\tilde{g}_k|]/(k+1)$ and $\Delta_{k+1} = \Delta$, 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 $i_E = 1$, $k = k+1$ and go to Step 1, otherwise go to Step 8.

Finally, if we use the algorithm for 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 absolute value of the quantity in (2.5). 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 with those obtained by a standard VM method with the update U8, controlled scaling and backward Taylor stabilization (see Lukšan 1994) and by the proximal bundle method PBL mentioned in Lukšan and Vlček 1998. Problems 1-16 are described in Mäkelä and Neittaanmäki 1992, problems 17-18 in Zowe 1985, problems 19-21 in Kiwiel 1989, problem 22 in Bihain 1984, problem 23 in Facchinei and Lucidi 1993, problems 24-28 in Lukšan 1986 and problem 25 also in Bandler, Srinivasan and Charalambous 1972.

In Table 1 we give optimal values of tested functions.

The parameters of the algorithm had the values $t_{\min} = 10^{-10}$, $t_{\max} = 10^3$, $c_1 = 10^{-4}$, $\varepsilon = 5 \cdot 10^{-7}$, $\varepsilon_f = 10^{-7}$, $\varrho = 2 \cdot 10^{-6}$, $L = 1$, $\sigma = 100$, $\mathcal{J}_k = \{\max[1, k-N-2], \dots, k\}$, $k \geq 1$ and $m_f = 2$ for problems 1-23 and 26-28, $m_f = 5$ for problem 24 and $m_f = 3$ for problem 25.

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

Table 1. Test problems

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 and D is the maximum allowable distance in one step (see Section 4); values of D were chosen experimentally. For better comparison, we give two time data in the last line - the first time concerns all 28 problems, the second one (in parentheses) only 27 problems, with problem 22 removed.

As a conclusion from our limited numerical experiments we may state that

- the standard VM method is able to find a solution to almost all problems (here it failed only once) and the computational time can be essentially less than for proximal bundle methods;
- our method is comparable with proximal bundle methods in the number of function and subgradient evaluations, but the computational time can be significantly less;
- although our method is designed for convex functions, it can be applied also to some nonconvex problems.

	Standard VM			VMC				PBL		
Nr.	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	163	226	-0.9999954	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		2297	2384			2018	2098	
	Time = 17.30 (3.29) sec			Time = 3.46 (2.55) sec				Time = 8.07 (6.40) sec		

Table 2. Our test results

Bibliography

[1] J.W. Bandler, T.V. Srinivasan, C. Charalambous, "Minimax optimization of networks by grazor search," *IEEE Trans. Microwawe Theory Tech.*, vol. MTT-20 (1972) 596-604.

[2] A. Bihain, "Optimization of upper semidifferentiable functions," *Journal of Optimization Theory and Applications* 4 (1984) 545-568.

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

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

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

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

[7] K.C. Kiwiel, "An ellipsoid trust region bundle method for nonsmooth convex minimization", *SIAM J. Control Optim.* 27 (1989) 737-757.

[8] C. Lemaréchal, "Numerical experiments in nonsmooth optimization", in: E.A. Nurminski (ed.), *Progress in nonsmooth optimization*, Proceedings CP 82.58, IIASA, 2361 Laxenburg (Austria), 61-84.

[9] C. Lemaréchal, C. Sagastizábal, "Variable metric bundle methods: From conceptual to implementable forms", *Mathematical Programming* 76 (1997) 393-410.

[10] L. Lukšan: "A compact variable metric algorithm for linear minimax approximation", *Computing* 36 (1986) 355-373.

[11] L. Lukšan: "Computational experience with known variable metric updates", *Journal of Optimization Theory and Applications* 83 (1994) 27-47.

[12] L. Lukšan, J. Vlček: "A bundle-Newton method for nonsmooth unconstrained minimization", to appear in Mathematical Programming.

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

[14] H. Schramm, J. Zowe, "A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results", *SIAM J. Optimization* 2 (1992) 121-152.

[15] J. Zowe, "Nondifferentiable Optimization", in: *Computational Mathematical Programming* (1985) 323-356.